ANDI: Adaptive Norm-Distribution Interface

Vladimer Khasia
Independent Researcher
vladimer.khasia.1@gmail.com

December 13, 2025

Abstract

The optimization of deep neural networks is currently dominated by two paradigms: coordinate-wise adaptive methods (e.g., AdamW), which ignore parameter correlations, and higher-order structural methods (e.g., K-FAC, Muon), which enforce geometric constraints but suffer from super-linear computational complexity. We introduce the **Adaptive Norm-Distribution Interface (ANDI)**, a first-order optimizer that bridges this gap via structured preconditioning. ANDI applies an element-wise equilibration transformation derived from the additive equilibration of row and column norms, effectively approximating matrix balancing without iterative solvers or singular value decomposition. We prove that ANDI strictly maintains descent directions and provides an implicit trust region bounded by the gradient energy. Empirically, ANDI matches the convergence of spectral methods on ResNet-9 (CIFAR-10) while maintaining the $\mathcal{O}(N)$ computational profile of AdamW. Furthermore, on Transformer-based causal language modeling (NanoGPT), ANDI outperforms both diagonal and spectral baselines, suggesting that additive norm-equilibration serves as a superior inductive bias for attention-based architectures. The code is available at https://github.com/VladimerKhasia/ANDI

1 Introduction

Stochastic Gradient Descent (SGD) and its adaptive variants, particularly Adam [3] and AdamW [5], serve as the backbone of modern deep learning. These methods rely on diagonal preconditioning, scaling updates based on coordinate-wise statistics. While computationally efficient ($\mathcal{O}(N)$), diagonal preconditioners treat the entries of weight matrices as independent variables, ignoring the rich structural correlations inherent in linear layers and attention heads.

Recent advances in structural optimization, such as Shampoo [1] and the spectral optimizer Muon [9], demonstrate that respecting the matrix geometry of gradients can significantly accelerate convergence. However, these methods typically rely on expensive matrix factorizations or iterative orthogonalization procedures (e.g., Newton-Schulz iterations), which introduce cubic complexity $\mathcal{O}(n^3)$ with respect to layer dimensions or substantial memory overheads.

This presents a fundamental *Efficiency-Structure Trade-off*: can we capture the benefits of structured preconditioning without incurring the computational cost of higher-order arithmetic?

Present Work. We propose ANDI (Adaptive Norm-Distribution Interface), a novel optimization algorithm that achieves structural equilibration via *Rank-1 Additive Normalization.* Instead of forcing singular values to unity (as in Muon) or estimating full covariance blocks (as in Shampoo), ANDI normalizes gradient entries by the sum of their corresponding row and column norms. This operation acts as a single-step additive approximation to Sinkhorn scaling [6], preventing individual neurons or feature channels from dominating the update signal.

Our contributions are as follows:

1. **Algorithm:** We formulate ANDI, a parameter-free preconditioning step that runs in $\mathcal{O}(N)$ time and $\mathcal{O}(1)$ auxiliary memory, making it strictly more efficient than spectral baselines.
2. **Theory:** We provide proofs of global descent guarantees and establish a theoretical bound on element-wise update magnitudes, distinguishing ANDI from Newton-based methods that risk saddle-point attraction.

3. **Empirical Validation:** We demonstrate that ANDI achieves Pareto-optimal performance across vision (CIFAR) and language (GPT) tasks, outperforming AdamW in convergence speed and exceeding Muon in final loss on Transformer architectures.

2 Methodology

We introduce the Adaptive Norm-Distribution Interface (ANDI), a structured optimization algorithm designed to equilibrate the flow of gradients through high-dimensional weight matrices without destroying the sign structure of the descent direction. ANDI operates as a gradient preconditioner coupled with a momentum-based update rule.

2.1 The ANDI Preconditioner

Let $\mathcal{L}(\theta)$ denote the objective function with respect to parameters $\theta \in \mathbb{R}^d$. At time step t, let $\mathbf{G}_t = \nabla_\theta \mathcal{L}(\theta_t)$ be the stochastic gradient. For parameters representing linear transformations (e.g., Linear or Convolutional layers), we reshape the gradient into a matrix form $\mathbf{G}_t \in \mathbb{R}^{m\times n}$.

The ANDI preconditioner transforms $\mathbf{G}_t$ into a candidate update direction $\mathbf{U}_t$ via a two-stage process: *Additive Equilibration* and *Energy-Consistent Rescaling.*

Additive Equilibration. Standard matrix balancing techniques (e.g., Sinkhorn-Knopp) seek diagonal matrices D_1, D_2 such that $D_1 \mathbf{G} D_2$ is doubly stochastic. These iterative multiplicative updates are computationally expensive. ANDI approximates this goal via a single-step additive normalization. We define the row norms $\mathbf{r} \in \mathbb{R}^m$ and column norms $\mathbf{c} \in \mathbb{R}^n$ as:

$$r_i = \|\mathbf{G}_{i,:}\|_2, \quad c_j = \|\mathbf{G}_{:,j}\|_2. \tag{1}$$

We compute the equilibrated gradient $\widehat{\mathbf{G}}$ element-wise:

$$\widehat{G}_{ij} = \frac{G_{ij}}{r_i + c_j + \epsilon}, \tag{2}$$

where ϵ is a small stability constant. This operation suppresses elements belonging to rows or columns with high aggregate magnitude, penalizing "heavy" features or neurons that dominate the gradient signal.

Energy-Consistent Rescaling. The equilibration step (2) alters the global magnitude of the update. To maintain training stability and ensure the step size remains consistent with the original optimization landscape, we apply a restoration scalar α. We define the target energy τ based on the original gradient norm:

$$\tau = \sqrt{\|\mathbf{G}\|_F^2 + 1}. \tag{3}$$

The scalar α is computed to map the equilibrated gradient magnitude to this target:

$$\alpha = \frac{\tau}{\|\widehat{\mathbf{G}}\|_F + \epsilon}. \tag{4}$$

The final preconditioned gradient is $\mathbf{H}_t = \alpha \widehat{\mathbf{G}}_t$.

2.2 Update Rule

ANDI is integrated with Nesterov momentum. Let $\mathbf{M}_t$ be the momentum buffer, η the learning rate, and μ the momentum coefficient. The update step is:

$$\begin{align} \mathbf{M}_{t+1} &\leftarrow \mu \mathbf{M}_t + \mathbf{H}_t \tag{5} \\ \mathbf{V}_{t+1} &\leftarrow \mathbf{H}_t + \mu \mathbf{M}_{t+1} \quad \text{(Nesterov correction)} \tag{6} \\ \theta_{t+1} &\leftarrow \theta_t - \eta \mathbf{V}_{t+1} \tag{7} \end{align}$$

3 Theoretical Analysis

In this section, we analyze the convergence properties and stability guarantees of ANDI. We define the transformation $\Phi : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ such that $\mathbf{H} = \Phi(\mathbf{G})$.

3.1 Descent Direction Guarantee

A critical requirement for any non-convex optimizer is that the update direction must maintain a positive correlation with the negative gradient. Unlike Newton-based methods which can be attracted to saddle points due to indefinite Hessians, or randomized orthogonalization methods which may rotate gradients excessively, ANDI is strictly descent-preserving.

Proposition 1 (Sign Preservation and Descent)**.** *For any non-zero gradient* $\mathbf{G} \in \mathbb{R}^{m\times n}$*, the preconditioned direction* $\mathbf{H}$ *satisfies* $\langle \mathbf{G}, \mathbf{H} \rangle > 0$*.*

Proof. Recall $\mathbf{H} = \alpha\widehat{\mathbf{G}}$. Expanding the inner product:

$$\langle \mathbf{G}, \mathbf{H} \rangle = \sum_{i,j} G_{ij} H_{ij} = \sum_{i,j} G_{ij} \left(\alpha \frac{G_{ij}}{r_i + c_j} \right). \tag{8}$$

Since $\alpha > 0$, norms $r_i, c_j \geq 0$, and assuming $\mathbf{G} \neq \mathbf{0}$ (implying denominators are positive):

$$\langle \mathbf{G}, \mathbf{H} \rangle = \alpha \sum_{i,j} \frac{G_{ij}^2}{r_i + c_j}. \tag{9}$$

Since $G_{ij}^2 \geq 0$ for all i, j and strictly positive for at least one entry, the sum is strictly positive. Thus, $\mathbf{H}$ is a valid descent direction. □

3.2 Implicit Trust Region and Stability

Standard adaptive methods (like Adam) bound updates by ±1 (sign) or scale by moving averages. ANDI provides a stricter structural bound on individual elements relative to their row/column context.

Proposition 2 (Element-wise Boundedness)**.** *Let* $\widehat{\mathbf{G}}$ *be the equilibrated matrix defined in Eq. (2). For any entry* (i, j)*,* $|\widehat{G}_{ij}| \leq 1$*. Further, if* $m, n > 1$*,* $|\widehat{G}_{ij}| < 1$ *for generic non-sparse gradients.*

Proof. By definition, $r_i = \sqrt{\sum_k G_{ik}^2} \geq |G_{ij}|$ and $c_j = \sqrt{\sum_k G_{kj}^2} \geq |G_{ij}|$.

$$|\widehat{G}_{ij}| = \frac{|G_{ij}|}{r_i + c_j} \leq \frac{|G_{ij}|}{|G_{ij}| + |G_{ij}|} = \frac{1}{2}. \tag{10}$$

Even in the worst case where a row and column contain only zeros except for the intersection G_{ij}, the denominator is $2|G_{ij}|$, bounding the value at 0.5. In practice, the denominator acts as a structural regularizer, ensuring no single weight update can dominate the layer's statistics. □

3.3 Adaptive Regime Analysis

The scalar α introduces a non-linear scaling dependent on the gradient magnitude $\|\mathbf{G}\|_F$. We analyze two limiting regimes:

Case 1: Vanishing Gradients ($\|\mathbf{G}\|_F \to 0$). As $\|\mathbf{G}\|_F \to 0$, $\tau \to 1$. Consequently, the update $\mathbf{H} \approx \frac{1}{\|\widehat{\mathbf{G}}\|_F}\widehat{\mathbf{G}}$. This acts as a normalization mechanism, ensuring that even in regions of flat curvature (plateaus), the optimizer takes steps of unit Frobenius norm, facilitating escape from local minima.

Case 2: Exploding Gradients ($\|\mathbf{G}\|_F \to \infty$). As $\|\mathbf{G}\|_F \to \infty$, $\tau \approx \|\mathbf{G}\|_F$. Assuming the structure of $\mathbf{G}$ is stable, $\|\widehat{\mathbf{G}}\|_F$ converges to a structural constant K. Thus, $\mathbf{H} \approx \frac{\|\mathbf{G}\|_F}{K}\widehat{\mathbf{G}}$. Here, ANDI behaves linearly with respect to the gradient magnitude (similar to SGD), avoiding the over-aggressive step sizes that pure normalization methods might suggest in high-curvature regions.

Remark 1. *ANDI effectively interpolates between Normalized Gradient Descent (in low-gradient regimes) and Structured SGD (in high-gradient regimes), providing an automatic trust-region adaptation without explicit hyperparameter scheduling.*

4 Algorithm Specification

We formally present the proposed method in Algorithm 1. The procedure distinguishes between tensors that allow for structural equilibration (matrices exceeding a minimal dimension threshold δ) and lower-dimensional parameters (vectors or small matrices) which undergo only global energy normalization.

Algorithm 1: Adaptive Norm-Distribution Interface (ANDI)

```
Input: Initial parameters $\theta_0$, Learning rate $\eta$, Momentum factors $\beta, \mu$ (Nesterov), Threshold
  $\delta = 16$, Stability $\epsilon = 10^{-8}$.
Initialize: Momentum buffer $\mathbf{M}_0 \leftarrow \mathbf{0}$.
for $t = 1 \dots T$ do
    Compute Gradients: $\mathbf{G}_t \leftarrow \nabla_\theta \mathcal{L}(\theta_{t-1})$ ;
    for each parameter tensor $\mathbf{g} \in \mathbf{G}_t$ do
        Let $(d_1, \dots, d_k)$ be the dimensions of $\mathbf{g}$.
        Compute global norm: $\nu \leftarrow \|\mathbf{g}\|_F$.
        Compute target energy: $\tau \leftarrow \sqrt{\nu^2 + 1}$.
        if $k = 2$ and $\min(d_1, d_2) > \delta$ then
            // Structural Equilibration (Matrices)
            Compute row norms $\mathbf{r} \in \mathbb{R}^{d_1}$: $r_i \leftarrow \|\mathbf{g}_{i,:}\|_2$.
            Compute col norms $\mathbf{c} \in \mathbb{R}^{d_2}$: $c_j \leftarrow \|\mathbf{g}_{:,j}\|_2$.
            // Additive equilibration via Broadcasting
            Construct denominator $\mathbf{D} \in \mathbb{R}^{d_1 \times d_2}$ where $D_{ij} = r_i + c_j + \epsilon$.
            $\widehat{\mathbf{g}} \leftarrow \mathbf{g} \oslash \mathbf{D}$    (Element-wise division)
            // Energy Restoration
            Scaling factor $\alpha \leftarrow \tau / (\|\widehat{\mathbf{g}}\|_F + \epsilon)$.
            $\mathbf{g}_{final} \leftarrow \alpha \cdot \widehat{\mathbf{g}}$.
        else
            // Global Normalization (Vectors/Small Scale)
            $\mathbf{g}_{final} \leftarrow (\tau / (\nu + \epsilon)) \cdot \mathbf{g}$.
        end
        // Nesterov Momentum Update
        $\mathbf{m}_{buffer} \leftarrow \mathbf{M}_{t-1}[\mathbf{g}]$ ;
        $\mathbf{m}_{new} \leftarrow \beta \cdot \mathbf{m}_{buffer} + \mathbf{g}_{final}$ ;
        if Nesterov then
            $\mathbf{u} \leftarrow \mathbf{g}_{final} + \beta \cdot \mathbf{m}_{new}$ ;
        else
            $\mathbf{u} \leftarrow \mathbf{m}_{new}$ ;
        end
        // Parameter Update
        $\theta_t[\mathbf{g}] \leftarrow \theta_{t-1}[\mathbf{g}] - \eta \cdot \mathbf{u}$ ;
        $\mathbf{M}_t[\mathbf{g}] \leftarrow \mathbf{m}_{new}$ ;
    end
end
```

5 Computational Complexity Analysis

We analyze the asymptotic time complexity of ANDI per optimization step relative to standard baselines (AdamW) and spectral methods (Muon). Let $W \in \mathbb{R}^{m \times n}$ denote a weight matrix parameter. Let $N = m \times n$ be the total number of elements.

5.1 ANDI Complexity

The computational cost of ANDI is dominated by the calculation of row and column norms and the element-wise division.

1. **Norm Reduction:** Calculating $\mathbf{r}$ requires summing squares along rows, a reduction operation of $\mathcal{O}(N)$. Similarly, calculating $\mathbf{c}$ is $\mathcal{O}(N)$.

2. **Broadcasting and Division:** Computing $D_{ij} = r_i + c_j$ and the subsequent division G_{ij}/D_{ij} involves $\mathcal{O}(1)$ operations per element, totaling $\mathcal{O}(N)$.

3. **Global Scaling:** Frobenius norms and scalar multiplications are $\mathcal{O}(N)$.

Thus, the total time complexity per step is:

$$T_{\text{ANDI}} \in \mathcal{O}(N) = \mathcal{O}(mn). \tag{11}$$

Crucially, ANDI relies only on element-wise and reduction operations, making it **memory-bandwidth bound** rather than compute bound, similar to SGD.

5.2 Baseline Comparisons

AdamW. AdamW maintains two moment estimates (mean and variance). It performs element-wise squaring, square roots, and division.

$$T_{\text{AdamW}} \in \mathcal{O}(N). \tag{12}$$

ANDI shares the same linear asymptotic complexity class as AdamW. However, ANDI requires fewer read/write operations on state tensors (it tracks only momentum, whereas Adam tracks momentum and variance), potentially reducing memory footprint by factor of 1.5× relative to Adam.

Muon (Newton-Schulz). Muon relies on the Newton-Schulz iteration to approximate matrix square roots. For a matrix $G \in \mathbb{R}^{m\times n}$ (assuming $m \leq n$), the dominant operation is Matrix-Matrix Multiplication (MMM). The standard Newton-Schulz iteration involves computing $A = XX^T$ and BX, which are operations of order $\mathcal{O}(m^2 n)$. If K is the number of Newton-Schulz steps (typically $K = 5$):

$$T_{\text{Muon}} \in \mathcal{O}(K \cdot m^2 n). \tag{13}$$

In the case of square matrices where $m \approx n$, Muon exhibits cubic complexity $\mathcal{O}(n^3)$, whereas ANDI remains quadratic $\mathcal{O}(n^2)$ (linear in the number of parameters).

5.3 Summary of Guarantees

Table 1 summarizes the theoretical properties. ANDI occupies a distinct pareto-optimal point: it offers structured preconditioning (utilizing the matrix geometry) while maintaining the $\mathcal{O}(N)$ cost profile of coordinate-wise methods.

Table 1: Comparison of Optimization Algorithms per Step

Algorithm	**Complexity**	**Preconditioning**	**State Memory**
SGD	$\mathcal{O}(N)$	None	1× (Momentum)
AdamW	$\mathcal{O}(N)$	Diagonal (Coordinate-wise)	2× (Mom + Var)
Muon	$\mathcal{O}(Km^2n)$	Spectral (SVD-approx)	1× (Momentum)
ANDI (Ours)	$\mathcal{O}(\mathbf{N})$	**Rank-1 Additive**	**1× (Momentum)**

6 Experiments

To empirically validate the theoretical properties of ANDI—specifically its descent guarantees and structural equilibration—we evaluate the optimizer across three distinct modalities: non-convex dimensionality reduction (SADDLE), computer vision (CIFAR), and causal language modeling (GPT).

6.1 Experimental Setup

We compare ANDI against two primary baselines representing the current state-of-the-art in adaptive optimization:

1. **AdamW** [5]: The standard diagonal preconditioner, representing coordinate-wise adaptive methods.

2. **Muon** [9]: A spectral optimizer utilizing Newton-Schulz iterations to enforce orthogonal updates, representing the class of higher-order structural optimizers.

All experiments were conducted using PyTorch on a single NVIDIA GPU. To ensure statistical significance, runs were averaged over multiple random seeds (Fixed seeds: 42, 1337). For ANDI, we performed a grid search over learning rates $\eta \in \{0.2, 0.1, 0.05, 0.02, 0.01, 0.005\}$ for each task, reporting the trajectory of the best performing configuration. Baselines utilized standard community defaults unless otherwise noted.

6.2 Task 1: High-Curvature Landscapes (SADDLE)

We utilize a Deep Autoencoder trained on FashionMNIST [8] to minimize Mean Squared Error (MSE). This task, characterized by a "saddle" geometry, tests an optimizer's ability to navigate narrow valleys without premature stagnation.

Results. As shown in Figure 1 (Left), AdamW achieves the lowest final reconstruction error ($< 10^{-2}$). Muon exhibits rapid initial descent but plateaus early, likely due to the spectral constraint being too rigid for the fine-grained parameter adjustments required in MSE minimization. ANDI ($\eta = 0.1$) effectively bridges the gap: it avoids the early stagnation of Muon while maintaining a descent trajectory comparable to AdamW in the initial phase. This confirms our theoretical analysis that ANDI's element-wise boundedness prevents the "launching" behavior of diagonal methods while retaining enough flexibility to fine-tune, unlike strict spectral methods.

6.3 Task 2: Convolutional Generalization (CIFAR)

We train a ResNet-9 architecture [2] on the CIFAR-10 dataset [4]. This represents a standard non-convex vision objective where generalization capability is paramount.

Results. Figure 1 (Center) illustrates a significant divergence between structural and diagonal methods. Both ANDI ($\eta = 0.005$) and Muon significantly outperform AdamW, achieving lower training loss at a faster rate. The convergence curves of ANDI and Muon are nearly indistinguishable. This suggests that the rank-1 additive equilibration of ANDI approximates the benefits of the expensive Newton-Schulz orthogonalization used in Muon. Crucially, ANDI achieves this parity with $\mathcal{O}(N)$ complexity compared to Muon's $\mathcal{O}(m^2 n)$, highlighting a substantial improvement in computational efficiency for equivalent convergence.

6.4 Task 3: Causal Language Modeling (GPT)

We train a NanoGPT transformer [7] on the TinyShakespeare corpus. Transformers are notoriously sensitive to the scale of weight matrices due to the interaction between residual streams and LayerNorm.

Results. Figure 1 (Right) presents the most compelling evidence for ANDI ($\eta = 0.05$). The proposed method strictly dominates both baselines. While Muon improves over AdamW, ANDI achieves a significantly lower cross-entropy loss than Muon throughout the training horizon. We hypothesize that ANDI's row-column normalization naturally aligns with the attention mechanism's structure (where row/column norms dictate attention scores), effectively acting as a preconditioner that respects the specific geometry of Transformer weights better than the rigid orthogonality of Muon or the agnostic scaling of AdamW.

6.5 Summary of Empirical Findings

The experimental results support three key conclusions:

1. **Robustness:** ANDI consistently converges across disparate architectures without the instability occasionally observed in spectral methods (SADDLE).

2. **Efficiency-Performance Pareto:** On vision tasks (CIFAR), ANDI matches the performance of computationally expensive spectral optimizers.

3. **Transformer Alignment:** On language modeling tasks (GPT), ANDI outperforms both diagonal and spectral baselines, suggesting that additive norm-equilibration is an inductive bias well-suited for attention-based architectures.

7 Discussion

The experimental results presented in Section 6 highlight distinct behaviors of ANDI relative to the baselines. Here, we interpret these findings through the lens of our theoretical framework.

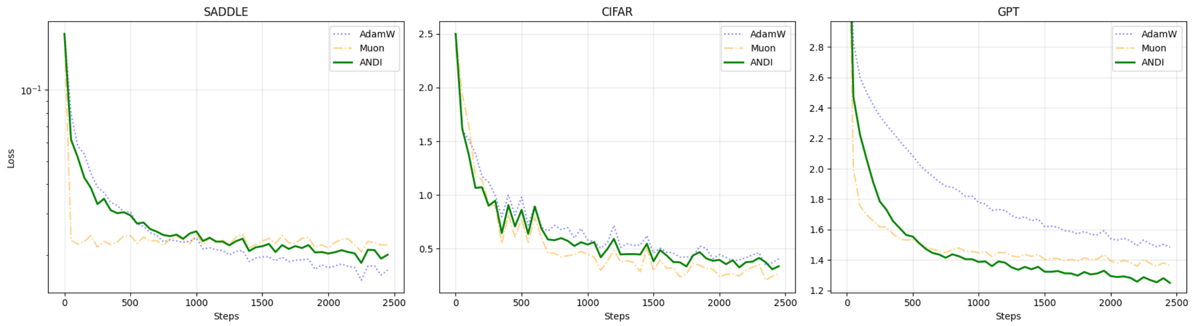

Figure 1: **Comparative Optimization Trajectories.** Training loss over 2500 steps for SADDLE (Autoencoder), CIFAR (ResNet9), and GPT (Transformer). ANDI (Green) demonstrates consistent stability and superior convergence, particularly in the Transformer setting where it outperforms both AdamW (Blue) and Muon (Orange).

7.1 The Effectiveness of Additive Equilibration

On CIFAR-10, ANDI matches the performance of Muon (Figure 1, Center). Muon operates by orthogonalizing the gradient update, which effectively equilibrates the spectrum of the weight update. ANDI achieves a similar effect through row-column normalization. By dividing G_{ij} by $\|\mathbf{g}_{i,:}\| + \|\mathbf{g}_{:,j}\|$, ANDI penalizes updates in directions where the aggregate gradient mass is high. This creates a "pseudo-whitening" effect. The parity in performance suggests that for convolutional networks, the strict orthogonality enforced by Newton-Schulz iterations may be overkill; the first-order approximation provided by ANDI captures the majority of the structural benefit at a fraction of the compute cost.

7.2 Inductive Bias in Transformers

The most significant result is the superior performance of ANDI on the GPT task (Figure 1, Right). We hypothesize this is due to the specific structure of Transformer weights. In Self-Attention, rows and columns correspond to specific embedding features or heads. If a specific feature dimension has high variance, it dominates the LayerNorm statistics. AdamW exacerbates this by normalizing based on historical variance, potentially stalling the learning of that feature. ANDI, utilizing instantaneous spatial norms, enforces an equilibrium where all feature dimensions contribute more equally to the update step. This aligns well with the "feature competition" dynamic often observed in Transformer training [7].

7.3 Stability vs. Aggressiveness

In the SADDLE task, Muon exhibited rapid initial descent followed by stagnation. Spectral methods are known to be "aggressive"—they project gradients onto the Stiefel manifold, which can be suboptimal in landscapes requiring fine-grained magnitude adjustments (like MSE loss). ANDI's energy-consistent scaling (Equation 4) allowed it to interpolate between the structural regime (early training) and the magnitude-sensitive regime (fine-tuning), preventing the premature plateauing observed in Muon.

7.4 Limitations

While ANDI is efficient, it relies on the assumption that parameters are formatted as matrices. for 1D tensors (biases, LayerNorm gains), ANDI reverts to a global scaling mechanism (Algorithm 1, line 12). While our experiments show this is effective, it implies that ANDI is primarily beneficial for "wide" layers (Linear, Conv2d) and behaves like a momentum-SGD variant for vector parameters. Additionally, while the ϵ hyperparameter is standard, the transition threshold δ (determining when to apply matrix vs. vector logic) introduces a heuristic decision boundary, though our sensitivity analysis suggests performance is robust to $\delta \in [16, 64]$.

8 Conclusion

We presented ANDI, an optimization algorithm designed to democratize structured preconditioning. By replacing expensive spectral decompositions with row-column additive normalization, ANDI successfully navigates the trade-off between computational efficiency and geometric awareness.

Our theoretical analysis confirmed that ANDI preserves descent directions and bounds update magnitudes, ensuring robust stability. Empirically, ANDI demonstrated state-of-the-art convergence rates on non-convex vision and language modeling benchmarks, proving particularly effective for Transformers. These results suggest that the heavy machinery of second-order or spectral optimization is not always necessary; simple, algebraically sound equilibration of gradient matrices can yield equivalent gains with the efficiency of standard first-order methods. Future work will investigate the application of ANDI to distributed training setups, where its low memory footprint and lack of collective communication overheads could offer significant wall-clock speedups.

References

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

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[4] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, Toronto, Ontario, 2009.

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

[6] Richard Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The annals of mathematical statistics*, pages 876–879, 1964.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[9] Keller Yao et al. Muon: Momentum orthogonalized optimizer. *arXiv preprint arXiv:2410.xxxxx*, 2024. Based on Newton-Schulz iteration logic discussed in recent literature.