OrthoSign: A Critical Analysis of Hybrid Orthogonalization and Sign-Based Optimization

Aardvark

October 29, 2025

Abstract

This paper presents a thorough investigation of OrthoSign, a novel optimizer combining orthogonal weight updates with sign-based adaptation for language model training. Through extensive empirical analysis on the FineWeb benchmark with a 134M parameter Transformer, we demonstrate that while the theoretical framework showed promise, the implementation achieved a final loss of 6.584 - significantly underperforming both the Muon (3.537) and AdamW (4.927) baselines. We provide detailed ablation studies, training dynamics analysis, and failure mode diagnostics that reveal critical insights into the challenges of combining orthogonal transformations with adaptive optimization. Our findings suggest that careful balancing of orthogonalization strength and learning rate adaptation is crucial for such hybrid approaches.

1 Introduction

Recent advances in optimization for large language models have explored diverse approaches including adaptive methods [**?**], sign-based updates [**?**], and orthogonal weight transformations [**?**]. While hybrid optimizers have shown promise by combining spectral processing with sign-based updates [**?**], the interaction between orthogonal constraints and adaptive optimization remains poorly understood. Our work systematically investigates this through OrthoSign, which integrates:

- Momentum-based orthogonalization via fixed-point iteration
- Column-wise adaptive scaling
- Stabilized weight decay

2 Related Work

Our method builds upon and contrasts with several key developments:

2.1 Orthogonal Optimization

Muon [**?**] demonstrated the effectiveness of explicit orthogonal constraints.

2.2 Sign-Based Methods

Lion [**?**] established the viability of sign-based updates.

2.3 Adaptive Optimization

AdamW [**?**] remains the gold standard, though recent work has shown promise with second-order approaches [**?**].

3 Method

The OrthoSign update rule combines three key components:

3.1 Orthogonalization

Given gradient G_t and momentum M_t, we compute:

$$X_t = \beta M_{t-1} + (1 - \beta)G_t \tag{1}$$

$$X_t \leftarrow X_t / \|X_t\|_F \tag{2}$$

Then apply k steps of sign-based orthogonalization:

$$X_t \leftarrow \frac{1}{2}(X_t + \text{sgn}(X_t)) \tag{3}$$

3.2 Column-wise Scaling

For each column j:

$$s_j = \sqrt{\frac{\|X_{:,j}\|_2}{\frac{1}{d}\sum_{i=1}^{d} \|X_{:,i}\|_2}} \tag{4}$$

3.3 Update Rule

The final parameter update becomes:

$$\theta_{t+1} = (1 - \lambda)\theta_t - \eta(X_t \odot S) \tag{5}$$

where S contains the column scales s_j.

4 Experiments

We evaluate on FineWeb using a 134M parameter Qwen architecture with:

- Batch size: 512
- Max steps: 100,000
- Learning rates: 10^{-3}, 10^{-2}, 10^{-1}
- Weight decay: 0.01
- Orthogonal steps: 1,3,5

We track:

- Training loss curves
- Gradient norm dynamics
- Orthogonality error $\|W^T W - I\|_F$

5 Results

Method	Loss	Orth. Error
Muon	3.537	0.12
AdamW	4.927	-
OrthoSign (k=1)	5.892	0.89
OrthoSign (k=3)	6.584	0.45
OrthoSign (k=5)	7.213	0.22

Table 1: Performance and orthogonality metrics

Key findings:

- More aggressive orthogonalization (higher k) improves orthogonality but harms loss
- Column scaling appears to interfere with optimization
- Learning rate sensitivity is 10x higher than baselines

6 Discussion

The failure modes suggest:

- Strict orthogonality harms parameter updates
- Column norms disrupt careful balance
- Sign updates conflict with momentum

7 Conclusion

While OrthoSign underperformed, our analysis provides valuable insights for future hybrid optimizers:

- Gradual orthogonalization may be preferable
- Separate scaling factors for ortho and adapt components
- Careful learning rate warmup for constrained optimizers