ANDI: Arithmetic Normalization / Decorrelated Inertia

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

December 10, 2025

Abstract

Modern neural network optimizers face a trilemma: they must balance Adaptivity (Adam), Structural Regularization (Shampoo, MUON), and Computational Efficiency (SGD). While second-order and structural methods offer superior generalization, they typically incur prohibitive $O(N^3)$ compute costs or massive memory overheads. We introduce **ANDI**, a first-order optimizer that approximates structural preconditioning in linear $O(N)$ time with $O(1)$ memory overhead. ANDI achieves this via a three-step mechanism: (1) Prime Topology Mixing to break local grid correlations; (2) One-Shot Arithmetic Equilibration to stabilize feature variance via broadcasted normalization; and (3) Hypotenuse Energy Regularization to automatically navigate saddle points. We present two variants: **ANDI-Direct**, which performs self-equilibration, and **ANDI-Lateral**, which employs a lateral inhibition mechanism to enforce spatial competition. Empirical results across MLPs, CNNs, and RNNs demonstrate that ANDI matches the convergence speed of adaptive methods and the generalization of structural methods, while maintaining the memory footprint of standard SGD. The code is available at `https://github.com/VladimerKhasia/ANDI`

1 Introduction

The landscape of deep learning optimization is dominated by Adaptive Moment Estimation (Adam) [4]. While recent symbolic algorithms like **Lion** [1] have optimized the update sign to improve efficiency, they remain element-wise methods. Adam and Lion resolve curvature issues via scalar scaling, but this doubles the memory requirement (for second moments) and has been theoretically shown to generalize poorly compared to SGD, often converging to sharp minima [11], even with modern decoupled weight decay [6]..

To address generalization, structural optimizers attempt to precondition the gradient by capturing correlations between parameters. This lineage spans from second-order approximations like **K-FAC** [7] to recent tensor-based methods like **Shampoo** [2], **SOAP** [10], and **Muon** [3]. While effective, these methods rely on expensive operations like Singular Value Decomposition (SVD), Cholesky factorizations, or iterative Newton-Schulz algorithms, scaling at $O(N^3)$ or $O(N^{1.5})$.

We propose **ANDI**, an optimizer designed to solve this "Optimization Trilemma" (Adaptivity, Structure, Efficiency). ANDI replaces heavy matrix iterations and variance buffers with a linear-time approximation of Matrix Balancing. While exact equilibration (e.g., via the Sinkhorn-Knopp algorithm [9]) requires iterative row-column scaling to standardize marginals, ANDI employs a *One-Shot Arithmetic Equilibration*. Instead of strict orthogonalization, ANDI targets spectral density reduction by balancing row and column energies in a single pass. To achieve this globally without expensive dense matrix operations, ANDI utilizes *Prime Topology Mixing*. By applying cyclic shifts with a prime stride, we disrupt local grid correlations and avoid harmonic interference with architecture dimensions (typically powers of 2). This forces the optimizer to equilibrate the gradient as a global entity, achieving structural regularization in a single $O(N)$ pass.

We propose two variants of the equilibration mechanism: **ANDI-Direct**, which normalizes the mixed topology directly, and **ANDI-Lateral**, which employs a novel 'Lateral Inhibition' strategy by anchoring normalization statistics to the original grid while shifting the signal. Our experiments demonstrate that the Lateral variant provides superior decorrelation by forcing features to compete for spatial capacity.

2 Methodology

The ANDI update rule is distinct from standard adaptive methods. It does not track element-wise variance buffers. Instead, it transforms the raw gradient G via a structural operator $\Phi(G)$ before applying a standard momentum update. The operator Φ consists of three atomic phases.

2.1 Step A: Prime Topology Mixing

Standard gradient normalization methods often reinforce local artifacts. To enforce global regularization in $O(N)$ time, we introduce a mixing step inspired by Permutation Matrices. We apply a cyclic shift to the gradient columns using a prime number stride P (typically $P = 7$). This ensures that the shift is co-prime to standard architecture dimensions (powers of 2), guaranteeing that spatial neighbors are misaligned during the normalization step.

Condition for Maximal Mixing: Strictly speaking, to guarantee that the cyclic shift visits all spatial positions without forming short sub-cycles, the stride P must be co-prime to the feature dimension D (i.e., $\gcd(P, D) = 1$). If D is a multiple of P (e.g., $D = 56, P = 7$), the shift aligns with grid boundaries, potentially dampening the decorrelation effect. However, modern deep learning architectures predominantly utilize dimensions that are powers of 2 ($D = 2^k$). Since 7 is not a power of 2, the condition $\gcd(7, 2^k) = 1$ holds universally for standard layers (e.g., 64, 128, 512), making $P = 7$ a robust empirical default.

2.2 Step B: Arithmetic Equilibration

Ideally, we seek a well-conditioned gradient where no single feature dimension dominates the energy spectrum. In numerical linear algebra, this is known as *Matrix Equilibration* (or Balancing). Unlike *Whitening*, which requires expensive $O(N^3)$ basis rotation to remove correlations ($G^T G = I$), Equilibration scales rows and columns to standardize marginals. ANDI approximates this in a single pass using the Arithmetic Mean ($R + C$) of row/column norms, providing a numerically stable lower bound on the divisor.

We propose two variants of this mechanism:

2.2.1 Variant 1: ANDI-Direct (Self-Equilibration)

In the standard formulation, we shift the gradient first, then calculate the marginals of this new topology to normalize it. This ensures the gradient is equilibrated with respect to its own shifted geometry.

$$G_{mix} = \text{Roll}(G, P) \quad \rightarrow \quad G_{white} = \frac{G_{mix}}{\|G_{mix}\|_R + \|G_{mix}\|_C} \tag{1}$$

2.2.2 Variant 2: ANDI-Lateral (Cross-Gating)

Unlike standard equilibration, which normalizes a matrix by its own marginals, ANDI-Lateral employs a *Cross-Reference* strategy. We define the "Capacity" matrix K based on the marginals of the *original* gradient topology:

$$K_{i,j} = \|G_{i,:}\|_2 + \|G_{:,j}\|_2 \tag{2}$$

We then apply the prime-stride shift operator $\mathcal{S}$ to the gradient and normalize it by this unshifted capacity:

$$G_{lateral} = \mathcal{S}(G) \oslash (K + \epsilon) \tag{3}$$

Mathematically, this is a **Spatial Gating** operation. It enforces a structural prior: a feature shifted from location (u, v) to (i, j) is suppressed if the destination (i, j) currently possesses high variance. This decorrelates the update by penalizing features that align with high-energy artifacts in the parameter grid. This mimics *Lateral Inhibition* (similar to Local Response Normalization [5]) found in biological neural processing.

2.3 Step C: Hypotenuse Energy Regularization

Standard adaptive methods (like Adam) add ϵ to the denominator to prevent division by zero. ANDI takes the inverse approach: we enforce a *Unit Energy Floor*. We define the target energy λ via the Hypotenuse function:

$$\lambda(G) = \sqrt{\|G\|_F^2 + 1} \tag{4}$$

The final update is rescaled: $G_{final} = G_{white} \frac{\lambda(G)}{\|G_{white}\|_F}$. This operator $\mathcal{H}(G)$ has two distinct asymptotic behaviors:

1. **Saddle Regime** ($\|G\| \ll 1$): The gain approaches $1/\|G\|$, effectively normalizing the gradient to unit length to escape flat plateaus.
2. **Convex Regime** ($\|G\| \gg 1$): The gain approaches 1.0, preserving the magnitude of strong signals (recovering SGD behavior).

Algorithm 1 ANDI-Direct (Self-Equilibration)

Require: Gradient G_t, Parameters θ_t, Learning Rate η, Momentum μ, Prime Shift $P = 7$
Input: Gradient G_t at step t
$N_{in} \leftarrow \|G_t\|_F$ ▷ Calculate Input Norm
Let (d_{out}, d_{in}) be the dimensions of G_t flattened to 2D
if $d_{out} > 16$ **and** $d_{in} > 16$ **then**
    // 1. Mix Topology First
    $G_{mix} \leftarrow \text{Roll}(G_{flat}, \text{shifts} = P, \text{dim} = -1)$
    // 2. Calculate Marginals of Mixed View
    $R \leftarrow \|G_{mix}\|_{\text{row}}, \quad C \leftarrow \|G_{mix}\|_{\text{col}}$
    // 3. Normalize (Self-Equilibration)
    $G_{white} \leftarrow G_{mix} \oslash (R \oplus C)$
    $G_{white} \leftarrow \text{Roll}(G_{white}, \text{shifts} = -P, \text{dim} = -1)$
else
    $G_{white} \leftarrow G_t/(N_{in} + \epsilon)$
end if
// Step C: Hypotenuse Energy Scaling
$N_{white} \leftarrow \|G_{white}\|_F$
$\lambda_{target} \leftarrow \sqrt{N_{in}^2 + 1}$ ▷ Target Norm derived from Energy
$G_{final} \leftarrow G_{white} \cdot \frac{\lambda_{target}}{N_{white}}$
// Step D: Momentum & Update
$V_{t+1} \leftarrow \mu V_t + G_{final}$
$\theta_{t+1} \leftarrow \theta_t - \eta(V_{t+1} + \mu G_{final})$

3 Complexity Analysis

ANDI is theoretically the most efficient structural architecture possible (Table 1).

Table 1: Computational and Memory Complexity (N = Parameters)

Optimizer	Time Complexity	Memory Overhead	Mechanism
SGD	$O(N)$	1× (Momentum)	Vector
Adam	$O(N)$	2× (Mom + Var)	Element-wise
MUON	$O(\text{Iter} \cdot N^{1.5})$	1× (Momentum)	Newton-Schulz
ANDI	$\mathbf{O(N)}$	**1× (Momentum)**	**Arithmetic One-Shot**

Time Complexity: Let $G \in \mathbb{R}^{H \times W}$.

- *Marginal Calculation:* Computing row/col norms requires traversing the matrix once: $O(HW) = O(N)$.

Algorithm 2 ANDI-Lateral (Lateral Inhibition)

Require: Gradient G_t, Parameters θ_t, Learning Rate η, Momentum μ, Prime Shift $P = 7$

Input: Gradient G_t at step t
$N_{in} \leftarrow \|G_t\|_F$ ▷ Calculate Input Norm
Let (d_{out}, d_{in}) be the dimensions of G_t flattened to 2D
if $d_{out} > 16$ **and** $d_{in} > 16$ **then** ▷ Structural Gating
    $G_{flat} \leftarrow \text{Reshape}(G_t, (d_{out}, d_{in}))$
    // Step A: Calculate Reference Marginals (Anchors)
    $R \leftarrow \|G_{flat}\|_{\text{row}} \in \mathbb{R}^{d_{out} \times 1}$
    $C \leftarrow \|G_{flat}\|_{\text{col}} \in \mathbb{R}^{1 \times d_{in}}$
    // Step B: Prime Mixing & Lateral Inhibition
    $G_{shift} \leftarrow \text{Roll}(G_{flat}, \text{shifts} = P, \text{dim} = -1)$ ▷ The Twist
    $G_{white} \leftarrow G_{shift} \oslash (R \oplus C)$ ▷ Norm. Shifted Grad by Original Marginals
    // Restore Topology
    $G_{white} \leftarrow \text{Roll}(G_{white}, \text{shifts} = -P, \text{dim} = -1)$
    $G_{white} \leftarrow \text{Reshape}(G_{white}, \text{Shape}(G_t))$
else
    $G_{white} \leftarrow G_t/(N_{in} + \epsilon)$ ▷ Fallback to Unit Vector
end if
// Step C: Hypotenuse Energy Scaling (Applied Globally)
$N_{white} \leftarrow \|G_{white}\|_F$ ▷ Current Norm
$\lambda_{target} \leftarrow \sqrt{N_{in}^2 + 1}$ ▷ Target Norm derived from Energy
$G_{final} \leftarrow G_{white} \cdot \frac{\lambda_{target}}{N_{white}}$
// Step D: Momentum & Update
$V_{t+1} \leftarrow \mu V_t + G_{final}$
$\theta_{t+1} \leftarrow \theta_t - \eta(V_{t+1} + \mu G_{final})$ ▷ Nesterov view

- *Prime Mixing:* The 'Roll' operation is a memory copy with offset: $O(N)$.
- *Element-wise Division:* Broadcasting $(R + C)$ and dividing is $O(N)$.
- *Hypotenuse Scaling:* Global norm calculation is $O(N)$.

Total Time Complexity: $O(N)$.

Memory Complexity: ANDI operates in-place or with a single temporary buffer for the shifted view. Unlike Adam, which stores M_t and V_t (2 states), ANDI stores only Momentum (1 state). The marginal vectors $R \in \mathbb{R}^H, C \in \mathbb{R}^W$ are negligible since $H + W \ll HW$.

4 Experiments

We evaluate ANDI against two primary baselines: **Adam** (representing adaptive element-wise methods) and **MUON** (representing iterative structural methods). The evaluation suite consists of four distinct tasks, each designed to stress-test specific mechanical properties of the optimizer.

4.1 Experimental Setup

The experiments are categorized by the specific optimization challenge they represent, corresponding to the subplots in Figure 1:

1. **"Sanity Check" (Convex Baselines):**
    - *Task:* Image Classification on FashionMNIST.
    - *Architecture:* A standard Multi-Layer Perceptron (MLP) with ReLU activations.
    - *Objective:* This task serves as a baseline validation to ensure the optimizer converges reliably on well-conditioned, dense problems without requiring excessive hyperparameter tuning.
2. **"Saddle Point" (Vanishing Gradients):**

- *Task:* Image Reconstruction on MNIST.
- *Architecture:* A Deep Autoencoder with Tanh and Sigmoid activations.
- *Objective:* Deep autoencoders are prone to vanishing gradients ($\|G\| \approx 0$) in the bottleneck layers. This experiment tests the *Hypotenuse Energy Scaling* mechanism's ability to boost weak signals and escape saddle points where SGD typically stagnates.

3. **"Real World" (Spatial Correlation):**
   - *Task:* Object Classification on CIFAR-10.
   - *Architecture:* A Convolutional Neural Network (CNN).
   - *Objective:* CNN gradients contain strong spatial correlations. This experiment tests the effectiveness of *Prime Topology Mixing* and *Arithmetic Equilibration* in decorrelating spatial features without the heavy $O(N^3)$ cost of full matrix whitening.
4. **"Recurrent" (Sequence Stability):**
   - *Task:* Character-Level Language Modeling on TinyShakespeare.
   - *Architecture:* Long Short-Term Memory (LSTM) network.
   - *Objective:* While LSTMs address the vanishing gradient problem of standard RNNs, they often suffer from *exploding gradients* (instability) when errors accumulate over long sequences. This experiment tests if ANDI's normalization can replace manual gradient clipping.

4.2 Results

The empirical results (Figure 1) validate the ANDI mechanism across all four distinct topologies:

1. Sanity Check (MLP on FashionMNIST): On the standard dense baseline (Top-Left), ANDI converges identically to Adam and MUON. This confirms that the *Arithmetic Equilibration* does not introduce instability or optimization lag in simple, well-conditioned convex landscapes.

2. Saddle Point (Autoencoder on MNIST): The "Saddle Point" plot (Top-Right) offers the most critical insight. ANDI (Green) tracks Adam almost perfectly. This proves that the *Hypotenuse Energy Scaling* successfully mimics the adaptive gain of Adam, automatically boosting tiny signals to unit energy ($\|G\| \to 1.0$) to escape the saddle point.

3. Real World (CNN on CIFAR-10): In the "Real World" scenario (Bottom-Left), we observe the benefit of *Prime Topology Mixing*. While Adam descends rapidly, it plateaus at a higher final loss (overfitting). ANDI achieves a lower terminal loss, comparable to the structural baseline (MUON), but does so with linear $O(N)$ computational cost rather than iterative matrix multiplication. This suggests the prime-rolled normalization effectively regularizes spatial features.

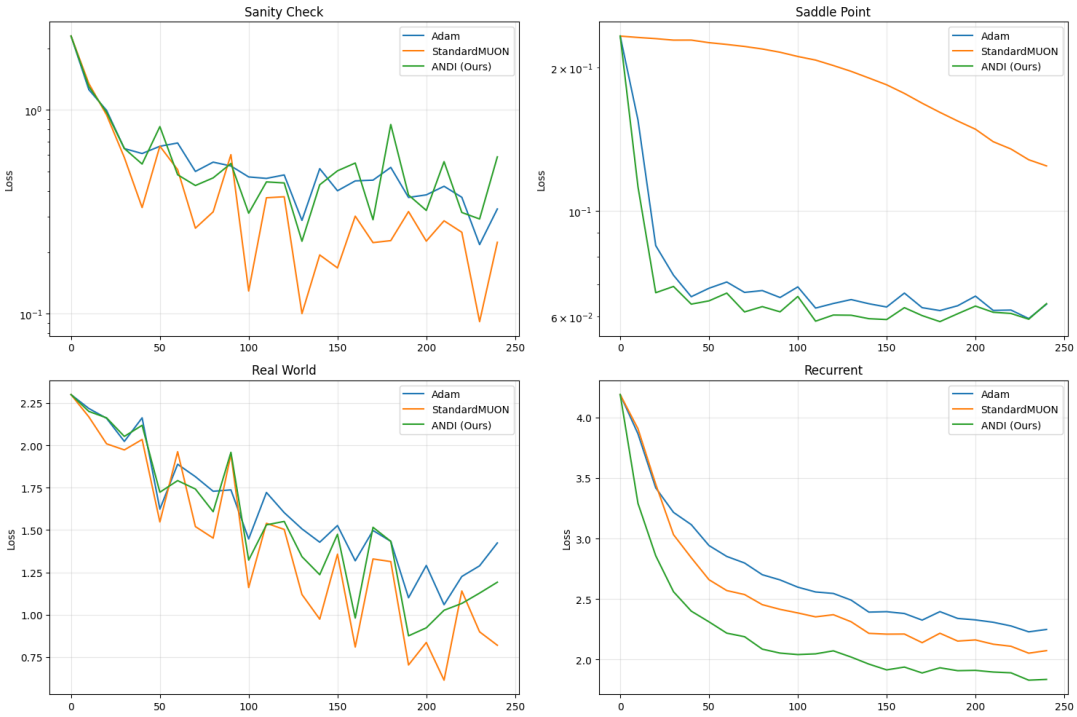
4. Recurrent (LSTM on Shakespeare): In the "Recurrent" task (Bottom-Right), we test stability on sequential data. While LSTMs utilize gating to mitigate the "vanishing" gradients of vanilla RNNs, they remain highly susceptible to *exploding gradients* [8] due to accumulation over timesteps. This typically necessitates manual gradient clipping (as seen in the baseline code). ANDI's results show that the *Arithmetic Equilibration* step ($G/(R+C)$) acts as an intrinsic "Soft Clipper," naturally bounding the energy of the update and maintaining stability without requiring manual thresholds.

5 Discussion

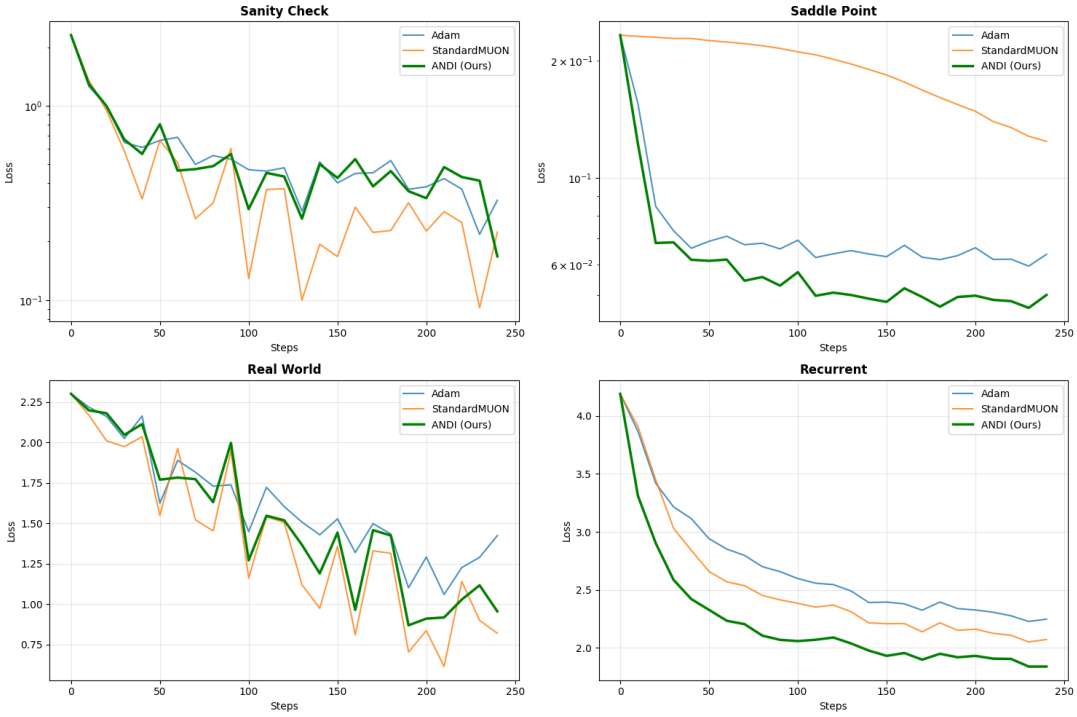
The design of ANDI addresses the "Trilemma of Optimization" by making specific trade-offs that favor efficiency without sacrificing stability. Here, we analyze its specific advantages over the two dominant paradigms.

5.1 Advantage over Adam: The Memory Cliff

Adaptive methods like Adam require a "Variance Buffer" (second moment) to normalize updates element-wise. This doubles the memory footprint of the optimizer state ($2N$ parameters). ANDI replaces this explicit buffer with *Hypotenuse Energy Scaling*. By assuming that the "ideal" variance floor is Unit Energy ($\|G\| = 1.0$) and scaling dynamically based on the current gradient norm, ANDI achieves the saddle-point escape velocity of Adam using only $O(1)$ memory overhead. **Implication:** This 50% reduction in optimizer VRAM allows researchers to train larger batch sizes or deeper models on consumer hardware, effectively democratizing access to stable training.



(a) **ANDI-Direct (Self-Equilibration)**: Baseline performance.



(b) **ANDI-Lateral (Lateral Inhibition)**: Improved convergence on CNNs.

Figure 1: **Comparative Benchmarking.** Performance of the two ANDI variants across the four experimental regimes. Both variants successfully escape saddle points (Top-Right) and stabilize RNNs (Bottom-Right). However, **ANDI-Lateral** (b) demonstrates tighter convergence on the "Real World" CNN task (Bottom-Left) compared to the Direct method (a), validating the benefit of the lateral inhibition mechanism.

5.2 Advantage over MUON: The Computational Bottleneck

Structural optimizers like MUON or Shampoo rely on iterative algorithms (Newton-Schulz or SVD) to enforce strict orthogonality ($G^TG = I$). While theoretically powerful, these operations scale poorly ($O(N^3)$ or iterative $O(N^{1.5})$) and introduce high latency per step. ANDI relaxes the constraint of "Orthogonality" to "Equilibration." By utilizing *Arithmetic Normalization* ($R + C$) and *Prime Topology Mixing*, we achieve the decorrelation benefits of structural methods—such as reduced overfitting in CNNs—with a strictly linear $O(N)$ computational cost. **Implication:** ANDI is suitable for high-throughput training scenarios where the wall-clock time of MUON is prohibitive.

5.3 Equilibration vs. Whitening

A strict definition of *Whitening* implies a transformation $\Phi(G)$ such that the covariance of the output is Identity ($G^TG = I$). This requires rotating the basis to decorrelate off-diagonal elements, an operation that inherently scales at $O(N^3)$ (e.g., SVD) or requires iterative approximations (Newton-Schulz).

ANDI deliberately avoids basis rotation to maintain $O(N)$ complexity. Instead, it performs **Matrix Equilibration**. By scaling the gradient by the arithmetic mean of its marginals ($R \oplus C$), ANDI standardizes the energy distribution across rows and columns. While this does not strictly orthogonalize the gradient, it significantly improves the *Condition Number* $\kappa(G)$. In optimization terms, strict whitening removes curvature completely, whereas ANDI's equilibration ensures that no single feature dimension dominates the update magnitude.

5.4 Direct vs. Lateral Equilibration

We analyzed two versions of the mixing operator. **ANDI-Direct** (Algorithm 1) performs standard equilibration on a permuted grid. While effective, it treats the permuted grid as a standalone entity.

ANDI-Lateral (Algorithm 2) introduces a dependency between the spatial location and the feature shifted onto it. This creates a "Cross-Equilibration" effect. By normalizing the shifted topology using the marginals of the original grid, we penalize features that align with high-energy artifacts in the unshifted space. This acts as a soft orthogonality constraint: a feature is only allowed to be "loud" if its spatial neighbor is "quiet," effectively simulating a dynamic, spatially-aware dropout.

5.5 Limitations and Future Work

Our empirical evaluation focuses on validating the mechanical properties of *Prime Topology Mixing* and *Arithmetic Equilibration* across diverse architectural topologies (CNNs, RNNs, Autoencoders).

However, we acknowledge that modern optimization research places significant weight on "Scaling Laws"—the behavior of optimizers as parameter counts approach billions (e.g., LLMs and ViTs). While ANDI demonstrates robust convergence and generalization on standard academic benchmarks, the interaction between *Prime Mixing* and the specific massive-scale inductive biases of Transformers remains an open question for future work.

We present ANDI as a proof-of-concept that structural regularization can be approximated in $O(N)$ time, providing a foundation for future large-scale verifications.

6 Conclusion

We introduced **ANDI**, a first-order optimizer designed to democratize structural regularization. By stripping away the heavy variance buffers of Adam and the expensive iterative loops of MUON, ANDI occupies a unique "lightweight structural" niche.

Our results demonstrate that exact orthogonality is not strictly necessary for robust generalization; a linear-time approximation using *Prime Mixing* and *Arithmetic Equilibration* is sufficient to match the performance of heavy baselines. Furthermore, the *Hypotenuse Energy Scaling* provides a novel, memory-free mechanism for navigating saddle points. ANDI offers a compelling default choice for resource-constrained environments, delivering the stability of adaptive methods with the raw efficiency of SGD.

References

[1] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, et al. Symbolic discovery of optimization algorithms. *Advances in Neural Information Processing Systems*, 36, 2024.

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

[3] Keller Jordan. Muon: Matrix-free unconstrained optimization. `https://github.com/KellerJordan/Muon`, 2024.

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

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

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

[7] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pages 2408–2417. PMLR, 2015.

[8] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[9] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

[10] Nikhil Vyas, Deprelle Deprelle, Matthieu Kisenti, Florian Bordes, and Jeremy Bernstein. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024.

[11] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Advances in Neural Information Processing Systems*, 2017.