NEWMUON: ADAM'S COMPLEXITY AND MUON'S PERFORMANCE

Hengjie Cao

1 INTRODUCTION

AdamW treats gradients element-wise, updating each coordinate independently and thus ignoring the inter-parameter correlations. In contrast, Muon constrains weight updates to induce nearly constant RMS changes in hidden states, yielding an orthogonalized update matrix that turns the originally highly anisotropic gradient singular-value spectrum into an isotropic one, redistributes update energy from dominant to small spectral components, and thereby achieves faster convergence and improved training stability. However, these benefits come with two key obstacles: high computational cost and poor distributed scalability. Both arise from Muon's reliance on Newton–Schulz (NS) iterations on full update matrices, which operationally entails cubic-time matrix multiplications on globally aggregated gradients, which in turn forces frequent, high-bandwidth synchronization across devices.

This work aims to achieve AdamW-like computational complexity while outperforming Muon in convergence. By analyzing the intrinsic semantic structure of the gradient spectrum, we find that a tiny low-rank head of singular directions primarily learns common features, whereas the remaining spectrum captures fine-grained semantic information that is crucial for loss reduction but whose updates are strongly suppressed by the dominant components. This distinction between spectral roles allows us to isolate and manipulate only this low-rank head at low cost, yielding a gradient singular-value distribution that is more favorable than full isotropy.

Inspired by these insights, we propose *NewMuon*, a novel optimizer designed to align with this semantic hierarchy. Diverging from Muon's approach of forcing a uniform spectrum via expensive global orthogonalization, NewMuon adopts a strategy of **Spectral Smooth Optimization**. We employ an efficient iterative method to explicitly extract the dominant subspace corresponding to the top 1.5% of singular values. By subtracting these high-magnitude components from the gradient, we effectively filter out the "loud" syntactic interference, isolating the spectral tail. This allows the optimization process to focus purely on the fine-grained semantic signals that were previously overshadowed. In addition, we propose a novel adjustment to momentum accumulation: instead of accumulating momentum based on the original gradient, which is highly anisotropic, momentum is now accumulated based on the smoothed spectral gradients. This modification ensures that momentum accumulation aligns with the spectral smoothness, leading to more stable updates. This approach yields two decisive advantages:

1. *Computational Efficiency*: By targeting only the narrow dominant subspace rather than processing the entire matrix, NewMuon circumvents the prohibitive $O(N^3)$ cost of Newton-Schulz iterations, reducing the spectral overhead to a computationally negligible level.

2. *Universal Distributed Scalability*: Since NewMuon operates on a low-rank decomposition of the dominant subspace, it minimizes communication volume. This enables seamless integration with Data Parallelism (DP), Pipeline Parallelism (PP), and Tensor Parallelism (TP), avoiding the high-bandwidth global synchronization required by full-matrix approaches.

3. *Momentum Alignment with Spectral Smoothness*: By adjusting the momentum accumulation to operate on the smoothed spectral gradients, NewMuon ensures that momentum updates are more aligned with the underlying optimization dynamics, further improving convergence stability and efficiency.

2 Muon Optimizer Algorithm

In this section, we describe the Muon optimizer, which improves the stability and convergence speed of gradient-based training algorithms by orthogonalizing the update direction. Muon is built upon the idea of momentum accumulation and orthogonalization via Newton-Schulz (NS) iteration.

2.1 Muon Algorithm with Momentum

Muon incorporates momentum in the update process. The momentum M_t accumulates gradients over time to smooth the update direction. However, unlike standard momentum-based optimizers, Muon performs an additional orthogonalization step to reduce the anisotropy of the gradient updates. The key idea is to apply Newton-Schulz to the momentum matrix M_t, which results in a near-orthogonal update matrix. The Muon algorithm with momentum is described below:

Algorithm 1 Muon Algorithm with Momentum

```
 1: Input: Initial weights $W_0$, learning rate schedule $\{\eta_t\}$, momentum coefficient $\beta \in [0, 1)$, batch
    size $B$
 2: for t = 0, 1, . . . , T - 1 do
 3:     Sample batch $\{\xi_{t,i}\}_{i=1}^{B}$ uniformly
 4:     Compute the gradient: $G_t = \frac{1}{B} \sum_{i=1}^{B} \nabla f(W_t; \xi_{t,i})$
 5:     if t ¿ 0 then
 6:         Update momentum: $M_t = \beta M_{t-1} + (1 - \beta) G_t$
 7:     else
 8:         Initialize momentum: $M_0 = G_0$
 9:     end if
10:     Perform Newton-Schulz on $M_t$: $U_t V_t^\top$ = Newton-Schulz($M_t$)
11:     Update the weights: $W_{t+1} = W_t - \eta_t U_t V_t^\top$
12: end for
```

Suppose the original gradient G_t is of rank r_t and the singular value decomposition (SVD) of G_t is given by: $G_t = U_t S_t V_t^\top$, where $S_t \in \mathbb{R}^{r_t \times r_t}$ is the diagonal matrix of the singular values of G_t, $U_t \in \mathbb{R}^{m \times r_t}$, and $V_t \in \mathbb{R}^{n \times r_t}$ are the left and right singular vector matrices of G_t, respectively.

In Muon, the update matrix O_t is constructed as: $O_t = U_t V_t^\top$, which is the nearest semi-orthogonal matrix to the original G_t. The purpose of this orthogonalization is to ensure that the update matrix has a uniform spectral distribution, mitigating the effect of dominant components in the gradient. By using $U_t V_t^\top$ instead of G_t, Muon improves the stability of the updates and promotes faster convergence.

2.2 Newton-Schulz Iteration for Gradient Update Refinement

While SVD provides an ideal orthogonal update matrix, it is computationally expensive, especially for large matrices. To address this, Muon uses Newton-Schulz iteration (NS iteration) to approximate the matrix square root, which efficiently orthogonalizes the momentum matrix without performing full SVD.

Newton-Schulz iteration is a method for iteratively approximating the matrix square root or inverse square root. It is applied to the momentum matrix M_t to produce a near-orthogonal matrix. The iterative process is described as follows:

Algorithm 2 Newton-Schulz Iteration for Matrix Orthogonalization

```
1: Input: Matrix $M$, number of steps steps $= 5$, small constant $\epsilon = 10^{-7}$
2: Initialize $X_0 = \frac{M}{\|M\|_F + \epsilon}$    (normalize by Frobenius norm)
3: for step = $1, \ldots,$ steps do
4:     $A = X_{step-1} X_{step-1}^\top$
5:     $B = b \cdot A + c \cdot A \cdot A$
6:     $X_{step} = a \cdot X_{step-1} + B \cdot X_{step-1}$
7: end for
8: Output: Matrix $X_{steps}$ as the approximate orthogonal matrix
```

3 Methodology: NewMuon Optimizer

In this section, we introduce the NewMuon optimizer, an enhanced version of the Muon optimizer designed to improve both computational efficiency and scalability in large-scale deep learning tasks. The NewMuon optimizer builds upon the core idea of the Muon algorithm, which orthogonalizes the update matrix to stabilize and accelerate convergence. However, NewMuon introduces several key modifications to improve the efficiency and scalability of the original Muon algorithm, making it more suitable for distributed training and large-scale models.

The NewMuon optimizer extends the core principles of the Muon optimizer by introducing efficient spectral shaping and orthogonal momentum. While Muon uses momentum accumulation and spectral orthogonalization, NewMuon enhances this process by replacing traditional full matrix decompositions (SVD) with a low-rank approximation. This change drastically reduces computational overhead, making NewMuon more efficient and scalable, especially for large-scale models and distributed training environments. Additionally, NewMuon allows for orthogonal momentum, further stabilizing training and improving convergence.

Compared to the original Muon, NewMuon introduces two major improvements:

1. Low-Rank Spectral Shaping: Instead of relying on expensive SVD to orthogonalize the gradient, NewMuon applies a low-rank approximation to the gradient and momentum matrices. This keeps the key spectral components intact while reducing the computational cost of matrix decompositions.

2. Orthogonal Momentum: NewMuon applies spectral shaping to the momentum matrix, ensuring that the updates are isotropic and stable. This step mitigates the risk of large gradient components overwhelming smaller, important updates.

These improvements make NewMuon more suitable for practical, large-scale training settings, offering faster convergence and reduced memory and computational costs.

The NewMuon optimizer is summarized in the following pseudocode:

Algorithm 3 NewMuon Algorithm with Spectral Shaping and Orthogonal Momentum

```
1: Input: Initial weights $W_0$, learning rate schedule $\{\eta_t\}$, momentum coefficient $\beta \in [0, 1)$, batch
   size $B$
2: for $t = 0, 1, \ldots, T-1$ do
3:     Sample batch $\{\xi_{t,i}\}_{i=1}^{B}$ uniformly
4:     Compute the gradient: $G_t = \frac{1}{B} \sum_{i=1}^{B} \nabla f(W_t; \xi_{t,i})$
5:     if t ¿ 0 then
6:         Update momentum: $M_t = \beta M_{t-1} + (1-\beta) G_t$
7:     else
8:         Initialize momentum: $M_0 = G_0$
9:     end if
10:    Apply spectral shaping to $M_t$: $M_t = \text{SpectralShaping}(M_t)$
11:    Update the weights: $W_{t+1} = W_t - \eta_t M_t$
12: end for
```

In NewMuon, the spectral shaping procedure replaces the need for full matrix decompositions, offering an efficient low-rank approximation. The key idea is to reshape the gradient matrix in such a way that its essential components are preserved, but its spectral energy is redistributed in a more balanced manner. This step reduces computational complexity while maintaining the key information in the gradient.

The spectral shaping procedure works as follows:$G_{\text{shaped}} = (U \cdot S_{\text{new}} \cdot V^{\top})$

The low-rank approximation ensures that the new gradient matrix has a reduced rank but still captures the essential structure of the original matrix. This is achieved by applying the spectral shaping function as follows:

Algorithm 4 Spectral Shaping Function

```
1: **Input:** Gradient matrix $G$, low-rank approximation ratio $k_{\text{ratio}}$
2: Compute the rank $k$ based on $k_{\text{ratio}}$
3: Perform low-rank SVD on $G$: $(U, S, V) = \text{SVD}(G)$
4: Adjust the singular values to match the RMS of the remaining components
5: Reconstruct the matrix with adjusted singular values
6: **Output:** Shaped gradient matrix $G_{\text{shaped}}$
```

4 Experiment Results

In this section, we present the experimental results of NewMuon and compare it against the baseline optimizers, including Adam and Muon. The experiments are conducted on several downstream tasks, and we evaluate both the loss and task performance.

Loss Comparison We first compare the loss values of NewMuon with and without momentum smoothing.

1. Without momentum smoothing: NewMuon achieves performance similar to Muon, with slightly better results, and outperforms Adam.

2. With momentum smoothing: The loss increases slightly but still remains better than Adam.

Downstream Task Performance We further evaluate the performance of NewMuon on several downstream tasks, comparing it with Adam and Muon. The tasks include ARC-C, ARC-E, BoolQ, HellaSwag, PIQA, RACE, and SIQA.

1. NewMuon outperforms both Adam and Muon in terms of task performance.

2. NewMuon_SM (with momentum smoothing) achieves the best performance across all tasks, significantly outperforming the other optimizers.

The results for the downstream tasks are summarized in the table below:

Table 1: Loss and performance on downstream tasks. Best performing method is NewMuon_SM, with significant improvement over other optimizers.

Method	ARC-C	ARC-E	BoolQ	HellaSwag	PIQA	RACE	SIQA	Avg.
Adam	0.197952	0.262205	0.378287	0.258913	0.525571	0.207540	0.340328	3.494
Muon	0.202218	0.268939	0.378287	0.254929	0.524483	0.208958	0.336745	3.362
NewMuon	0.210751	0.251684	0.378287	0.257817	0.533188	0.211188	0.330092	3.363
NewMuon_SM	0.225256	0.236953	0.621713	0.253734	0.541349	0.211390	0.333675	3.41