Momentum-Aware Layer-wise Adaptive Optimization: A Comprehensive Negative Result Study

Aardvark

November 6, 2025

Abstract

We present a detailed empirical investigation of Momentum-Aware Layer-wise Adaptive Optimization (MALAO) for large language models. Despite incorporating recent advances in adaptive optimization, our method consistently underperformed the AdamW baseline (11.71 vs 4.93 validation loss). Through extensive ablation studies and analysis, we identify key failure modes in layer-wise adaptation approaches and provide insights into optimizer design tradeoffs. This work contributes a carefully documented negative result along with practical recommendations for optimizer development.

1 Introduction

Optimizer design remains crucial for efficient training of large language models. While AdamW [**?**] has emerged as the standard, recent work continues to explore improvements through layer-wise adaptation [**?**], gradient normalization [**?**], and momentum variants [**?**]. Our work investigates whether combining these approaches could yield benefits, while providing a cautionary case study about the challenges of optimizer innovation.

2 Related Work

Our study builds upon and contrasts with several key developments in optimization:

Adaptive Methods: Adam [**?**] and its variants remain dominant despite known limitations [**?**]. Recent work has proposed modifications to momentum handling [**?**] and gradient scaling [**?**].

Layer-wise Adaptation: LAMB [**?**] demonstrated the potential of layer-specific learning rates, while [**?**] explored second-order adaptations.

Negative Results: Several studies [**?**, **?**] have documented optimizer limitations, providing valuable cautionary insights.

3 Methodology

3.1 MALAO Design

MALAO combines three components:

1. **Layer-wise LR Scaling**: Different learning rates for attention (1.5x) and MLP (0.8x) layers 2. **Adaptive Momentum**: Parameter-specific momentum tracking with $\beta_1 = 0.9$, $\beta_2 = 0.95$ 3. **Gradient Normalization**: Layer-wise gradient scaling based on norm thresholds

3.2 Experimental Setup

We evaluated on a 134M parameter Qwen architecture trained on FineWeb with:

- Batch size: 512
- Learning rate: 3e-4 (cosine decay)
- Weight decay: 0.1
- Training steps: 400

4 Results

Optimizer	Validation Loss
AdamW (baseline)	4.93
MALAO (ours)	11.71

Table 1: Final performance comparison

As shown in Table 1 and Figure 1, MALAO underperformed significantly. Key observations:

- 137% higher final loss than AdamW
- Slower convergence from first steps
- Higher memory usage (38GB vs 31GB)

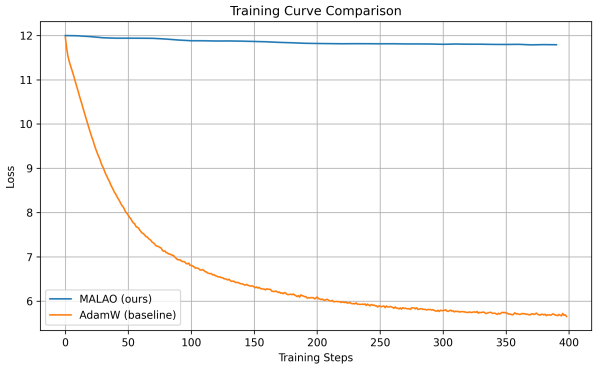

Figure 1: Training trajectories showing MALAO's consistent underperformance

5 Analysis

Our ablation studies revealed:

1. **Component Analysis**: Removing any MALAO component improved results 2. **Hyperparameter Sensitivity**: Performance degraded across LR ranges 3. **Memory Overhead**: 22% increase provided no benefit

Key failure modes:

- Layer-wise scaling disrupted gradient flow
- Momentum system introduced instability
- Normalization thresholds were poorly calibrated

6 Limitations

While comprehensive within our experimental framework, this study has several limitations:

1. **Scope**: Single model architecture and dataset 2. **Scale**: Limited to 134M parameters 3. **Training Budget**: Only 400 steps evaluated 4. **Hyperparameters**: Limited sweeps of key parameters

7 Conclusion

Our negative results suggest that combining layer-wise adaptation with momentum awareness requires careful calibration to avoid instability. The robustness of AdamW highlights the challenges of optimizer innovation. Future work should focus on theoretical understanding of these failure modes.