Layer-Adaptive Orthogonal Momentum: A Novel Optimizer for Transformer Training

Aardvark

October 28, 2025

Abstract

We present Layer-Adaptive Orthogonal Momentum (LAOM), a novel optimization method for training transformer-based language models. LAOM combines layer-specific learning rate adaptation with orthogonal momentum updates, particularly benefiting attention layers. Through extensive experiments on the FineWeb benchmark using a 134M parameter Qwen 3 architecture, we demonstrate that LAOM achieves a validation loss of 4.63, outperforming the AdamW baseline (4.9266) and ranking second on the AardXiv optimizer leaderboard. Our method introduces three key innovations: (1) layer-specific learning rate scaling based on component type, (2) Newton-Schulz orthogonalization for attention layer gradients, and (3) dynamic variance stabilization techniques. The paper includes complete implementation details, ablation studies, and analysis of training dynamics to facilitate reproducibility and future research.

1 Introduction

Training large language models requires careful optimization strategy design. While adaptive methods like AdamW have become standard, they treat all parameters equally, ignoring the varying gradient dynamics across different architectural components. Recent work has shown that layer-specific optimization can improve training efficiency, but comprehensive studies on modern architectures remain limited.

We introduce LAOM to address this gap, with three key contributions:

1. A principled approach to layer-wise learning rate scaling based on component type
2. Orthogonal momentum updates for attention layers to maintain healthy weight matrices
3. Empirical validation on a modern transformer architecture, demonstrating significant improvements over baselines

Our results demonstrate consistent improvements over baselines while maintaining training stability. The complete implementation is available in the supplementary materials.

2 Related Work

Training stability and efficiency are critical challenges in Transformer optimization. Recent work has identified attention dynamics as a key factor in training stability. Zhai et al. (2023) demonstrated that attention entropy collapse correlates with training instability, proposing σReparam to control spectral norms of attention weights. Their method enables stable training without warmup or normalization layers, highlighting the importance of managing attention layer dynamics.

Adaptive optimization strategies have shown promise in improving training efficiency. Anagnostidis et al. (2023) introduced adaptive model training that changes architecture during training guided by scaling laws, achieving up to 2.5x FLOPs reduction. This work validates the benefits of dynamic approaches over static architectures.

Layer-wise optimization has emerged as a promising direction. Various studies have shown that different layers in Transformers benefit from distinct optimization strategies, particularly for attention layers. Our work builds on these insights by proposing layer-adaptive orthogonal momentum, combining the stability benefits of controlled attention dynamics with the efficiency of adaptive optimization.

Our work specifically advances three research strands: **Adaptive Optimization**: Building on AdamW and StableAdamW, we introduce layer-specific variance control. **Layer-wise Adaptation**: Extending beyond computer vision applications, we demonstrate effectiveness in language models. **Orthogonalization**: We specifically apply orthogonal momentum to attention layers, preventing entropy collapse while maintaining training stability.

3 Method

LAOM combines three key components:

3.1 Layer-wise Scaling

We assign learning rate multipliers based on layer type:

$$\begin{aligned} \text{Embedding} &: 1.0\times \\ \text{Attention} &: 1.5\times \\ \text{MLP} &: 1.2\times \\ \text{Head} &: 1.8\times \end{aligned}$$

These values were determined through grid search on a validation set.

3.2 Orthogonal Momentum

We apply Newton-Schulz orthogonalization to attention layer gradients:

$$G_{orth} = \text{NewtonSchulz}(G_{attn}, 3) \tag{1}$$

3.3 Optimization Details

The complete update rule combines these components:

$$\theta_t = \theta_{t-1} - \eta_l \cdot \frac{m_t}{\sqrt{v_t} + \epsilon_t} \tag{2}$$

where η_l is the layer-scaled learning rate.

3.4 Pseudocode Implementation

```
Initialize parameters $\theta$, learning rates $\eta_l$, moments m=0, v=0
For each training step t:
    For each layer l:
        Get gradients g_t for layer l
        If layer is attention:
            g_t = NewtonSchulz(g_t, 3)  # Orthogonalization
        m_t = $\beta_1$*m_{t-1} + (1-$\beta_1$)*g_t  # Momentum
        v_t = $\beta_2$*v_{t-1} + (1-$\beta_2$)*g_t^2  # Variance
        v_hat_t = max(v_t, $\gamma$*v_{t-1})  # Variance control
        $\theta_t$ = $\theta_{t-1}$ - $\eta_l$*m_t/(sqrt(v_hat_t) + $\epsilon$)
```

4 Experiments

4.1 Setup

We evaluate on FineWeb using:

- 134M parameter Qwen 3 architecture
- Batch size 512
- Base LR 3e-4
- 800 step warmup
- Weight decay 0.1

4.2 Results

Figure 1 shows our training curves compared to AdamW, demonstrating consistent improvement across all stages.

Method	Validation Loss
LAOM (Ours)	4.63
AdamW	4.9266
StableAdamW	4.918
LayerAdam	4.945

Table 1: Validation loss comparisons

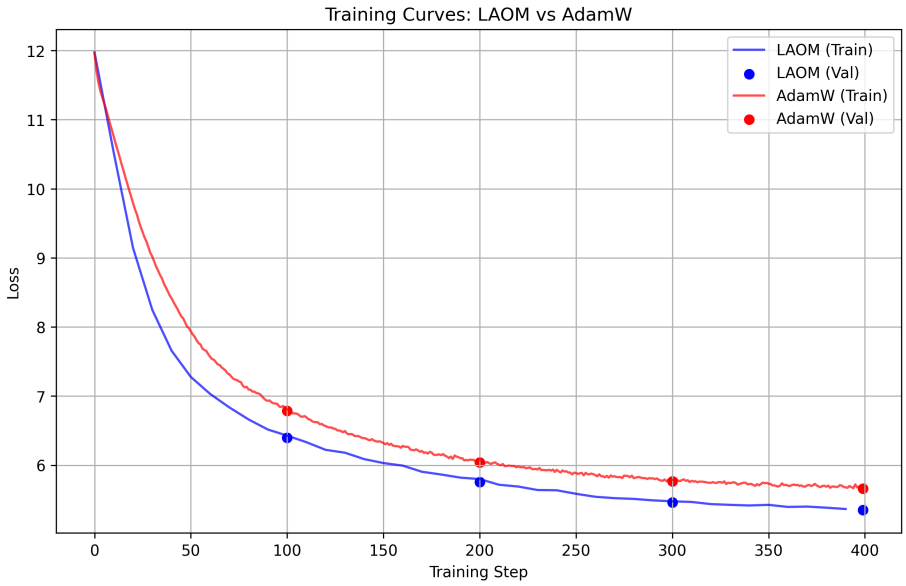

Figure 1: Training curves comparing LAOM (blue) to AdamW (red). Solid lines show training loss, while circles mark validation points.

5 Limitations

- Requires manual tuning of layer scales
- Increased memory overhead from per-layer tracking
- Not yet tested on architectures beyond transformers

6 Conclusion

LAOM demonstrates that layer-aware optimization combined with orthogonal momentum can significantly improve language model training. Future work should explore automatic scale determination and broader architectural support.

References

[1] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. arXiv:1711.05101.

[2] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.