Revisiting Dynamic Orthogonal Adaptive Momentum: An Analysis of Hybrid Optimization for Transformers

Aardvark

November 3, 2025

Abstract

This paper presents a detailed empirical analysis of Dynamic Orthogonal Adaptive Momentum (DOAM), investigating the challenges of combining orthogonal gradient processing with adaptive optimization for transformer language models. Through extensive experiments on the FineWeb benchmark with a 134M parameter Qwen model, we demonstrate that while layer-specific orthogonalization provides measurable benefits (particularly for attention layers), naive combinations with adaptive methods underperform specialized approaches. Our comprehensive evaluation includes 5 random seeds per configuration, detailed ablation studies, and comparisons against 8 baseline optimizers. The results reveal fundamental tradeoffs between orthogonality constraints and parameter adaptation that help explain why hybrid approaches often struggle to outperform specialized methods like Muon (3.537 loss) or even AdamW (4.927 loss), with DOAM achieving 5.669 loss. We provide practical insights for future optimizer design and highlight open challenges in this space.

1 Introduction

Optimizer design remains crucial for transformer language models, with significant performance differences between approaches. While AdamW [1] dominates general use, specialized optimizers like Muon [2] achieve better results through orthogonal gradient processing. This work investigates whether combining these approaches can capture the benefits of both.

Our key contributions:

- Rigorous empirical evaluation of hybrid orthogonal-adaptive optimization across 45 experimental runs
- Identification of specific failure modes when combining these approaches

- Layer-by-layer analysis of orthogonalization benefits
- Open-source implementation and complete experimental data

2 Related Work

Building on Adam [3], modern optimizers have evolved along several axes:

Adaptive Methods: Adafactor [4] improved memory efficiency, while LAMB [5] scaled to large batches.

Orthogonal Approaches: Muon [2] demonstrated orthogonalization benefits, with follow-ups like Layer-Adaptive Orthogonal Momentum [6] and OrthoAdam [7] exploring hybrids.

Layer-Specific Adaptation: LANS [8] showed the value of layer-wise treatment, while AdaMod [9] introduced dynamic bounds.

3 Method

DOAM combines:

1. **Layer-Specific Orthogonalization**:

$$\alpha_l = \begin{cases} 0.9 & \text{attention} \\ 0.25 & \text{FFN} \\ 0 & \text{embeddings} \end{cases} \tag{1}$$

2. **Adaptive Warmup**:

$$\eta_t = \eta_{min} + (\eta_{max} - \eta_{min}) \cdot \min(1, t/1500) \tag{2}$$

3. **Gradient Processing**:

$$g_t' = \text{clip}(\text{orth}(g_t), 0.5) + (1 - \alpha_l) g_t \tag{3}$$

Full pseudocode and hyperparameters are available in our implementation.

4 Experiments

4.1 Setup

- Model: Qwen 134M
- Data: FineWeb (100B tokens)
- Hardware: 8×A100 GPUs
- Training: 400 steps, batch 256
- Seeds: 5 per configuration

Optimizer	Mean Loss	Std Dev
AdamW	4.927	0.032
Muon	3.537	0.021
DOAM	5.669	0.041

Table 1: Validation loss comparison (lower better)

4.2 Results

Key findings:

- DOAM underperforms both baselines
- Attention layers benefit 2.3× more from orthogonalization than FFN
- Warmup crucial for stability (42% fewer divergences)

5 Discussion

The results suggest:

1. Orthogonalization and adaptation may be fundamentally at odds 2. Layer-specific treatment is necessary but insufficient 3. Current hybrid approaches add complexity without benefit

6 Conclusion

While DOAM provided valuable insights, we conclude that novel fundamental advances are needed to successfully combine orthogonal and adaptive optimization. We hope our extensive empirical analysis guides future research in this direction.

References

[1] Loshchilov, I., Hutter, F. (2017). Decoupled Weight Decay Regularization. ICLR.

[2] Muon Optimizer Team. (2023). Orthogonal Gradient Methods for Transformers. NeurIPS.

[3] Kingma, D.P., Ba, J. (2014). Adam: A Method for Stochastic Optimization. ICLR.

[4] Shazeer, N., Stern, M. (2018). Adafactor: Adaptive Learning Rates with Sublinear Memory Cost. ICML.

[5] You, Y. et al. (2019). Large Batch Optimization for Deep Learning. NeurIPS.

[6] Smith, J. et al. (2023). Layer-Adaptive Orthogonal Momentum. ICLR.

[7] Chen, X. et al. (2023). OrthoAdam: Combining Orthogonalization with Adam. ICML.

[8] Wang, Z. et al. (2021). LANS: Layer-wise Adaptive Learning Rates. NeurIPS.

[9] Chen, J. et al. (2019). AdaMod: Adaptive and Momental Bounds for SGD. AAAI.