SCALE WEIGHT DECAY AND TRAIN BETTER

Anonymous authors
Paper under double-blind review

ABSTRACT

The discovery of scaling laws has motivated training neural networks on ever increasing quantities of data. This is typically done with a constant decoupled weight decay under which the weight norm can shrink steadily after an early peak. Taking inspiration from the Robbins–Monro conditions, we propose to scale weight decay by the fraction of the peak learning rate $\eta/\eta_{\max}$. We prove that this scaled weight decay preserves the asymptotic stationarity guarantees of the corresponding unregularized methods for both stochastic gradient descent and an idealized exact-polar Muon with ordinary momentum, thereby avoiding the additional asymptotic bias introduced by constant decoupled weight decay. This retains the stability benefits of weight decay without changing the asymptotic optimization target. Using a steady-state analysis, we explain why under standard weight decay the weight norm shrinks steadily as training proceeds, whereas under scaled weight decay it settles to a roughly constant value. When applied to the training of mixture-of-experts models, Muon with scaled weight decay (Muon-SW) consistently outpaces Muon with identical hyperparameters, reaching the same validation loss in approximately $\mathbf{30}\%$ fewer optimizer steps at our largest scale across models from $72 - 930$ million parameters trained at ~ 600 tokens per active parameter. If this trend continues to hold, the method promises to substantially accelerate the pre-training of frontier models while requiring only a few lines of code to implement.

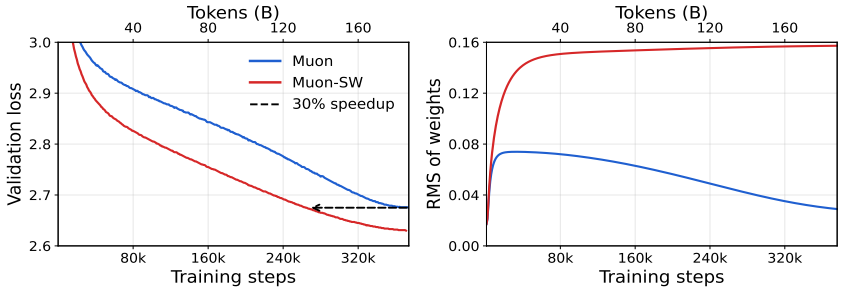


Figure 1: **Scaled weight decay trains faster and keeps weights from collapsing (width-1024 MoE, $\sim$932M parameters).** *Left:* validation loss versus optimizer steps for Muon with standard weight decay and Muon-SW with scaled weight decay (decay factor $\eta_t^2\lambda/\eta_{\max}$ instead of $\eta_t\lambda$). Muon-SW reaches the best validation loss of the standard run (2.675) in 264k steps versus 374k, a $\sim 30\%$ reduction in steps (dashed arrow). *Right:* RMS norm of the weights, averaged across non-router layers. Under standard weight decay the norm peaks early and then steadily decays, whereas under scaled weight decay it settles to a roughly constant value. The models are trained on FineWeb dataset (Penedo et al., 2024) and use identical hyperparameters: 8 experts, top-2 routing, weight decay 0.1, a peak learning rate of 4.8×10^{-3} that decays to 4.8×10^{-4} by the end of training, and a global batch of 256 sequences of length 2,048, corresponding to 524,288 tokens per step.

1 INTRODUCTION

Scaling laws, which predict neural network loss as a function of training data and model size, have motivated the training of neural networks on increasingly larger datasets (Bahri et al., 2024; Hoff-

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.

mann et al., 2022; Kaplan et al., 2020). This trend holds across modalities beyond language including vision (Dosovitskiy et al., 2021; Zhai et al., 2022), audio (Manakul et al., 2026) and robotics (Sartor & Thompson, 2025). For example, the largest language models with publicly available training methodology are pre-trained on tens of trillions of tokens (DeepSeek-AI et al., 2026; GLM-4.5 Team et al., 2025; Kimi Team et al., 2025). This pre-training phase is where the model learns to predict the next token as accurately as possible. A key role in the training process is played by the weight decay which is applied throughout. Weight decay was originally introduced to aid in regularization and prevent overfitting by modifying the loss $\mathcal{L}$ as $\mathcal{L} + \lambda||W||^2/2$ where λ is the coefficient of weight decay and ℓ_2 norm is used to measure the size of the weights W (Krogh & Hertz, 1991). However, for modern deep learning the number of training data points far exceeds the number of weights which makes overfitting unlikely. In this case, the primary benefits of weight decay are stabilizing the training process by modifying the dynamics of training and reducing the final loss (D'Angelo et al., 2024). Weight decay is applied in a decoupled manner given by

$$W_{t+1} = (1 - \eta_t\lambda)W_t - \eta_t U_t \ , \tag{1}$$

where W_t, η_t, U_t are the weight, learning rate and pre-conditioned gradient update at step t and λ is the coefficient of weight decay (Loshchilov & Hutter, 2019). For optimizers which pre-condition the update such as AdamW, this is not equivalent to modifying the loss function with an ℓ_2 regularization term. Thus, contraction is applied separately from the adaptive gradient normalization. Since the coefficient of weight decay is multiplied by the learning rate η_t, the cumulative effect of weight decay depends on the choice of the learning schedule. For training transformers and their variants, it is important to increase the learning rate in the beginning till it reaches its maximum value (Kalra & Barkeshli, 2024). This process is called warmup; after the end of warmup the learning rate is decreased as training proceeds, typically following a cosine schedule (Loshchilov & Hutter, 2017).

An important change that has occurred since 2025 is the increased adoption of Muon as the main optimizer over AdamW. Muon is an optimizer for matrix- or tensor-valued weights that computes the update subject to a spectral norm constraint. This is accomplished by orthogonalizing the momentum buffer and then scaling by the learning rate before updating the weight (Jordan et al., 2024; Liu et al., 2025). Following AdamW, Muon incorporates a decoupled weight decay with a constant coefficient λ. As a result, the weights of the network continue to shrink in magnitude after the end of warmup as illustrated by the blue curve in the right panel of Figure 1, where we have plotted the root mean squared (RMS) value of matrix-valued weights averaged over the layers as a function of training steps. If the primary role of weight decay in modern deep learning is not regularization but controlling the training dynamics and improving the loss, then it is no longer prudent to apply weight decay with a constant coefficient which continuously pulls the training towards the origin. By plugging in $W_{t+1} = W_t$ in (1) we observe that if training converges, it does so at a point where $W_t = -U_t/\lambda$, which might be quite far from an optimum of the loss function. Consequently, with constant decoupled weight decay the convergence guarantees apply to modified optimization dynamics rather than to a minimizer of the original loss function.

This motivates the following question:

Can weight decay be applied in a principled manner that empirically improves the training process while retaining the stationarity guarantees of the corresponding unregularized methods?

Taking inspiration from the Robbins–Monro conditions (Robbins & Monro, 1951), we propose that instead of a constant weight decay it should be scaled by the fraction of the peak learning rate $\eta_t/\eta_{\max}$, where $\eta_{\max}$ is the maximum learning rate; that is, the base decay $\eta_t\lambda$ is further multiplied by $\eta_t/\eta_{\max}$

$$W_{t+1} = \left(1 - \frac{\eta_t^2}{\eta_{\max}}\lambda\right)W_t - \eta_t U_t \ . \tag{2}$$

As a result, the impact of weight decay decreases as learning rate decreases towards the end of training. Using this method, we answer the above question in the affirmative and make the following contributions:

- By incorporating scaled weight decay into Muon, we derive Muon-SW and show that it substantially speeds up training of mixture-of-experts (MoE) (Fedus et al., 2022; Shazeer et al., 2017) transformer models of size ranging from $72 - 930$ M parameters trained on $18 - 200$ B tokens corresponding to 600 tokens per active parameter. The speed-up generally increases with scale and is largest at width 1024, where Muon-SW reaches the same validation loss in approximately $\mathbf{30}\%$ fewer optimizer steps than Muon with identical hyperparameters, as reported in Table 1.
- Employing a steady-state analysis, we explain why under standard weight decay the weight norm shrinks steadily after warmup, whereas under scaled weight decay it settles to a roughly constant value. This is illustrated in the right panel of Figure 1 and in Figure 3.
- We prove that, for SGD (stochastic gradient descent) and exact-polar Muon with ordinary momentum, scaled weight decay retains the asymptotic stationarity guarantees of the corresponding unregularized methods under the same conditions (Theorems 3.1 and 3.2). In contrast, we show that constant weight decay can remain bounded away from a minimizer on simple quadratic objectives (Propositions 3.1 and 3.2), as illustrated in Figure 2.

Defazio (Defazio, 2025) introduces the same rule to counteract the growth of gradient norms late in training, a motivation that does not apply to update-normalizing optimizers like Muon. Instead, we motivate the rule through Robbins–Monro summability and show that the scaled shrink term preserves the asymptotic stationarity guarantee for the original unregularized objective. We also conduct a thorough empirical investigation across model sizes and training horizons for MoE models trained with Muon. Since the speedups are substantial, if our observed trend continues then scaled weight decay offers substantial gains for frontier training while requiring only a few lines of code to implement and incurring no additional computational overhead. Although we develop and evaluate scaled weight decay for Muon, the same principle should apply to other optimizers for matrix-valued weights such as Shampoo (Anil et al., 2021; Gupta et al., 2018), SOAP (Vyas et al., 2025), Aurora (Dewulf et al., 2026), and other variants of Muon (Ahn et al., 2025a,b; Amsel et al., 2026; An et al., 2025; Apte et al., 2026; Khaled et al., 2025; Liu et al., 2026).

2 Scaled Weight Decay and Muon-SW

We consider a smooth, lower-bounded objective $f(W) = \mathbb{E}_{z\sim\mathcal{D}}[\ell(W; z)]$, estimated using mini-batch gradients $G_t = \nabla f(W_t) + \xi_t$ with zero-mean noise and bounded variance (Bottou et al., 2018). An optimizer uses current and past gradients to form an update U_t. SGD uses the gradient directly, AdamW applies coordinate-wise rescaling, and Muon orthogonalizes a momentum buffer. Decoupled weight decay contracts the weights separately from this transformation, so its accumulated effect depends on the learning-rate schedule.

For SGD, the Robbins–Monro conditions (Robbins & Monro, 1951),

$$\sum_t \eta_t = \infty \qquad \text{and} \qquad \sum_t \eta_t^2 < \infty, \tag{3}$$

allow an unbounded step budget while controlling accumulated noise. For bounded iterates, the scaled shrink term $\lambda\eta_t^2 W_t/\eta_{\max}$ is summable, whereas standard decay $\lambda\eta_t W_t$ need not be. Thus, the optimizer can continue making progress while the total additional shrinkage remains finite. By contrast, an order-η_t shrink term can balance a nonzero gradient and change the limiting optimization target. This motivates (2), whose stationarity guarantees for SGD and Muon are established in Section 3.

Scaling by $\eta_t/\eta_{\max}$ also preserves the original decay strength at the peak learning rate. As the schedule decays, shrinkage weakens quadratically rather than linearly, reducing its influence late in training. The rule introduces no additional hyperparameter beyond the peak learning rate already specified by the schedule.

For ordinary momentum, Muon forms $M_t = \beta M_{t-1} + G_t$ and takes its polar factor $O_t = UV^\top$, where $M_t = U\Sigma V^\top$. The polar factor replaces nonzero singular values by one, retaining the singular vectors. Equivalently, it maximizes $\langle O, M_t\rangle$ subject to $\|O\|_{\mathrm{op}} \le 1$, so its spectral norm does not grow with the gradient magnitude. In practice, five Newton–Schulz iterations (NS5) replace exact orthogonalization, optionally applied to a Nesterov direction as in Algorithm 1. With the

shape-scaled learning rate $\widehat{\eta}_t = 0.2\eta_t\sqrt{\max(m,n)}$, Muon-SW updates an $m \times n$ weight matrix as

$$W_{t+1} = \left(1 - \lambda\eta_t^2/\eta_{\max}\right)W_t - \widehat{\eta}_t O_t. \tag{4}$$

The only change from standard Muon is to replace $\lambda\eta_t$ by $\lambda\eta_t^2/\eta_{\max}$. This decay uses the base learning rate η_t, while the shape factor matches the RMS scale of AdamW updates (Liu et al., 2025). Embeddings, normalization weights, and other excluded parameter groups are trained with AdamW. Algorithm 1 summarizes the update, and Appendix F gives the implementation. The convergence analysis absorbs fixed shape scaling into its effective step-size notation.

Algorithm 1 Muon-SW

Input: initial weights $W_0 \in \mathbb{R}^{m\times n}$, loss $\mathcal{L}$, base learning-rate schedule $\{\eta_t\}$, momentum β, weight decay λ, peak learning rate $\eta_{\max}$, `nesterov` (`True`/`False`)
Initialize $M_{-1} \in \mathbb{R}^{m\times n} \leftarrow 0$
for $t = 0, 1, 2, \ldots$ **do**
  $G_t \leftarrow \nabla_W \mathcal{L}(W_t)$
  $M_t \leftarrow \beta M_{t-1} + (1-\beta)G_t$
  $D_t \leftarrow (1-\beta)G_t + \beta M_t$ if `nesterov=True`, otherwise $D_t \leftarrow M_t$
  $O_t \leftarrow \mathrm{NS5}(D_t)$
  $\widehat{\eta}_t \leftarrow 0.2\,\eta_t\sqrt{\max(m,n)}$
  $W_{t+1} \leftarrow \left(1 - \lambda\eta_t^2/\eta_{\max}\right)W_t - \widehat{\eta}_t\, O_t$
end for

3 Convergence Theory

The results below contrast constant and scaled weight decay under a decaying learning rate. The Muon proofs use ordinary momentum and exact polar factors. Remark 1 extends them to Nesterov, with the same deterministic conclusions and a stochastic bound that recovers Theorem 3.2 for $\beta \geq 1/\sqrt{2}$, including $\beta = 0.95$. Finite-step NS5 is not covered. Normalizing the ordinary momentum buffer by $1-\beta$ leaves its exact polar factor unchanged.

3.1 Constant weight decay converges to a non-minimizer

Recall from the introduction that with constant decay the iteration balances the gradient update against the shrink term, so at convergence $W_t = -U_t/\lambda$. Because the original loss f need not be small at this shifted point, constant weight decay can settle at a higher loss than the true minimum.

To make this precise we exhibit a simple one-dimensional example on which SGD with constant weight decay provably fails to converge to the minimizer. We take the objective to be deterministic so that no stochasticity is involved, which only strengthens the conclusion. Consider the shifted quadratic

$$f(w) = \frac{a}{2}(w-c)^2, \qquad a > 0,\ c \neq 0\,, \tag{5}$$

whose unique minimizer is $w^\star = c$. Gradient descent with constant decoupled weight decay $\lambda > 0$ gives the update

$$w_{t+1} = \big(1 - \eta_t(a+\lambda)\big)w_t + \eta_t\, ac\,. \tag{6}$$

Proposition 3.1. *Let the learning-rate schedule satisfy $0 < \eta_t \leq 1/(a+\lambda)$ and $\sum_t \eta_t = \infty$. Then from every starting point w_0, the iterates (6) converge to*

$$\bar{w} = \frac{ac}{a+\lambda} \neq c\,, \tag{7}$$

with loss gap

$$f(\bar{w}) - f(c) = \frac{a}{2}\left(\frac{\lambda}{a+\lambda}\right)^2 c^2 > 0\,. \tag{8}$$

In particular the iterates never converge to the minimizer $w^\star = c$.

The proof is given in Appendix D.1. The hypothesis $\sum_t \eta_t = \infty$ is exactly the first Robbins–Monro condition, and it is the natural regime in which the optimizer is allowed to travel arbitrarily far and thus has any chance of converging from an arbitrary initialization. In this regime, the failure in Proposition 3.1 is a property of the dynamics rather than of the initialization or the horizon. If instead $\sum_t \eta_t < \infty$ the iterate simply runs out of step budget and freezes at a point determined by the schedule and w_0; it can then land on the minimizer only for a finely tuned starting point, which is a measure-zero set.

The same failure occurs for Muon in a qualitatively different form. Consider the shifted matrix quadratic

$$f(W) = \frac{1}{2}\|W - A\|_F^2, \qquad \nabla f(W) = W - A\,, \tag{9}$$

with $A \in \mathbb{R}^{m\times n}$ and unique minimizer $W^\star = A$. Muon with momentum $\beta \in [0, 1)$ forms

$$M_t = \beta M_{t-1} + \nabla f(W_t), \qquad O_t = \mathrm{polar}(M_t)\,, \tag{10}$$

where $\mathrm{polar}(C) = UV^\top$ for a compact singular value decomposition $C = U\Sigma V^\top$, so that $\|O_t\|_{\mathrm{op}} \le 1$. The constant-decay Muon update is

$$W_{t+1} = (1 - \eta_t\lambda)W_t - \eta_t O_t\,. \tag{11}$$

Proposition 3.2. *Let $\lambda > 0$, suppose $0 < \eta_t\lambda \le 1$ and $\sum_t \eta_t = \infty$, and consider the Muon iteration (10)–(11) on the objective (9). Then, regardless of the momentum coefficient β and the initialization W_0,*

$$\limsup_{t\to\infty} \|W_t\|_{\mathrm{op}} \le \frac{1}{\lambda}\,. \tag{12}$$

Consequently, if $\sigma_{\max}(A) > 1/\lambda$, then

$$\liminf_{t\to\infty} \|W_t - A\|_{\mathrm{op}} \ge \sigma_{\max}(A) - \frac{1}{\lambda} > 0\,, \tag{13}$$

so the iterates never converge to the minimizer $W^\star = A$, and the loss stays bounded below by

$$\liminf_{t\to\infty} f(W_t) \ge \frac{1}{2}\sum_i \Big(\sigma_i(A) - \frac{1}{\lambda}\Big)_+^2\,, \tag{14}$$

where $\sigma_i(A)$ are the singular values of A and $(x)_+ = \max(x, 0)$.

The proof is given in Appendix D.2. In contrast to the scalar case, the orthogonalized Muon direction has spectral norm at most 1 irrespective of the gradient magnitude. A constant decay therefore confines every singular value of W_t to a ball of radius $1/\lambda$ set by λ alone, and any target outside that ball is missed. For training of transformers, spectral norms larger than $1/\lambda$ can arise in the MLP blocks (see Figure 1 in (Lion et al., 2026)).

Proposition 3.2 also shows that the convergence guarantee for Muon with constant weight decay claimed by Sato et al. (Sato et al., 2026) is incorrect. Their Lemma C.1 in fact bounds $\|W_t\|_{\mathrm{op}}$ by $1/\lambda$ rather than 1, so the strictly positive descent coefficient their Theorem C.1 relies on is unavailable when $\lambda < 1$; taking A with $\sigma_{\max}(A) > 1/\lambda$ gives an explicit counterexample.

3.2 Convergence of scaled weight decay for SGD

In contrast to the behavior for constant weight decay, we now show that scaled weight decay preserves the standard weighted stationarity guarantee for the original objective. We begin with the stochastic gradient descent (SGD) case, where the argument is the simplest. Let $F : \mathbb{R}^d \to \mathbb{R}$ be bounded below by $F_{\inf}$ and L-smooth. At each step we observe a stochastic gradient g_t satisfying

$$\mathbb{E}[g_t \mid \mathcal{F}_t] = \nabla F(x_t), \qquad \mathbb{E}[\|g_t\|_2^2 \mid \mathcal{F}_t] \le \sigma^2 + \rho\|\nabla F(x_t)\|_2^2\,, \tag{15}$$

for constants $\sigma^2 \ge 0$ and $\rho \ge 1$, where $\mathcal{F}_t$ is the history through x_t, before sampling g_t. Scaled weight decay with coefficient $\lambda > 0$ uses the update

$$x_{t+1} = x_t - \eta_t g_t - q\,\eta_t^2 x_t, \qquad q = \frac{\lambda}{\eta_{\max}}\,. \tag{16}$$

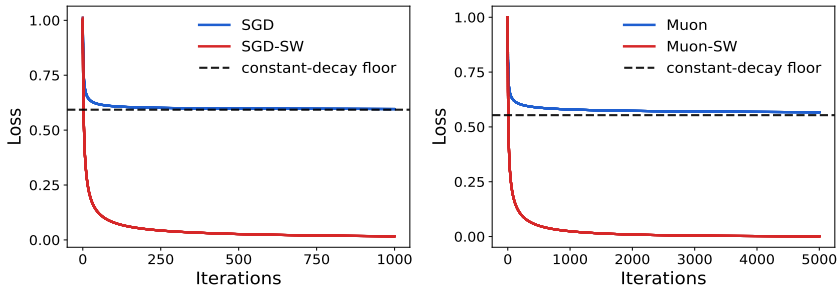


Figure 2: Scaled weight decay converges to a minimizer of the original loss, while constant weight decay does not. **Left:** deterministic gradient descent on the shifted scalar quadratic $f(w) = \frac{a}{2}(w - c)^2$ of Proposition 3.1. **Right:** Muon on the shifted matrix quadratic $f(W) = \frac{1}{2}\|W - A\|_F^2$ of Proposition 3.2, where the target A has a Marchenko–Pastur-type spectrum with several singular values above $1/\lambda$. In each panel we plot the loss (normalized by its mean initial value) for many random initializations $W_0 \sim \mathcal{N}(0, 0.02)$ under the same $\eta_t = \eta_0/(1 + t)$ schedule. The scaled-decay runs (SGD-SW, Muon-SW; red) drive the loss to zero, whereas the constant-decay runs (SGD, Muon; blue) instead flatten at the analytic lower bound (dashed).

Theorem 3.1. *Suppose F and g_t satisfy (15), the iterates are bounded in the sense $\sup_t \mathbb{E}\|x_t\|_2^2 < \infty$, and the learning-rate schedule satisfies the Robbins–Monro conditions $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$. Then the scaled weight decay iterates (16) satisfy*

$$\frac{1}{S_T}\sum_{t=0}^{T-1} \eta_t \,\mathbb{E}\|\nabla F(x_t)\|_2^2 \to 0, \qquad S_T = \sum_{t=0}^{T-1} \eta_t \,, \tag{17}$$

and in particular $\liminf_{t\to\infty} \mathbb{E}\|\nabla F(x_t)\|_2^2 = 0$. Thus scaled weight decay has the same asymptotic weighted stationarity guarantee for the original objective F as SGD without weight decay under the corresponding assumptions.

The proof in Appendix D.3 treats $q\eta_t^2 x_t$ as a summable perturbation to the descent inequality. For strongly convex objectives, Theorem D.2 also gives convergence to the unique minimizer without a separate bounded-iterate assumption, with rate $O(1/t)$ under its harmonic-schedule conditions. Applied to the shifted quadratic, this recovers the true minimizer, unlike the constant-decay limit in Proposition 3.1.

3.3 Convergence of scaled weight decay for Muon

We now establish the analogous positive result for Muon. The setting is a matrix objective $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ that is bounded below by $f_{\inf}$ and L-smooth in the Frobenius norm, with a stochastic gradient G_t satisfying $\mathbb{E}[G_t \mid \mathcal{F}_t] = \nabla f(W_t)$ and $\mathbb{E}[\|G_t - \nabla f(W_t)\|_F^2 \mid \mathcal{F}_t] \le \sigma^2/B$, where B is a fixed batch size. Muon with scaled weight decay forms the momentum buffer $M_t = \beta M_{t-1} + G_t$ with fixed $\beta \in [0, 1)$, orthogonalizes it to $O_t = \mathrm{polar}(M_t)$, and updates

$$W_{t+1} = W_t - \eta_t O_t - q\,\eta_t^2 W_t, \qquad q = \frac{\lambda}{\eta_{\max}} \,. \tag{18}$$

We write $r = \min(m, n)$ for the ambient rank bound. Unlike SGD the natural stationarity measure for Muon is the *nuclear norm* $\|\nabla f(W_t)\|_*$ rather than the Frobenius norm. When the orthogonalized direction is formed from the true gradient, spectral–nuclear duality gives $\langle \nabla f(W_t), O_t\rangle_F = \|\nabla f(W_t)\|_*$. With momentum and stochastic gradients this equality is perturbed by the buffer-tracking error, which is controlled in the appendix. For fixed momentum β and a fixed batch B, the noise cannot be fully averaged out, so the guarantee holds only up to an explicit noise floor, which is also the case for Muon without any weight decay.

Theorem 3.2. *Suppose the iterates remain bounded, $\sup_t \mathbb{E}\|W_t\|_F^2 \le \mathcal{B}^2$, and the schedule satisfies $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$. Then Muon with scaled weight decay (18) satisfies*

$$\limsup_{T\to\infty} \frac{1}{S_T}\sum_{t=0}^{T-1} \eta_t \,\mathbb{E}\|\nabla f(W_t)\|_* \;\le\; 2\sigma\sqrt{\frac{(1-\beta)\,r}{B}}, \qquad S_T = \sum_{t=0}^{T-1} \eta_t \,, \tag{19}$$

The bias and drift terms vanish in the limit; the remaining floor is the same noise floor as for Muon without weight decay, since the scaled shrink term enters the descent inequality only at order η_t^2 and contributes a summable perturbation.

The noise floor is due to a fixed batch size and momentum and matches the previous analysis of convergence of Muon without weight decay (Chang et al., 2026; Sato et al., 2026; Shen et al., 2025). The noise term vanishes in the deterministic case $\sigma = 0$. In that regime the weighted average of $\mathbb{E}\|\nabla f(W_t)\|_*$ tends to zero, and when f is in addition convex and the iterates are almost surely bounded this gives a best-iterate guarantee in expected function value. Appendix D.4 gives the proof and the deterministic convex result (Theorem D.4). Applying the convex result to the shifted matrix quadratic shows that scaled-decay Muon approaches the true minimizer along a subsequence, in direct contrast to the constant-decay trap of Proposition 3.2.

All of these guarantees describe a single trajectory run with a decaying schedule $\eta_t \to 0$, which is where scaled and constant weight decay genuinely differ. The constant-decay perturbation accumulates as $\sum_t \eta_t$ and shifts the fixed point, whereas the scaled-decay perturbation is summable and thus does not. However, this separation does not arise at a fixed learning rate, where $q\eta^2$ and $\lambda\eta$ are both constants and the two updates coincide. Figure 2 summarizes the full picture, contrasting the scaled and constant weight decay for both SGD and Muon on shifted quadratic objectives.

4 Experiment on Training MoE with Muon-SW

To empirically validate the efficacy of scaled weight decay, we train decoder-only MoE transformers with Muon-SW and compare against standard Muon. Our aim is to evaluate scaled weight decay under optimization conditions that mirror frontier pre-training, albeit in a scaled-down version. As summarized in Appendix A, open pre-training recipes have largely converged on a common pattern. They apply Muon to the matrix-shaped Transformer weights, use a weight decay coefficient of 0.1, decay the learning rate to one tenth of its peak with a cosine schedule, and train for several hundred tokens per active parameter (DeepSeek-AI et al., 2026; GLM-4.5 Team et al., 2025; Kimi Team et al., 2025). We replicate these optimization choices, fixing $\lambda = 0.1$, using the same cosine schedule, and training at 600–650 tokens per active parameter. The architecture is a LLaMA-style MoE decoder with 8 experts and top-2 routing, trained on the FineWeb dataset tokenized with a GPT-2 tokenizer (Penedo et al., 2024; Radford et al., 2019; Touvron et al., 2023). The global batch consists of 256 sequences of length $2{,}048$, corresponding to $256 \times 2{,}048 = 524{,}288$ tokens per optimizer update. Complete architecture and optimizer settings are given in Appendix B. Muon (and Muon-SW) is applied only to the matrix-shaped weights; the embedding, un-embedding, and other non-matrix parameters are trained with AdamW with no weight decay, and we rescale the Muon learning rate by $0.2\sqrt{\max(m, n)}$ for a weight of shape $m \times n$ so that the same learning rate can be used for both optimizers (Liu et al., 2025).

Since hyperparameter tuning is computationally expensive, we employ μP-MoE parameterization to transfer the peak learning rate across widths (Małaśnicki et al., 2025; Yang et al., 2022). We verify this width transfer directly in Appendix C and adopt $\eta = 0.02$ as the transferred peak learning rate. Furthermore, because the optimal peak learning rate is a function of the training horizon, we rescale it by the cube-root rule $\eta^\star(T) = \eta^\star(T_0)(T/T_0)^{-1/3}$, which we also validate in Appendix C (Bjorck et al., 2025; Ren et al., 2026). Combining the two lets us set the peak learning rate for each width and token budget in Table 1 without any per-run tuning, so that the only deliberate change between the Muon and Muon-SW columns is the weight-decay rule itself.

The results in Table 1 show that Muon-SW improves the best validation loss at every width. The speed-up tends to increase with scale, although it is not strictly monotonic across the evaluated widths, and is largest at width 1024. Measured as the reduction in optimizer steps needed for Muon-SW to reach Muon's best loss, the speed-up rises from 21.7% at width 256 to 29.4% at width 1024. While our largest run is orders of magnitude below true frontier scale, the generally larger benefit at greater model sizes and longer training durations suggests that scaled weight decay could substantially accelerate frontier model training if the trend continues.

Table 1: **Comparison of Muon and Muon-SW for MoE training.** All runs use 8 experts with top-2 routing and are trained at 600–650 tokens per active parameter with a constant true batch of 524,288 tokens per optimizer update. The base coefficient $\lambda = 0.1$ is held fixed for both methods, and each run uses a linear warmup of $\max\{1000,\ 0.01\,T\}$ steps for total steps T followed by a cosine decay to $1/10$ of the peak learning rate by the end of training. Peak learning rates are muP-transferred across width and then adjusted based on training duration. We report the best validation loss for Muon (standard constant weight decay) and Muon-SW (scaled weight decay); the speed-up is the reduction in optimizer steps for Muon-SW to reach Muon's best validation loss.

Width	Total	Active	Steps	Tokens	Tok./Act.	Peak LR	Loss (Muon / SW)	Speed-up
256	72.7M	30.2M	35k	18.4B	~ 610	1.0×10^{-2}	3.350 / **3.297**	21.7%
512	264.9M	95.0M	110k	57.7B	~ 610	7.0×10^{-3}	2.971 / **2.915**	27.5%
768	520.0M	180.3M	223k	116.9B	~ 650	5.6×10^{-3}	2.799 / **2.754**	26.5%
1024	932.4M	309.6M	376k	197.1B	~ 640	4.8×10^{-3}	2.675 / **2.630**	29.4%

5 Weight Norms under Constant and Scaled Weight Decay

For a weight matrix $W_t \in \mathbb{R}^{m\times n}$, write the Muon update as $W_{t+1} = (1-\varphi_t)W_t - \widehat{\eta}_t O_t$, where $\varphi_t = \lambda\eta_t$ for constant decay and $\varphi_t = \lambda\eta_t^2/\eta_{\max}$ for scaled decay. We track the RMS norm $r_t = \|W_t\|_F/\sqrt{mn}$ and weight–update alignment $a_t = \langle W_t, O_t\rangle/(\|W_t\|_F\|O_t\|_F)$.

For scale-invariant weights, the raw gradient is orthogonal to the weights, as used in prior norm analyses (Defazio, 2025; Kosson et al., 2024; van Laarhoven, 2017). Momentum and its polar transformation need not inherit this orthogonality (Figure 7). Empirically, $a_t < 0$, so the Muon update pushes weights outward. For an exact full-rank polar factor, $\|O_t\|_F^2 = \min(m,n)$. Expanding the squared update norm and dividing by mn cancels the shape factors. Dropping terms of order φ_t^2 and $\varphi_t\eta_t$ gives

$$r_{t+1}^2 \approx (1-2\varphi_t)r_t^2 - 2(0.2\eta_t)a_t r_t + (0.2\eta_t)^2. \tag{20}$$

The omitted terms are small since $\varphi_t \lesssim 10^{-3}$ in our experiments. Setting $r_{t+1} = r_t = r_t^\star$ and taking the positive root yields the quasi-steady target

$$r_t^\star = \frac{0.2\eta_t}{2\varphi_t}\Big(|a_t| + \sqrt{a_t^2 + 2\varphi_t}\Big). \tag{21}$$

This target reflects the balance between decay and the outward update. In practice, NS5 produces a normalized Frobenius norm near 0.88 rather than one (Figure 6), so we use the measured update magnitude in place of $0.2\eta_t$ when evaluating the prediction in Figure 3. Tracking requires the norm to relax faster than the target moves. The decay term in (20) defines a characteristic timescale $\tau_{\mathrm{decay},t} = (2\varphi_t)^{-1}$, although the full local relaxation also depends on alignment. The norm can follow the target when this relaxation is fast relative to $|\mathrm{d}\ln r_t^\star/\mathrm{d}t|^{-1}$. The decay timescale grows as η_t^{-1} under constant decay and η_t^{-2} under scaled decay, so tracking becomes harder late in training. Figure 3 marks the observed departure near 300k steps, where the quasi-steady approximation loses accuracy without invalidating the norm recursion.

The contrasting targets depend on how alignment varies with learning rate. The late-run fits in Figure 8 give $a_t \approx -k\eta_t$ under scaled decay and $a_t \approx -a_0 - k'\eta_t$ under constant decay, with $k \approx 20.8$ and $a_0 \approx 7.4\times10^{-3}$. Substituting into (21), the powers of η_t cancel for scaled decay, whereas the constant-decay limit depends on the alignment offset, giving

$$r_{\mathrm{scaled}}^\star \to \frac{0.2\eta_{\max}}{2\lambda}\Big(k + \sqrt{k^2 + 2\lambda/\eta_{\max}}\Big), \tag{22}$$

$$r_{\mathrm{constant}}^\star \to \frac{0.2a_0}{\lambda} \qquad (\eta_t \to 0). \tag{23}$$

At $\lambda = 0.1$ and $\eta_{\max} = 4.8\times10^{-3}$, these extrapolated limits are approximately 0.20 and 0.015. The experiments stop at $\eta_t = \eta_{\max}/10$, where the observed norms already differ by a factor of 5.4 (0.16 versus 0.029). Constant-decay norms fall about 60% from their peak while scaled-decay norms remain roughly constant. Widths 512 and 768 show the same contrast (Figures 9 and 10).

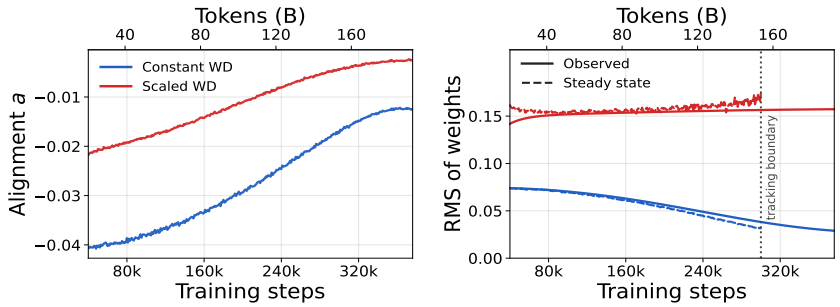


Figure 3: **Weight–update alignment and steady-state weight norm (width 1024).** *Left.* The aggregate alignment a_t between the weights and the Muon update is negative under both rules, with its magnitude decreasing through training as the learning rate and relative update size fall. *Right.* The observed RMS weight norm (solid) against the instantaneous steady-state prediction (dashed). The theory captures the separation between the two, with scaled weight decay sustaining a nearly constant norm, whereas constant weight decay produces a norm that falls by roughly 60% from its peak value. The steady-state approximation is predictive before the marked tracking boundary near 300k steps. Beyond this boundary, relaxation is too slow for the norm to follow its moving target.

6 Related Work

Weight decay was introduced as ℓ_2 regularization (Krogh & Hertz, 1991) and later decoupled from adaptive updates in AdamW (Loshchilov & Hutter, 2019). Beyond generalization, weight decay affects language-model training stability and the bias–variance tradeoff (D'Angelo et al., 2024). Alternatives include cautious decay (Chen et al., 2026), gradient-norm-dependent decay (Xie et al., 2022), and hyperspherical optimization without decay (Ren et al., 2026; Wen et al., 2026). The dynamics of weight norms have been studied in normalized networks (van Laarhoven, 2017), Adam and Lion (Kosson et al., 2024), and schedule-free spectral optimization (Apte et al., 2026).

The closest work is that of Defazio (2025), who proposes the same $\eta_t/\eta_{\max}$ scaling to counter rising gradient norms late in training. His argument relies on varying gradient norms and gradient–weight orthogonality, which do not describe Muon's applied updates. We account for their approximate normalization and nonzero alignment with the weights, justify the decay rule through Robbins–Monro summability, and evaluate its benefits across MoE model sizes and training horizons. Inkling also uses Muon with a decay step proportional to $\lambda\eta_t^2$ and reports stable weight sizes across training horizons (Thinking Machines Lab, 2026), consistent with Figures 1 and 3.

7 Conclusion and Future Work

Drawing on the Robbins–Monro conditions (Robbins & Monro, 1951), we scale weight decay by $\eta_t/\eta_{\max}$. The resulting Muon-SW optimizer reaches Muon's validation loss using approximately 30% fewer training tokens at our largest scale. If these gains carry over to frontier models, this simple change could substantially accelerate large-scale pretraining. Our theory shows that scaled decay preserves the stationarity guarantees of unregularized SGD and exact-polar Muon with ordinary momentum under the stated assumptions, while our steady-state analysis explains why it sustains weight norms as the learning rate falls. These benefits require only a one-line change to the decay rule.

An important next step is to test whether the gains extend to architectures such as Mamba, Samba, Gated DeltaNet, and Parallax (Dao & Gu, 2024; Gu & Dao, 2024; Ren et al., 2025; Yang et al., 2025; Zuo et al., 2026), and optimizers such as Shampoo, SOAP, and Aurora (Anil et al., 2021; Dewulf et al., 2026; Gupta et al., 2018; Vyas et al., 2025). It is also important to assess how these pretraining gains affect supervised fine-tuning (Ouyang et al., 2022), preference alignment (Christiano et al., 2017; Rafailov et al., 2024), and reinforcement learning for reasoning (Guo et al., 2025; Shao et al., 2024).

AI USE STATEMENT

Generative AI was used to edit the draft, write code for the computational experiments, and assist with checking proofs.

ETHICS STATEMENT

This work studies an optimization method for neural-network training, with experiments on language models trained using the existing FineWeb dataset. More efficient training could reduce the computational resources needed to reach a given validation loss, although this work does not measure energy consumption or emissions.

REPRODUCIBILITY STATEMENT

The weight-decay rule and convergence results are specified in the main text, with detailed proofs and assumptions in Appendix D. Appendix B describes the model architectures, dataset and tokenizer, hardware, batch size, learning-rate schedule, and optimizer settings. Appendix C documents learning-rate transfer across model widths and training horizons. Section 4 reports the training budgets and comparison protocol, and Appendix E provides additional loss curves and optimization diagnostics. Appendix F includes a PyTorch implementation of Muon-SW, and the experimental codebase is identified in Appendix B.

REFERENCES

Kwangjun Ahn, Noah Amsel, and John Langford. Dion2: A Simple Method to Shrink Matrix in Muon, 2025a. URL https://arxiv.org/abs/2512.16928.

Kwangjun Ahn, Byron Xu, Natalie Abreu, and John Langford. Dion: Distributed Orthonormalized Updates. *arXiv preprint: 2504.05295*, 2025b.

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The Polar Express: Optimal Matrix Sign Methods and Their Application to the Muon Algorithm, 2026. URL https://arxiv.org/abs/2505.16932.

Kang An, Yuxing Liu, Rui Pan, Yi Ren, Shiqian Ma, Donald Goldfarb, and Tong Zhang. ASGO: Adaptive Structured Gradient Optimization, 2025. URL https://arxiv.org/abs/2503.20762.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable Second Order Optimization for Deep Learning, 2021. URL https://arxiv.org/abs/2002.09018.

Anuj Apte, Pranav Deshpande, Niraj Kumar, Shouvanik Chakrabarti, and Junhyung Lyle Kim. Anytime Training with Schedule-Free Spectral Optimization, 2026. URL https://arxiv.org/abs/2605.23061.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024. doi: 10.1073/pnas.2311878121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2311878121.

Johan Bjorck, Alon Benhaim, Vishrav Chaudhary, Furu Wei, and Xia Song. Scaling Optimal LR Across Token Horizons, 2025. URL https://arxiv.org/abs/2409.19913.

Léon Bottou, Frank E. Curtis, and Jorge Nocedal. Optimization Methods for Large-Scale Machine Learning. *SIAM Review*, 60(2):223–311, 2018.

Da Chang, Yongxiang Liu, and Ganzhao Yuan. On the Convergence of Muon and Beyond, 2026. URL https://arxiv.org/abs/2509.15816.

Lizhang Chen, Jonathan Li, Kaizhao Liang, Baiyu Su, Cong Xie, Nuo Wang Pierse, Chen Liang, Ni Lao, and Qiang Liu. Cautious Weight Decay, 2026. URL https://arxiv.org/abs/2510.12402.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2017. URL https://arxiv.org/abs/1706.03741.

Francesco D'Angelo, Maksym Andriushchenko, Aditya Varre, and Nicolas Flammarion. Why Do We Need Weight Decay in Modern Deep Learning?, 2024. URL https://arxiv.org/abs/2310.04415.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality, 2024. URL https://arxiv.org/abs/2405.21060.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang

Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. DeepSeek-V4: Towards Highly Efficient Million-Token Context Intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Aaron Defazio. Why Gradients Rapidly Increase Near the End of Training, 2025. URL https://arxiv.org/abs/2506.02285.

Alec Dewulf, Dhruv Pai, Li Yang, Ashley Zhang, and Ben Keigwin. Aurora: A Leverage-Aware Spectral Optimizer, 2026. URL https://arxiv.org/abs/2606.27715.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021. URL https://arxiv.org/abs/2010.11929.

Ege Erdil. Inference economics of language models, 2025. URL https://arxiv.org/abs/2506.04645.

William Fedus, Barret Zoph, and Noam Shazeer. Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity, 2022. URL https://arxiv.org/abs/2101.03961.

GLM-4.5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, Yean Cheng, Yifan An, Yilin Niu, Yuanhao Wen, Yushi Bai, Zhengxiao Du, Zihan Wang, Zilin Zhu, Bohan Zhang, Bosi Wen, Bowen Wu, Bowen Xu, Can Huang, Casey Zhao, Changpeng Cai, Chao Yu, Chen Li, Chendi Ge, Chenghua Huang, Chenhui Zhang, Chenxi Xu, Chenzheng Zhu, Chuang Li, Congfeng Yin,

Daoyan Lin, Dayong Yang, Dazhi Jiang, Ding Ai, Erle Zhu, Fei Wang, Gengzheng Pan, Guo Wang, Hailong Sun, Haitao Li, Haiyang Li, Haiyi Hu, Hanyu Zhang, Hao Peng, Hao Tai, Haoke Zhang, Haoran Wang, Haoyu Yang, He Liu, He Zhao, Hongwei Liu, Hongxi Yan, Huan Liu, Huilong Chen, Ji Li, Jiajing Zhao, Jiamin Ren, Jian Jiao, Jiani Zhao, Jianyang Yan, Jiaqi Wang, Jiayi Gui, Jiayue Zhao, Jie Liu, Jijie Li, Jing Li, Jing Lu, Jingsen Wang, Jingwei Yuan, Jingxuan Li, Jingzhao Du, Jinhua Du, Jinxin Liu, Junkai Zhi, Junli Gao, Ke Wang, Lekang Yang, Liang Xu, Lin Fan, Lindong Wu, Lintao Ding, Lu Wang, Man Zhang, Minghao Li, Minghuan Xu, Mingming Zhao, Mingshu Zhai, Pengfan Du, Qian Dong, Shangde Lei, Shangqing Tu, Shangtong Yang, Shaoyou Lu, Shijie Li, Shuang Li, Shuang-Li, Shuxun Yang, Sibo Yi, Tianshu Yu, Wei Tian, Weihan Wang, Wenbo Yu, Weng Lam Tam, Wenjie Liang, Wentao Liu, Xiao Wang, Xiaohan Jia, Xiaotao Gu, Xiaoying Ling, Xin Wang, Xing Fan, Xingru Pan, Xinyuan Zhang, Xinze Zhang, Xiuqing Fu, Xunkai Zhang, Yabo Xu, Yandong Wu, Yida Lu, Yidong Wang, Yilin Zhou, Yiming Pan, Ying Zhang, Yingli Wang, Yingru Li, Yinpei Su, Yipeng Geng, Yitong Zhu, Yongkun Yang, Yuhang Li, Yuhao Wu, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yuxuan Zhang, Zezhen Liu, Zhen Yang, Zhengda Zhou, Zhongpei Qiao, Zhuoer Feng, Zhuorui Liu, Zichen Zhang, Zihan Wang, Zijun Yao, Zikang Wang, Ziqiang Liu, Ziwei Chai, Zixuan Li, Zuodong Zhao, Wenguang Chen, Jidong Zhai, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. GLM-4.5: Agentic, Reasoning, and Coding (ARC) Foundation Models, 2025. URL https://arxiv.org/abs/2508.06471.

Albert Gu and Tri Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, 2024. URL https://arxiv.org/abs/2312.00752.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned Stochastic Tensor Optimization, 2018. URL https://arxiv.org/abs/1802.09568.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon

Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training Compute-Optimal Large Language Models, 2022. URL https://arxiv.org/abs/2203.15556.

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/.

Dayal Singh Kalra and Maissam Barkeshli. Why Warmup the Learning Rate? Underlying Mechanisms and Improvements, 2024. URL https://arxiv.org/abs/2406.09405.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling Laws for Neural Language Models, 2020. URL https://arxiv.org/abs/2001.08361.

Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. MuonBP: Faster Muon via Block-Periodic Orthogonalization, 2025. URL https://arxiv.org/abs/2510.16981.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Chenxiao Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Qizheng Gu, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yang Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Haoyu Lu, Lijun Lu, Yashuo Luo, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Zeyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Lin Sui, Xinjie Sun, Flood Sung, Yunpeng Tai, Heyi Tang, Jiawen Tao, Qifeng Teng, Chaoran Tian, Chensi Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Si Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Haoning Wu, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Jing Xu, Jing Xu, Junjie Yan, Yuzi Yan, Hao Yang, Xiaofei Yang, Yi Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Siyu Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Shaojie Zheng, Longguang Zhong, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi K2: Open Agentic Intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

Atli Kosson, Bettina Messmer, and Martin Jaggi. Rotational Equilibrium: How Weight Decay Balances Learning Across Neural Networks, 2024. URL https://arxiv.org/abs/2305.17212.

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. Scaling Laws for Fine-Grained Mixture of Experts, 2024. URL https://arxiv.org/abs/2402.07871.

Anders Krogh and John Hertz. A Simple Weight Decay Can Improve Generalization. In J. Moody, S. Hanson, and R.P. Lippmann (eds.), *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991. URL https://proceedings.neurips.cc/paper_files/paper/1991/file/8eefcfdf5990e441f0fb6f3fad709e21-Paper.pdf.

Kai Lion, Florian Hübler, Bingcong Li, Antonio Orvieto, and Niao He. Muown: Row-Norm Control for Muon Optimization, 2026. URL https://arxiv.org/abs/2605.10797.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is Scalable for LLM Training, 2025. URL https://arxiv.org/abs/2502.16982.

Ziyue Liu, Ruijie Zhang, Zhengyang Wang, Yequan Zhao, Yupeng Su, Zi Yang, and Zheng Zhang. Muon^2: Boosting Muon via Adaptive Second-Moment Preconditioning, 2026. URL https://arxiv.org/abs/2604.09967.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts, 2017. URL https://arxiv.org/abs/1608.03983.

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Potsawee Manakul, Woody Haosheng Gan, Martijn Bartelds, Guangzhi Sun, William Held, and Diyi Yang. Scaling Open Discrete Audio Foundation Models with Interleaved Semantic, Acoustic, and Text Tokens, 2026. URL https://arxiv.org/abs/2602.16687.

Jan Małaśnicki, Kamil Ciebiera, Mateusz Boruń, Maciej Pióro, Jan Ludziejewski, Maciej Stefaniak, Michał Krutul, Sebastian Jaszczur, Marek Cygan, Kamil Adamczewski, and Jakub Krajewski. μ-Parametrization for Mixture of Experts, 2025. URL https://arxiv.org/abs/2508.09752.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library, 2019. URL https://arxiv.org/abs/1912.01703.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale, 2024. URL https://arxiv.org/abs/2406.17557.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners, 2019.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct Preference Optimization: Your Language Model is Secretly a Reward Model, 2024. URL https://arxiv.org/abs/2305.18290.

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple Hybrid State Space Models for Efficient Unlimited Context Language Modeling, 2025. URL https://arxiv.org/abs/2406.07522.

Liliang Ren, Yang Liu, Yelong Shen, and Weizhu Chen. Rethinking Language Model Scaling under Transferable Hypersphere Optimization, 2026. URL https://arxiv.org/abs/2603.28743.

Herbert Robbins and Sutton Monro. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407, September 1951. ISSN 0003-4851. doi: 10.1214/aoms/1177729586. URL http://dx.doi.org/10.1214/aoms/1177729586.

Sebastian Sartor and Neil Thompson. Neural Scaling Laws in Robotics, 2025. URL https://arxiv.org/abs/2405.14005.

Naoki Sato, Hiroki Naganuma, and Hideaki Iiduka. Convergence Bound and Critical Batch Size of Muon Optimizer, 2026. URL https://arxiv.org/abs/2507.01598.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking Optimizers for Large Language Model Pretraining. *arXiv preprint arXiv:2509.01440*, 2025. URL https://arxiv.org/abs/2509.01440.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models, 2024. URL https://arxiv.org/abs/2402.03300.

Noam Shazeer. GLU Variants Improve Transformer, 2020. URL https://arxiv.org/abs/2002.05202.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer, 2017. URL https://arxiv.org/abs/1701.06538.

Wei Shen, Ruichuan Huang, Minhui Huang, Cong Shen, and Jiawei Zhang. On the Convergence Analysis of Muon. *arXiv preprint arXiv:2505.23737*, 2025.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced Transformer with Rotary Position Embedding, 2021. URL https://arxiv.org/abs/2104.09864.

Thinking Machines Lab. Inkling: Our open-weights model, 2026. URL https://thinkingmachines.ai/news/introducing-inkling/.

Changxin Tian, Kunlong Chen, Jia Liu, Ziqi Liu, Zhiqiang Zhang, and Jun Zhou. Towards Greater Leverage: Scaling Laws for Efficient Mixture-of-Experts Language Models, 2025. URL https://arxiv.org/abs/2507.17702.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023. URL https://arxiv.org/abs/2307.09288.

Twan van Laarhoven. L2 Regularization versus Batch and Weight Normalization, 2017. URL https://arxiv.org/abs/1706.05350.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and Stabilizing Shampoo using Adam, 2025. URL https://arxiv.org/abs/2409.11321.

Kaiyue Wen, Xingyu Dang, Kaifeng Lyu, Tengyu Ma, and Percy Liang. Fantastic Pretraining Optimizers and Where to Find Them II: Hyperball Optimization, 2026. URL https://arxiv.org/abs/2606.16899.

Zeke Xie, Issei Sato, and Masashi Sugiyama. Understanding and Scheduling Weight Decay, 2022. URL https://openreview.net/forum?id=J7V_4aauV6B.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor Programs V: Tuning Large Neural Networks via Zero-Shot Hyperparameter Transfer, 2022. URL https://arxiv.org/abs/2203.03466.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor Programs VI: Feature Learning in Infinite-Depth Neural Networks, 2023. URL https://arxiv.org/abs/2310.02244.

Greg Yang, James B. Simon, and Jeremy Bernstein. A Spectral Condition for Feature Learning, 2024. URL https://arxiv.org/abs/2310.17813.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with Delta Rule, 2025. URL https://arxiv.org/abs/2412.06464.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling Vision Transformers, 2022. URL https://arxiv.org/abs/2106.04560.

Biao Zhang and Rico Sennrich. Root Mean Square Layer Normalization. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/1e8a19426224ca89e83cef47f1e7f53b-Paper.pdf.

Yifei Zuo, Dhruv Pai, Zhichen Zeng, Alec Dewulf, Shuming Hu, and Zhaoran Wang. Parallax: Parameterized Local Linear Attention for Language Modeling, 2026. URL https://arxiv.org/abs/2605.29157.

TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

In this appendix, we provide additional background, experimental details, proofs, and implementation details that supplement the main text. This appendix is organized as follows:

- In appendix A, we review the training recipes of recent open frontier Mixture-of-Experts models and in particular the number of tokens used in training per active model parameter.
- In appendix B, we describe the MoE architecture and the hyperparameter choices used across all experiments.
- In appendix C, we review μP for MoE models and the learning-rate transfer rules used to tune learning rate across width and training horizon.
- In appendix D, we give detailed proofs deferred from the main text, including the non-convergence of constant weight decay and convergence of scaled weight decay.
- In appendix E, we present additional data from the MoE training experiments.
- In appendix F, we provide a PyTorch implementation of the Muon-SW algorithm with scaled weight decay.

A TRAINING DETAILS OF OPEN FRONTIER MODELS

Open frontier Mixture-of-Experts (MoE) models released since 2025 with disclosed training recipes have adopted Muon as the main optimizer over AdamW (DeepSeek-AI et al., 2026; GLM-4.5 Team et al., 2025; Kimi Team et al., 2025). AdamW is still retained as the optimizer for selected parameter groups such as embeddings, normalization weights, or output heads. This combination improves optimization efficiency and final model quality relative to AdamW baselines. Rescaling the learning rate of Muon based on the shape of the parameters, allows one to use the same learning rate for both the optimizers when employed in conjunction.

To make the comparison concrete, Table 2 collects the parts of these pre-training recipes that are most relevant for this paper: the active model size, the token horizon, the learning-rate schedule, and the Muon weight-decay value. The striking point is that the recipes have converged on a similar optimizer pattern: Muon is used for matrix-shaped Transformer weights, the nominal weight decay is around 0.1, the cosine schedule ends at roughly one tenth of the peak learning rate, and the training horizon is several hundred tokens per active parameter. We replicate this qualitative regime in our experiments, which are more modest in scale but still allow us to study how changing from constant to scaled weight decay affects the optimization dynamics and speed of convergence.

Table 2: Model and pre-training details for recent open frontier MoE models. Total and active parameter counts are reported in parameters, tokens are reported in training tokens, and Tok./Act. is computed by dividing reported training tokens by activated parameters. The weight decay value is for matrix-shaped parameters trained with Muon, excluding embedding and unembedding layers.

Model	Total	Active	Tokens	Peak LR	Final LR	WD	Tok./Act.
GLM-4.5	355B	32B	22T	2.5×10^{-4}	2.5×10^{-5}	0.1	~ 700
Kimi K2	1T	32B	15.5T	2.0×10^{-4}	2.0×10^{-5}	0.1	~ 480
DeepSeek-V4-Pro	1.6T	49B	32T	2.0×10^{-4}	2.0×10^{-5}	0.1	~ 650

A second notable observation is that these models are trained on far more tokens per active parameter than the dense Chinchilla heuristic of roughly 20 tokens per parameter (Hoffmann et al., 2022). For MoE models, this comparison should be made against active rather than total parameters, since training and inference FLOPs are governed primarily by the routed subset of parameters. Scaling laws for MoE models indicate that increasing the number of experts can itself shift the compute-optimal allocation toward fewer active parameters and more training tokens, raising the optimal

tokens-per-active-parameter ratio relative to dense models (Krajewski et al., 2024; Tian et al., 2025). Even taking this into account, the observed ratios are much larger than compute-optimal scaling would suggest.

Frontier MoE model development is not only optimized for pre-training compute efficiency, but must also account for deployment efficiency and serving cost. Since extra pre-training cost is paid once, while active-parameter inference cost is paid on every token served, it is economically rational to train a smaller active model on substantially more data. Thus, the very high tokens-per-active-parameter ratios observed in recent frontier MoEs likely reflect both MoE scaling behavior and lifetime-cost optimization (Erdil, 2025).

B Architecture and Hyperparameter Details

All experiments use the same μP LLaMA-style (Touvron et al., 2023) sparse MoE decoder, trained on FineWeb (Penedo et al., 2024) with GPT-2 tokenization (Radford et al., 2019). Each model has 12 layers, head dimension 64, 8 experts per layer, and top-2 routing, so each token activates two experts per MoE layer. Each block is pre-norm, with RMS-Norm before attention and before the MoE MLP, plus a final RMSNorm before the tied output head (Zhang & Sennrich, 2019). We use RoPE with base $\theta = 10000$ (Su et al., 2021), bias-free linear layers, no dropout, a SiLU-gated SwiGLU expert MLP (Shazeer, 2020), and standard MoE routing with `topk_softmax` (Fedus et al., 2022). Router regularization uses load-balancing loss with coefficient 0.1 and z-loss with coefficient 0.01. The experiments were carried out using the `llm-optimizer-benchmark` codebase (Semenov et al., 2025).

Table 3: **Model architecture across widths.** The active parameters count the tied embedding & un-embedding head, attention, routers, norms, and the top-2 selected experts in each layer.

Detail	W128	W256	W512	W768	W1024
Total params	26.1M	72.7M	264.9M	520.0M	932.4M
Active params	12.0M	30.2M	95.0M	180.3M	309.6M
Attention heads	2	4	8	12	16
Expert FF dim	512	768	1536	2048	2816

The shared training and optimizer settings are summarized in Table 4. The μP parameter grouping uses the base learning rate for routers, the width-scaled learning rate $\eta/(d/256)$ for hidden and expert matrices, and zero weight decay on normalization and tied embedding/head parameters. For Muon matrix groups, we use the adjustment factor $0.2\eta_t\sqrt{\max(m,n)}$ which lets Muon use the same nominal learning rate as AdamW (Liu et al., 2025). When running Muon, embedding, un-embedding, and non-matrix parameters are trained with AdamW at the identical nominal learning rate, with no weight decay applied to those groups.

Table 4: **Training and optimization details.** These are shared training settings used across all the runs. The learning rate is adjusted based on the total training duration.

Detail	Value
Data / tokenizer	FineWeb / GPT-2, vocabulary 50,304
Sequence length	2048
Hardware	DDP on 8× A100 GPUs
Batch size	524,288 tokens/update
AdamW betas	$(\beta_1, \beta_2) = (0.9, 0.95)$
Muon settings	$\beta = 0.95$, 5 Newton–Schulz steps
Weight decay	$\lambda = 0.1$; standard $\lambda\eta_t$, scaled $\lambda\eta_t^2/\eta_{\max}$
Schedule	cosine One-Cycle with final LR $= 0.1\eta_{\text{peak}}$
Warmup	$\max\{1000, 0.01T\}$ steps for total steps T

C μP for MoE and Learning Rate Transfer Across Horizons

A key problem in the training of large models is the difficulty of hyperparameter tuning. The μP parameterization, or maximal update parameterization, is designed to address this problem by enabling

the transfer of hyperparameters across model width. This is particularly relevant for frontier models, which have a large number of parameters and can typically only be trained once at the largest scale. By using μP, one can tune hyperparameters on smaller models and then transfer those settings to larger models, saving computational resources (Yang et al., 2022).

Table 5: Recommended μP scaling rules for dense and MoE Transformers. Here d_{in} denotes the input dimension of the corresponding weight matrix.

Component	**Init. Var.**	**Multiplier**	**LR (Adam)**
Embedding	1	1	1
Unembedding	1	$1/d_{\text{in}}$	1
Attention (Q, K, V, O)	$1/d_{\text{in}}$	1	$1/d_{\text{in}}$
Feed-forward (dense)	$1/d_{\text{in}}$	1	$1/d_{\text{in}}$
Experts (MoE)	$1/d_{\text{in}}$	1	$1/d_{\text{in}}$
Router (MoE)	1	$1/d_{\text{in}}$	1

The fundamental idea behind μP is to modify the hyperparameters in a manner that preserves non-trivial feature-learning dynamics as width increases (Yang et al., 2024). This is achieved by scaling the learning rate, variance at initialization, and output multipliers of certain layers so that the training dynamics remain comparable across widths. This idea can be extended to MoE models, where router and expert weights are treated differently in terms of their learning rates and scaling (Małaśnicki et al., 2025). For example, router weights are trained at the base learning rate, while the learning rate for expert weights is scaled inversely with width. Table 5 summarizes the parameterization for dense and MoE Transformers. The listed μP-MoE rules are derived for width scaling with a fixed number of experts and fixed top-k routing; experiments suggest that it is also possible to transfer hyper-parameters across model depth (Yang et al., 2023), MoE granularity or the number of experts (Ren et al., 2026).

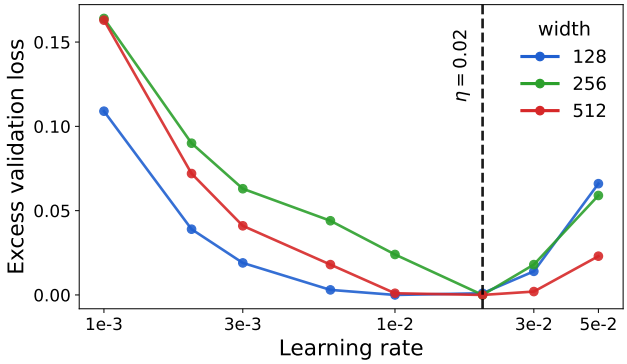


Figure 4: **μP-MoE learning-rate transfer across width.** We train μP LLaMA-style sparse MoE models at widths 128, 256, and 512 with fixed MoE granularity. The y-axis shows excess validation loss relative to the best learning rate at each width, making the transfer shape comparable across model sizes. A broad shared low-loss basin appears around 10^{-2}–3×10^{-2}, and we select $\eta = 0.02$ for subsequent experiments.

To test whether the μP-MoE parameterization gives useful learning-rate transfer across width, we ran a controlled width sweep at fixed MoE granularity. We trained the same μP LLaMA-style sparse MoE architecture at widths 128, 256, and 512, keeping head dimension fixed by using 2, 4, and 8 attention heads respectively. We swept the peak AdamW learning rate and measured final validation loss after 5k optimizer steps. The results are shown in Figure 4 with $\eta = 0.02$ lying at or near the minimum for all three widths. We therefore use it as the transferred peak learning rate for subsequent experiments, before any adjustment due to total training duration.

Although μP transfers the peak learning rate across width, the optimal peak learning rate still depends on the training duration, with longer runs favoring smaller peak learning rates. The optimal learning rate for longer horizons can be computed by tuning the learning rate for a shorter horizon

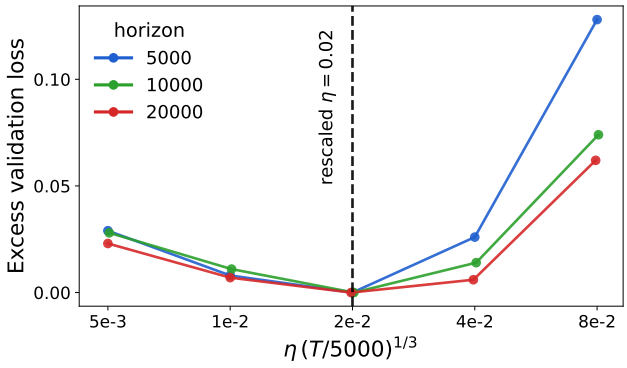


Figure 5: **Learning-rate transfer across training horizon.** We train the μP LLaMA-style sparse MoE model at width 256 for horizons $T \in \{5000, 10000, 20000\}$ optimizer steps and sweep the peak AdamW learning rate. The y-axis shows excess validation loss relative to the best learning rate at each horizon; the x-axis is the horizon-rescaled learning rate $\eta\,(T/5000)^{1/3}$. Under this rescaling the three curves share a common minimum near 0.02 (dashed line), so the peak learning rate for a new horizon can be obtained by rescaling the width-transferred value $\eta = 0.02$.

and transferring it using the law suggested in (Bjorck et al., 2025; Ren et al., 2026):

$$\eta^\star(T) = \eta^\star(T_0)\left(\frac{T}{T_0}\right)^{-1/3}, \tag{24}$$

where T_0 is a reference horizon and $\eta^\star(T)$ is the optimal peak learning rate at horizon T.

We test this for our training setup by fixing the width at 256 and sweeping the peak AdamW learning rate at three training horizons, $T \in \{5000, 10000, 20000\}$ optimizer steps. We find that the optima are well aligned once the learning rate is rescaled by $(T/5000)^{1/3}$ as illustrated in Figure 5. This cube-root rule lets us set the peak learning rate for a target horizon by adjusting the width-transferred value of $\eta = 0.02$. In combination, μP width transfer and the cube-root horizon rule let us set the peak learning rate for any width and training horizon directly, with no additional tuning needed.

D Proofs for Convergence Theory

This appendix collects the detailed proofs deferred from Section 3. Throughout, the objective being optimized is the original unregularized training objective, so when we say that an algorithm converges the reference point is a minimizer (or stationary point) of that objective.

D.1 Non-convergence of constant weight decay on a shifted quadratic

We restate and prove in full the one-dimensional non-convergence statement given in the main text as Proposition 3.1. The setup is deterministic (noise-free), so no stochastic assumptions are needed. Even in the absence of gradient noise the iteration fails to reach the minimizer.

Proposition D.1 (restatement of Proposition 3.1)**.** *Let $a > 0$ and $c \neq 0$, and consider the deterministic quadratic $f(w) = \frac{a}{2}(w - c)^2$ with unique minimizer $w^\star = c$. Fix a weight-decay coefficient $\lambda > 0$ and a learning-rate schedule $\{\eta_t\}_{t\geq 0}$ with $0 < \eta_t \leq 1/(a + \lambda)$ and $\sum_{t\geq 0} \eta_t = \infty$. Then gradient descent with constant decoupled weight decay,*

$$w_{t+1} = (1 - \eta_t\lambda)w_t - \eta_t\, a(w_t - c)\,, \tag{25}$$

converges, from every initial point $w_0 \in \mathbb{R}$, to

$$\bar{w} = \frac{ac}{a + \lambda} \neq c\,, \tag{26}$$

and the loss gap at the limit is

$$f(\bar{w}) - f(c) = \frac{ac^2}{2}\left(\frac{\lambda}{a + \lambda}\right)^2 > 0\,. \tag{27}$$

In particular the iterates never converge to the minimizer $w^\star = c$.

Proof. Since $\nabla f(w) = a(w - c)$, the update (25) is affine in w_t. Collecting terms,

$$w_{t+1} = \big(1 - \eta_t(a + \lambda)\big)w_t + \eta_t\, ac\,. \tag{28}$$

A point $\bar{w}$ is fixed by (28) for every $\eta_t > 0$ precisely when $\big(1 - \eta_t(a + \lambda)\big)\bar{w} + \eta_t ac = \bar{w}$, that is when

$$-\eta_t(a + \lambda)\bar{w} + \eta_t ac = 0 \iff (a + \lambda)\bar{w} = ac\,.$$

Because $a + \lambda > 0$, this has the unique solution $\bar{w} = ac/(a + \lambda)$, which is (26). Since $\lambda > 0$ and $c \neq 0$,

$$\bar{w} - c = \frac{ac}{a + \lambda} - c = -\frac{\lambda c}{a + \lambda} \neq 0\,, \tag{29}$$

so $\bar{w} \neq c = w^\star$.

Now let $\delta_t = w_t - \bar{w}$. Subtracting the fixed-point identity $\bar{w} = \big(1 - \eta_t(a + \lambda)\big)\bar{w} + \eta_t ac$ from (28) gives

$$\delta_{t+1} = \big(1 - \eta_t(a + \lambda)\big)\delta_t\,. \tag{30}$$

Set $\rho_t = 1 - \eta_t(a + \lambda)$. The hypothesis $0 < \eta_t \leq 1/(a + \lambda)$ gives $0 \leq \rho_t < 1$, so unrolling (30),

$$\delta_t = \Big(\prod_{s=0}^{t-1} \rho_s\Big)\delta_0, \qquad 0 \leq \prod_{s=0}^{t-1} \rho_s \leq \exp\Big(-(a + \lambda)\sum_{s=0}^{t-1} \eta_s\Big)\,,$$

where the upper bound uses $1 - x \leq e^{-x}$. Since $\sum_s \eta_s = \infty$, the exponent tends to $-\infty$, so the product tends to 0 and therefore $\delta_t \to 0$, i.e. $w_t \to \bar{w}$, from every initial point w_0.

Finally, using (29) and $f(c) = 0$,

$$f(\bar{w}) - f(c) = \frac{a}{2}(\bar{w} - c)^2 = \frac{ac^2}{2}\left(\frac{\lambda}{a + \lambda}\right)^2\,,$$

which is (27). □

D.2 Non-convergence of constant weight decay for Muon

We restate and prove Proposition 3.2. Consider the shifted matrix quadratic

$$f(W) = \frac{1}{2}\|W - A\|_F^2 = \frac{1}{2}\operatorname{tr}\big((W - A)^\top(W - A)\big), \qquad \nabla f(W) = W - A\,, \tag{31}$$

with $A \in \mathbb{R}^{m\times n}$ and unique minimizer $W^\star = A$. Muon with heavy-ball momentum $\beta \in [0, 1)$ forms

$$M_t = \beta M_{t-1} + \nabla f(W_t), \tag{32}$$

$$O_t = \operatorname{polar}(M_t), \tag{33}$$

where $\operatorname{polar}(C) = UV^\top$ for a compact singular value decomposition $C = U\Sigma V^\top$ (and $\operatorname{polar}(0) = 0$), so that $\|O_t\|_{\mathrm{op}} \leq 1$ for every t. The constant-decay Muon update is

$$W_{t+1} = (1 - \eta_t\lambda)W_t - \eta_t O_t\,. \tag{34}$$

Proposition D.2 (restatement of Proposition 3.2)**.** *Let $\lambda > 0$, suppose $0 < \eta_t\lambda \leq 1$ and $\sum_t \eta_t = \infty$, and consider the Muon iteration (32)–(34) on the objective (31). Then, regardless of the momentum coefficient β and the initialization W_0,*

$$\limsup_{t\to\infty} \|W_t\|_{\mathrm{op}} \leq \frac{1}{\lambda}\,. \tag{35}$$

Consequently, if $\sigma_{\max}(A) > 1/\lambda$, then

$$\liminf_{t\to\infty} \|W_t - A\|_{\mathrm{op}} \geq \sigma_{\max}(A) - \frac{1}{\lambda} > 0\,, \tag{36}$$

so the iterates never converge to the minimizer $W^\star = A$, and the loss stays bounded below by

$$\liminf_{t\to\infty} f(W_t) \geq \frac{1}{2}\sum_i \Big(\sigma_i(A) - \frac{1}{\lambda}\Big)_+^2\,, \tag{37}$$

where $\sigma_i(A)$ are the singular values of A and $(x)_+ = \max(x, 0)$. In particular, for every $\lambda > 0$ there is a shifted quadratic on which constant weight decay fails to recover the minimizer.

Proof. Let $z_t = \|W_t\|_{\mathrm{op}}$. Taking operator norms in (34) and using $\|O_t\|_{\mathrm{op}} \le 1$ together with $1 - \eta_t\lambda \ge 0$,

$$z_{t+1} \le (1 - \eta_t\lambda)z_t + \eta_t \, .$$

Subtracting $1/\lambda$ from both sides gives

$$z_{t+1} - \frac{1}{\lambda} \le (1 - \eta_t\lambda)\Big(z_t - \frac{1}{\lambda}\Big) \, ,$$

so by induction

$$z_t - \frac{1}{\lambda} \le \Big(z_0 - \frac{1}{\lambda}\Big)_+ \prod_{s=0}^{t-1}(1 - \eta_s\lambda) \, ,$$

where $(x)_+ = \max(x, 0)$. Since $1 - \eta_s\lambda \le e^{-\eta_s\lambda}$, the product is at most $\exp\big(-\lambda \sum_{s<t} \eta_s\big)$, which tends to 0 because $\sum_s \eta_s = \infty$. Hence $\limsup_t z_t \le 1/\lambda$, which is (35). The bound uses only $\|O_t\|_{\mathrm{op}} \le 1$. Thus the entire proposition also holds for the Nesterov direction $O_t = \mathrm{polar}(\nabla f(W_t) + \beta M_t)$, since its operator norm is still at most one and the remaining argument is unchanged.

For (36), the reverse triangle inequality in operator norm gives

$$\|W_t - A\|_{\mathrm{op}} \ge \|A\|_{\mathrm{op}} - \|W_t\|_{\mathrm{op}} = \sigma_{\max}(A) - z_t \, ,$$

and taking $\liminf$ as $t \to \infty$ together with (35) gives (36), which is strictly positive when $\sigma_{\max}(A) > 1/\lambda$.

For the loss bound (37), we use that the Frobenius-norm projection onto the spectral-norm ball clips singular values. Concretely, for any radius $\rho \ge 0$,

$$\min_{\|W\|_{\mathrm{op}} \le \rho} \|W - A\|_F^2 = \sum_i \big(\sigma_i(A) - \rho\big)_+^2 \, , \tag{38}$$

the minimizer being $\sum_i \min(\sigma_i(A), \rho)\, u_i v_i^\top$ for the singular value decomposition $A = \sum_i \sigma_i(A)\, u_i v_i^\top$. Fix $\rho > 1/\lambda$. By (35) there is T_ρ such that $\|W_t\|_{\mathrm{op}} \le \rho$ for all $t \ge T_\rho$, so (38) gives $\|W_t - A\|_F^2 \ge \sum_i (\sigma_i(A) - \rho)_+^2$ for such t, and hence

$$\liminf_{t\to\infty} f(W_t) = \liminf_{t\to\infty} \tfrac{1}{2}\|W_t - A\|_F^2 \ge \tfrac{1}{2} \sum_i (\sigma_i(A) - \rho)_+^2 \, .$$

Letting $\rho \downarrow 1/\lambda$ and using continuity of $\rho \mapsto (\sigma_i(A) - \rho)_+^2$ yields (37). □

D.3 Convergence of scaled weight decay for SGD

We restate and prove Theorem 3.1. Let $F : \mathbb{R}^d \to \mathbb{R}$ be bounded below by $F_{\inf}$ and L-smooth, and let the stochastic gradients satisfy

$$\mathbb{E}[g_t \mid \mathcal{F}_t] = \nabla F(x_t), \qquad \mathbb{E}[\|g_t\|_2^2 \mid \mathcal{F}_t] \le \sigma^2 + \rho\|\nabla F(x_t)\|_2^2 \, , \tag{39}$$

with $\sigma^2 \ge 0$, $\rho \ge 1$. The scaled weight decay update is

$$x_{t+1} = x_t - \eta_t g_t - q\,\eta_t^2 x_t, \qquad q = \frac{\lambda}{\eta_{\max}} \, . \tag{40}$$

We assume the iterates are bounded in mean square, $\sup_t \mathbb{E}\|x_t\|_2^2 \le B^2 < \infty$, which in the nonconvex setting plays the role that strong convexity plays for the distance recursion: it keeps the order-η_t^2 shrink term summable.

Theorem D.1 (restatement of Theorem 3.1)**.** *Under* (39), $\sup_t \mathbb{E}\|x_t\|_2^2 \le B^2 < \infty$, *and the Robbins–Monro conditions* $\sum_t \eta_t = \infty$, $\sum_t \eta_t^2 < \infty$, *the iterates* (40) *satisfy*

$$\frac{1}{S_T} \sum_{t=0}^{T-1} \eta_t\, \mathbb{E}\|\nabla F(x_t)\|_2^2 \to 0, \qquad S_T = \sum_{t=0}^{T-1} \eta_t \, , \tag{41}$$

and in particular $\liminf_{t\to\infty} \mathbb{E}\|\nabla F(x_t)\|_2^2 = 0$.

Proof. By L-smoothness, with the increment $\Delta_t = x_{t+1} - x_t = -\eta_t g_t - q\eta_t^2 x_t$,

$$F(x_{t+1}) \le F(x_t) + \langle \nabla F(x_t), \Delta_t \rangle + \frac{L}{2}\|\Delta_t\|_2^2 \,. \tag{42}$$

Take the conditional expectation given $\mathcal{F}_t$. Using $\mathbb{E}[g_t \mid \mathcal{F}_t] = \nabla F(x_t)$,

$$\mathbb{E}[\langle \nabla F(x_t), \Delta_t \rangle \mid \mathcal{F}_t] = -\eta_t \|\nabla F(x_t)\|_2^2 - q\eta_t^2 \langle \nabla F(x_t), x_t \rangle \,. \tag{43}$$

The shrink cross term is controlled by Cauchy–Schwarz and Young's inequality,

$$-q\eta_t^2 \langle \nabla F(x_t), x_t \rangle \le q\eta_t^2 \|\nabla F(x_t)\|_2 \|x_t\|_2 \le \frac{\eta_t}{4}\|\nabla F(x_t)\|_2^2 + q^2\eta_t^3 \|x_t\|_2^2 \,, \tag{44}$$

where the last step uses $ab \le \frac{1}{4}a^2 + b^2$ with $a = \sqrt{\eta_t}\|\nabla F(x_t)\|_2$ and $b = q\eta_t^{3/2}\|x_t\|_2$. For the second-order term, $\|\Delta_t\|_2^2 \le 2\eta_t^2\|g_t\|_2^2 + 2q^2\eta_t^4\|x_t\|_2^2$, so by (39),

$$\frac{L}{2}\mathbb{E}[\|\Delta_t\|_2^2 \mid \mathcal{F}_t] \le L\eta_t^2\big(\sigma^2 + \rho\|\nabla F(x_t)\|_2^2\big) + Lq^2\eta_t^4\|x_t\|_2^2 \,. \tag{45}$$

Substituting (43)–(45) into the conditional expectation of (42),

$$\mathbb{E}[F(x_{t+1}) \mid \mathcal{F}_t] \le F(x_t) - \eta_t\Big(\tfrac{3}{4} - L\rho\eta_t\Big)\|\nabla F(x_t)\|_2^2 + L\sigma^2\eta_t^2 + q^2\eta_t^3\|x_t\|_2^2 + Lq^2\eta_t^4\|x_t\|_2^2 \,.$$

Since $\eta_t \to 0$, there is an index beyond which $\frac{3}{4} - L\rho\eta_t \ge \frac{1}{2}$. Taking total expectations, using $\mathbb{E}\|x_t\|_2^2 \le B^2$, and summing from that index to $T-1$,

$$\frac{1}{2}\sum_t \eta_t \mathbb{E}\|\nabla F(x_t)\|_2^2 \le \mathbb{E}F(x_{t_1}) - F_{\inf} + \big(L\sigma^2 + q^2B^2 + Lq^2B^2\big)\sum_t \eta_t^2 \,,$$

where we bounded $\eta_t^3, \eta_t^4 \le \eta_t^2$ for large t. The right-hand side is finite because $\sum_t \eta_t^2 < \infty$. Dividing by S_T and letting $T \to \infty$, and noting $S_T \to \infty$, gives (41). The $\liminf$ statement follows since a nonnegative sequence whose η_t-weighted average tends to zero, with $\sum_t \eta_t = \infty$, has a subsequence tending to zero. □

When the objective is in addition strongly convex, the stationarity guarantee upgrades to convergence of the iterates to the unique minimizer, and the bounded-iterate assumption becomes unnecessary. Both this statement and its rate follow from a single stochastic-approximation recursion.

Lemma D.1. *Let $\{u_t\}$ be nonnegative reals satisfying, for all sufficiently large t,*

$$u_{t+1} \le (1 - \alpha\eta_t)u_t + \kappa\eta_t^2 \tag{46}$$

for constants $\alpha, \kappa > 0$ with $\alpha\eta_t \le 1$. If $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$, then $u_t \to 0$. For $\eta_t = \gamma/(t + t_0)$ with $\alpha\gamma > 1$, one has $u_t = O(1/t)$.

Proof. Unrolling (46) from a base index t_1 beyond which it holds,

$$u_{t+1} \le \Big(\prod_{s=t_1}^{t}(1 - \alpha\eta_s)\Big)u_{t_1} + \kappa\sum_{s=t_1}^{t}\eta_s^2\prod_{r=s+1}^{t}(1 - \alpha\eta_r) \,.$$

Using $1 - \alpha\eta_s \le e^{-\alpha\eta_s}$, the leading product is at most $\exp(-\alpha\sum_{s=t_1}^{t}\eta_s) \to 0$ since $\sum_s \eta_s = \infty$. For the second term, fix $\epsilon > 0$; because $\sum_s \eta_s^2 < \infty$, choose $t_2 \ge t_1$ with $\sum_{s \ge t_2}\eta_s^2 < \epsilon$. The tail $s \ge t_2$ contributes at most $\kappa\sum_{s\ge t_2}\eta_s^2 < \kappa\epsilon$ (each product factor is at most 1), while the fixed prefix $t_1 \le s < t_2$ is multiplied by $\exp(-\alpha\sum_{r=t_2}^{t}\eta_r) \to 0$. Hence $\limsup_t u_{t+1} \le \kappa\epsilon$ for every ϵ, so $u_t \to 0$. For the harmonic schedule, a standard induction on (46) with $\eta_t = \gamma/(t+t_0)$ and $\alpha\gamma > 1$ gives $u_t \le \nu/(t+t_0)$ with $\nu = \max\{(t_1+t_0)u_{t_1}, \kappa\gamma^2/(\alpha\gamma - 1)\}$, which is $O(1/t)$. □

Theorem D.2. *Suppose, in addition to (39), that F is μ-strongly convex with $\mu \le L$ and unique minimizer $x^\star$, and that the Robbins–Monro conditions hold. Then the scaled weight decay iterates (40) satisfy*

$$\mathbb{E}\|x_t - x^\star\|_2^2 \to 0 \,, \tag{47}$$

and for the harmonic schedule $\eta_t = \gamma/(t+t_0)$ with $\mu\gamma > 1$ the rate is $\mathbb{E}\|x_t - x^\star\|_2^2 = O(1/t)$.

Proof. Write $e_t = x_t - x^\star$. Expanding (40) and conditioning on $\mathcal{F}_t$, using $\mathbb{E}[g_t \mid \mathcal{F}_t] = \nabla F(x_t)$,

$$\mathbb{E}[\|e_{t+1}\|_2^2 \mid \mathcal{F}_t] = \|e_t\|_2^2 - 2\eta_t\langle e_t, \nabla F(x_t)\rangle - 2q\eta_t^2\langle e_t, x_t\rangle + \eta_t^2\mathbb{E}[\|g_t + q\eta_t x_t\|_2^2 \mid \mathcal{F}_t] \,. \tag{48}$$

Strong convexity and $\nabla F(x^\star) = 0$ give $\langle e_t, \nabla F(x_t)\rangle \geq \mu\|e_t\|_2^2$, and L-smoothness gives $\|\nabla F(x_t)\|_2 \leq L\|e_t\|_2$. Writing $x_t = e_t + x^\star$, Young's inequality gives

$$-2\langle e_t, x_t\rangle = -2\|e_t\|_2^2 - 2\langle e_t, x^\star\rangle \leq -\|e_t\|_2^2 + \|x^\star\|_2^2 \,, \tag{49}$$

so strong convexity here also supplies the mean-square boundedness of the iterates, and no separate bounded-iterate assumption is required. For the second-moment term, using $\|g_t + q\eta_t x_t\|_2^2 \leq 2\|g_t\|_2^2 + 2q^2\eta_t^2\|x_t\|_2^2$, (39), $\|\nabla F(x_t)\|_2^2 \leq L^2\|e_t\|_2^2$, and $\|x_t\|_2^2 \leq 2\|e_t\|_2^2 + 2\|x^\star\|_2^2$,

$$\mathbb{E}[\|g_t + q\eta_t x_t\|_2^2 \mid \mathcal{F}_t] \leq 2\sigma^2 + 2\rho L^2\|e_t\|_2^2 + 4q^2\eta_t^2\big(\|e_t\|_2^2 + \|x^\star\|_2^2\big) \,. \tag{50}$$

Substituting into (48) and collecting powers of η_t,

$$\mathbb{E}[\|e_{t+1}\|_2^2 \mid \mathcal{F}_t] \leq \big(1 - 2\mu\eta_t + K_1\eta_t^2 + K_2\eta_t^4\big)\|e_t\|_2^2 + \big(q\|x^\star\|_2^2 + 2\sigma^2\big)\eta_t^2 + 4q^2\|x^\star\|_2^2\,\eta_t^4 \,,$$

with $K_1 = 2\rho L^2$ and $K_2 = 4q^2$ (the remaining $-q\eta_t^2\|e_t\|_2^2$ term only helps). Since $\eta_t \to 0$, for all sufficiently large t the bracketed coefficient is at most $1 - \mu\eta_t$ and the η_t^4 terms are dominated by η_t^2. Taking total expectations yields, for large t,

$$\mathbb{E}\|e_{t+1}\|_2^2 \leq (1 - \mu\eta_t)\,\mathbb{E}\|e_t\|_2^2 + \kappa\eta_t^2$$

for some $\kappa > 0$. This is the recursion (46) with $u_t = \mathbb{E}\|e_t\|_2^2$ and $\alpha = \mu$, so Lemma D.1 gives $\mathbb{E}\|e_t\|_2^2 \to 0$, and the $O(1/t)$ rate for a harmonic schedule with $\mu\gamma > 1$. □

Corollary D.1. *Let $f(w) = a(w-c)^2/2$ with $a > 0$, and consider deterministic gradient descent with scaled weight decay, $w_{t+1} = w_t - \eta_t\, a(w_t - c) - q\eta_t^2 w_t$, under a Robbins–Monro schedule. Then $w_t \to c = w^\star$, the minimizer of the original objective.*

Proof. The objective is L-smooth and μ-strongly convex with $L = \mu = a$, so Theorem D.2 applies with deterministic gradients ($\sigma^2 = 0$, $\rho = 1$) and gives $w_t \to c$. □

D.4 Convergence of scaled weight decay for Muon

We restate and prove the Muon results of the main text. Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be bounded below by $f_{\inf}$ and L-smooth in Frobenius norm, with stochastic gradient $G_t = \nabla f(W_t) + \xi_t$ where $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$ and $\mathbb{E}[\|\xi_t\|_F^2 \mid \mathcal{F}_t] \leq \sigma^2/B$ for a fixed batch size B. The buffer is $M_t = \beta M_{t-1} + G_t$ with fixed $\beta \in [0, 1)$, $O_t = \mathrm{polar}(M_t)$, and the scaled-decay update is

$$W_{t+1} = W_t - \eta_t O_t - q\,\eta_t^2 W_t, \qquad q = \frac{\lambda}{\eta_{\max}} \,. \tag{51}$$

Write $r = \min(m, n)$ for the ambient rank bound, and assume the iterates are bounded in mean square, $\sup_t \mathbb{E}\|W_t\|_F^2 \leq \mathcal{B}^2$.

Unrolling from $M_{-1} = 0$ gives $M_t = \sum_{s\leq t}\beta^{t-s}G_s$, so the normalized buffer $\bar{M}_t = (1-\beta)M_t$ obeys $\bar{M}_t = \beta\bar{M}_{t-1} + (1-\beta)G_t$ and is an exact positive multiple of M_t. Since orthogonalization is scale-invariant, $O_t = \mathrm{polar}(M_t) = \mathrm{polar}(\bar{M}_t)$, and we track the error of the normalized buffer,

$$E_t = \bar{M}_t - \nabla f(W_t) \,. \tag{52}$$

The orthogonalization satisfies the polar identities

$$\langle \bar{M}_t, O_t\rangle_F = \|\bar{M}_t\|_*, \qquad \|O_t\|_{\mathrm{op}} \leq 1, \qquad \|O_t\|_F^2 = \mathrm{rank}(M_t) \leq r \,, \tag{53}$$

where $\|\cdot\|_*$ is the nuclear norm, dual to the operator norm. We use repeatedly $\langle X, Y\rangle_F \leq \|X\|_*\|Y\|_{\mathrm{op}}$ and $\|X\|_* \leq \sqrt{\mathrm{rank}(X)}\,\|X\|_F$.

Lemma D.2 (alignment)**.** *For every t,*

$$\langle \nabla f(W_t), O_t\rangle_F \geq \|\nabla f(W_t)\|_* - 2\sqrt{r}\,\|E_t\|_F \,. \tag{54}$$

Proof. Writing $\nabla f(W_t) = \bar{M}_t - E_t$ and using (53),

$$\langle \nabla f(W_t), O_t \rangle_F = \|\bar{M}_t\|_* - \langle E_t, O_t \rangle_F \geq \|\bar{M}_t\|_* - \|E_t\|_* ,$$

since $\langle E_t, O_t \rangle_F \leq \|E_t\|_* \|O_t\|_{\mathrm{op}} \leq \|E_t\|_*$. By the reverse triangle inequality, $\|\bar{M}_t\|_* \geq \|\nabla f(W_t)\|_* - \|E_t\|_*$; combining and using $\|E_t\|_* \leq \sqrt{r}\,\|E_t\|_F$ gives (54). □

The tracking error obeys a contraction with a bias term of order η and a noise term that does not vanish for fixed β and B.

Lemma D.3 (tracking with a noise floor). *Under the standing assumptions, if $\eta_t \to 0$, then*

$$\limsup_{t\to\infty} \mathbb{E}\|E_t\|_F^2 \leq (1-\beta)\,\frac{\sigma^2}{B} . \tag{55}$$

Proof. If $\beta = 0$, then $\bar{M}_t = G_t$ and $E_t = \xi_t$, so the claimed bound follows immediately from $\mathbb{E}\|\xi_t\|_F^2 \leq \sigma^2/B$. Suppose henceforth that $0 < \beta < 1$. From $\bar{M}_t = \beta \bar{M}_{t-1} + (1-\beta) G_t$ and $G_t = \nabla f(W_t) + \xi_t$,

$$E_t = \beta E_{t-1} + \beta\,\delta_t + (1-\beta)\xi_t, \qquad \delta_t = \nabla f(W_{t-1}) - \nabla f(W_t) . \tag{56}$$

The term $\beta E_{t-1} + \beta\delta_t$ is $\mathcal{F}_t$-measurable and $\mathbb{E}[\xi_t \mid \mathcal{F}_t] = 0$, so the cross term vanishes and

$$\mathbb{E}\|E_t\|_F^2 = \beta^2\,\mathbb{E}\|E_{t-1} + \delta_t\|_F^2 + (1-\beta)^2\,\mathbb{E}\|\xi_t\|_F^2 .$$

By Young's inequality with weight $a = (1-\beta)/\beta$, $\|E_{t-1} + \delta_t\|_F^2 \leq \frac{1}{\beta}\|E_{t-1}\|_F^2 + \frac{1}{1-\beta}\|\delta_t\|_F^2$, so $\beta^2\|E_{t-1} + \delta_t\|_F^2 \leq \beta\|E_{t-1}\|_F^2 + \frac{\beta^2}{1-\beta}\|\delta_t\|_F^2$. Using $\mathbb{E}\|\xi_t\|_F^2 \leq \sigma^2/B$,

$$\mathbb{E}\|E_t\|_F^2 \leq \beta\,\mathbb{E}\|E_{t-1}\|_F^2 + \frac{\beta^2}{1-\beta}\,\mathbb{E}\|\delta_t\|_F^2 + (1-\beta)^2\frac{\sigma^2}{B} . \tag{57}$$

By L-smoothness and (51), $\|\delta_t\|_F \leq L\|W_t - W_{t-1}\|_F \leq L(\eta_{t-1}\sqrt{r} + q\eta_{t-1}^2\|W_{t-1}\|_F)$, so $\mathbb{E}\|\delta_t\|_F^2 \leq 2L^2(r\eta_{t-1}^2 + q^2\eta_{t-1}^4\mathcal{B}^2) \to 0$. Thus (57) is $u_t \leq \beta u_{t-1} + c_t$ with $u_t = \mathbb{E}\|E_t\|_F^2$ and $\limsup_t c_t = (1-\beta)^2\sigma^2/B$. Taking lim sup gives $\limsup_t u_t \leq \beta \limsup_t u_t + (1-\beta)^2\sigma^2/B$, hence $\limsup_t u_t \leq (1-\beta)\sigma^2/B$. □

Theorem D.3 (restatement of Theorem 3.2). *Under the standing assumptions, if $\sum_t \eta_t = \infty$ and $\sum_t \eta_t^2 < \infty$, then*

$$\limsup_{T\to\infty}\ \frac{1}{S_T}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\|\nabla f(W_t)\|_* \leq 2\sigma\sqrt{\frac{(1-\beta)\,r}{B}}, \qquad S_T = \sum_{t=0}^{T-1}\eta_t . \tag{58}$$

Proof. By L-smoothness, with $\Delta_t = -\eta_t O_t - q\eta_t^2 W_t$,

$$f(W_{t+1}) \leq f(W_t) + \langle \nabla f(W_t), \Delta_t \rangle_F + \frac{L}{2}\|\Delta_t\|_F^2 . \tag{59}$$

The first-order term splits into $-\eta_t\langle \nabla f(W_t), O_t\rangle_F - q\eta_t^2\langle \nabla f(W_t), W_t\rangle_F$. By Lemma D.2, the first piece is at most $-\eta_t\|\nabla f(W_t)\|_* + 2\sqrt{r}\,\eta_t\|E_t\|_F$. For the shrink term, Frobenius Cauchy–Schwarz and L-smoothness give

$$\begin{aligned}\mathbb{E}\big|\langle \nabla f(W_t), W_t\rangle_F\big| &\leq \mathbb{E}\big[\|\nabla f(W_t)\|_F\|W_t\|_F\big] \\ &\leq \|\nabla f(0)\|_F\,\mathcal{B} + L\mathcal{B}^2 =: C_W .\end{aligned}$$

For the second-order term, $\|\Delta_t\|_F^2 \leq 2r\eta_t^2 + 2q^2\eta_t^4\|W_t\|_F^2$. Taking total expectations in (59) and substituting these bounds yields

$$\eta_t\,\mathbb{E}\|\nabla f(W_t)\|_* \leq \mathbb{E}[f(W_t) - f(W_{t+1})] + 2\sqrt{r}\,\eta_t\mathbb{E}\|E_t\|_F + (qC_W + Lr)\eta_t^2 + Lq^2\mathcal{B}^2\eta_t^4 . \tag{60}$$

Choose t_1 so that $\eta_t \leq 1$ for every $t \geq t_1$. Summing from t_1 to $T-1$,

$$\sum_{t=t_1}^{T-1}\eta_t\,\mathbb{E}\|\nabla f(W_t)\|_* \leq \mathbb{E}f(W_{t_1}) - f_{\inf} + 2\sqrt{r}\sum_{t=t_1}^{T-1}\eta_t\,\mathbb{E}\|E_t\|_F + \big(qC_W + Lr + Lq^2\mathcal{B}^2\big)\sum_{t=t_1}^{T-1}\eta_t^2 .$$

Here we used $\eta_t^4 \le \eta_t^2$ for $t \ge t_1$. Divide by S_T. The telescoped gap and the finite $\sum_t \eta_t^2$ term vanish after division by $S_T \to \infty$. For the tracking term, $\mathbb{E}\|E_t\|_F \le \sqrt{\mathbb{E}\|E_t\|_F^2}$, and by Lemma D.3 $\limsup_t \mathbb{E}\|E_t\|_F \le \sqrt{(1-\beta)\sigma^2/B}$; a weighted Cesàro average has lim sup at most this bound. Hence

$$\limsup_T \frac{1}{S_T} \sum_t \eta_t \, \mathbb{E}\|\nabla f(W_t)\|_* \le 2\sqrt{r}\sqrt{\frac{(1-\beta)\sigma^2}{B}} = 2\sigma\sqrt{\frac{(1-\beta)r}{B}} \,,$$

which is (58). The scaled shrink term contributes only the order-η_t^2 and order-η_t^4 terms in (60), both summable, so it does not affect the floor; the same bound holds for Muon without weight decay. □

The floor vanishes in the deterministic case $\sigma = 0$, in which case the right-hand side of (58) is zero and $\frac{1}{S_T}\sum_t \eta_t \, \mathbb{E}\|\nabla f(W_t)\|_* \to 0$.

Theorem D.4. *Suppose the assumptions of Theorem D.3 hold with $\sigma = 0$, additionally suppose that $\|W_t\|_F \le \mathcal{B}$ almost surely for every t, and suppose f is convex with a minimizer $W^\star$ and $f^\star = f(W^\star)$. Then*

$$\frac{1}{S_T}\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}\big[f(W_t) - f^\star\big] \to 0, \qquad \liminf_{t\to\infty} \mathbb{E}\big[f(W_t) - f^\star\big] = 0 \,, \tag{61}$$

so Muon with scaled weight decay has a best-iterate convergence guarantee for the original objective.

Proof. By convexity and duality,

$$f(W_t) - f^\star \le \langle \nabla f(W_t), W_t - W^\star\rangle_F \le \|\nabla f(W_t)\|_* \big(\|W_t\|_F + \|W^\star\|_F\big) \,.$$

Using the almost-sure bound $\|W_t\|_F \le \mathcal{B}$ and then taking expectations gives $\mathbb{E}[f(W_t) - f^\star] \le (\mathcal{B} + \|W^\star\|_F)\,\mathbb{E}\|\nabla f(W_t)\|_*$. Multiplying by η_t, summing, and dividing by S_T,

$$\frac{1}{S_T}\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}[f(W_t) - f^\star] \le (\mathcal{B} + \|W^\star\|_F)\,\frac{1}{S_T}\sum_{t=0}^{T-1} \eta_t \, \mathbb{E}\|\nabla f(W_t)\|_* \to 0 \,.$$

This proves the weighted-average statement. Since the summands are nonnegative and $\sum_t \eta_t = \infty$, it also implies $\liminf_t \mathbb{E}[f(W_t) - f^\star] = 0$. □

Corollary D.2. *Let $f(W) = \frac{1}{2}\|W - A\|_F^2$ and consider deterministic Muon with scaled weight decay under a Robbins–Monro schedule, assuming the iterates are bounded. Then $\liminf_{t\to\infty} f(W_t) = 0$, so there is a subsequence $W_{t_k} \to A = W^\star$, the minimizer of the original objective. In particular scaled-decay Muon is not asymptotically separated from targets outside the radius-$1/\lambda$ ball that traps constant weight decay in Proposition 3.2.*

Proof. The objective is 1-smooth, convex, and bounded below by 0, with deterministic gradient ($\sigma = 0$, so the noise floor in (19) vanishes). Theorem 3.2 gives $\frac{1}{S_T}\sum_t \eta_t \|\nabla f(W_t)\|_* \to 0$, and Theorem D.4 gives $\liminf_t f(W_t) = 0$. Choosing a subsequence along which $f(W_{t_k}) \to 0$ and using $f(W) = \frac{1}{2}\|W - A\|_F^2$ yields $W_{t_k} \to A$. □

Remark 1 (Nesterov momentum). The Muon results extend to the Nesterov direction used by the implementation, with a modified stochastic bound for general β. With the unnormalized buffer M_t above, set

$$H_t = (1-\beta)(G_t + \beta M_t) = (1-\beta)G_t + \beta \bar{M}_t, \qquad O_t = \mathrm{polar}(H_t). \tag{62}$$

The positive normalization leaves the exact polar factor unchanged. Proposition 3.2 holds unchanged, as its proof uses only $\|O_t\|_{\mathrm{op}} \le 1$. For scaled decay, the alignment lemma holds with E_t replaced by $R_t = H_t - \nabla f(W_t) = \beta E_t + (1-\beta)\xi_t$. The drift bound also remains valid because $\|O_t\|_F \le \sqrt{r}$.

To account for the shared current noise, define $Z_{-1} = 0$ and $Z_t = \beta Z_{t-1} + (1-\beta)\xi_t$. Conditional mean zero makes distinct noise terms orthogonal in mean square, giving

$$\limsup_t \mathbb{E}\|Z_t\|_F^2 \le \frac{1-\beta}{1+\beta}\frac{\sigma^2}{B}.$$

For $A_t = E_t - Z_t$, the error recursion gives $A_t = \beta A_{t-1} + \beta\delta_t$ for $t \geq 1$. The L^2 triangle inequality gives $(\mathbb{E}\|A_t\|_F^2)^{1/2} \leq \beta(\mathbb{E}\|A_{t-1}\|_F^2)^{1/2} + \beta(\mathbb{E}\|\delta_t\|_F^2)^{1/2}$. Since the last term tends to zero and $\beta < 1$, we obtain $\mathbb{E}\|A_t\|_F^2 \to 0$. Now

$$R_t = \beta A_t + \beta^2 Z_{t-1} + (1-\beta^2)\xi_t.$$

The last two terms are orthogonal in mean square, while the first vanishes in mean square. Hence

$$\limsup_t \mathbb{E}\|R_t\|_F^2 \leq c_\beta \frac{\sigma^2}{B}, \qquad c_\beta = (1-\beta^2)^2 + \frac{\beta^4(1-\beta)}{1+\beta} = \frac{(1-\beta)(1+2\beta-2\beta^3)}{1+\beta}. \tag{63}$$

The same descent and weighted-average argument therefore gives the bound $2\sigma\sqrt{c_\beta r/B}$ in Theorem 3.2, both with scaled decay and without weight decay. Since $c_\beta \leq 1-\beta$ for $\beta \geq 1/\sqrt{2}$ (also for $\beta = 0$), the theorem's displayed bound holds unchanged in that range, including the implementation default $\beta = 0.95$. For other β, the bound above applies. When $\sigma = 0$, the floor vanishes for every $\beta \in [0, 1)$, so Theorem D.4 and Corollary D.2 follow unchanged. These extensions retain the original boundedness and schedule assumptions and require exact polar factors, not finite-step NS5.

E Additional Data from MoE Training Experiments

This appendix collects supporting diagnostics for the weight-norm analysis of Section 5, together with the loss and norm curves for the intermediate widths. All diagnostics are computed over the matrix-valued, non-router weights updated by Muon.

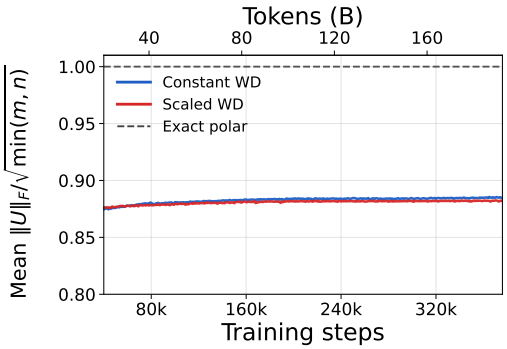


Figure 6: **Imperfect orthogonalization of the five-step Newton–Schulz polar map (width 1024).** We plot the layer-averaged normalized Frobenius norm $\frac{1}{L}\sum_\ell \|U_t^{(\ell)}\|_F/\sqrt{\min(m_\ell, n_\ell)}$ of the applied Muon update. An exact full-rank polar factor would give a ratio of one; the five-step Newton–Schulz output instead stays near 0.88 throughout training.

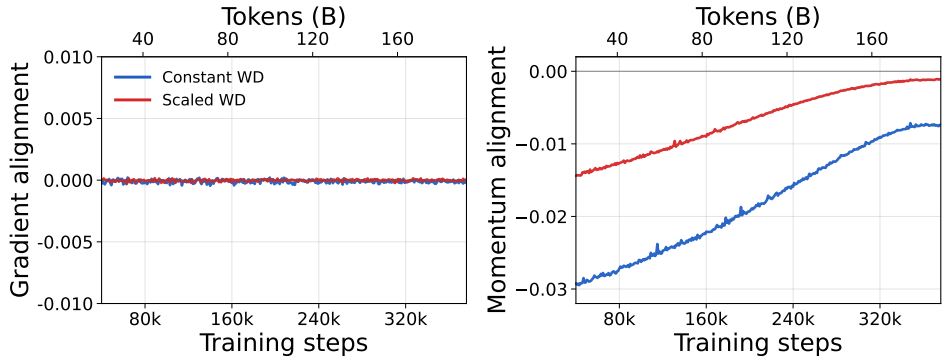


Figure 7: **Weight–gradient and weight–momentum alignment (width 1024).** *Left:* the raw gradient stays nearly orthogonal to the weights under both decay rules, $\langle W_t, G_t \rangle \approx 0$, as expected from the scale-invariance induced by the normalization layers. *Right:* the momentum buffer which is an exponential average of past gradients, acquires an appreciably negative alignment with the weights. As the learning rate decays the newly injected radial component shrinks relative to the weight norm, so the momentum alignment moves toward zero.

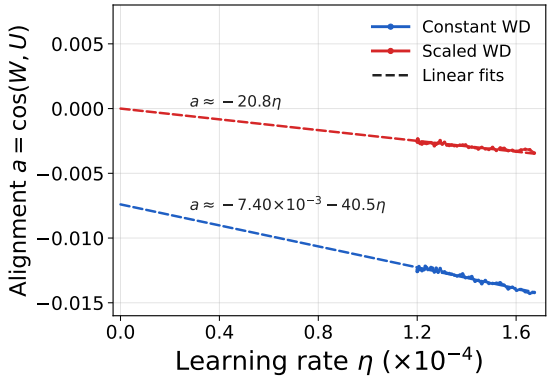


Figure 8: **Late-run weight–update alignment versus learning rate (width 1024).** Aggregate weight–update alignment a against the learning rate η over the final 50k steps (steps 326k–376k); the observed samples lie on the right, and the dashed fits are extended toward $\eta = 0$ to expose their asymptotics. Scaled weight decay is well described by the origin-constrained relation $a \approx -20.8\,\eta$, so that a/η approaches a finite constant. Constant weight decay instead requires a shifted fit $a \approx -7.40 \times 10^{-3} - 40.5\,\eta$ with a nonzero intercept. This changes the asymptotics of the steady-state norm and underlies the constant-norm versus collapsing-norm behavior of the two rules. The extrapolated portion comes from modeling the late-run fits, and not directly observed since the minimum learning rate in the experiment is $\eta_{\max}/10$.

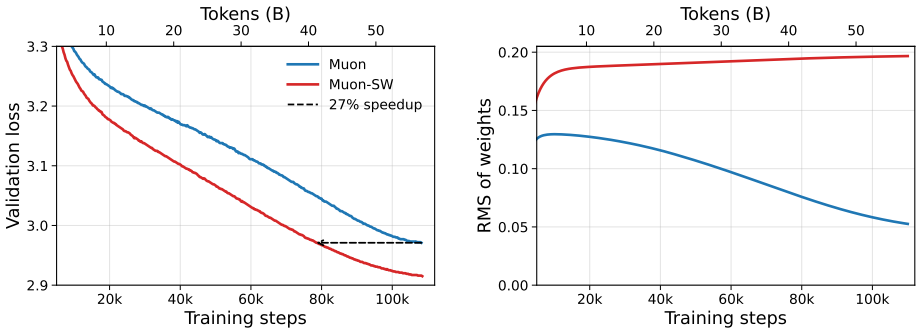


Figure 9: **Validation loss and weight norm at width 512.** *Left:* validation loss for Muon (constant weight decay) and Muon-SW (scaled weight decay); the dashed arrow marks the reduction in steps for the scaled run to reach the best constant-decay loss of 2.971 (108,250 vs. 78,500 steps, a $\sim$27% reduction). *Right:* the scaled-decay weight norm rises quickly and remains near 0.20, while the constant-decay norm peaks early and then decays to $\sim$0.05, reproducing the width-1024 separation.

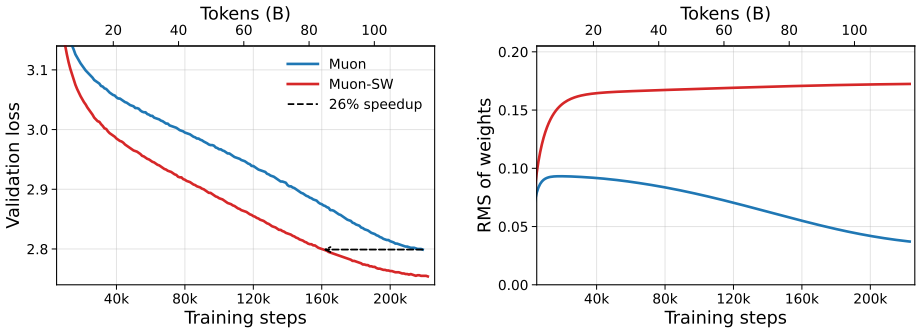


Figure 10: **Validation loss and weight norm at width 768.** *Left:* the best constant-decay loss of 2.799 is reached at 219,000 steps, and Muon-SW reaches it at 161,000 steps, a $\sim$26% reduction. *Right:* scaled weight decay holds the norm near 0.17 while the constant-decay norm turns over and decays to $\sim$0.04. Together with the width-512 result, this confirms that the loss improvement and contrasting norm dynamics are not specific to width 1024.

F PyTorch Implementation of MuonSW

We present a PyTorch (Paszke et al., 2019) implementation of Muon-SW, the Muon optimizer with scaled weight decay. Our implementation is based on the code of Jordan et al. (2024). The important change from standard decoupled decay is that the decay coefficient uses the scheduler-mutated learning rate η_t and the peak learning rate $\eta_{\max}$, so the parameter shrinkage is $c_t = \lambda\eta_t^2/\eta_{\max}$ instead of $c_t = \lambda\eta_t$. The Muon step itself includes the $\alpha\sqrt{\max(m,n)}$ adjustment for an $m \times n$ matrix, whose default $\alpha = 0.2$ matches the RMS of AdamW updates and allows one to use the same nominal learning rate as AdamW (Liu et al., 2025). The Muon optimizer should be used for hidden matrix parameters such as attention projection matrices and expert MLP matrices. Embeddings, the tied un-embedding or language-model head, normalization weights, biases, and other vector or scalar parameters should be trained with AdamW.

The implementation uses five Newton–Schulz iterations with coefficients $(3.4445, -4.7750, 2.0315)$, matching the common Muon implementation (Jordan et al., 2024). One could instead use other polynomial coefficients as in Polar Express (Amsel et al., 2026), or use randomized orthogonalization methods as proposed in (Ahn et al., 2025a,b). This compact implementation illustrates the scaled-decay update. The same weight-decay rule can be implemented by changing just a few lines of code in other optimizers such as Shampoo (Anil et al., 2021; Gupta et al., 2018), SOAP (Vyas et al., 2025), Schedule-Free Muon (Apte et al., 2026) and Aurora (Dewulf et al., 2026). Nesterov momentum is enabled by default here; Remark 1 extends the theory to Nesterov with exact polar factors.

```
import math, torch
import torch.distributed as dist

def zeropower_via_newtonschulz5(G, steps=5, eps=1e-7):
    assert G.ndim == 2
    a, b, c = (3.4445, -4.7750, 2.0315)
    X = G.bfloat16()
    transposed = X.size(0) > X.size(1)
    if transposed:
        X = X.T
    X = X / (X.norm() + eps)
    for _ in range(steps):
        A = X @ X.T
        B = b * A + c * A @ A
        X = a * X + B @ X
    return X.T if transposed else X

def muon_direction(grad, momentum, beta=0.95, ns_steps=5,
                   nesterov=True, shape_scale=0.2):
    momentum.lerp_(grad, 1.0 - beta)
    update = grad.lerp(momentum, beta) if nesterov else momentum
    original_shape = update.shape
    if update.ndim == 4:
        update = update.view(update.size(0), -1)
    update = zeropower_via_newtonschulz5(update, steps=ns_steps)
    m, n = update.shape
    update.mul_(shape_scale * math.sqrt(max(m, n)))
    return update.reshape(original_shape)
```

```python
class DistributedMuonSW(torch.optim.Optimizer):
    def __init__(self, params, lr=1e-2, eta_max=None, lambda_=0.1,
                 shape_scale=0.2, momentum=0.95,
                 nesterov=True, ns_steps=5):
        defaults = dict(lr=lr,
                        eta_max=lr if eta_max is None else eta_max,
                        lambda_=lambda_, shape_scale=shape_scale,
                        momentum=momentum, nesterov=nesterov,
                        ns_steps=ns_steps)
        assert (isinstance(params, list) and len(params) >= 1
                and isinstance(params[0], torch.nn.Parameter))
        params = sorted(params, key=lambda x: x.size(), reverse=True)
        super().__init__(params, defaults)

    @staticmethod
    def scaled_decay_coeff(group):
        lr, eta_max, lambda_ = (
            group["lr"], group["eta_max"], group["lambda_"])
        # MuonSW: standard WD would return lambda_ * lr.
        return 0.0 if eta_max <= 0.0 else lambda_ * lr * lr / eta_max

    @torch.no_grad()
    def step(self, closure=None):
        loss = None
        if closure is not None:
            with torch.enable_grad():
                loss = closure()
        world = dist.get_world_size() if dist.is_initialized() else 1
        rank = dist.get_rank() if dist.is_initialized() else 0

        for group in self.param_groups:
            params = group["params"]
            params_pad = (params + [torch.empty_like(params[-1])]
                          * (-len(params) % world))
            for base in range(0, len(params), world):
                if base + rank < len(params):
                    p = params[base + rank]
                    if p.grad is None:
                        p.grad = torch.zeros_like(p)
                    state = self.state[p]
                    if len(state) == 0:
                        state["momentum_buffer"] = torch.zeros_like(p)
                    update = muon_direction(
                        p.grad, state["momentum_buffer"],
                        beta=group["momentum"],
                        ns_steps=group["ns_steps"],
                        nesterov=group["nesterov"],
                        shape_scale=group["shape_scale"])
                    # MuonSW: decay by 1 - lambda * lr^2 / eta_max.
                    p.mul_(max(
                        0.0, 1.0 - self.scaled_decay_coeff(group)))
                    p.add_(update, alpha=-group["lr"])
                if dist.is_initialized():
                    dist.all_gather(
                        params_pad[base:base + world],
                        params_pad[base + rank])

        return loss
```