# ORTHOGONAL WITNESS CONTROL FOR MUON OPTIMIZATION VIA SIGMOID SPECTRAL RESHAPING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Matrix-valued optimizers such as Muon exploit the spectral structure of neural network updates through Newton–Schulz orthogonalization, but their near-flattening of the singular spectrum discards relative magnitude information across gradient modes. We introduce *Soren* (**S**pectral **O**rthogonal **Re**shapi**n**g), a matrix-valued optimizer that preserves the singular subspaces of the gradient while applying a bounded, monotone sigmoid transformation to its singular values. This smoothly compresses dominant modes without fully flattening the spectrum. We interpret Soren as a positive-definite preconditioned gradient method and establish convergence guarantees under relative smoothness and metric Polyak–Łojasiewicz geometry. To avoid explicit singular value decomposition, we further develop a finite-depth Soft Newton–Schulz (SNS) polynomial realization of the sigmoid spectral map and characterize how its spectral approximation affects the induced convergence geometry. Experiments across LLM pre-training, supervised fine-tuning, and direct preference optimization demonstrate the effectiveness and robustness of Soren against established optimizers.

## 1 INTRODUCTION

Optimization methods for deep neural networks increasingly exploit the matrix structure of neural network parameters rather than treating all parameters as unstructured vectors. Muon (Jordan et al., 2024) has recently emerged as a matrix-valued optimizer for hidden layers. It first constructs a momentum-based update following SGD (Robbins & Monro, 1951) with momentum (Polyak, 1964), and then applies Newton–Schulz iterations to approximately orthogonalize the update. This preserves the singular subspaces of the update while flattening its active singular spectrum.

This spectral transformation can mitigate the dominance of a small number of singular modes and produce more balanced update directions. However, flattening the spectrum also discards relative magnitude information across gradient modes, pushing weak and strong active directions toward a similar scale. This motivates a softer form of spectral transformation that can control spectral imbalance while retaining information about the relative strengths of different modes.

We introduce *Soren* (**S**pectral **O**rthogonal **Re**shapi**n**g), a sigmoid-based matrix-valued optimizer that preserves the singular subspaces of the gradient while smoothly reshaping their magnitudes. The bounded and monotone sigmoid transformation compresses dominant singular modes while preserving their ordering and the non-uniformity of the spectrum. In contrast to SGD-type updates, which retain the original spectral imbalance, and Muon, which strongly flattens the active spectrum, Soren provides controlled spectral shaping while retaining relative spectral structure.

We study Soren from both optimization and computational perspectives. The sigmoid spectral transformation admits a positive-definite preconditioning interpretation, enabling convergence analysis under relative smoothness and metric Polyak–Łojasiewicz geometry. To avoid the explicit SVD required by the exact spectral map, we further develop a finite-depth Soft Newton–Schulz (SNS) polynomial realization. We also establish a convergence guarantee for the resulting finite-step SNS update. More specifically, our contributions are as follows:

- We introduce *Soren*, a sigmoid-based spectral reshaping optimizer that preserves the singular subspaces of matrix-valued gradients while smoothly compressing dominant singular modes without fully flattening the spectrum.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- We establish convergence guarantees for Soren under relative smoothness and metric Polyak–Łojasiewicz geometry. By interpreting the sigmoid spectral transformation as a positive-definite preconditioner, our analysis yields a product-form convergence bound for iteration-dependent metrics and a linear convergence rate under uniform metric bounds.
- We develop a finite-depth Soft Newton–Schulz polynomial realization of the sigmoid spectral map that avoids explicit SVD computation, and establish a corresponding convergence guarantee for the finite-step SNS update.
- We evaluate Soren across LLM pre-training, supervised fine-tuning, and direct preference optimization, demonstrating consistent improvements over established optimizers across diverse training settings.

## 2 Related work

**Classical and adaptive optimization.** First-order optimization remains the foundation of modern neural network training. Stochastic gradient descent (SGD) provides the canonical baseline for large-scale optimization (Robbins & Monro, 1951), with momentum methods accelerating gradient descent by accumulating past gradients (Polyak, 1964), while Nesterov acceleration introduces a look-ahead correction that improves the behavior of momentum-based updates (Nesterov, 1983). In deep learning, the choice of momentum and initialization has been shown to strongly affect trainability and convergence (Sutskever et al., 2013). Adaptive optimizers further improve robustness by rescaling updates using moment estimates, with Adam (Kingma & Ba, 2015) and AdamW (Loshchilov & Hutter, 2019b) becoming standard choices for large-scale training and fine-tuning. These methods, however, mainly treat parameters as unstructured vectors, leaving matrix structure in neural network layers largely unused.

**Matrix-valued spectral optimization.** Recent work has increasingly exploited the matrix structure of neural network parameters for optimization. Muon (Jordan et al., 2024) applies Newton–Schulz iterations to momentum-based updates of matrix-valued parameters, effectively approximating their polar factors and flattening the nonzero singular values. HTMuon (Pang et al., 2026) extends this framework by reshaping the singular spectrum through a heavy-tailed transformation. More recently, (Qi et al., 2026) provide a systematic spectral analysis of Muon and investigate a broader family of spectral transformations. Related to polar decomposition and Newton–Schulz orthogonalization (Björck & Bowie, 1971), Soren instead applies a smooth sigmoid-based spectral reshaping that compresses dominant singular modes while retaining the relative ordering and non-uniformity of the spectrum.

**Theoretical foundations for spectral preconditioning.** The spectral interpretation of Muon connects naturally to curvature-aware and preconditioned optimization. Local quadratic models and Gauss–Newton approximations provide standard frameworks for analyzing optimization dynamics near a current iterate (Martens, 2010). Relative smoothness (Bauschke et al., 2017; Lu et al., 2018) and Polyak–Łojasiewicz (PL) geometry (Polyak, 1963; Karimi et al., 2016) provide a complementary framework for establishing convergence beyond the standard strongly convex setting. Building on these tools, we interpret Soren's sigmoid spectral reshaping as a controlled positive-definite preconditioning of the gradient and analyze both its exact and finite-step polynomial realizations under the corresponding metric conditions.

## 3 Preliminaries

### 3.1 Muon Optimizer

At iteration $t$, given the current weight $\boldsymbol{W}_{t-1}$, let the full gradient be $\boldsymbol{G}_t = \nabla \mathcal{L}(\boldsymbol{W}_{t-1})$. Following the spectral discussion in the introduction, we first present Muon in its clean orthogonal-gradient form:

$$\begin{aligned} \boldsymbol{O}_t &= \text{Newton-Schulz}\left(\boldsymbol{G}_t\right), \\ \boldsymbol{W}_t &= \boldsymbol{W}_{t-1} - \eta \boldsymbol{O}_t. \end{aligned} \tag{1}$$

Here $\eta$ denotes the learning rate. The Newton–Schulz iteration approximates the polar factor of its input (Jordan et al., 2024). In particular, if $\boldsymbol{G}_t = U\Sigma V^\top$ is the singular value decomposition of $\boldsymbol{G}_t$, then the corresponding ideal polar update is $\left(\boldsymbol{G}_t\boldsymbol{G}_t^\top\right)^{-1/2}\boldsymbol{G}_t = UV^\top$. Therefore, the Muon step preserves the singular subspaces of the gradient while replacing its singular spectrum by a flat one.

The expression above is the form most convenient for analysis. In practical implementations, however, Muon is commonly applied to a momentum-based update rather than the raw gradient itself.

### 3.2 Momentum-Induced Gradient Updates

Momentum-based gradient descent (Polyak, 1964) accumulates past gradients through the state

$$\begin{aligned}\boldsymbol{M}_t &= \mu\boldsymbol{M}_{t-1} + (1-\mu)\boldsymbol{G}_t,\\ \boldsymbol{W}_t &= \boldsymbol{W}_{t-1} - \eta\boldsymbol{M}_t,\end{aligned} \tag{2}$$

where $\eta$ is the learning rate and $\mu$ is the momentum coefficient. The momentum buffer is updated using linear interpolation. The state $\boldsymbol{M}_t$ smooths stochastic fluctuations while retaining information from previous iterations.

Nesterov accelerated gradient (NAG) (Nesterov, 1983) introduces a look-ahead correction to the momentum direction:

$$\boldsymbol{N}_t = (1-\mu)\boldsymbol{G}_t + \mu\boldsymbol{M}_t. \tag{3}$$

Practical Muon implementations often apply Newton–Schulz orthogonalization to this Nesterov-style update (Jordan et al., 2024):

$$\begin{aligned}\boldsymbol{O}_t &= \text{Newton-Schulz}\left(\boldsymbol{N}_t\right),\\ \boldsymbol{W}_t &= \boldsymbol{W}_{t-1} - \eta\boldsymbol{O}_t.\end{aligned} \tag{4}$$

It combines the current gradient with the accumulated momentum before the final step is formed. Thus, $\boldsymbol{G}_t$, $\boldsymbol{M}_t$, and $\boldsymbol{N}_t$ denote different update quantities, while Muon refers to the spectral transformation applied before updating the weights.

## 4 Theoretical Analysis of Sigmoid Spectral Reshaping

### 4.1 A Geometric Perspective on Spectral Updates

Let $G_t = \nabla\mathcal{L}(W_{t-1}) = U\Sigma V^\top = \sum_i \sigma_i u_i v_i^\top$ be the singular value decomposition of the gradient, and define $A_i = u_i v_i^\top$. These rank-one matrices are orthonormal under the Frobenius inner product: $\langle A_i, A_j\rangle_F = \mathrm{Tr}(A_i^\top A_j) = \delta_{ij}$, where $\delta_{ij}$ denotes the Kronecker delta. Thus, the gradient admits an orthogonal decomposition into spectral directions $A_i$, with $\sigma_i$ specifying the magnitude of the gradient along each direction. For an update $O_t$ that preserves these spectral directions, we can similarly write $O_t = \sum_i \alpha_i A_i$, where $\alpha_i$ denotes the magnitude of the update along the $i$-th spectral direction. With $W_t = W_{t-1} - \eta O_t$, the resulting parameter displacement is

$$\Delta W_t = W_t - W_{t-1} = -\eta O_t = -\eta\sum_i \alpha_i A_i.$$

Hence, from a spectral perspective, an optimizer can be viewed as transforming the singular-value coefficients $\{\sigma_i\}$ of the gradient into the update coefficients $\{\alpha_i\}$, while preserving the underlying spectral directions $\{A_i\}$. Under a first-order Taylor approximation around $W_{t-1}$, the corresponding loss change is:

$$\mathcal{L}(W_t) - \mathcal{L}(W_{t-1}) \approx \langle G_t, \Delta W_t\rangle_F = -\eta\,\langle G_t, O_t\rangle_F\,.$$

Substituting the spectral decompositions of $G_t$ and $O_t$, and using the orthonormality of the spectral directions $\{A_i\}$, we obtain:

$$\langle G_t, O_t\rangle_F = \left\langle \sum_i \sigma_i A_i, \sum_j \alpha_j A_j \right\rangle_F = \sum_i\sum_j \sigma_i\alpha_j\,\langle A_i, A_j\rangle_F = \sum_i \sigma_i\alpha_i.$$

Hence, the first-order loss change is governed by the interaction between the gradient coefficients $\{\sigma_i\}$ and the update coefficients $\{\alpha_i\}$, with the contribution of each spectral mode to the local descent determined by the product $\sigma_i\alpha_i$.

This spectral view provides a common interpretation of SGD and Muon. SGD sets $\alpha_i = \sigma_i$, giving $\langle G_t, O_t\rangle_F = \sum_i \sigma_i^2$, so dominant modes contribute quadratically to first-order descent. Muon sets $\alpha_i = 1$ on active modes, giving $\langle G_t, O_t\rangle_F = \sum_i \sigma_i$, thereby flattening the relative magnitudes across active modes. In contrast, Soren uses the spectral transformation $\alpha_i = s(\sigma_i)$, where $s(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, giving:

$$O_t = \sum_i s(\sigma_i) A_i, \qquad \langle G_t, O_t\rangle_F = \sum_i \sigma_i s(\sigma_i).$$

Because $s(\cdot)$ is strictly increasing and bounded, Soren preserves the ordering of spectral modes while compressing their relative magnitudes and bounding the update coefficients. We next establish convergence guarantees for this spectral update.

### 4.2 Convergence under Relative Smoothness and PL Geometry

The spectral reshaping operation in Soren admits an equivalent positive-definite preconditioning view, which allows us to analyze convergence under relative smoothness and metric Polyak–Łojasiewicz (PL) geometry. Let $G_t \in \mathbb{R}^{m\times n}$, $m \le n$, have full row rank at every analyzed step, with thin SVD $G_t = U_t\Sigma_t V_t^\top$ and $\Sigma_t = \operatorname{diag}(\sigma_{t,1}, \dots, \sigma_{t,m}) \succ 0$. The spectral map $s(G_t) = U_t s(\Sigma_t) V_t^\top$ induces the following positive-definite preconditioner:

$$P_t = U_t s(\Sigma_t)\Sigma_t^{-1} U_t^\top = U_t \operatorname{diag}\left(\frac{s(\sigma_{t,i})}{\sigma_{t,i}}\right) U_t^\top \succ 0. \tag{5}$$

Thus, $s(G_t) = P_t G_t$. Writing $w_t = \operatorname{vec}(W_t)$, $g_t = \operatorname{vec}(G_t)$, where $\operatorname{vec}(\cdot)$ denotes column-wise vectorization, and defining $\mathcal{P}_t = I_n \otimes P_t$, the update can be written exactly as:

$$w_t = w_{t-1} - \eta_t \mathcal{P}_t g_t. \tag{6}$$

The derivation is given in Appendix A.1. For any positive-definite matrix $M$, define $\|v\|_M^2 = v^\top M v$. We measure parameter displacements in the $\mathcal{P}_t^{-1}$ norm and gradients in the dual $\mathcal{P}_t$ norm. The metric is determined by the gradient at $w_{t-1}$ and held fixed throughout the one-step analysis. Let $\Omega_t$ be a convex region containing the update segment. Assume that $\mathcal{L}$ is twice continuously differentiable on a neighborhood of $\Omega_t$, with full Hessian $H(z) = \nabla^2\mathcal{L}(z)$ and finite optimal value $\mathcal{L}^*$.

With $\mathcal{P}_t$ fixed, define $\widetilde{\beta}_t$ as the smallest relative-smoothness constant:

$$\widetilde{\beta}_t = \inf\left\{\beta > 0 : H(z) \preceq \beta \mathcal{P}_t^{-1} \text{ for all } z \in \Omega_t\right\}, \tag{7}$$

and define $\widetilde{\alpha}_t$ as the largest metric Polyak–Łojasiewicz (PL) constant:

$$\widetilde{\alpha}_t = \inf_{\substack{w\in\Omega_t \\ \mathcal{L}(w)>\mathcal{L}^*}} \frac{\frac{1}{2}\|\nabla\mathcal{L}(w)\|_{\mathcal{P}_t}^2}{\mathcal{L}(w) - \mathcal{L}^*}. \tag{8}$$

Equation 7 characterizes relative smoothness with respect to the quadratic reference induced by $\mathcal{P}_t^{-1}$ (Bauschke et al., 2017; Lu et al., 2018), while Equation 8 defines the corresponding $\mathcal{P}_t$-metric PL constant (Karimi et al., 2016). Together, they determine the per-step contraction factor $1 - \widetilde{\alpha}_t/\widetilde{\beta}_t$. Under the step size specified below, the descent lemma together with the lower bound $\mathcal{L} \ge \mathcal{L}^*$ ensures that $\widetilde{\alpha}_t \le \widetilde{\beta}_t$. The metric formulations and descent lemma are derived in Appendix A.2, with the convergence proof given in Appendix A.3.

**Theorem 1** (Soren convergence)**.** *Under the assumptions above, suppose that the line segment between $w_{t-1}$ and $w_t$ lies in $\Omega_t$ for every iteration $t$, and that $0 < \widetilde{\alpha}_t < \infty$ and $0 < \widetilde{\beta}_t < \infty$. With the step size $\eta_t = 1/\widetilde{\beta}_t$, the exact Soren update in Eq. 6 satisfies, for any integer $T \ge 1$,*

$$\mathcal{L}(w_T) - \mathcal{L}^* \le \left[\prod_{t=1}^{T}\left(1 - \frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t}\right)\right](\mathcal{L}(w_0) - \mathcal{L}^*). \tag{9}$$

*Here, $t$ indexes the optimization iterations and $T$ denotes the number of iterations under consideration. If there exist constants $0 < \overline{\alpha} \le \overline{\beta} < \infty$ such that*

$$\widetilde{\alpha}_t \ge \overline{\alpha}, \qquad \widetilde{\beta}_t \le \overline{\beta}, \qquad \forall t, \tag{10}$$

*then*

$$\mathcal{L}(w_T) - \mathcal{L}^* \le \left(1 - \frac{\overline{\alpha}}{\overline{\beta}}\right)^T (\mathcal{L}(w_0) - \mathcal{L}^*). \tag{11}$$

The product bound allows the metric to vary across iterations, whereas the uniform bounds yield a linear convergence rate. The proof in Appendix A.3 combines the full-Hessian descent lemma with the metric PL condition.

### 4.3 Polynomial Realization via Soft Newton–Schulz Iteration

To avoid an explicit SVD, we use a finite-depth *Soft Newton–Schulz (SNS) iteration*, inspired by Newton–Schulz orthogonalization (Björck & Bowie, 1971). The identity $\mathrm{sigmoid}(x) = \frac{1}{2}(1 + \tanh(x/2))$ motivates two polynomial streams. Initialize $Q_0 = G_t/(\|G_t\|_F + \epsilon)$ and $T_0 = G_t/4$, and apply

$$\Phi(X) = \frac{1}{2}(3I - XX^\top)X. \tag{12}$$

Applying $K$ iterations to $Q$ and two iterations to $T$, we define

$$\widehat{s}(G_t) = \frac{1}{2}(Q_K + T_2). \tag{13}$$

Algorithms 1–2 in Appendix E provide pseudocode, while Figure 3 compares SNS with the sigmoid map. Both streams remain diagonal in the singular-vector basis of $G_t = U_t \Sigma_t V_t^\top$, yielding coefficients $\widehat{s}_{t,i}$. To compare SNS with the exact map at the same gradient iterate, assume

$$|\widehat{s}_{t,i} - s(\sigma_{t,i})| \le \rho_t s(\sigma_{t,i}), \qquad 0 \le \rho_t < 1, \qquad \forall i. \tag{14}$$

Under the full-row-rank assumption of Section 4.2, this ensures a positive-definite SNS preconditioner. Let $\widehat{\alpha}_t$ and $\widehat{\beta}_t$ denote the corresponding SNS metric PL and relative-smoothness constants, respectively, defined as in equation 7–8 using the SNS metric on the same region $\Omega_t$ and with the same $\mathcal{L}^*$. Appendix A.4 gives the explicit definitions and preconditioner bounds.

**Theorem 2** (Finite-step geometry preservation)**.** *Under equation 14 and $0 < \widetilde{\alpha}_t \le \widetilde{\beta}_t < \infty$, the corresponding exact and SNS constants satisfy*

$$\frac{1-\rho_t}{1+\rho_t}\frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t} \le \frac{\widehat{\alpha}_t}{\widehat{\beta}_t} \le \frac{1+\rho_t}{1-\rho_t}\frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t}. \tag{15}$$

Thus, the guaranteed per-step progress retains at least a fraction $(1-\rho_t)/(1+\rho_t)$ of that for the exact map at the same iterate. The individual metric bounds and proof appear in Appendices A.4–A.9. These bounds directly yield the following convergence guarantee for the finite-step SNS update.

**Corollary 1** (SNS convergence)**.** *Under the assumptions of Section 4.2, suppose each SNS update segment lies in $\Omega_t$ and, along the SNS trajectory, Eq. (14) holds with $\rho_t \le \overline{\rho} < 1$, $\widetilde{\alpha}_t \ge \overline{\alpha} > 0$, and $\widetilde{\beta}_t \le \overline{\beta} < \infty$. Then, with $\eta_t = 1/\widehat{\beta}_t$,*

$$\mathcal{L}(w_T) - \mathcal{L}^* \le \left[1 - \frac{1-\overline{\rho}}{1+\overline{\rho}}\frac{\overline{\alpha}}{\overline{\beta}}\right]^T (\mathcal{L}(w_0) - \mathcal{L}^*). \tag{16}$$

The resulting factor $(1-\overline{\rho})/(1+\overline{\rho})$ quantifies the degradation in the guaranteed progress induced by the finite-step approximation. This convergence guarantee is conditional on the spectral-error and uniform metric bounds; the proof is given in Appendix A.10.

## 5 Experiments

We evaluate Soren across pre-training and post-training, including supervised fine-tuning (SFT) (Ouyang et al., 2022) and direct preference optimization (DPO) (Rafailov et al., 2023). We compare against optimizers that are closely related to our work, including AdamW (Loshchilov & Hutter, 2019a), Muon (Jordan et al., 2024), and HTMuon (Pang et al., 2026).

### 5.1 LLM PRE-TRAINING

**Experimental Setup.** We pretrain various LLaMA-family models (LLaMA-60M, LLaMA-135M, LLaMA-350M, LLaMA-1B (Touvron et al., 2023)) on the C4 dataset (Raffel et al., 2020).

Table 1: Comparison of established pretraining optimizers across LLaMA models of varying sizes on the C4 dataset. Lower perplexity indicates better performance. Detailed hyperparameter configurations are reported in Tables 6–8 in Appendix C.

| | **LLaMA-60M** | **LLaMA-135M** | **LLaMA-350M** | **LLaMA-1B** |
|---|---|---|---|---|
| AdamW | 44.54 | 36.96 | 23.49 | 20.82 |
| Muon | 36.86 | 27.42 | 21.73 | 18.41 |
| HTMuon | 35.32 | **26.15** | 22.58 | 18.44 |
| **Soren** | **35.14** | **26.15** | **21.53** | **18.29** |

Table 1 compares Soren with established pre-training optimizers across four LLaMA model scales on the C4 dataset. Soren achieves the lowest perplexity on three of the four model scales and ties for the best result on the remaining scale, demonstrating consistently strong performance across model sizes. Compared with Muon, Soren reduces perplexity by 1.72, 1.57, 0.20, and 0.12 on LLaMA-60M, LLaMA-135M, LLaMA-350M, and LLaMA-1B, respectively. Relative to AdamW, Soren achieves reductions of 9.40, 10.81, 1.96, and 2.53, respectively. Compared with HTMuon, Soren achieves lower perplexity on LLaMA-60M, LLaMA-350M, and LLaMA-1B by 0.18, 1.05, and 0.15, respectively, while matching HTMuon on LLaMA-135M at a perplexity of 26.15. Overall, Soren consistently outperforms AdamW and Muon across all model scales while achieving comparable or better performance than HTMuon, indicating that Soren provides a competitive optimization approach for LLaMA pre-training.

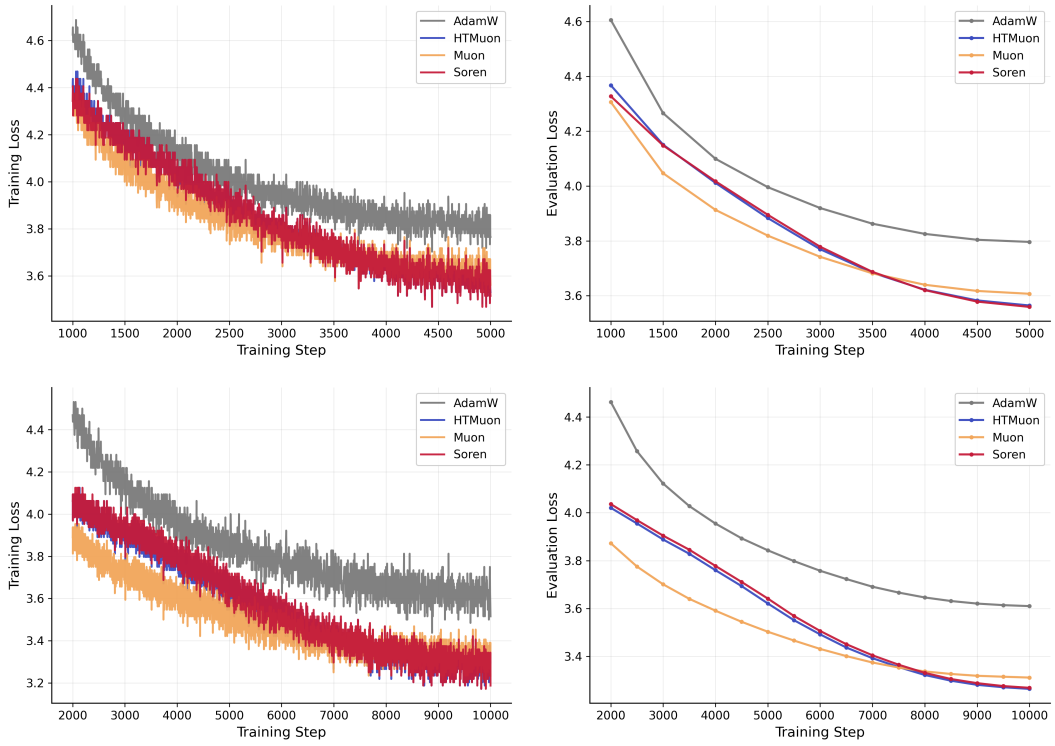



Figure 1: Training and evaluation loss curves for the LLaMA-60M (top row) and LLaMA-135M (bottom row) experiments.

In figure 1, we present the training and evaluation loss curves for LLaMA-60M and LLaMA-135M on the C4 dataset. Soren exhibits convergence behavior closely aligned with HTMuon, particularly

Table 2: Zero-shot evaluation results (↑) on six commonsense reasoning benchmarks for the Qwen3-0.6B model pretrained with different optimizers.

| Optimizer | ARC-c | ARC-e | PIQA | HellaSwag | OBQA | WinoGrande | Avg. |
|---|---|---|---|---|---|---|---|
| AdamW | 23.13 | 25.05 | 50.16 | 25.49 | 24.00 | 48.80 | 32.77 |
| Muon | 23.46 | 26.89 | 50.11 | 26.97 | 24.40 | 50.51 | 33.72 |
| HTMuon | **25.77** | 26.81 | 50.71 | 27.25 | 24.60 | 49.88 | 34.17 |
| Soren | 25.51 | **27.36** | **51.25** | **28.47** | **25.80** | **51.99** | **35.06** |

in the later stages of training, with comparable training and evaluation loss trajectories across both model scales. Compared with Muon, Soren achieves similar or lower losses toward the end of training, while consistently outperforming AdamW in the later stages. Notably, Soren reaches evaluation losses comparable to or slightly lower than HTMuon toward the end of training, consistent with the final perplexity results in Table 1. In figure 4 in appendix B, we provide the training loss curves for LLaMA-350M and LLaMA-1B. These results indicate that Soren exhibits stable convergence behavior and competitive optimization performance during pre-training.

**Downstream Tasks** To further assess the applicability of Soren to broader NLP tasks, we evaluate a Qwen3-0.6B (Yang et al., 2025) model pre-trained with Soren on six commonsense reasoning benchmarks using the lm-eval-harness framework. For all tasks, we use the default prompts and conduct evaluation in a zero-shot setting.

As shown in Table 2, Soren achieves the best performance on five out of six commonsense reasoning benchmarks and obtains the highest average score of $35.06$ across all tasks. Compared with HTMuon, which achieves the second-highest average score of $34.17$, Soren improves the overall performance by $0.89$ points. Soren also consistently outperforms AdamW and Muon across all six benchmarks, while achieving particularly strong gains on ARC-e, PIQA, HellaSwag, OBQA, and WinoGrande. These results highlight the effectiveness of Soren for pre-training language models and its ability to improve performance across a diverse set of commonsense reasoning tasks.

### 5.2 LLM Supervised Fine-tuning

**Experimental Setup.** We conduct supervised fine-tuning experiments on Qwen2.5-3B (Qwen et al., 2025) using the UltraChat 200k dataset (Ding et al., 2023). We use the train_sft split for training and follow the standard data format provided by the dataset. To assess the general capabilities of the resulting models, we follow the Hugging Face Open LLM Leaderboard evaluation protocol (Beeching et al., 2023) and use the Language Model Evaluation Harness (Gao et al., 2021). We evaluate performance on six benchmarks spanning commonsense reasoning, multi-task understanding, factuality, and mathematical reasoning: ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2020), MMLU (Hendrycks et al., 2021), TruthfulQA (Lin et al., 2022), and GSM8K (Cobbe et al., 2021).

Table 3: Evaluation results of different optimizers for SFT on Qwen2.5-3B. Detailed hyperparameter configurations are reported in Table 9 in Appendix C.

| **Method** | **HellaSwag** | **ARC** | **MMLU** | **TruthfulQA** | **Winogrande** | **GSM8K** | **Average** |
|---|---|---|---|---|---|---|---|
| AdamW | $73.08_{\pm 0.44}$ | $55.97_{\pm 1.45}$ | $61.79_{\pm 0.39}$ | $46.75_{\pm 1.48}$ | $68.90_{\pm 1.30}$ | $64.44_{\pm 1.32}$ | 61.82 |
| Muon | $73.72_{\pm 0.44}$ | $56.14_{\pm 1.45}$ | $62.83_{\pm 0.39}$ | $47.44_{\pm 1.49}$ | $68.90_{\pm 1.30}$ | $\underline{69.83}_{\pm 1.26}$ | 63.14 |
| HTMuon | $\underline{73.75}_{\pm 0.44}$ | $\underline{56.31}_{\pm 1.45}$ | $\underline{62.93}_{\pm 0.39}$ | $\underline{47.50}_{\pm 1.48}$ | $\underline{70.24}_{\pm 1.30}$ | $69.22_{\pm 1.27}$ | $\underline{63.33}$ |
| **Soren (Ours)** | $\mathbf{74.31}_{\pm 0.44}$ | $\mathbf{57.17}_{\pm 1.45}$ | $\mathbf{65.62}_{\pm 0.38}$ | $\mathbf{48.14}_{\pm 1.50}$ | $\mathbf{71.11}_{\pm 1.27}$ | $\mathbf{72.55}_{\pm 1.23}$ | **64.82** |

Soren achieves the highest score on all six evaluation benchmarks as well as the highest average score of 64.82, outperforming AdamW, Muon, and HTMuon by 3.00, 1.68, and 1.49 points, respectively. Compared with HTMuon, Soren improves performance by 0.56 points on HellaSwag, 0.86 points on ARC, 2.69 points on MMLU, 0.64 points on TruthfulQA, 0.87 points on Winogrande, and 2.72 points on GSM8K. Notably, the gains are particularly pronounced on MMLU and GSM8K, demonstrating consistent improvements across diverse capabilities, including commonsense rea-

soning, knowledge-intensive understanding, factuality, and mathematical reasoning. The sigmoid spectral reshaping in Soren smoothly compresses dominant singular modes while preserving relative differences among weaker modes, which can produce more balanced gradient updates during fine-tuning and contribute to the observed performance improvements.

### 5.3 LLM Direct Preference Optimization

**Experimental Setup.** We conduct direct preference optimization experiments starting from the HuggingFaceH4/mistral-7b-sft-alpha checkpoint, a supervised fine-tuned version of Mistral-7B-v0.1 (Jiang et al., 2023). We use this SFT checkpoint as the initial policy for subsequent preference optimization stage. Specifically, we perform DPO on the UltraFeedback Binarized dataset (Cui et al., 2024), which provides preference pairs consisting of chosen and rejected responses. For evaluation, we follow the same evaluation protocol and benchmark suite used in the SFT experiments described above.

Table 4: Evaluation results of different optimizers for DPO on Mistral 7B. Detailed hyperparameter configurations are reported in Table 9 in Appendix C.

| Method | HellaSwag | ARC | MMLU | TruthfulQA | Winogrande | GSM8K | Average |
|---|---|---|---|---|---|---|---|
| AdamW | $80.69_{\pm 0.36}$ | $61.09_{\pm 1.42}$ | $51.09_{\pm 0.39}$ | $45.32_{\pm 1.58}$ | $76.40_{\pm 1.14}$ | $30.93_{\pm 1.27}$ | 57.59 |
| Muon | $\underline{81.89_{\pm 0.38}}$ | $\underline{61.95_{\pm 1.42}}$ | $55.07_{\pm 0.40}$ | $\mathbf{49.56}_{\pm 1.57}$ | $\underline{77.35_{\pm 1.18}}$ | $\underline{31.69_{\pm 1.28}}$ | $\underline{59.59}$ |
| HTMuon | $81.85_{\pm 0.38}$ | $61.86_{\pm 1.42}$ | $\underline{55.13_{\pm 0.40}}$ | $49.52_{\pm 1.57}$ | $76.95_{\pm 1.18}$ | $31.31_{\pm 1.28}$ | 59.44 |
| **Soren (Ours)** | $\mathbf{83.25}_{\pm 0.37}$ | $\mathbf{62.46}_{\pm 1.42}$ | $\mathbf{57.21}_{\pm 0.39}$ | $\underline{49.54_{\pm 1.56}}$ | $\mathbf{77.74}_{\pm 1.17}$ | $\mathbf{36.32}_{\pm 1.32}$ | **61.09** |

On Mistral-7B, Soren achieves the highest average performance of 61.09, outperforming AdamW, Muon, and HTMuon by 3.50, 1.50, and 1.65 points, respectively. Soren also achieves the best performance on five of the six benchmarks, including HellaSwag (83.25), ARC (62.46), MMLU (57.21), Winogrande (77.74), and GSM8K (36.32), while remaining highly competitive on TruthfulQA (49.54), where it is only marginally below Muon (49.56). Compared with HTMuon, Soren yields improvements of 1.40 points on HellaSwag, 0.60 points on ARC, 2.08 points on MMLU, 0.79 points on Winogrande, and 5.01 points on GSM8K. The particularly large gains on MMLU and GSM8K demonstrate the effectiveness of Soren for preference optimization across knowledge-intensive and mathematical reasoning tasks.

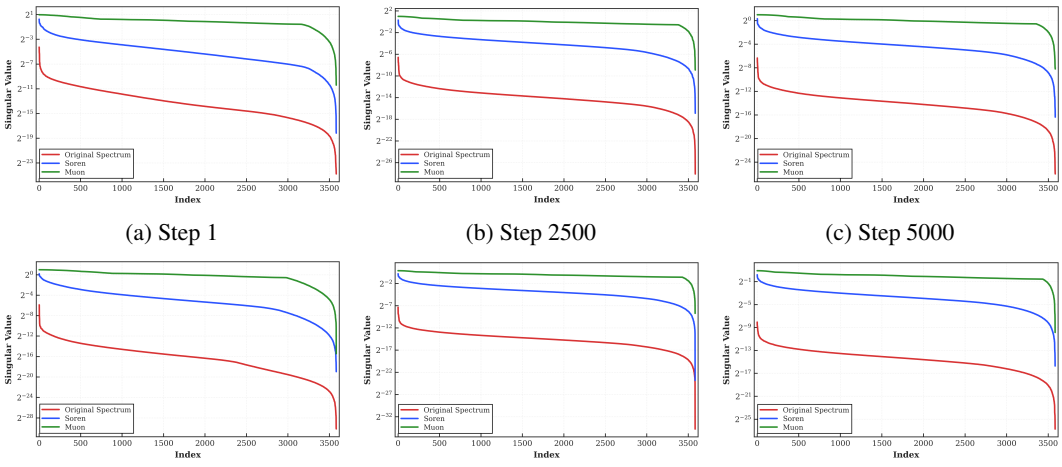
### 5.4 Singular Spectrum Analysis

Figure 2 visualizes the singular spectra of representative gradient matrices from the first and final layers of the LLaMA-60M model at different stages of training. As shown in the figure, the original gradient exhibits a pronounced spectral decay, with a large gap between dominant and weaker singular modes, causing the update to be disproportionately influenced by a small number of high-magnitude directions while suppressing the relative contribution of weaker modes. In contrast, Muon largely flattens the spectrum by assigning nearly equal magnitude to the active singular directions, removing much of the relative spectral structure of the original gradient. Soren exhibits an intermediate behavior: it substantially compresses the gap between large and small singular values while preserving their ordering and non-uniformity. Thus, dominant modes remain the primary contributors to the update, but their relative dominance is reduced, allowing weaker modes to contribute more substantially without being forced to the same magnitude. This spectral behavior, consistently observed across layers and training stages, highlights Soren's ability to mitigate excessive spectral imbalance while retaining informative relative differences among singular directions.

### 5.5 Soft Newton–Schulz Iteration Analysis

We empirically evaluate how closely the finite-step Soft Newton–Schulz (SNS) iteration approximates the exact sigmoid spectral map using the maximum relative error (MaxRelErr). Specifically, for singular values $x \in \mathcal{X} \subset [0.01, 1]$, we define $s(x) = \mathrm{sigmoid}(x)$ and denote the corresponding SNS approximation by $\widehat{s}(x)$. We measure the maximum pointwise relative error as:

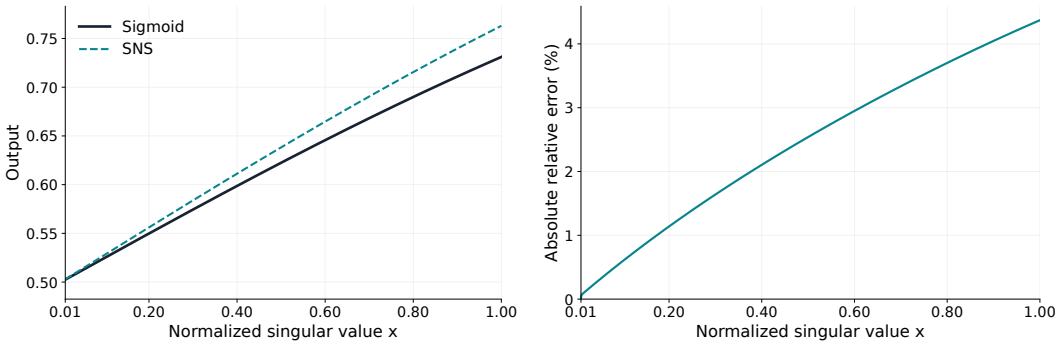
$$\mathrm{MaxRelErr} = \max_{x \in \mathcal{X}} \frac{|\widehat{s}(x) - s(x)|}{s(x)}.$$




(a) Step 1 (b) Step 2500 (c) Step 5000

(d) Step 1 (e) Step 2500 (f) Step 5000

Figure 2: Spectral transformation of a representative gradient matrix from the Llama 60M model, with singular values sorted in descending order. The top and bottom rows correspond to the first and final layers, respectively, while the columns show the initial, intermediate, and final training steps. Soren smoothly compresses dominant modes while preserving weaker-mode structure, in contrast to Muon-style orthogonalization, which flattens active modes toward a nearly uniform scale.




(a) SNS and sigmoid on normalized singular values. (b) Absolute relative error $|\widehat{s}(x) - s(x)|/s(x)$.

Figure 3: Comparison between SNS and the sigmoid approximation.

Figure 3a compares the two scalar mappings, while Figure 3b shows their pointwise relative error. Over the evaluated range, the SNS approximation closely follows the exact sigmoid mapping, with a maximum relative error of $4.37\%$. This metric has the same form as the relative spectral error introduced in Eq. (14), providing an empirical assessment of the approximation quality at the scalar level. Beyond this scalar-level analysis, we provide spectral comparisons on LLM training in Appendix B.3 and evaluate mean and maximum absolute errors on CNN gradients in Appendix D.

## 6 Conclusion

We introduced *Soren*, a spectral reshaping optimizer that applies a sigmoid transformation to the singular values of gradient updates, providing a smooth alternative to both magnitude-preserving and spectral orthogonalization updates. We established convergence guarantees under relative smoothness and Polyak–Łojasiewicz geometry, and developed a finite-step Soft Newton–Schulz realization that avoids explicit SVD computation. Experiments across language model pre-training, supervised fine-tuning, and direct preference optimization demonstrate that Soren achieves competitive or improved performance over established optimizers. Overall, our results show that controlled spectral reshaping is a promising approach for optimization in neural language model training and alignment.

### AI USE STATEMENT

In this work, we used generative AI tools for minor modifications to existing baseline code. We have not used generative AI tools for developing research ideas or devising mathematical transformations or derivation strategies. Additionally, we used generative AI tools for English language editing and LaTeX preparation. We have reviewed all AI-assisted work, with team members and the supervisor checking the wording, LaTeX, and modified code functions. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### REPRODUCIBILITY STATEMENT

To support reproducibility, we state the assumptions underlying our theoretical results in Section 4 and provide proofs of the convergence and geometry-preservation results in Appendices A.3, A.9, and A.10. Pseudocode for Soren, the Soft Newton–Schulz iteration, and the baseline optimizers is provided in Appendix E. Section 5 describes the models, datasets, and evaluation protocols, while Appendix C reports model configurations, learning-rate settings, and other training hyperparameters. Additional experimental results and computational-efficiency measurements are provided in Appendix B, and the procedure for evaluating SNS approximation errors is described in Appendix D.

## REFERENCES

Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond lipschitz gradient continuity: First-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017. doi: 10.1287/moor.2016.0817.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open llm leaderboard. `https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard`, 2023.

Å. Björck and C. Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971. doi: 10.1137/0708036. URL `https://doi.org/10.1137/0708036`.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved AI? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. URL `https://arxiv.org/abs/2110.14168`.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 9722–9744. PMLR, 21–27 Jul 2024. URL `https://proceedings.mlr.press/v235/cui24f.html`.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3029–3051, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL `https://aclanthology.org/2023.emnlp-main.183/`.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021. URL `https://doi.org/10.5281/zenodo.5371628`.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://iclr.cc/virtual/2021/poster/2962.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. URL https://kellerjordan.github.io/posts/muon/.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak–łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases – European Conference, ECML PKDD 2016, Proceedings, Part I*, volume 9851 of *Lecture Notes in Computer Science*, pp. 795–811. Springer, 2016. doi: 10.1007/978-3-319-46128-1_50.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86:2278–2324, 1998. URL https://api.semanticscholar.org/CorpusID:14542261.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019a. URL https://iclr.cc/virtual/2019/poster/935.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019b. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Haihao Lu, Robert M. Freund, and Yurii E. Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018. doi: 10.1137/16M1099546.

James Martens. Deep learning via hessian-free optimization. In *International Conference on Machine Learning*, 2010. URL https://icml.cc/Conferences/2010/papers/458.pdf.

Yurii Nesterov. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269:543–547, 1983. URL https://api.semanticscholar.org/CorpusID:145918791.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-2011.

URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

Tianyu Pang, Yujie Fang, Zihang Liu, Shenyang Deng, Lei Hsiung, Shuhua Yu, and Yaoqing Yang. HTMuon: Improving muon via heavy-tailed spectral correction. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36504–36535, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.1819. URL https://aclanthology.org/2026.findings-acl.1819/.

Boris Polyak. Gradient methods for the minimisation of functionals. *Ussr Computational Mathematics and Mathematical Physics*, 3:864–878, 1963. URL https://api.semanticscholar.org/CorpusID:119473154.

Boris Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4:1–17, 1964. URL https://api.semanticscholar.org/CorpusID:120243018.

Xianbiao Qi, Marco Chen, Jiaquan Ye, Yelin He, and Rong Xiao. Delving into muon and beyond: Deep analysis and extensions. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951. URL https://api.semanticscholar.org/CorpusID:16945044.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In Sanjoy Dasgupta and David McAllester (eds.), *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/sutskever13.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu,

Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Shixuan Gao, Shuang Luo, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Tang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhou, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019.

# A Mathematical proofs and derivations

## A.1 Exact spectral preconditioner and vectorization

This appendix derives the preconditioned form used in Section 4.2. Let $G_t \in \mathbb{R}^{m\times n}$ with $m \le n$ have full row rank and thin SVD

$$G_t = U_t \Sigma_t V_t^\top,$$

where $U_t \in \mathbb{R}^{m\times m}$ is orthogonal, $V_t \in \mathbb{R}^{n\times m}$ has orthonormal columns, and $\Sigma_t = \mathrm{diag}(\sigma_{t,1}, \dots, \sigma_{t,m})$ with $\sigma_{t,i} > 0$. The exact spectral Soren map is

$$s(G_t) = U_t s(\Sigma_t) V_t^\top.$$

Using Eq. (5),

$$P_t G_t = U_t s(\Sigma_t)\Sigma_t^{-1} U_t^\top U_t \Sigma_t V_t^\top = U_t s(\Sigma_t) V_t^\top = s(G_t). \tag{17}$$

Since $s(\sigma_{t,i}) > 0$ and $\sigma_{t,i} > 0$, every eigenvalue $s(\sigma_{t,i})/\sigma_{t,i}$ of $P_t$ is positive, hence $P_t \succ 0$.

Let $w_t = \mathrm{vec}(W_t)$ and $g_t = \mathrm{vec}(G_t)$. The identity

$$\mathrm{vec}(ABC) = (C^\top \otimes A)\,\mathrm{vec}(B)$$

gives

$$\mathrm{vec}(P_t G_t) = \mathrm{vec}(P_t G_t I_n) = (I_n \otimes P_t)\,\mathrm{vec}(G_t) = \mathcal{P}_t g_t. \tag{18}$$

Therefore the matrix update $W_t = W_{t-1} - \eta_t P_t G_t$ is equivalent to Eq. (6).

## A.2 Metric relative smoothness and metric PL geometry

Fix an iteration $t$ and hold $\mathcal{P}_t$ constant. The associated quadratic reference function is

$$h_t(w) = \frac{1}{2} w^\top \mathcal{P}_t^{-1} w.$$

Then $\nabla^2 h_t(w) = \mathcal{P}_t^{-1}$. Thus

$$H(z) \preceq \widetilde{\beta}_t \mathcal{P}_t^{-1}$$

is precisely the Hessian form of relative smoothness with respect to the quadratic reference $h_t$ (Bauschke et al., 2017; Lu et al., 2018).

The primal metric norm and its dual norm are

$$\|d\|^2_{\mathcal{P}_t^{-1}} = d^\top \mathcal{P}_t^{-1} d, \qquad \|g\|^2_{\mathcal{P}_t} = g^\top \mathcal{P}_t g.$$

For fixed $\mathcal{P}_t$, the linear coordinate transformation $w = \mathcal{P}_t^{1/2} y$ gives

$$\nabla_y \mathcal{L}(\mathcal{P}_t^{1/2} y) = \mathcal{P}_t^{1/2} \nabla_w \mathcal{L}(w),$$

and hence

$$\|\nabla_y \mathcal{L}\|_2^2 = \nabla_w \mathcal{L}(w)^\top \mathcal{P}_t \nabla_w \mathcal{L}(w).$$

The standard PL/gradient-dominance condition (Karimi et al., 2016), applied to the transformed objective $y \mapsto \mathcal{L}(\mathcal{P}_t^{1/2} y)$, therefore takes the form of Eq. (8) in the original coordinates. This change of coordinates explains the metric formulation; positivity of the metric PL constant remains an assumption of our analysis.

We next derive the descent inequality used in Theorem 1. Let $w, w' = w + d \in \Omega_t$. Since $\Omega_t$ is convex, $w + \tau d \in \Omega_t$ for $\tau \in [0, 1]$. The integral form of Taylor's theorem gives

$$\mathcal{L}(w + d) = \mathcal{L}(w) + \nabla \mathcal{L}(w)^\top d + \int_0^1 (1 - \tau) d^\top H(w + \tau d) d \, d\tau. \tag{19}$$

By Eq. (7),

$$d^\top H(w + \tau d) d \leq \widetilde{\beta}_t d^\top \mathcal{P}_t^{-1} d.$$

Using $\int_0^1 (1 - \tau) d\tau = 1/2$ in Eq. (19),

$$\mathcal{L}(w + d) \leq \mathcal{L}(w) + \nabla \mathcal{L}(w)^\top d + \frac{\widetilde{\beta}_t}{2} \|d\|^2_{\mathcal{P}_t^{-1}}. \tag{20}$$

This inequality follows directly from the Hessian bound along the segment and is valid without a quadratic approximation of the objective.

### A.3 Proof of Theorem 1

*Proof.* Set

$$d_t = w_t - w_{t-1} = -\eta_t \mathcal{P}_t g_t, \qquad g_t = \nabla \mathcal{L}(w_{t-1}).$$

Applying Eq. (20) at $w_{t-1}$ gives

$$\mathcal{L}(w_t) \leq \mathcal{L}(w_{t-1}) - \eta_t g_t^\top \mathcal{P}_t g_t + \frac{\widetilde{\beta}_t}{2} \eta_t^2 (\mathcal{P}_t g_t)^\top \mathcal{P}_t^{-1} (\mathcal{P}_t g_t) \tag{21}$$

$$= \mathcal{L}(w_{t-1}) - \eta_t \left(1 - \frac{\eta_t \widetilde{\beta}_t}{2}\right) \|g_t\|^2_{\mathcal{P}_t}. \tag{22}$$

With $\eta_t = 1/\widetilde{\beta}_t$,

$$\mathcal{L}(w_t) \leq \mathcal{L}(w_{t-1}) - \frac{1}{2\widetilde{\beta}_t} \|g_t\|^2_{\mathcal{P}_t}. \tag{23}$$

The full-row-rank assumption gives $g_t \neq 0$, hence $\mathcal{L}(w_{t-1}) > \mathcal{L}^*$, since a differentiable objective has zero gradient at a global minimizer. Combining Eq. (23) with $\mathcal{L}(w_t) \geq \mathcal{L}^*$ and Eq. (8) therefore yields

$$\widetilde{\alpha}_t \leq \frac{\frac{1}{2}\|g_t\|^2_{\mathcal{P}_t}}{\mathcal{L}(w_{t-1}) - \mathcal{L}^*} \leq \widetilde{\beta}_t.$$

Thus the contraction factor below lies in $[0, 1)$; strict inequality between the two constants is not required. By Eq. (8), evaluated at $w_{t-1}$,

$$\frac{1}{2}\|g_t\|^2_{\mathcal{P}_t} \geq \widetilde{\alpha}_t \left(\mathcal{L}(w_{t-1}) - \mathcal{L}^*\right).$$

Substituting this bound into Eq. (23) yields

$$\mathcal{L}(w_t) - \mathcal{L}^* \leq \left(1 - \frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t}\right) \left(\mathcal{L}(w_{t-1}) - \mathcal{L}^*\right).$$

Recursively applying the one-step inequality for $t = 1, \ldots, T$ proves Eq. (9).

Under Eq. (10),

$$\frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t} \geq \frac{\overline{\alpha}}{\overline{\beta}},$$

so every factor in the product satisfies

$$1 - \frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t} \leq 1 - \frac{\overline{\alpha}}{\overline{\beta}}.$$

Applying this bound to Eq. (9) proves Eq. (11). □

### A.4 SNS METRIC DEFINITIONS AND COMPARISON BOUNDS

For the finite-step SNS output in Section 4.3, singular-subspace preservation gives

$$\widehat{s}(G_t) = U_t \widehat{S}_t V_t^\top, \qquad \widehat{S}_t = \operatorname{diag}(\widehat{s}_{t,1}, \dots, \widehat{s}_{t,m}). \tag{24}$$

Under the full-row-rank assumption of Section 4.2, define

$$\widehat{P}_t = U_t \operatorname{diag}\left(\frac{\widehat{s}_{t,i}}{\sigma_{t,i}}\right) U_t^\top, \qquad \widehat{\mathcal{P}}_t = I_n \otimes \widehat{P}_t. \tag{25}$$

Then $\widehat{s}(G_t) = \widehat{P}_t G_t$. The relative error bound in Eq. (14) makes all coefficients positive and hence both preconditioners positive definite.

**Lemma 1** (Preconditioner sandwich)**.** *Under Eq. (14), the exact and finite-step preconditioners satisfy:*

$$(1-\rho_t)\mathcal{P}_t \preceq \widehat{\mathcal{P}}_t \preceq (1+\rho_t)\mathcal{P}_t, \tag{26}$$

*and, consequently,*

$$\frac{1}{1+\rho_t}\mathcal{P}_t^{-1} \preceq \widehat{\mathcal{P}}_t^{-1} \preceq \frac{1}{1-\rho_t}\mathcal{P}_t^{-1}. \tag{27}$$

To compare the two metrics under the same local geometry, we use the same region $\Omega_t$ and optimal value $\mathcal{L}^*$ to define the SNS constants:

$$\widehat{\beta}_t = \inf\left\{\beta > 0 : H(z) \preceq \beta \widehat{\mathcal{P}}_t^{-1} \text{ for all } z \in \Omega_t\right\}, \tag{28}$$

$$\widehat{\alpha}_t = \inf_{\substack{w\in\Omega_t \\ \mathcal{L}(w)>\mathcal{L}^*}} \frac{\frac{1}{2}\|\nabla\mathcal{L}(w)\|^2_{\widehat{\mathcal{P}}_t}}{\mathcal{L}(w)-\mathcal{L}^*}. \tag{29}$$

The individual metric bounds underlying Theorem 2 are

$$(1-\rho_t)\widetilde{\alpha}_t \le \widehat{\alpha}_t \le (1+\rho_t)\widetilde{\alpha}_t, \tag{30}$$

$$(1-\rho_t)\widetilde{\beta}_t \le \widehat{\beta}_t \le (1+\rho_t)\widetilde{\beta}_t. \tag{31}$$

The following subsections establish the spectral representation, preconditioner sandwich, and metric bounds, then prove the theorem and convergence corollary.

### A.5 SINGULAR-SUBSPACE PRESERVATION OF FINITE-STEP SNS

Represent a stream iterate in the fixed singular-vector basis of its input as

$$X_k = U D_k V^\top, \qquad D_k = \operatorname{diag}(d_{k,1}, \dots, d_{k,m}).$$

The diagonal entries of $D_k$ are spectral coefficients and need not be nonnegative after a polynomial step. Since $U$ is orthogonal and $V$ has orthonormal columns,

$$X_k X_k^\top = U D_k^2 U^\top.$$

Using Eq. (12),

$$\Phi(X_k) = \frac{1}{2}(3I - U D_k^2 U^\top) U D_k V^\top = U\left[\frac{1}{2}(3I - D_k^2)D_k\right] V^\top.$$

Thus, the recurrence acts only on the diagonal coefficients in the fixed basis $U, V$. Both $Q_0$ and $T_0$ are scalar multiples of $G_t$, so they share the input basis $U_t, V_t$. Induction over any finite number of steps in each stream gives

$$Q = U_t D_Q V_t^\top, \qquad T = U_t D_T V_t^\top.$$

Consequently,

$$\frac{1}{2}(Q+T) = U_t\left[\frac{1}{2}(D_Q + D_T)\right] V_t^\top,$$

which proves Eq. (24).

### A.6 SNS PRECONDITIONER REPRESENTATION AND SANDWICH

From Eq. (25),

$$\widehat{P}_t G_t = U_t \operatorname{diag}\left(\frac{\widehat{s}_{t,i}}{\sigma_{t,i}}\right) U_t^\top U_t \Sigma_t V_t^\top = U_t \widehat{S}_t V_t^\top = \widehat{s}(G_t).$$

Equation (14) is equivalent to

$$(1-\rho_t)s(\sigma_{t,i}) \le \widehat{s}_{t,i} \le (1+\rho_t)s(\sigma_{t,i}).$$

Dividing by $\sigma_{t,i} > 0$ and using the common eigenbasis $U_t$ gives

$$(1-\rho_t)P_t \preceq \widehat{P}_t \preceq (1+\rho_t)P_t.$$

Taking the Kronecker product with $I_n \succeq 0$ preserves Loewner order and proves Eq. (26). Since every matrix is positive definite and inversion reverses Loewner order,

$$\widehat{\mathcal{P}}_t \preceq (1+\rho_t)\mathcal{P}_t \quad \Longrightarrow \quad \frac{1}{1+\rho_t}\mathcal{P}_t^{-1} \preceq \widehat{\mathcal{P}}_t^{-1},$$

and

$$(1-\rho_t)\mathcal{P}_t \preceq \widehat{\mathcal{P}}_t \quad \Longrightarrow \quad \widehat{\mathcal{P}}_t^{-1} \preceq \frac{1}{1-\rho_t}\mathcal{P}_t^{-1}.$$

Together these inequalities prove Eq. (27).

### A.7 PRESERVATION OF THE METRIC PL CONSTANT

For every $w \in \Omega_t$ with $\mathcal{L}(w) > \mathcal{L}^*$, Eq. (26) implies

$$(1-\rho_t)\|\nabla\mathcal{L}(w)\|_{\mathcal{P}_t}^2 \le \|\nabla\mathcal{L}(w)\|_{\widehat{\mathcal{P}}_t}^2 \le (1+\rho_t)\|\nabla\mathcal{L}(w)\|_{\mathcal{P}_t}^2.$$

Dividing by $2(\mathcal{L}(w)-\mathcal{L}^*) > 0$ yields

$$(1-\rho_t)\frac{\frac{1}{2}\|\nabla\mathcal{L}(w)\|_{\mathcal{P}_t}^2}{\mathcal{L}(w)-\mathcal{L}^*} \le \frac{\frac{1}{2}\|\nabla\mathcal{L}(w)\|_{\widehat{\mathcal{P}}_t}^2}{\mathcal{L}(w)-\mathcal{L}^*} \le (1+\rho_t)\frac{\frac{1}{2}\|\nabla\mathcal{L}(w)\|_{\mathcal{P}_t}^2}{\mathcal{L}(w)-\mathcal{L}^*}.$$

Taking the infimum over the same set of $w$ proves Eq. (30).

### A.8 PRESERVATION OF THE RELATIVE-SMOOTHNESS CONSTANT

By the definition of $\widetilde{\beta}_t$,

$$H(z) \preceq \widetilde{\beta}_t \mathcal{P}_t^{-1}, \qquad \forall z \in \Omega_t.$$

The left inequality in Eq. (27) is equivalent to

$$\mathcal{P}_t^{-1} \preceq (1+\rho_t)\widehat{\mathcal{P}}_t^{-1}.$$

Hence

$$H(z) \preceq (1+\rho_t)\widetilde{\beta}_t \widehat{\mathcal{P}}_t^{-1}, \qquad \forall z \in \Omega_t,$$

so the minimality of $\widehat{\beta}_t$ gives

$$\widehat{\beta}_t \le (1+\rho_t)\widetilde{\beta}_t.$$

Conversely, by the definition of $\widehat{\beta}_t$,

$$H(z) \preceq \widehat{\beta}_t \widehat{\mathcal{P}}_t^{-1}.$$

The right inequality in Eq. (27) gives

$$\widehat{\mathcal{P}}_t^{-1} \preceq \frac{1}{1-\rho_t}\mathcal{P}_t^{-1}.$$

Therefore

$$H(z) \preceq \frac{\widehat{\beta}_t}{1-\rho_t}\mathcal{P}_t^{-1},$$

and the minimality of $\widetilde{\beta}_t$ yields

$$\widetilde{\beta}_t \le \frac{\widehat{\beta}_t}{1-\rho_t}.$$

Equivalently,

$$(1-\rho_t)\widetilde{\beta}_t \le \widehat{\beta}_t.$$

Combining the two directions proves Eq. (31).

### A.9 Proof of Theorem 2

Equation (30) follows from Appendix A.7, and Eq. (31) follows from Appendix A.8. Since all constants are positive,

$$\frac{\widehat{\alpha}_t}{\widehat{\beta}_t} \geq \frac{(1-\rho_t)\widetilde{\alpha}_t}{(1+\rho_t)\widetilde{\beta}_t} = \frac{1-\rho_t}{1+\rho_t}\frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t},$$

and similarly

$$\frac{\widehat{\alpha}_t}{\widehat{\beta}_t} \leq \frac{(1+\rho_t)\widetilde{\alpha}_t}{(1-\rho_t)\widetilde{\beta}_t} = \frac{1+\rho_t}{1-\rho_t}\frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t}.$$

This proves Eq. (15).

### A.10 Proof of Corollary 1

*Proof.* The assumed segment containment allows us to apply the descent argument of Appendix A.3 along the SNS trajectory, using $\widehat{\mathcal{P}}_t$, $\widehat{\alpha}_t$, and $\widehat{\beta}_t$. With $\eta_t = 1/\widehat{\beta}_t$,

$$\mathcal{L}(w_t) - \mathcal{L}^* \leq \left(1 - \frac{\widehat{\alpha}_t}{\widehat{\beta}_t}\right)\left(\mathcal{L}(w_{t-1}) - \mathcal{L}^*\right).$$

From Theorem 2, $\rho_t \leq \overline{\rho}$, and the uniform exact bounds,

$$\frac{\widehat{\alpha}_t}{\widehat{\beta}_t} \geq \frac{1-\rho_t}{1+\rho_t}\frac{\widetilde{\alpha}_t}{\widetilde{\beta}_t} \geq \frac{1-\overline{\rho}}{1+\overline{\rho}}\frac{\overline{\alpha}}{\overline{\beta}}. \tag{32}$$

Therefore every contraction factor is bounded above by

$$1 - \frac{1-\overline{\rho}}{1+\overline{\rho}}\frac{\overline{\alpha}}{\overline{\beta}}.$$

Recursion for $t = 1, \ldots, T$ gives Eq. (16). □

## B More Experimental Details

### B.1 Training and Evaluation Loss Curve

Figure 4 presents the training and evaluation loss curves for LLaMA-350M and LLaMA-1B on the C4 dataset. We observe that Soren exhibits a convergence trend similar to Muon and HTMuon in terms of training loss, while achieving a consistently lower evaluation loss in the later stages of training for both model scales. In particular, the gap between Soren and the other matrix-based optimizers becomes more apparent toward the end of training, whereas AdamW maintains a noticeably higher evaluation loss throughout training. These results indicate that Soren maintains competitive optimization dynamics while achieving improved generalization performance at the end of pre-training.

### B.2 More Experiments Results

**Time Complexity** We measure the wall-clock training time of AdamW, Muon, HTMuon, and Soren to evaluate their practical computational efficiency. We conduct these measurements across pre-training: Llama-60M on C4 using 1 GPU H100, SFT: Qwen2.5-3B on UltraChat 200k using 1 GPU H100, and DPO: Mistral-7B-v0.1 on UltraFeedback Binarized using 2 GPUs H100. For each setting, we adopt the optimal learning rate and hyperparameter configuration reported in Tables 7 and 9, respectively.

Table 5 reports the wall-clock training time of different optimizers across pre-training, SFT, and DPO settings. Soren introduces a moderate overhead compared with AdamW and Muon, with a relative training time of $1.69\times$ that of AdamW on LLaMA-60M. However, this overhead remains substantially smaller than that of HTMuon, which requires $5.39\times$ the training time of AdamW in the same setting. For the larger SFT and DPO experiments, the training time of Soren remains close to that of AdamW and Muon. Overall, these results highlight a favorable computational trade-off, where Soren achieves improved performance while maintaining a moderate wall-clock training cost.

We also provide the FLOPs analysis for Muon and Soren. Let the initial weight matrix be $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$.

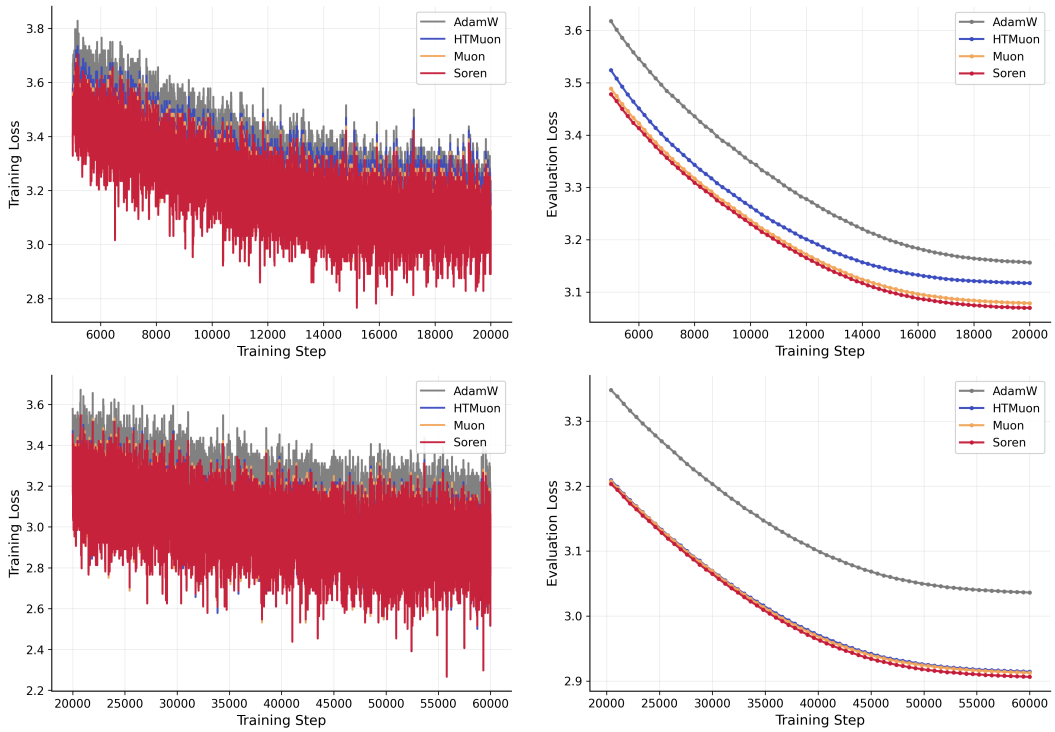



Figure 4: Training and evaluation loss curves for the LLaMA-350M (top row) and LLaMA-1B (bottom row) experiments.

Table 5: Comparison of the wall-clock training time of different optimizers. Relative training time is normalized with respect to AdamW.

| | LLaMA-60M | | Qwen2.5-3B | Mistral-7B-v0.1 |
|---|---|---|---|---|
| **Optimizer** | **Time** | **Relative to AdamW** | **Time** | **Time** |
| AdamW | 0h23m | 1.00x | 5h23m | 10h3m |
| Muon | 0h28m | 1.22x | 5h33m | 10h11m |
| HTMuon | 2h04m | 5.39x | 5h38m | 11h19m |
| Soren | 0h39m | 1.69x | 5h33m | 10h29m |

- **FLOPs analysis for Muon (Algorithm 4):**

$$F_{\text{Muon}} = 20mnr + O(mn). \tag{33}$$

- **FLOPs analysis for Soren (Algorithm 2):**

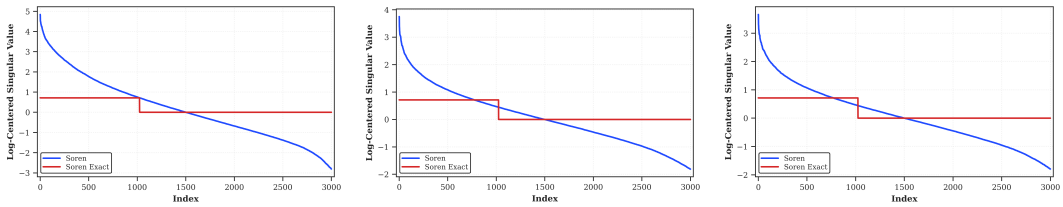
$$F_{\text{Soren}} = 4mnr(S+2) + O(mn). \tag{34}$$

where $r = \min(m, n)$ and $S$ denotes the number of SNS iterations in the $Q$ stream.

Compared with Muon, Soren incurs additional computational cost primarily due to the extra Newton–Schulz iterations introduced by the SNS procedure.

### B.3 Spectral Approximation of the Soren Transform in LLM Training

Figure 5 compares the log-centered singular spectral produced by SNS and the exact sigmoid spectral map at different training stages. The finite-step SNS approximation closely matches the exact sigmoid transformation across all three snapshots, reproducing the overall spectral shape and the relative compression of dominant and weaker modes. This agreement persists throughout training,




(a) Step 1 (b) Step 2500 (c) Step 5000

Figure 5: Log-centered singular spectra of the exact sigmoid spectral map and its finite-step Soft Newton–Schulz (SNS) approximation for a representative gradient matrix from the final layer of the Llama-135M model. Log singular values are centered by subtracting the median of the $\log_2$ singular values to isolate relative spectral shape from overall scale. The columns show the initial, intermediate, and later training steps. SNS closely tracks the exact sigmoid transformation across training stages, preserving its relative spectral structure.

demonstrating that a small number of Newton–Schulz iterations is sufficient to provide a faithful polynomial approximation to the exact sigmoid spectral map without explicit SVD computation.

# C Detailed Hyperparameters

## C.1 Hyperparameter settings for LLM pretraining

Our practical Soren implementation follows the hybrid Muon–AdamW formulation (Jordan et al., 2024), using standard AdamW hyperparameters for the subset of parameters updated with AdamW. For the C4 experiments, we follow the LLaMA architecture configurations reported in Table 6 to ensure reproducibility. All model variants are trained with a maximum sequence length of 256 and a batch size of 512, corresponding to approximately 13K tokens per batch. We train LLaMA-60M and LLaMA-135M on a single H100 GPU, while LLaMA-350M and LLaMA-1B are trained on four L40S GPUs.

Table 6: Hyperparameters of LLaMA models.

| Params | Hidden | Intermediate | Heads | Blocks | Steps | Data amount | Batch Size |
|---|---|---|---|---|---|---|---|
| 60M | 512 | 1376 | 8 | 8 | 5K | 0.5B | 512 |
| 135M | 768 | 2048 | 12 | 12 | 10K | 1B | 512 |
| 350M | 1024 | 2736 | 16 | 24 | 20K | 2B | 512 |
| 1B | 2048 | 5461 | 32 | 24 | 60K | 6B | 512 |

For HTMuon, we use $p = 0.125$ throughout all LLM pre-training experiments, following the setting identified as optimal in (Pang et al., 2026). The Muon update is applied to both the embedding and output layers during training. Further details on the hyperparameter configurations are provided in Tables 7 and 8.

Table 7: Hyperparameters for LLaMA-60M and LLaMA-135M on the C4 dataset. Boldface values denote the optimal hyperparameters.

| | LLaMA-60M | | LLaMA-135M | |
|---|---|---|---|---|
| Optimizer | LR | WD | LR | WD |
| AdamW | **1e-3** | 0.1 | **1e-3** | 0.1 |
| Muon | {**0.01**, 0.02, 0.03, 0.04} | 0.1 | {**0.01**, 0.02, 0.03, 0.04} | 0.1 |
| HTMuon | {0.01, 0.02, **0.03**, 0.04} | 0.1 | {0.01, 0.02, **0.03**, 0.04} | 0.1 |
| Soren | {0.01, 0.02, **0.03**, 0.04} | 0.1 | {0.01, 0.02, **0.03**, 0.04} | 0.1 |

Table 8: Hyperparameters for LLaMA-350M and LLaMA-1B on the C4 dataset. Boldface values denote the optimal hyperparameters.

| | LLaMA-350M | | LLaMA-1B | |
|---|---|---|---|---|
| Optimizer | LR | WD | LR | WD |
| AdamW | **1e-3** | 0.1 | **6e-4** | 0.1 |
| Muon | {0.0025, **0.005**, 0.01, 0.015} | 0.1 | {**0.005**, 0.01} | 0.1 |
| HTMuon | {0.0025, **0.005**, 0.01, 0.015} | 0.1 | {**0.005**, 0.01} | 0.1 |
| Soren | {0.0025, 0.005, **0.01**, 0.015} | 0.1 | {0.005, **0.01**} | 0.1 |

### C.2 Hyperparameter Settings for LLM Supervised Fine-Tuning and LLM Direct Preference Optimization

We report the hyperparameter configurations used for supervised fine-tuning (SFT) and direct preference optimization (DPO) experiments in Table 9. For SFT, we use Qwen2.5-3B on the UltraChat 200k dataset, while for DPO, we use Mistral-7B-v0.1 on the UltraFeedback Binarized dataset. For each optimizer, we evaluate a set of learning rates and select the configuration achieving the best validation performance. We use a weight decay of $0.1$ for all experiments. Boldface values denote the selected learning rates.

Table 9: Hyperparameters for Qwen2.5-3B and Mistral-7B-v0.1 on the UltraChat 200k and UltraFeedback Binarized datasets. Boldface values denote the selected learning rates.

| | Qwen2.5-3B | | Mistral-7B-v0.1 | |
|---|---|---|---|---|
| Optimizer | LR | WD | LR | WD |
| AdamW | {1e-6, 5e-6, 2e-5, **3e-5**} | 0.1 | {**1e-7**, 4e-7, 5e-7} | 0.1 |
| Muon | {1e-6, 5e-6, **2e-5**, 3e-5} | 0.1 | {1e-7, 4e-7, **5e-7**} | 0.1 |
| HTMuon | {1e-6, 5e-6, **2e-5**, 3e-5} | 0.1 | {1e-7, 4e-7, **5e-7**} | 0.1 |
| Soren | {**1e-6**, 5e-6, 2e-5, 3e-5} | 0.1 | {1e-7, **4e-7**, 5e-7} | 0.1 |

## D More Evaluation Details

As a complementary evaluation, we measure the mean absolute error (MAE) and maximum absolute error (MaxErr) of the SNS approximation on gradients extracted from a convolutional network (LeCun et al., 1998) trained on CIFAR-10 (Krizhevsky, 2009). For each validation mini-batch, we compute the output-layer gradient $G$ and normalize it as $\bar{G} = G/(\|G\|_F + \varepsilon)$. We then obtain the singular value decomposition $\bar{G} = U\,\mathrm{diag}(\sigma_i)V^\top$ and compare the exact sigmoid coefficients $s_i = \mathrm{sigmoid}(\sigma_i)$ with their SNS approximations $\widehat{s}_i$. We report the following absolute-error metrics:

$$\mathrm{MAE} = \frac{1}{r}\sum_{i=1}^{r} |s_i - \widehat{s}_i|\,, \qquad \mathrm{MaxErr} = \max_{1\le i\le r} |s_i - \widehat{s}_i|\,.$$

Table 10 summarizes the absolute approximation errors across training. These results complement the scalar relative-error analysis in Section 5.5 by evaluating the approximation on gradient matrices encountered during training. Importantly, these metrics measure absolute approximation errors and therefore do not directly quantify the maximum relative error on CNN gradients or establish the relative-error bound assumed in the convergence analysis.

## E Optimization Algorithms

In this section, we provide the algorithms for all optimizers evaluated in our study. We use the following notation throughout: $\mathbf{W}_t$ denotes the model parameters at step $t$, $\mathbf{G}_t$ the corresponding gradient, $\eta$ the learning rate, $\lambda$ the weight decay coefficient, $\beta_1$ and $\beta_2$ the first- and second-moment

Table 10: Approximation error of the SNS iteration relative to the exact sigmoid spectral map. Each entry reports the mean ± standard deviation over validation mini-batches.

| Metric | Epoch 2 | Epoch 4 | Epoch 6 | Epoch 8 | Epoch 10 |
|---|---|---|---|---|---|
| MAE | $0.0138_{\pm 0.0008}$ | $0.0139_{\pm 0.0005}$ | $0.0155_{\pm 0.0007}$ | $0.0158_{\pm 0.0010}$ | $0.0163_{\pm 0.0010}$ |
| MaxErr | $0.0902_{\pm 0.0100}$ | $0.0893_{\pm 0.0069}$ | $0.0924_{\pm 0.0095}$ | $0.0950_{\pm 0.0076}$ | $0.0971_{\pm 0.0112}$ |

decay rates, respectively, and $\epsilon$ a numerical stability constant. We further use $\|\mathbf{G}_t\|$ to denote the gradient norm, and $\mathbf{m}_t$ and $\mathbf{v}_t$ to denote the first and second moment estimates, respectively. Unless otherwise specified, all operations are element-wise.

### E.1 Soren Optimizer

We summarize the proposed Soren optimizer and its polynomial realization in the following algorithms. Algorithm 1 defines the Soft Newton–Schulz (SNS) iteration, denoted by $\mathcal{SNS}(\cdot)$, which approximates the sigmoid spectral map without requiring an explicit SVD. Algorithm 2 presents the complete Soren update, applying $\mathcal{SNS}(\cdot)$ to the Nesterov-adjusted momentum.

**Algorithm 1** Soft Newton–Schulz Iteration $\mathcal{SNS}(\boldsymbol{X})$

**Require:** Input matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, steps $K$, numerical stability constant $\epsilon = 10^{-8}$
1: $\widehat{\boldsymbol{X}} \leftarrow \boldsymbol{X}/(\|\boldsymbol{X}\|_F + \epsilon)$
2: $\boldsymbol{Q} \leftarrow \widehat{\boldsymbol{X}}$
3: **for** $k = 1, \ldots, K$ **do**
4: $\quad \boldsymbol{Q} \leftarrow \frac{1}{2}(3\boldsymbol{I} - \boldsymbol{Q}\boldsymbol{Q}^\top)\boldsymbol{Q}$
5: **end for**
6: $\boldsymbol{T} \leftarrow \frac{1}{4}\boldsymbol{X}$
7: **for** $j = 1, 2$ **do**
8: $\quad \boldsymbol{T} \leftarrow \frac{1}{2}(3\boldsymbol{I} - \boldsymbol{T}\boldsymbol{T}^\top)\boldsymbol{T}$
9: **end for**
10: **return** $\frac{1}{2}\boldsymbol{Q} + \frac{1}{2}\boldsymbol{T}$

**Algorithm 2** Soren

**Require:** Learning rate $\eta$, momentum coefficient $\mu$
1: Initialize $\boldsymbol{W}_0$ and $\boldsymbol{M}_0 \leftarrow \boldsymbol{0}$
2: **for** $t = 1, \ldots$ **do**
3: $\quad \boldsymbol{G}_t \leftarrow \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}_{t-1})$
4: $\quad \boldsymbol{M}_t \leftarrow \mu \boldsymbol{M}_{t-1} + (1-\mu)\boldsymbol{G}_t$
5: $\quad \boldsymbol{N}_t \leftarrow (1-\mu)\boldsymbol{G}_t + \mu \boldsymbol{M}_t$
6: $\quad \boldsymbol{O}_t \leftarrow \mathcal{SNS}(\boldsymbol{N}_t)$
7: $\quad \boldsymbol{W}_t \leftarrow \boldsymbol{W}_{t-1} - \eta \boldsymbol{O}_t$
8: **end for**
9: **return** $\boldsymbol{W}_t$

### E.2 Baseline Optimizers

This section presents the algorithmic details of the baseline optimizers considered in our experiments. We provide the update procedures for HTMuon, Muon, and AdamW to clarify the optimization steps used throughout our empirical evaluation. The algorithms are presented using a unified notation consistent with the formulation of Soren above.

**Algorithm 3** HTMuon

**Require:** Initial weights $\boldsymbol{W}_0 \in \mathbb{R}^{m\times n}$, loss function $\mathcal{L}$, learning rate $\eta$, momentum coefficient $\beta$, weight decay $\lambda$, power $p \in (0,1)$

Initialize $\boldsymbol{M}_0 \leftarrow \boldsymbol{0}$
**for** $t = 1, 2, \ldots$ **do**
  $\boldsymbol{G}_t \leftarrow \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}_{t-1})$
  $\boldsymbol{M}_t \leftarrow \beta \boldsymbol{M}_{t-1} + (1-\beta)\boldsymbol{G}_t$
  $\boldsymbol{U}_t, \boldsymbol{\Sigma}_t, \boldsymbol{V}_t^\top \leftarrow \mathrm{SVD}(\boldsymbol{M}_t)$
  $\boldsymbol{O}_t \leftarrow \boldsymbol{U}_t \boldsymbol{\Sigma}_t^p \boldsymbol{V}_t^\top$
  $s \leftarrow \sqrt{\max\left(1, \frac{m}{n}\right)}$
  $\boldsymbol{W}_t \leftarrow \boldsymbol{W}_{t-1} - \eta\lambda \boldsymbol{W}_{t-1} - \eta s \boldsymbol{O}_t$
**end for**
**return** $\boldsymbol{W}_t$

**Algorithm 4** Muon

**Require:** Learning rate $\eta$, momentum coefficient $\mu$

Initialize $\boldsymbol{M}_0 \leftarrow \boldsymbol{0}$
**for** $t = 1, \ldots$ **do**
  $\boldsymbol{G}_t \leftarrow \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}_{t-1})$
  $\boldsymbol{M}_t \leftarrow \mu \boldsymbol{M}_{t-1} + \boldsymbol{G}_t$
  $\boldsymbol{O}_t \leftarrow \mathrm{NewtonSchulz}(\boldsymbol{M}_t)$
  $\boldsymbol{W}_t \leftarrow \boldsymbol{W}_{t-1} - \eta \boldsymbol{O}_t$
**end for**
**return** $\boldsymbol{W}_t$

**Algorithm 5** AdamW

**Require:** Learning rate $\eta$, weight decay $\lambda$, moment coefficients $\beta_1, \beta_2$, numerical stability constant $\epsilon$

Initialize $\boldsymbol{W}_0$, $\boldsymbol{m}_0 \leftarrow \boldsymbol{0}$, $\boldsymbol{v}_0 \leftarrow \boldsymbol{0}$
**for** $t = 1, \ldots$ **do**
  $\boldsymbol{G}_t \leftarrow \nabla_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W}_{t-1})$
  $\boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1-\beta_1)\boldsymbol{G}_t$
  $\boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1-\beta_2)\boldsymbol{G}_t^2$
  $\widehat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t / (1-\beta_1^t)$
  $\widehat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t / (1-\beta_2^t)$
  $\boldsymbol{O}_t \leftarrow \widehat{\boldsymbol{m}}_t / \left(\sqrt{\widehat{\boldsymbol{v}}_t} + \epsilon\right)$
  $\boldsymbol{W}_t \leftarrow \boldsymbol{W}_{t-1} - \eta \boldsymbol{O}_t - \eta\lambda \boldsymbol{W}_{t-1}$
**end for**
**return** $\boldsymbol{W}_t$