WHEN SCHEDULE-FREE FAILS: ONLINE LEARNING RATES FOR ANYTIME TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Schedule-free (SF) optimization avoids a predefined training horizon, yet its performance can worsen with continued training at a fixed learning rate. We introduce a new view of SF based on the pending update, which accumulates new updates before they are gradually applied to the output iterate. This view motivates **ORA, Online Learning Rates for Anytime Training**, which selects learning rates to control the size of new updates relative to the pending update. Our local analysis in an aligned subspace of a deterministic two-layer linear model shows that SF-Muon generally converges to a nonstationary point while ORA-Muon converges to an optimum. In noisy two-layer regression, smaller fixed learning rates delay SF-Muon's risk increase, while ORA closely tracks the lower envelope of SF-Muon curves. In 124M language model pretraining, ORA reaches the same validation loss as SF-Muon with **42–51% fewer tokens**. Compared with SF-Muon, ORA more closely follows the empirical loss scaling law under cosine decay and yields more accurate extrapolation. These results support studying anytime methods through scaling laws rather than only at a single horizon. ORA also matches the loss of strong Muon cosine baselines with **23% and 33% fewer tokens** on 124M and 720M models, respectively. Extrapolation suggests that these token savings grow further with longer training.

1 INTRODUCTION

Learning rate decay is central to language model training, but a cosine schedule requires the training length to be specified in advance (Loshchilov & Hutter, 2017; Hägele et al., 2024). Schedule-free (SF) methods such as SF-AdamW avoid this requirement by using a constant learning rate after warmup (Defazio et al., 2024; Loshchilov & Hutter, 2019). Recent work combines SF with Muon (Jordan et al., 2024) to bring the benefits of Muon to anytime language model training (Kim et al., 2026; Apte et al., 2026).

Our noisy two-layer linear regression experiments expose a failure of SF: after substantial initial improvement, further training leads to worse performance. Smaller fixed learning rates only delay this deterioration in SF-Muon. A fixed learning rate that works well for a short run can therefore become unsuitable when training is extended, leaving its choice dependent on the training horizon. SF-AdamW behaves similarly, even with the Polyak-based learning rates from SF+ and SPSsafe (Defazio, 2026; Oikonomou et al., 2026).

To address this failure, we take a new view of SF. Prior work commonly describes the output iterate X_t as an average of the base iterates Z_t (Defazio et al., 2024; Kim et al., 2026; Apte et al., 2026). We instead write the SF iteration in terms of X_t and the *pending update* $P_t = Z_t - X_t$. Each new update is added to P_t, with a fraction c_t of the sum applied to X_t and the remainder retained for future steps. The norm of P_t therefore provides a natural reference for the size of new updates. This view motivates **ORA, Online Learning Rates for Anytime Training**, which bounds the new update norm by $\sqrt{c_t}\,\|P_t\|$ after warmup and chooses the learning rate accordingly. ORA requires neither a target loss nor a predefined training length. Its learning rate tends to decrease later in training, without an imposed decay schedule.

In the noisy regression experiments, ORA-Muon continues to reduce risk and closely tracks the lower envelope of SF-Muon curves. To investigate this difference theoretically, we study the same model with exact gradients and Muon without internal momentum. Under common local conditions

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.

in an aligned subspace, we prove that SF-Muon generally converges to a nonstationary point, even at arbitrarily small fixed learning rates. Moreover, changing only how SF averages the base iterates cannot remove this error. ORA-Muon instead converges to an optimum from the same local region. Experiments from shared random initializations agree with the theoretical predictions.

We evaluate ORA on 124M and 720M Llama-based models using the optimizer benchmark setup (Semenov et al., 2025), following AMUSE (Kim et al., 2026). At 124M, ORA-Muon reaches SF-Muon's final validation loss with **42% fewer tokens** at the same peak learning rate. This gain comes solely from reducing the learning rate after warmup. Increasing only ORA's peak learning rate raises the saving to **51%**. The same ORA rule yields substantial gains with AdamW and AMUSE and remains effective at larger batch sizes.

To assess anytime training beyond a single horizon, we study empirical loss scaling laws with training length, following prior work on cosine decay and SF training (Bu et al., 2026; Defazio, 2026). Compared with SF-Muon, ORA-Muon more closely follows the loss scaling law under cosine decay and yields more accurate and stable extrapolation. ORA's learning rates and loss curves become similar later in training despite different peak learning rates, and the fitted curves have nearly equal limiting losses. Thus, ORA is largely horizon-free with respect to its peak learning rate. These results provide a more reliable basis for studying anytime methods through loss scaling laws. As an example, we use this approach to study the distinct effects of decay on the output parameters and the pending update.

To our knowledge, *ORA-Muon is the first anytime method to decisively outperform tuned Muon cosine baselines over long training horizons in both observations and extrapolation*. ORA-Muon matches the loss of the strong Muon cosine baseline with **23% fewer tokens** on the 124M model. Keeping the same optimizer hyperparameters and rescaling only the warmup length, ORA-Muon achieves a **33% saving** on the 720M model without further tuning. Extrapolation suggests that these token savings over cosine decay grow further with longer training.

Relation to prior work. Iterate averaging can improve generalization under constant or cyclical learning rates (Izmailov et al., 2018) and reproduce aspects of learning rate decay in simplified models (Sandler et al., 2023). Experiments on language models and AlgoPerf nevertheless find that averaging does not fully replace explicit decay (Hägele et al., 2024; Ajroldi et al., 2025). Unlike in ordinary iterate averaging, SF's averaged iterate affects the training trajectory itself. For SF-GD and SF-SGD, Chae & Kim (2026) establish stationarity guarantees for the gradient evaluation iterates Y_t on smooth nonconvex objectives. Our local analysis shows that SF-Muon's output X_t can instead converge to a nonstationary point at a fixed learning rate.

AMUSE adjusts the gradient evaluation point Y_t by varying the interpolation coefficient β_t (Kim et al., 2026). SF-NorMuon (Apte et al., 2026) combines SF with NorMuon (Li et al., 2026). SF+ and SPSsafe use Polyak-based learning rates derived from loss and gradient information (Defazio, 2026; Oikonomou et al., 2026). ORA instead chooses the learning rate to control the size of new updates relative to the pending update.

2 From Iterate Averaging to Pending Updates

A schedule-free (SF) optimizer maintains an output iterate X_t used for evaluation, a gradient evaluation point Y_t, and a base iterate Z_t updated by a base optimizer (Defazio et al., 2024). Let D_t denote the quantity computed by the base optimizer, including weight decay, before multiplication by the learning rate η_t. At step t, SF computes

$$\begin{aligned} Y_t &= \beta X_t + (1-\beta) Z_t, \\ Z_{t+1} &= Z_t - \eta_t D_t, \\ X_{t+1} &= (1-c_t) X_t + c_t Z_{t+1}. \end{aligned} \tag{1}$$

Here $X_1 = Z_1$, $\beta \in (0,1)$, and $c_t \in (0,1]$. The base optimizer uses the gradient at Y_t to compute D_t. Typically, η_t is held constant after warmup.

SF is commonly interpreted through iterate averaging: Z_t follows the base optimizer, while X_t averages the resulting base iterates with weights determined by c_t (Defazio et al., 2024; Kim et al.,

2026; Apte et al., 2026). Here, we take a different view of the same algorithm by tracking how newly generated updates are applied to the output iterate over time.

Let $S_t = -\eta_t D_t$ denote the update scaled by the learning rate at step t, and define the *pending update* $P_t = Z_t - X_t$. With these definitions, we can rewrite iteration (1) in terms of X_t and P_t:

$$\boxed{\begin{aligned} X_{t+1} &= X_t + c_t(P_t + S_t), \\ P_{t+1} &= (1 - c_t)(P_t + S_t). \end{aligned}} \tag{2}$$

At step t, the new update S_t is added to the pending update P_t. A fraction c_t of $P_t + S_t$ is applied to X_t, while the remaining fraction $1 - c_t$ is retained in P_{t+1} for future steps. Initially, $P_1 = 0$ because $X_1 = Z_1$. The gradient evaluation point can be written as $Y_t = X_t + (1 - \beta)P_t$.

The term *pending* has a precise meaning. Unrolling iteration (2) along the computed trajectory shows how much of each update remains pending. For $i \le t$, the coefficient of S_i in P_{t+1} is $\prod_{j=i}^{t}(1 - c_j)$, while the coefficient of S_i in $X_{t+1} - X_1$ is $1 - \prod_{j=i}^{t}(1 - c_j)$. If $\sum_{j=1}^{\infty} c_j = \infty$, then for each fixed i the coefficient of S_i in P_{t+1} tends to zero as $t \to \infty$. Thus, each fixed update is fully applied to X in the limit. Appendix A.1 gives the derivation.

Because the new update S_t is added directly to P_t, the pending update provides a natural reference for its size. The coefficient c_t determines how much of $P_t + S_t$ is applied to X_t. We use these relationships to design the online learning rate in the next section.

3 ORA: Online Learning Rates for Anytime Training

3.1 Choosing the Size of New Updates

To choose the size of the new update S_t, we first examine the change in the pending update P_t. From $P_{t+1} = (1 - c_t)(P_t + S_t)$ in (2), we obtain

$$\|P_{t+1}\|^2 - \|P_t\|^2 = -(2c_t - c_t^2)\,\|P_t\|^2 + 2(1 - c_t)^2\langle P_t, S_t\rangle + (1 - c_t)^2\,\|S_t\|^2 . \tag{3}$$

For stochastic updates, the inner product term can fluctuate across steps. As a heuristic, we treat this term as small in expectation. We compare the nonnegative contribution $(1 - c_t)^2\,\|S_t\|^2$ with the reduction $(2c_t - c_t^2)\,\|P_t\|^2$ in (3). For small c_t, choosing comparable magnitudes suggests $\|S_t\|^2 \asymp c_t\,\|P_t\|^2$. This motivates a reference scale of $\sqrt{Kc_t}\,\|P_t\|$, where $K > 0$ is a constant.

To choose a default value of K, we also consider the change in the output iterate X_t. Since $X_{t+1} - X_t = c_t P_t + c_t S_t$, the contribution from the pending update is $c_t P_t$. This gives two reference lengths: $\|P_t\|$, the size of the pending update, and $c_t\,\|P_t\|$, the size of the portion applied to X_t at step t. For small c_t, we seek a new update whose norm is small compared with $\|P_t\|$, yet substantially exceeds $c_t\,\|P_t\|$. A natural reference scale for $\|S_t\|$ satisfying both requirements is their geometric mean:

$$\sqrt{(c_t\,\|P_t\|)\,\|P_t\|} = \sqrt{c_t}\,\|P_t\| . \tag{4}$$

This choice recovers the square-root scale suggested above and corresponds to $K = 1$. We adopt this as a natural default, without claiming that the exponent or coefficient is optimal for every problem.

3.2 The ORA Learning Rate Rule

We use this reference scale to set an upper bound on the norm of the new update. After warmup, we choose the largest learning rate $\eta_t \le \eta_{\max}$ that satisfies

$$\|S_t\| \le \sqrt{Kc_t}\,\|P_t\| , \tag{5}$$

where $K = 1$ by default. For $\|D_t\| > 0$, substituting $S_t = -\eta_t D_t$ gives the learning rate rule for **ORA (Online Learning Rates for Anytime Training)**:

$$\boxed{\eta_t = \min\left\{\eta_{\max}, \frac{\sqrt{Kc_t}\,\|P_t\|}{\|D_t\|}\right\}.} \tag{6}$$

ORA requires neither a predefined training length nor a target loss. It does not enforce learning rate decay, but we observe that η_t tends to decrease later in training and is then determined online, independently of $\eta_{\max}$ (Figure 4(c)).

Warmup and c_t. In our experiments, we warm up the learning rate linearly to $\eta_{\max}$ over T_{w} steps, using $\eta_t^{\mathrm{w}} = \eta_{\max} t/T_{\mathrm{w}}$ for $t \le T_{\mathrm{w}}$. Let $u_t = \min\{t/T_{\mathrm{w}}, 1\}$. We compute c_t using the squared learning-rate weights of SF (Defazio et al., 2024), as also used by AMUSE and SF-NorMuon (Kim et al., 2026; Apte et al., 2026):

$$c_t = \frac{(\eta_{\max} u_t)^2}{\sum_{i=1}^{t} (\eta_{\max} u_i)^2} = \frac{u_t^2}{\sum_{i=1}^{t} u_i^2}. \tag{7}$$

We use c_t from (7) throughout training, independently of ORA's learning rate adjustments.

ORA with Muon and AdamW. For ORA-Muon and ORA-AdamW, we compute the norms in (6) over different sets of parameters. Here, $\mathcal{I}$ denotes the selected parameter set, and $\|\cdot\|_{\mathcal{I}}$ denotes the Euclidean norm over all entries corresponding to the parameters in $\mathcal{I}$.

For ORA-Muon, we follow prior work on SF-Muon (Kim et al., 2026; Apte et al., 2026) and use auxiliary AdamW without momentum for the parameters not updated by Muon (Jordan et al., 2024; Liu et al., 2025). In our experiments, the components of D_t typically have larger magnitudes on these auxiliary parameters than on the Muon parameters, partly due to gradient noise. Including the auxiliary parameters can make the learning rate overly small. ORA-Muon therefore determines the learning rate using only the Muon parameters and applies it to all parameter groups.

For ORA-AdamW, we retain momentum in AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019), as it can improve SF-AdamW's performance and robustness to larger learning rates (Defazio, 2026, Section 3). ORA-AdamW determines the learning rate using all trainable parameters. Algorithm 1 gives the complete procedure, with further implementation details in Appendix A.2.

Algorithm 1 ORA: Online Learning Rates for Anytime Training

Require: Initial parameters X_1, base optimizer, β, $\eta_{\max}$, K, $\mathcal{I}$, $\{c_t\}_{t\ge1}$, LR sequence $\{\eta_t^{\mathrm{w}}\}_{t=1}^{T_{\mathrm{w}}}$

Initialize $P_1 = 0$ and the base optimizer state
for $t = 1, 2, \ldots$ until stopped **do**
  Compute the gradient at $Y_t = X_t + (1-\beta)P_t$
  Compute D_t using the base optimizer, including weight decay
  if $t \le T_{\mathrm{w}}$ **then**
    $\eta_t \leftarrow \eta_t^{\mathrm{w}}$
  else
    $\eta_t \leftarrow \min\{\eta_{\max}, \sqrt{K c_t}\, \|P_t\|_{\mathcal{I}} \,/\, \|D_t\|_{\mathcal{I}}\}$
  end if
  $S_t \leftarrow -\eta_t D_t$ for all parameters
  $X_{t+1} \leftarrow X_t + c_t(P_t + S_t)$
  $P_{t+1} \leftarrow (1 - c_t)(P_t + S_t)$
end for
return Current output iterate X

Unless otherwise stated, we use $K = 1$, linear LR warmup with $\eta_t^{\mathrm{w}} = \eta_{\max} t/T_{\mathrm{w}}$, and c_t defined by (7). For ORA-Muon, $\mathcal{I}$ contains the Muon parameters. For ORA-AdamW, it contains all parameters.

4 When Schedule-Free Training Fails

Linear networks combine nonconvex objectives and nonlinear learning dynamics with a structure that permits rigorous analysis (Saxe et al., 2014; Arora et al., 2019; Dominé et al., 2023). Even two-layer networks can exhibit persistent oscillations under gradient descent with a sufficiently large fixed learning rate (Ghosh et al., 2025). We study SF and ORA on $f_X(\mathbf{a}) = X^{(2)}X^{(1)}\mathbf{a}$, where $X^{(1)} \in \mathbb{R}^{n\times d}$ and $X^{(2)} \in \mathbb{R}^{d\times n}$. For a target matrix $M_\star \in \mathbb{R}^{d\times d}$, we minimize the risk

$$\mathcal{R}(X) = \frac{1}{2}\left\| X^{(2)}X^{(1)} - M_\star \right\|_F^2. \tag{8}$$

We omit weight decay because it can prevent exact recovery of $M_\star$.

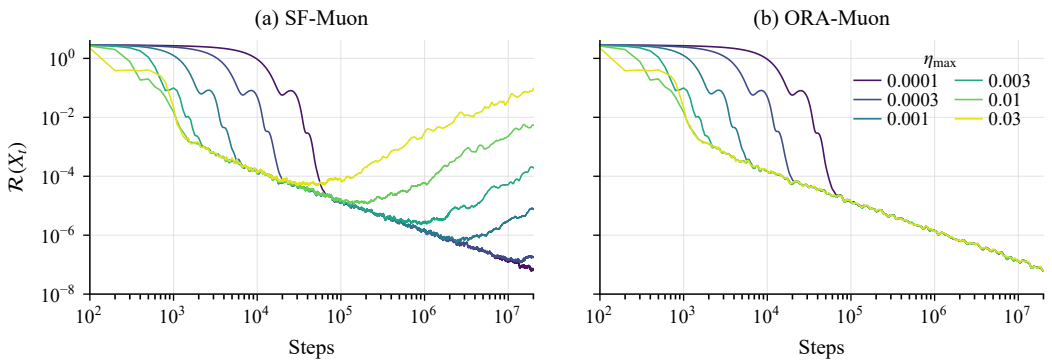

Figure 1: **Risk in noisy two-layer linear regression at six values of $\eta_{\max}$.** Only the learning rate rule after warmup differs. Smaller fixed learning rates delay SF-Muon's late risk increase, while ORA-Muon closely tracks the lower envelope of the SF-Muon curves.

4.1 Noisy Two-Layer Linear Regression

We compare SF-Muon and ORA-Muon on a $16 \to 24 \to 16$ network with a randomly generated full-rank target matrix $M_\star$. At each step, we sample a fresh minibatch of standard Gaussian inputs and add independent zero-mean Gaussian noise to the corresponding target outputs (Corlouer et al., 2026). We compute the gradient of the minibatch squared error at Y_t and evaluate $\mathcal{R}(X_t)$ exactly. The methods share random initialization and minibatches, and differ only in their learning rate rule after warmup (Appendix C.1).

In Figure 1, larger fixed learning rates give SF-Muon lower risk earlier but lead to later increases or fluctuations. Smaller learning rates delay this behavior, so the best fixed rate changes as training continues. After the initial transients, ORA-Muon's risk decreases along nearly overlapping curves close to the SF-Muon lower envelope. The smallest fixed rate remains competitive at 20M steps, but extending the runs at the two smallest learning rates to 100M reveals their separation from ORA-Muon (Appendix C.2). Similar late risk increases occur with SF-AdamW and persist under the Polyak learning rates from SF+ (Defazio, 2026) and SPSsafe (Oikonomou et al., 2026), while ORA-AdamW continues to reduce risk (Appendix C.3).

4.2 Local Convergence in the Deterministic Model

To study the limiting behavior of the same model, we use exact gradients $\nabla\mathcal{R}(Y_t)$ and Muon polar factors (Jordan et al., 2024), removing internal momentum to simplify the analysis. Let $n \geq d > 1$ and write $M_\star = U\,\mathrm{diag}(r_i^2)V^\top$, where U, V are orthogonal and the nonzero r_i have arbitrary fixed signs. For fixed $Q \in \mathbb{R}^{n\times d}$ with $Q^\top Q = I_d$, define the subspace

$$\mathcal{S} = \left\{ \left(Q\,\mathrm{diag}(x)V^\top,\ U\,\mathrm{diag}(x)Q^\top\right) : x \in \mathbb{R}^d \right\}. \tag{9}$$

Proposition 1. *If $X_N, P_N \in \mathcal{S}$, both SF-Muon and ORA-Muon keep $X_t, P_t \in \mathcal{S}$ for all $t \geq N$.*

Let x, p denote the coordinates of X, P in $\mathcal{S}$. The coordinates of Y are $y = x + \alpha p$, where $\alpha = 1-\beta$. The risk is $\mathcal{R}(X) = \frac{1}{2}\sum_i (x_i^2 - r_i^2)^2$. For the chosen zero-risk solution $r = (r_i)_{i=1}^d$, fix $\theta \in (0, 1/2)$ and $\eta_{\max} > 0$ with $B = 2\alpha\eta_{\max} < \frac{1}{2}\min_i |r_i|$. We compare the two methods in a nonempty open region where both x and y are close to r:

$$\Omega_{r,B} = \left\{ (x,p) : \|x - r\| < B, \quad \|x - r + \alpha p\|_\infty < \frac{\theta\alpha}{\sqrt{d}}\,\|p\| \right\}. \tag{10}$$

Theorem 1 (Local risk separation)**.** *There exists $c_* = c_*(\alpha, d, \theta) > 0$ such that the following holds. Suppose c_t satisfies*

$$0 < c_{t+1} \leq c_t \leq c_* \quad (t \geq N), \qquad c_t \to 0, \qquad \sum_{t=N}^{\infty} c_t = \infty. \tag{11}$$

Starting from the same state $(x_N, p_N) \in \Omega_{r,B}$ defined in (10) with the same $\eta_{\max}$, we have:

- ***SF-Muon:*** $x_t \to x_\infty$ *and* $\lim_{t\to\infty} \mathcal{R}(X_t) > 0$ *for almost every initial state.*
- ***ORA-Muon:*** $x_t \to r$ *and* $\lim_{t\to\infty} \mathcal{R}(X_t) = 0$ *for every initial state.*

In Theorem 1, SF-Muon generally converges to a nonoptimal point that is not even stationary. This falls short of first-order stationarity, a standard convergence target in nonconvex optimization (Nocedal & Wright, 2006). Our c_t in (7) satisfies (11) for sufficiently large N, and $\eta_{\max}$ can be arbitrarily small, with the local region shrinking accordingly. Proofs are in Appendix B.

Why the limiting risks differ. SF constructs X by averaging Z (Defazio et al., 2024), and later work studies the averaging weights (Song et al., 2025; Defazio, 2026). In our local deterministic setting, SF-Muon keeps each coordinate $z_i = x_i + p_i$ of Z on a fixed grid of spacing $\eta_{\max}$. Suppose, for contradiction, that $x_{t,i} \to r_i$, where r_i lies strictly between two adjacent grid points and is not their midpoint. Then $y_{t,i} - r_i$ eventually has the same sign as $z_{t,i} - r_i$, forcing $z_{t,i}$ to alternate between these two points. Under (11), averaging forces $x_{t,i}$ to converge to their midpoint, contradicting $x_{t,i} \to r_i$. Holding the local initial state and $\eta_{\max}$ fixed, all c_t satisfying (11) give the same limit (Appendix B.3), so changing the averaging weights alone cannot remove the error.

ORA-Muon satisfies $\|S_t\| \le \sqrt{c_t}\,\|P_t\|$ and preserves $\|y_t - r\| \le \rho\,\|x_t - r\|$ in $\Omega_{r,B}$ for a fixed $\rho < 1$. Iteration (2) gives

$$x_{t+1} = (1-\kappa_t)x_t + \kappa_t y_{t+1}, \qquad \kappa_t = \frac{c_t}{\alpha + \beta c_t}, \tag{12}$$

which moves x_t toward the more accurate y_{t+1}. Since (11) ensures $\sum_t \kappa_t = \infty$, we obtain $x_t \to r$ and zero limiting risk.

Validation of the local analysis. We test the local predictions in deterministic experiments from shared random initializations in the subspace (9), with $P_1 = 0$. All tested ORA-Muon trajectories eventually satisfy the local conditions of Theorem 1. The theoretical SF-Muon risk limits agree with the observed late risks. Details are in Appendix C.4.

5 Language Model Pretraining

5.1 Experimental Setup

Following AMUSE (Kim et al., 2026), we train 124M and 720M Llama-based models on FineWeb (Penedo et al., 2024), using the architectures and training setup from the LLM optimizer benchmark of Semenov et al. (2025). For Muon with cosine decay, we adopt the benchmark's selected D-Muon configuration, also used by AMUSE. D-Muon extends weight decay to Muon's matrix parameters, improving performance in the benchmark (Semenov et al., 2025, Figure 4). We refer to this baseline as Muon throughout.

Our 124M AMUSE baseline uses its published hyperparameters. SF-Muon uses the same settings but fixes $\beta = 0.95$, following AMUSE's SF-Muon comparison (Kim et al., 2026, Appendices E.2 and F.6). ORA-Muon only replaces SF-Muon's fixed learning rate after warmup with the online rule in (6). These 124M SF-Muon and ORA-Muon runs share the same initialization and training data order. Both use the weight decay term λZ_t unless stated otherwise, following AMUSE and SF-NorMuon (Kim et al., 2026; Apte et al., 2026). At each model size, we evaluate all methods on the same validation subset, using the output iterate X for SF and ORA. Appendix D.1 gives the full training configurations and evaluation details.

5.2 Improving Token Efficiency

To isolate the benefit of replacing the fixed learning rate with ORA, we compare 124M SF-Muon and ORA-Muon at the same peak learning rate $\eta_{\max} = 0.01$ in Figure 2(a). After their common warmup, SF-Muon keeps $\eta_t = 0.01$, while ORA selects $\eta_t \le 0.01$, with all other settings unchanged. Their losses remain close until ORA starts reducing the learning rate, after which ORA's advantage grows. ORA reaches SF-Muon's 128k validation loss with **42% fewer tokens** (Appendix D.2). This gain comes solely from reducing the learning rate after warmup.

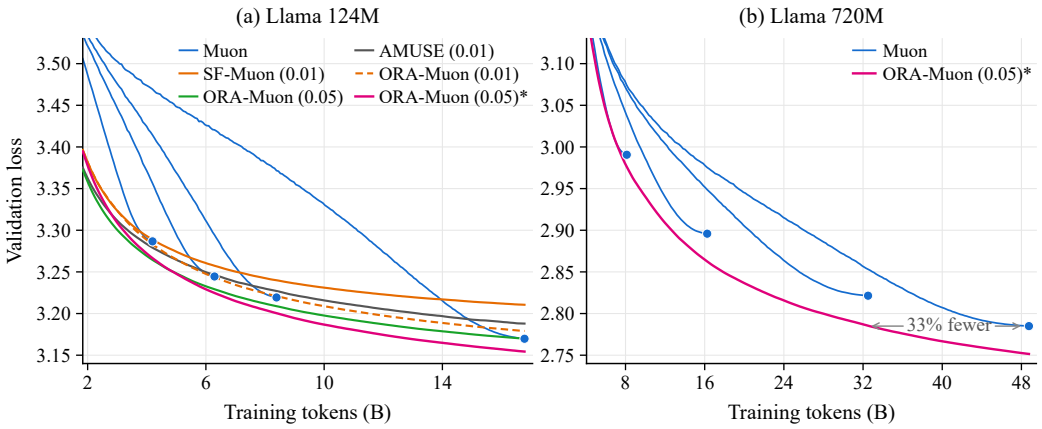

Figure 2: **Validation loss in language model pretraining.** Muon cosine runs end at 32k, 48k, 64k, and 128k steps (left) and 8k, 16k, 32k, and 48k (right). The arrow marks a 33% token saving at matched validation loss. Legend values denote $\eta_{\max}$. Starred ORA-Muon curves use $\lambda_X = 0.05$ and $\lambda_P = 0.02625$. Other ORA-Muon curves use $\lambda_X = \lambda_P = 0.05$.

Online control also allows a larger peak learning rate without retaining it throughout training. Increasing only $\eta_{\max}$ to 0.05 raises the token saving to **51%**. ORA-Muon outperforms AMUSE and achieves comparable or lower loss than Muon with cosine decay (Figure 2(a)).

With $\eta_{\max} = 0.05$ and Z decay, ORA's advantage over Muon with cosine decay narrows as training continues, and the two methods are nearly tied at 128k steps. Extrapolating the loss curves (Section 5.3) suggests that Muon may overtake this ORA run in longer training. To further improve late performance, we examine weight decay in terms of the output iterate and the pending update. We write D_t in (1) as

$$D_t = \mathrm{Opt}_t(Y_t) + \lambda_X X_t + \lambda_P P_t. \tag{13}$$

Here, $\mathrm{Opt}_t(Y_t)$ is computed from the gradient at Y_t and the optimizer state, before applying the learning rate or weight decay. Weight decay at $Z = X + P$ uses $\lambda_X = \lambda_P = \lambda$. Weight decay at $Y = X + (1-\beta)P$, used in the original SF experiments (Defazio et al., 2024), instead gives $\lambda_X = \lambda$ and $\lambda_P = (1-\beta)\lambda$.

Figure 2(a) also shows ORA with $\lambda_P = (1-\beta/2)\lambda = 0.02625$. Changing only λ_P initially gives higher loss than ORA with Z decay but improves later performance. It reaches the 128k Muon cosine loss with **23% fewer tokens**. Section 5.5 examines how λ_P controls forgetting of pending updates. ORA also improves AMUSE and SF-AdamW and remains effective with larger batches (Appendix D.3). ORA-AdamW outperforms the Polyak-based methods (Appendix D.4).

For 720M, we transfer the starred 124M ORA-Muon settings without further tuning, rescaling only the warmup to 4k steps to approximately preserve warmup tokens per parameter. ORA-Muon matches the final losses of the 16k, 32k, and 48k Muon cosine runs with **20%, 30%, and 33% fewer tokens**, respectively (Figure 2(b)).

5.3 Loss Scaling Laws and Extrapolation

We next ask whether observed loss curves can predict performance at longer training horizons. Empirical scaling laws describe how language model loss varies with model size and training data (Kaplan et al., 2020; Hoffmann et al., 2022). Here we study loss scaling laws with training length using the 124M experiments. Bu et al. (2026, Section 4.2) report that an inverse-square-root function closely fits the final losses of cosine runs. SF+ uses a horizontally shifted version to fit and extrapolate SF loss curves (Defazio, 2026, Section 9). We adopt this form:

$$\widehat{L}(T) = b + \frac{A}{\sqrt{T+s}}. \tag{14}$$

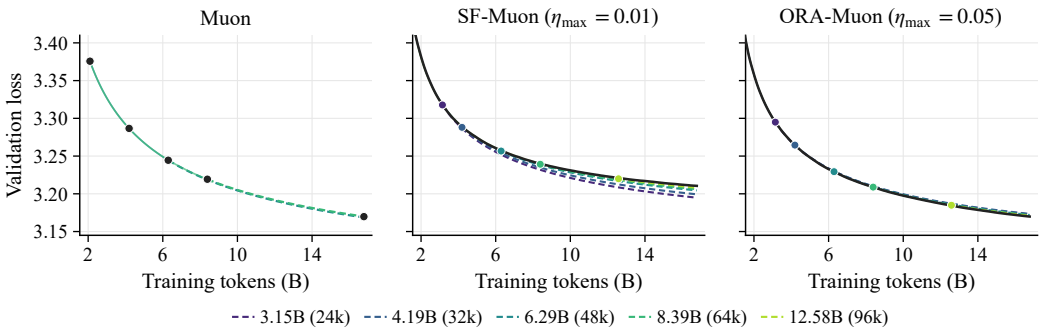

Figure 3: **Loss extrapolation.** Black marks or curves show observations, colored solid lines show fits, and dashed lines show predictions. Colors indicate fitting cutoffs.

Here T is the number of training tokens and s is the horizontal shift, both in billions of tokens. The parameter b is the limiting loss of the fitted curve. At each cutoff, we fit A, s, b to observations up to that point and evaluate predictions on later observations.

In Figures 3–5, each ORA fit starts at the first validation point after the learning rate has remained below $\eta_{\max}$ for 3k steps. This start remains fixed as the cutoff advances. In Figure 3, SF-Muon and ORA-Muon, both with Z decay, use the same fitting interval from 10.5k steps to each cutoff. Muon fits use the cosine endpoints available at each cutoff. All fits use unweighted least squares. Appendix E.1 describes the fitting procedure, and Appendix E.3 examines sensitivity to the 3k delay.

Figure 3 shows that fits to Muon's cosine endpoints accurately predict loss at longer training horizons. Predictions for SF-Muon underestimate later loss and move upward as more observations are included, whereas those for ORA-Muon are more accurate and stable. Across the tested cutoffs, ORA-Muon reduces long-range prediction RMSE by a median **60%** relative to SF-Muon. Appendix E.2 provides further results at 124M and 720M.

For the starred runs in Figure 2, extrapolation using (14) predicts that ORA's token savings over Muon with cosine decay continue to grow with training (Appendix E.4).

5.4 Learning Rates and the Training Horizon

We use the loss scaling law in (14) to study how learning rates affect performance across training horizons. In our experiments, the fitted shifts s are small relative to T later in training. The curves are then approximately $b + A/\sqrt{T}$, so their shape is largely determined by A and b. As the inverse-square-root term becomes smaller, differences in b become more prominent.

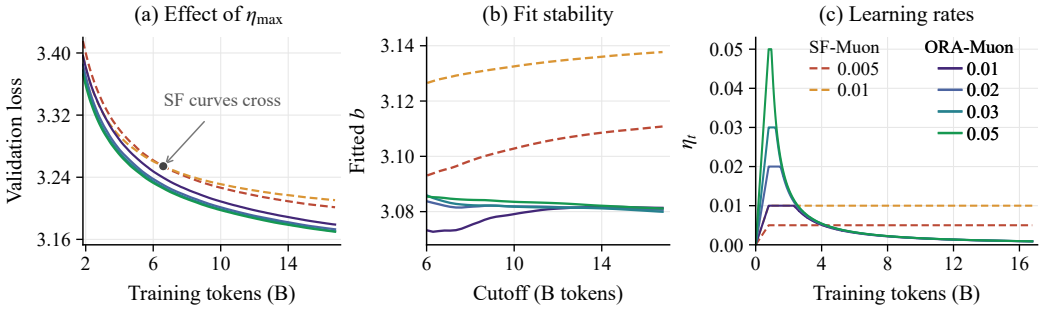

Figure 4: **Learning rates and late performance.** Left: validation loss. Middle: fitted b as the cutoff advances. Right: learning rates η_t. Legend values denote $\eta_{\max}$. All runs use Z decay.

Fixed learning rates. In Figure 4(a), SF-Muon with $\eta_{\max} = 0.01$ gives lower loss early, but the run with $\eta_{\max} = 0.005$ overtakes it around 6.6B tokens (about 50k steps). The fits through 128k

steps yield a larger A and a lower b for $\eta_{\max} = 0.005$, consistent with the observed crossing. For both learning rates, the fitted b also rises overall as the cutoff advances (Figure 4(b)), consistent with the upward revisions to SF-Muon's predictions in Figure 3. The preferred fixed learning rate therefore depends on the training horizon.

The horizon-free property. We use *horizon-free* to describe hyperparameter selection without this tradeoff between performance at shorter and longer training horizons. For example, if changing a hyperparameter mainly affects A, with b approximately unchanged, a smaller A gives lower loss across longer training horizons.

Online learning rates. Despite different values of $\eta_{\max}$, ORA-Muon automatically selects similar learning rates later in training (Figure 4(c)). Its loss curves also become close (Figure 4(a)). The fitted b values approach similar levels below those of SF-Muon and become more stable as the cutoff advances (Figure 4(b)). Jointly fitting the four curves through 128k steps with a shared b increases the pooled fitting RMSE by only 3.6% compared with separate fits (Appendix E.3).

ORA supports more accurate extrapolation (Section 5.3) and is largely horizon-free with respect to $\eta_{\max}$. These results provide a more reliable basis for studying anytime methods through loss scaling laws. We next examine whether reducing the decay of the pending update can further lower b.

5.5 Forgetting Pending Updates

In (13), λ_X acts on the output parameters, while λ_P shrinks the existing pending update to $(1 - \eta_t\lambda_P)P_t$. This removes part of the pending update without applying it to X, which we interpret as forgetting. Appendix F provides further details and discussion.

Figure 5 compares different fixed values of λ_P, with all other settings unchanged. Smaller λ_P gives higher loss early but lower loss later. The corresponding larger A and lower b capture this tradeoff, unlike changes to $\eta_{\max}$ under ORA, which leave b nearly unchanged (Figure 4(b)). The choice of λ_P is therefore not horizon-free. For the starred runs in Figure 2, we use $\lambda_P = (1 - \beta/2)\lambda_X$ as a simple intermediate choice. In the 124M experiment, this gives a fitted b close to Muon's and a smaller A, consistent with the observed token savings. Adapting λ_P online to control forgetting is a natural direction for future work.

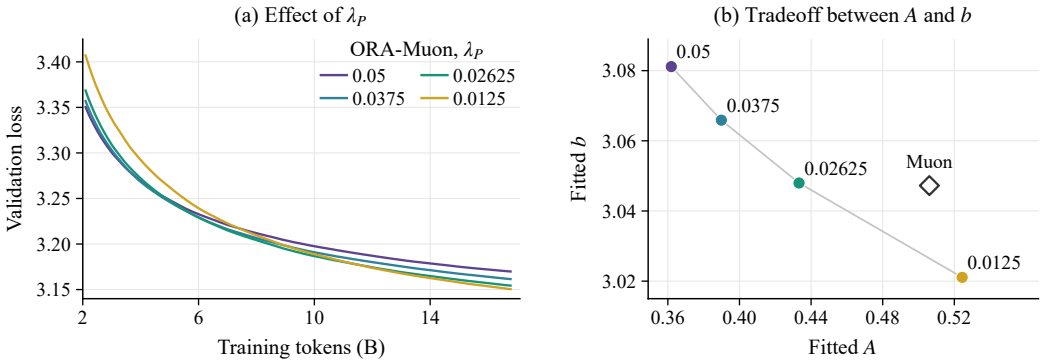

Figure 5: **Forgetting pending updates.** Left: validation loss. Right: fitted A, b through 128k steps, with Muon for comparison. All ORA-Muon runs use $\lambda_X = 0.05$ and $\eta_{\max} = 0.05$.

6 Conclusion

Our regression experiments and local theory expose failures of SF with constant learning rates. ORA controls new updates relative to the pending update, yielding largely horizon-free online learning rates. It improves token efficiency over strong Muon cosine baselines at 124M and 720M, with extrapolation predicting further savings in longer training. These results support studying anytime methods through loss scaling laws rather than performance at a single horizon.

AI USE STATEMENT

We used large language models to help organize and revise the manuscript, search for related work, implement and run experiments, and analyze and visualize results. AI tools also assisted with checking mathematical arguments and editing proofs. The authors directed the research, reviewed the AI-assisted work, and take responsibility for the final content.

REPRODUCIBILITY STATEMENT

Implementation details and proofs are provided in Appendices A and B. Appendices C–F describe the experimental settings and analyses. We provide supplementary code for the optimizers, experiments, and plotting, together with lightweight experimental logs.

REFERENCES

Niccolò Ajroldi, Antonio Orvieto, and Jonas Geiping. When, where and why to average weights? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 925–941. PMLR, 2025. URL https://proceedings.mlr.press/v267/ajroldi25a.html.

Anuj Apte, Pranav Deshpande, Niraj Kumar, Shouvanik Chakrabarti, and Junhyung Lyle Kim. Anytime training with schedule-free spectral optimization. *arXiv preprint arXiv:2605.23061*, 2026. URL https://arxiv.org/abs/2605.23061v1.

Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1810.02281.

Zhiqi Bu, Shiyun Xu, and Jialin Mao. Convex dominance in deep learning I: A scaling law of loss and learning rate. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/af288c3b7336851931b9424b30f073ac-Abstract-Conference.html.

Jiseok Chae and Donghwan Kim. Understanding schedule-free methods in nonconvex optimization: Rate guarantees and escaping saddles. *arXiv preprint arXiv:2607.09167*, 2026. URL https://arxiv.org/abs/2607.09167v1.

Guillaume Corlouer, Avi Semler, Alexander Strang, and Alexander Gietelink Oldenziel. Stochastic gradient descent in the saddle-to-saddle regime of deep linear networks. *arXiv preprint arXiv:2604.06366*, 2026. URL https://arxiv.org/abs/2604.06366v1.

Aaron Defazio. ScheduleFree+: Scaling learning-rate-free & schedule-free learning to large language models. *arXiv preprint arXiv:2605.19095*, 2026. URL https://arxiv.org/abs/2605.19095v1.

Aaron Defazio, Xingyu (Alice) Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The road less scheduled. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/136b9a13861308c8948cd308ccd02658-Abstract-Conference.html.

Clémentine C. J. Dominé, Lukas Braun, James E. Fitzgerald, and Andrew M. Saxe. Exact learning dynamics of deep linear networks with prior knowledge. *Journal of Statistical Mechanics: Theory and Experiment*, 2023(11):114004, 2023. doi: 10.1088/1742-5468/ad01b8. URL https://doi.org/10.1088/1742-5468/ad01b8.

Avrajit Ghosh, Soo Min Kwon, Rongrong Wang, Saiprasad Ravishankar, and Qing Qu. Learning dynamics of deep matrix factorization beyond the edge of stability. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/2d43f7a61b57f83619f82c971e4bddc0-Paper-Conference.pdf.

Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 76232–76264, 2024. doi: 10.52202/079017-2427. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/8b970e15a89bf5d12542810df8eae8fc-Abstract-Conference.html.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence*, pp. 876–885, 2018. URL https://auai.org/uai2018/proceedings/papers/313.pdf.

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

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. *arXiv preprint arXiv:2001.08361*, 2020. URL https://arxiv.org/abs/2001.08361.

Jueun Kim, Baekrok Shin, Jihun Yun, Beomhan Baek, Minhak Song, and Chulhee Yun. AMUSE: Anytime muon with stable gradient evaluation. *arXiv preprint arXiv:2605.22432*, 2026. URL https://arxiv.org/abs/2605.22432v2.

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

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. NorMuon: Making Muon more efficient and scalable. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/61880.

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. *arXiv preprint arXiv:2502.16982*, 2025. URL https://arxiv.org/abs/2502.16982v1.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1608.03983v5.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101v3.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5. URL https://doi.org/10.1007/978-0-387-40065-5.

Dimitris Oikonomou, Matthew Buchholz, Yuen-Man Pun, Robert M. Gower, and Nicolas Loizou. Taking the road less scheduled with adaptive Polyak steps. *arXiv preprint arXiv:2511.07767*, 2026. URL https://arxiv.org/abs/2511.07767v2.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Mark Sandler, Andrey Zhmoginov, Max Vladymyrov, and Nolan Miller. Training trajectories, mini-batch losses and the curious role of the learning rate. *arXiv preprint arXiv:2301.02312*, 2023. URL https://arxiv.org/abs/2301.02312.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014. URL https://www.saxelab.org/assets/papers/Saxe2014.pdf.

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/2509.01440v1.

Minhak Song, Beomhan Baek, Kwangjun Ahn, and Chulhee Yun. Through the river: Understanding the benefit of schedule-free methods for language model training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/b94ab4933fea38629a1308fb78cce2cc-Abstract-Conference.html.

A Algorithm Details

A.1 Accumulation and Application of Updates

Starting from $P_1 = 0$, expanding iteration (2) gives

$$\begin{aligned} P_{t+1} &= \sum_{i=1}^{t} \left[\prod_{j=i}^{t} (1 - c_j) \right] S_i, \\ X_{t+1} - X_1 &= \sum_{i=1}^{t} \left[1 - \prod_{j=i}^{t} (1 - c_j) \right] S_i. \end{aligned} \tag{15}$$

The first identity follows by repeated substitution. For the second, adding the two lines of (2) gives $X_{t+1} + P_{t+1} = X_t + P_t + S_t$. Summing over steps yields $X_{t+1} + P_{t+1} = X_1 + \sum_{i=1}^{t} S_i$. Substituting the first identity proves the second.

If $0 < c_j \leq 1$ and $\sum_{j=1}^{\infty} c_j = \infty$, then for each fixed i,

$$0 \leq \prod_{j=i}^{t} (1 - c_j) \leq \exp\left(- \sum_{j=i}^{t} c_j \right) \longrightarrow 0. \tag{16}$$

Here we used $1 - u \leq e^{-u}$ for $u \geq 0$. Thus, the coefficient of S_i in P_{t+1} tends to zero, and its coefficient in $X_{t+1} - X_1$ tends to one. This describes how each fixed update is applied to X, without requiring convergence of the sequence of iterates.

A.2 Implementation Details

For an efficient implementation, we store Y_t as the model parameters and Z_t in the optimizer state, as in the SF code.[1] We initialize $Y_1 = Z_1 = X_1$. For fixed $\beta \in (0, 1)$,

$$P_t = \frac{Z_t - Y_t}{\beta}, \qquad X_t = Z_t - P_t. \tag{17}$$

Thus, neither X_t nor P_t needs to be stored separately between steps.

Algorithm 2 implements one step of Algorithm 1 using these stored parameters and the same hyperparameters.

Algorithm 2 Implementation of an ORA step

Require: $Y = Y_t$, $Z = Z_t$, base optimizer state

Compute the gradient at Y and apply clipping if specified
Update the base optimizer state and compute D_t, including weight decay in (13)
if $t \leq T_{\mathrm{w}}$ **then**
    $\eta_t \leftarrow \eta_t^{\mathrm{w}}$
else
    $\eta_t \leftarrow \min\{\eta_{\max}, \sqrt{K c_t}\, \|Z - Y\|_{\mathcal{I}} / (\beta \|D_t\|_{\mathcal{I}})\}$
end if
$S_t \leftarrow -\eta_t D_t$ for all parameters
$Y \leftarrow (1 - c_t) Y + c_t Z + (1 - \beta + \beta c_t) S_t$
$Z \leftarrow Z + S_t$

If $\|D_t\|_{\mathcal{I}} = 0$ after warmup, we set $\eta_t = \eta_{\max}$.

The last two lines implement iteration (1) in place, with Y updated before Z. For validation, we temporarily load X_t from (17) as the model parameters and restore the saved Y_t afterward.

[1] https://github.com/facebookresearch/schedule_free

B LOCAL CONVERGENCE: SF-MUON VERSUS ORA-MUON

This appendix proves Proposition 1 and Theorem 1 for the deterministic two-layer model in Section 4.2. We use exact gradients and Muon polar factors, without internal momentum or weight decay. The analysis is restricted to the aligned subspace $\mathcal{S}$ in (9). The convergence results use the local region $\Omega_{r,B}$ in (10) and the conditions on c_t in (11).

Throughout, $\eta = \eta_{\max} > 0$, $\alpha = 1 - \beta$ with $0 < \beta < 1$, and $\|\cdot\|$ denotes the Euclidean norm over all entries.

B.1 INVARIANCE OF THE ALIGNED SUBSPACE

Proof of Proposition 1. If $X_t, P_t \in \mathcal{S}$, then $Y_t = X_t + \alpha P_t \in \mathcal{S}$. Let $y_t = x_t + \alpha p_t$ be its coordinates. The two layer gradients are

$$\begin{aligned} \nabla_{Y^{(1)}} \mathcal{R}(Y_t) &= Q \operatorname{diag}\big(y_{t,i}(y_{t,i}^2 - r_i^2)\big) V^\top, \\ \nabla_{Y^{(2)}} \mathcal{R}(Y_t) &= U \operatorname{diag}\big(y_{t,i}(y_{t,i}^2 - r_i^2)\big) Q^\top. \end{aligned} \tag{18}$$

The polar factor preserves the singular vectors and replaces nonzero singular values by one, leaving zero singular values unchanged. Thus $D_t, S_t \in \mathcal{S}$, and the coordinates s_t of $S_t = -\eta_t D_t$ satisfy

$$s_{t,i} = -\eta_t \operatorname{sign}\big(y_{t,i}(y_{t,i}^2 - r_i^2)\big). \tag{19}$$

Iteration (2) then gives $X_{t+1}, P_{t+1} \in \mathcal{S}$. □

Since both methods preserve $\mathcal{S}$, we can study their convergence using the coordinates x_t, p_t. Iteration (2) becomes

$$\begin{aligned} y_t &= x_t + \alpha p_t, & z_t &= x_t + p_t, \\ x_{t+1} &= x_t + c_t(p_t + s_t), & p_{t+1} &= (1 - c_t)(p_t + s_t). \end{aligned} \tag{20}$$

Let $k_t = \#\{i : y_{t,i}(y_{t,i}^2 - r_i^2) \neq 0\}$. The joint norms over both layers satisfy $\|P_t\|^2 = 2\|p_t\|^2$ and $\|D_t\|^2 = 2k_t$. Thus the learning rate rules for $t \geq N$ are

$$\eta_t^{\mathrm{SF}} = \eta, \qquad \eta_t^{\mathrm{ORA}} = \min\left\{\eta, \frac{\sqrt{c_t}\,\|p_t\|}{\sqrt{k_t}}\right\} \quad (k_t > 0). \tag{21}$$

When $k_t = 0$, both methods have $s_t = 0$. The risk is exactly $\mathcal{R}(X_t) = \frac{1}{2}\sum_i (x_{t,i}^2 - r_i^2)^2$.

B.2 LOCAL BOUNDS AND CONTRACTION

For the local analysis, fix $\theta \in (0, 1/2)$ and $B = 2\alpha\eta < \frac{1}{2}\min_i |r_i|$ as in Theorem 1. Define

$$e = x - r, \qquad a = y - r = e + \alpha p, \qquad E = \|e\|, \qquad \Delta = \|p\|, \tag{22}$$

and set

$$b_0 = \frac{\theta\alpha}{\sqrt{d}}, \qquad \rho = \frac{\theta}{1 - \theta} < 1. \tag{23}$$

We omit the step index in single-step calculations and use a superscript $+$ for the next state. The region $\Omega_{r,B}$ in (10) is given by $E < B$ and $\|a\|_\infty < b_0\Delta$. It is open and nonempty.

If $\|a\|_\infty \leq b_0\Delta$, the identity $e = a - \alpha p$ gives

$$\|a\| \leq \theta\alpha\Delta, \qquad \alpha(1 - \theta)\Delta \leq E \leq \alpha(1 + \theta)\Delta, \qquad \|a\| \leq \rho E. \tag{24}$$

In $\Omega_{r,B}$, we have $E, \Delta > 0$ and $\|a\|_\infty < B$. If $\|a\|_\infty \leq B$, then $|y_i - r_i| \leq B < |r_i|/2$ for all i. Hence y_i and $y_i + r_i$ have the same sign as r_i, giving $y_i(y_i + r_i) > 0$ and

$$\operatorname{sign}\big(y_i(y_i + r_i)(y_i - r_i)\big) = \operatorname{sign}(a_i). \tag{25}$$

For $0 < c < 1$, (20) and (12) give

$$\begin{aligned} a^+ &= a + \beta c p + (\alpha + \beta c)s, \\ e^+ &= (1 - \kappa)e + \kappa a^+, \qquad \kappa = \frac{c}{\alpha + \beta c} \in (0, 1). \end{aligned} \tag{26}$$

Lemma 1 (Local boundedness of SF-Muon)**.** *The set*

$$\mathcal{K}_B = \{(x,p) : \|e\|_\infty \le B,\ \|a\|_\infty \le B\} \tag{27}$$

contains $\Omega_{r,B}$*. For any* $(x,p) \in \mathcal{K}_B$ *and*

$$0 < c \le c_{\mathrm{SF}} := \min\left\{\frac{1}{2}, \frac{\alpha}{5\beta}\right\}, \tag{28}$$

the SF-Muon iteration with learning rate η *gives* $(x^+, p^+) \in \mathcal{K}_B$*.*

Proof. The inclusion $\Omega_{r,B} \subset \mathcal{K}_B$ follows from (24). In $\mathcal{K}_B$, (25) gives $s_i = -\eta\,\mathrm{sign}(a_i)$, and $|p_i| = |a_i - e_i|/\alpha \le 4\eta$. With $\gamma = (\alpha+\beta c)\eta \ge B/2$, (26) yields

$$\begin{aligned} |a_i^+| &\le \left|a_i - \gamma\,\mathrm{sign}(a_i)\right| + 4\beta c\eta \\ &\le \max\{B-\gamma, \gamma\} + 4\beta c\eta \\ &= \alpha\eta + 5\beta c\eta \le B. \end{aligned} \tag{29}$$

By (26), $|e_i^+| \le B$ as well. □

Lemma 2 (Local contraction of ORA-Muon)**.** *There exists* $c_{\mathrm{ORA}} = c_{\mathrm{ORA}}(\alpha, d, \theta) \in (0,1)$ *such that, for any* $(x,p) \in \Omega_{r,B}$ *and* $0 < c \le c_{\mathrm{ORA}}$*, the ORA-Muon iteration gives* $(x^+, p^+) \in \Omega_{r,B}$ *and*

$$E^+ \le \frac{1-\kappa}{1-\rho\kappa} E < E, \qquad \kappa = \frac{c}{\alpha+\beta c}. \tag{30}$$

Proof. Write $\tau = \sqrt{c}$ and first require $\tau < (1-\theta)/2$. By (24),

$$\Delta \le \frac{B}{\alpha(1-\theta)} = \frac{2\eta}{1-\theta}. \tag{31}$$

By (25), $k = \#\{i : a_i \ne 0\}$. If $k > 0$, then

$$\frac{\tau\Delta}{\sqrt{k}} \le \frac{2\eta\tau}{1-\theta} < \eta. \tag{32}$$

Thus the online term in (21) determines the learning rate, and

$$s = -\frac{\tau\Delta}{\sqrt{k}}\,\mathrm{sign}(a), \qquad \|s\| = \tau\Delta. \tag{33}$$

If $k = 0$, then $s = 0$, so $\|s\| \le \tau\Delta$ in every case.

For $k > 0$, set $\gamma = (\alpha + \beta\tau^2)\tau\Delta/\sqrt{k}$. Since $1 \le k \le d$,

$$\frac{\alpha\tau\Delta}{\sqrt{d}} \le \gamma \le (\alpha+\beta\tau^2)\tau\Delta. \tag{34}$$

Using $|a_i - \gamma\,\mathrm{sign}(a_i)| \le \max\{b_0\Delta - \gamma, \gamma\}$ and (34) in (26) gives

$$\left\|a^+\right\|_\infty \le \Delta\left[\max\left\{b_0 - \frac{\alpha}{\sqrt{d}}\tau, (\alpha+\beta\tau^2)\tau\right\} + \beta\tau^2\right]. \tag{35}$$

If $k = 0$, then $a = 0$ and $s = 0$, so $a^+ = \beta c p$ satisfies the same bound. Also, $\|s\| \le \tau\Delta$ gives

$$\Delta^+ = (1-c)\,\|p+s\| \ge (1-\tau^2)(1-\tau)\Delta =: L(\tau)\Delta > 0. \tag{36}$$

To obtain $\|a^+\|_\infty < b_0\Delta^+$ from (35) and (36), it suffices that

$$\begin{aligned} b_0 - \frac{\alpha}{\sqrt{d}}\tau + \beta\tau^2 &< b_0 L(\tau), \\ (\alpha+\beta\tau^2)\tau + \beta\tau^2 &< b_0 L(\tau). \end{aligned} \tag{37}$$

Both inequalities hold for all sufficiently small positive τ, because

$$\begin{aligned}\lim_{\tau\downarrow 0}\frac{b_0L(\tau)-b_0+\alpha\tau/\sqrt{d}-\beta\tau^2}{\tau}&=\frac{\alpha}{\sqrt{d}}-b_0>0,\\ \lim_{\tau\downarrow 0}\big[b_0L(\tau)-(\alpha+\beta\tau^2)\tau-\beta\tau^2\big]&=b_0>0.\end{aligned}\tag{38}$$

Choose $0<\tau_*<(1-\theta)/2$ so that (37) holds for every $0<\tau\le\tau_*$, and set $c_{\mathrm{ORA}}=\tau_*^2$. This choice depends only on α,d,θ and gives

$$\big\|a^+\big\|_\infty<b_0\Delta^+.\tag{39}$$

By (24), $\|a^+\|\le\rho E^+$. Thus (26) gives

$$E^+\le(1-\kappa)E+\rho\kappa E^+.$$

Rearranging proves (30). In particular, $E^+<B$, which together with (39) shows that $(x^+,p^+)\in\Omega_{r,B}$. □

For Theorem 1, we fix

$$c_*=\min\{c_{\mathrm{SF}},c_{\mathrm{ORA}}\}.\tag{40}$$

For $0<c\le c_*$, Lemma 1 applies to states in $\mathcal{K}_B$, and Lemma 2 to states in $\Omega_{r,B}$.

B.3 SF-Muon: Convergence to a Nonstationary Point

In $\mathcal{K}_B$, each coordinate of SF-Muon follows the scalar iteration below with $e_t=x_{t,i}-r_i$ and $v_t=z_{t,i}-r_i$, by (20) and (25).

Lemma 3 (Limit of the scalar SF iteration). *Let $(e_t,v_t)_{t\ge N}$ be a bounded sequence satisfying*

$$v_{t+1}=v_t-\eta\,\mathrm{sign}(\beta e_t+\alpha v_t),\qquad e_{t+1}=(1-c_t)e_t+c_tv_{t+1},\tag{41}$$

where $c_t>0$ is nonincreasing, $c_t\to 0$, and $\sum_{t=N}^\infty c_t=\infty$. For $u=(-v_N)\bmod\eta\in[0,\eta)$, define

$$\begin{gathered}\mu=\frac{\eta}{2}-u,\qquad L_u=-\frac{\alpha}{\beta}(\eta-u),\qquad U_u=\frac{\alpha}{\beta}u,\\ \xi(u)=\max\{L_u,\min\{U_u,\mu\}\}.\end{gathered}\tag{42}$$

Then $e_t\to\xi(u)$.

Proof. All v_t lie in $\mathcal{L}=-u+\eta\mathbb{Z}$. To bound e_t from above, fix $A>\xi(u)$ with $-\beta A/\alpha\notin\mathcal{L}$. Choose $\ell\in\mathcal{L}$ such that

$$\ell<-\frac{\beta}{\alpha}A<\ell+\eta,\qquad \mu_A=\ell+\frac{\eta}{2}.\tag{43}$$

We have $\mu_A\le\xi(u)<A$. Indeed, $A>L_u$ implies $\ell\le-u$, hence $\mu_A\le\mu$. This suffices when $\mu\le\xi(u)$. Otherwise $\xi(u)=U_u$ and $A>U_u$ implies $\ell\le-u-\eta$, so $\mu_A\le\mu-\eta<0\le\xi(u)$.

To prove $\limsup_t e_t\le A$, define the auxiliary function

$$\Psi_A(v)=\frac{(v-\ell-\eta)^2}{2\eta}.\tag{44}$$

By (41), $e_t\ge A$ and $v_{t+1}\ge v_t$ imply $v_t\le\ell$. Expanding (44) for the three possible increments $-\eta,0,\eta$ therefore gives

$$v_{t+1}-\mu_A\le\Psi_A(v_t)-\Psi_A(v_{t+1})\qquad\forall\,t\in\{j\ge N:e_j\ge A\}.\tag{45}$$

Set $V_t=e_t+c_t\Psi_A(v_t)$. By (41), (45), and $c_{t+1}\le c_t$,

$$\begin{aligned}V_{t+1}-V_t&\le c_t(\mu_A-e_t)+(c_{t+1}-c_t)\Psi_A(v_{t+1})\\ &\le c_t(\mu_A-e_t)\qquad\forall\,t\in\{j\ge N:e_j\ge A\}.\end{aligned}\tag{46}$$

Since (e_t,v_t) is bounded, there is a constant $C>0$, independent of t, such that $0\le V_t-e_t\le Cc_t$ and $|e_{t+1}-e_t|\le Cc_t$ for all $t\ge N$. If $e_t\ge A$ eventually, (46) and $\sum_t c_t=\infty$ force $V_t\to-\infty$, contradicting its boundedness. Thus $e_t<A$ at arbitrarily late steps.

Consider any subsequent interval with $e_{k-1} < A$ and $e_j \geq A$ for all $j = k, \ldots, t$. By (46),

$$e_t \leq V_t \leq V_k \leq A + C(c_{k-1} + c_k). \tag{47}$$

Every sufficiently late step with $e_t \geq A$ belongs to such an interval. As these steps tend to infinity, the corresponding starting indices k also tend to infinity. Since $c_t \to 0$, (47) gives $\limsup_t e_t \leq A$. The excluded values of A form a discrete set. Letting $A \downarrow \xi(u)$ gives $\limsup_t e_t \leq \xi(u)$.

For the lower bound, $(-e_t, -v_t)$ satisfies the same scalar iteration with phase $u' = (-u) \bmod \eta$. Formula (42) gives $\xi(u') = -\xi(u)$, including when $u = 0$. Applying the upper bound to this sequence yields $\liminf_t e_t \geq \xi(u)$, proving convergence. □

Proof of Theorem 1 for SF-Muon. Since $(x_N, p_N) \in \Omega_{r,B} \subset \mathcal{K}_B$ and $c_t \leq c_*$ for all $t \geq N$, Lemma 1 gives $(x_t, p_t) \in \mathcal{K}_B$ for all $t \geq N$. Thus (25) holds, and each coordinate follows (41) with $e_{t,i} = x_{t,i} - r_i$ and $v_{t,i} = z_{t,i} - r_i$. These scalar sequences are bounded: $|e_{t,i}|, |a_{t,i}| \leq B$ and $v_{t,i} = (a_{t,i} - \beta e_{t,i})/\alpha$. Let $u_i = (r_i - z_{N,i}) \bmod \eta$. Lemma 3 gives

$$\begin{aligned} x_{t,i} &\longrightarrow x_{\infty,i} = r_i + \xi(u_i), \\ \mathcal{R}(X_t) &\longrightarrow \frac{1}{2} \sum_i \big[\xi(u_i)(2r_i + \xi(u_i))\big]^2. \end{aligned} \tag{48}$$

Since $|x_{\infty,i} - r_i| \leq B$, each $x_{\infty,i}$ is nonzero and has the same sign as r_i. Zero risk is therefore equivalent to $\xi(u_i) = 0$ for all i. By (42), this holds exactly when each u_i is 0 or $\eta/2$, or equivalently

$$x_N + p_N - r \in \frac{\eta}{2}\mathbb{Z}^d. \tag{49}$$

This set is a countable union of d-dimensional affine subspaces in the $2d$-dimensional coordinates (x_N, p_N), so its intersection with $\Omega_{r,B}$ has Lebesgue measure zero.

For every other initial state in $\Omega_{r,B}$,

$$\|\nabla \mathcal{R}(X_\infty)\|^2 = 2 \sum_i x_{\infty,i}^2 (x_{\infty,i}^2 - r_i^2)^2 > 0. \tag{50}$$

Hence X_∞ is nonstationary for the full matrix objective (8). The limiting risk in (48) depends on the initial grid, η, and β, but not on the particular sequence c_t satisfying (11). □

The same argument proves the SF-Muon conclusions for initial states in the larger region $\mathcal{K}_B$.

B.4 ORA-Muon: Convergence to a Zero-Risk Solution

Proof of Theorem 1 for ORA-Muon. Since $(x_N, p_N) \in \Omega_{r,B}$ and $c_t \leq c_*$ for all $t \geq N$, Lemma 2 gives $(x_t, p_t) \in \Omega_{r,B}$ and (30) at every step $t \geq N$. The contraction factor satisfies

$$\frac{1 - \kappa_t}{1 - \rho\kappa_t} = 1 - \frac{(1-\rho)\kappa_t}{1 - \rho\kappa_t} \leq \exp\big(-(1-\rho)\kappa_t\big). \tag{51}$$

Since $\kappa_t \geq c_t$ and $\sum_{t=N}^{\infty} c_t = \infty$, iterating (30) gives

$$E_t \leq E_N \exp\left(-(1-\rho) \sum_{j=N}^{t-1} \kappa_j\right) \longrightarrow 0. \tag{52}$$

By (24), $p_t \to 0$ as well. Thus $x_t, y_t, z_t \to r$ and $\mathcal{R}(X_t) \to 0$. □

C Additional Two-Layer Linear Experiments

C.1 Noisy Model and Experimental Setup

The noisy experiments use the $16 \to 24 \to 16$ network and risk (8). We use a randomly generated full-rank target $M_\star$ with largest singular value one and condition number four. We independently sample the entries of $X_1^{(1)}$ from $\mathcal{N}(0, 0.2^2/d)$ and those of $X_1^{(2)}$ from $\mathcal{N}(0, 0.2^2/n)$. We initialize $P_1 = 0$ and the optimizer moments to zero. All experiments use double precision, zero weight decay, and no gradient clipping.

At step t, we draw $m = 64$ fresh pairs with $\mathbf{a}_{t,j} \sim \mathcal{N}(0, I_d)$ and $\mathbf{b}_{t,j} = M_\star \mathbf{a}_{t,j} + \boldsymbol{\varepsilon}_{t,j}$, where $\boldsymbol{\varepsilon}_{t,j} \sim \mathcal{N}(0, 0.25^2 I_d)$. Inputs and noise are independent across samples and steps. Writing

$$f_t(X) = \frac{1}{2m} \sum_{j=1}^{m} \left\| X^{(2)} X^{(1)} \mathbf{a}_{t,j} - \mathbf{b}_{t,j} \right\|^2, \qquad g_t = \nabla f_t(Y_t), \quad Y_t = X_t + (1-\beta) P_t, \tag{53}$$

we have $\mathbb{E}[g_t] = \nabla \mathcal{R}(Y_t)$, where the expectation is over the fresh minibatch with Y_t held fixed.

Within each comparison, all methods and learning rate choices share the target, initialization, and minibatches. SF-Muon and ORA-Muon use internal Muon momentum 0.95, $\beta = 0.95$, and c_t from (7) with a 500-step linear warmup. ORA-Muon starts its online rule at step 501, uses $K = 1$, and determines the learning rate using all parameters. At a common $\eta_{\max}$, the only difference within each SF/ORA pair is the learning rate rule after warmup.

C.2 Longer Runs at Small Learning Rates

At 20M steps in Figure 1, SF-Muon's risk is still decreasing at the smallest learning rate, while a slight increase is already visible at the next smallest. To examine their behavior over longer training, we continue these runs to 100M steps (Figure 6). SF-Muon's risk eventually increases, while ORA-Muon's risk continues to decrease along nearly coincident curves. Thus a fixed learning rate that looks competitive over a shorter run can lose that advantage when the same run is continued.

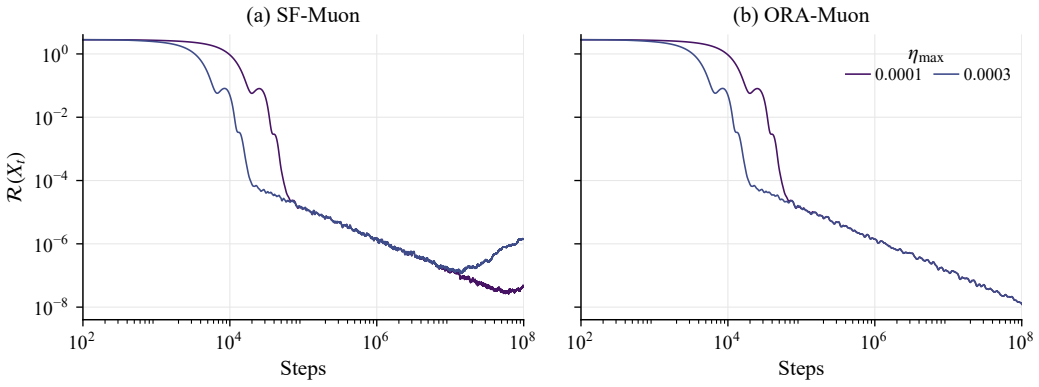

Figure 6: **The two smallest Muon learning rates extended to 100M steps.** Smaller fixed learning rates delay the separation from ORA-Muon.

C.3 AdamW and Polyak-Based Learning Rates

Figure 7 extends the comparison to AdamW over 100M steps. SF-AdamW and ORA-AdamW retain the warmup and c_t settings above, with $\beta = 0.9$, Adam moment coefficients $(0.9, 0.95)$, and $\epsilon = 10^{-8}$. As with Muon, larger fixed learning rates lead to late risk increases or oscillations, while smaller rates delay this behavior. ORA-AdamW's risk continues to decrease along similar curves across different values of $\eta_{\max}$.

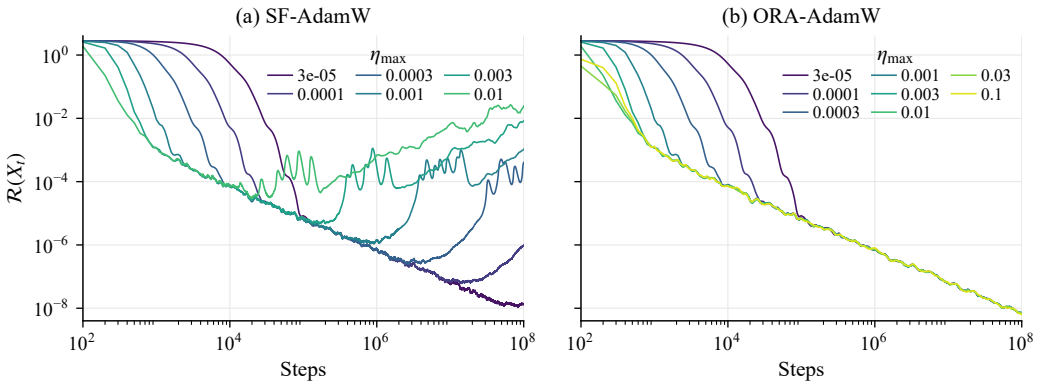

Figure 7: **SF-AdamW and ORA-AdamW over 100M steps.** Matching values of $\eta_{\max}$ share a color. ORA additionally uses 0.03 and 0.1.

Polyak-based learning rates. We also compare ORA with two Polyak-based learning rates: the one proposed in SF+ (Defazio, 2026, Algorithm 1) and the safeguarded learning rate used by SF-Adam-SPSsafe (Oikonomou et al., 2026, Algorithm 3). We fix $\beta = 0.9$ and set the supplied loss lower bound to $f^\star = 0$, following the settings used in both works.

We apply a constant global multiplier ρ to the SF+ Polyak and SPSsafe learning rates. Setting $\rho = 1$ causes early risk increases in both methods, so we use smaller multipliers to examine their behavior over longer runs. This scaling is separate from the warmup factor $u_t = \min\{t/500, 1\}$. In SF+, ρ corresponds to the base `lr` in its code.[2] For SPSsafe, we introduce ρ as an additional global multiplier, which does not appear in the original paper (Oikonomou et al., 2026).

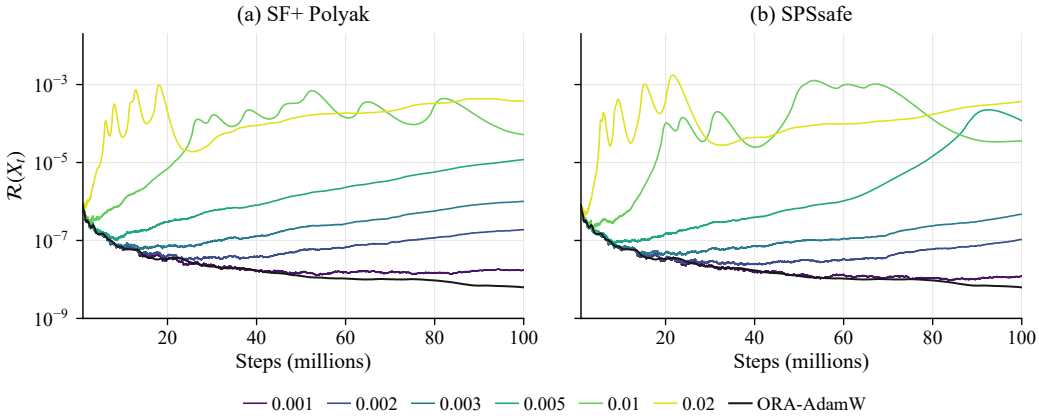

Figure 8: **Polyak-based learning rates over 100M steps.** Legend values denote the global multiplier ρ. Black shows ORA-AdamW with $\eta_{\max} = 0.1$ from Figure 7 as a reference.

For the SF+ Polyak learning rate, Adam uses the settings above. We set $\nu_0 = 0$, $\nu_t = 0.9\nu_{t-1} + 0.1\sqrt{\pi/2}\,\|g_t\|_1$, and $\hat{\nu}_t = \nu_t/(1-0.9^t)$. Writing $[a]_+ = \max\{a, 0\}$, the learning rate is

$$\eta_t^{\mathrm{SF+}} = \rho u_t \frac{[f_t(Y_t) - f^\star + \beta\langle g_t, P_t\rangle]_+}{\hat{\nu}_t}. \tag{54}$$

We use the c_t rule of Defazio (2026, Algorithm 1), with $c_t = 1$ during the 500-step warmup.

For SPSsafe, we use Adam without first-moment momentum. For each parameter coordinate j, we set $v_{0,j} = 0$, $v_{t,j} = 0.98v_{t-1,j} + 0.02g_{t,j}^2$, and $\hat{v}_{t,j} = v_{t,j}/(1-0.98^t)$. Setting $h_{t,j} = \sqrt{\hat{v}_{t,j}} + 10^{-8}$

[2] https://github.com/facebookresearch/schedule_free

gives

$$\eta_t^{\mathrm{SPSsafe}} = \rho u_t \frac{[f_t(Y_t) - f^\star + \beta\langle g_t, P_t\rangle]_+}{\max\left\{\sum_j g_{t,j}^2/h_{t,j},\, M\right\}}. \tag{55}$$

We set $M = 10$ and use $c_t = 1/t$, with steps indexed from one (Oikonomou et al., 2026, Algorithm 3).

The late risk increase persists with both Polyak-based learning rates, although smaller ρ delays its onset (Figure 8). ORA-AdamW continues to reduce risk.

C.4 Validation of the Local Analysis

We use the deterministic vector iteration (20) from random initialization to test whether trajectories eventually satisfy the local conditions and whether SF-Muon's observed risks agree with its predicted limits. The five settings are $(d, n) \in \{(16, 16), (16, 24), (32, 48)\}$ with $r_i = 1$, and $(d, n) = (16, 24)$ with r_i^2 spaced linearly from one to $1/2$ or geometrically from one to $1/128$. We set $p_1 = 0$ and generate x_1 as the singular values of an $n \times d$ matrix with independent $\mathcal{N}(0, \sigma^2/n)$ entries, using four draws at each $\sigma \in \{0.05, 0.1, 0.3\}$. Both methods share the initializations and use the same values of $\eta_{\max} \in \{0.001, 0.003, 0.01\}$, giving 36 pairs per setting and 180 in total. All runs use $\beta = 0.95$, a 1000-step linear warmup, c_t from (7), and 2M steps.

The SF-Muon risk limits can be predicted immediately after the 1000-step warmup. We compute the dotted lines in Figure 9(a) from $z_{1001} = x_{1001} + p_{1001}$ using (48). Across all SF trajectories, the mean risk over steps 1.6M–2M differs from these early predictions by at most 0.43%.

With $\theta = 0.2$ and $B = 2(1 - \beta)\eta_{\max}$, Figure 9(b) shows the fraction of ORA-Muon trajectories satisfying the local conditions of Theorem 1. All ORA trajectories satisfy these conditions by 106k steps and continue to satisfy them through 2M steps. All SF-Muon trajectories enter and remain in the larger region $\mathcal{K}_B$ defined in (27) of Lemma 1. The SF-Muon convergence result also applies to initial states in $\mathcal{K}_B$ (Appendix B.3).

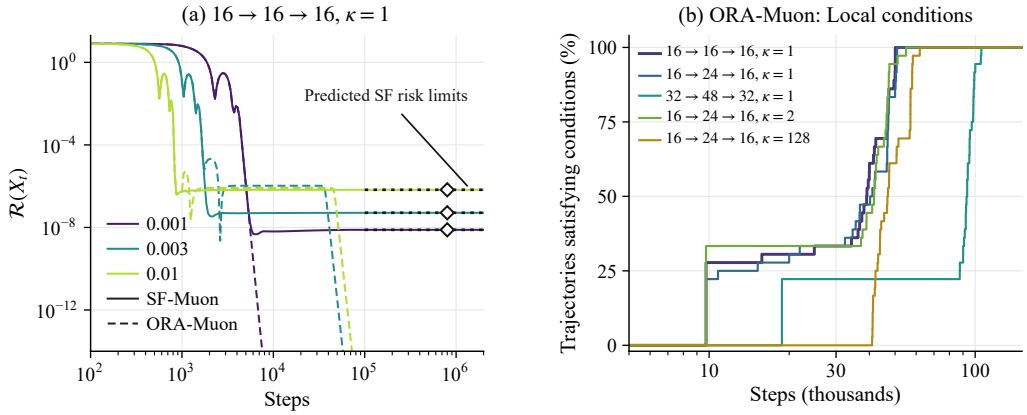

Figure 9: **Risk trajectories and validation of local conditions.** (a) Colors indicate $\eta_{\max}$. Dotted lines and diamonds mark the SF risk limits from (48). (b) Fractions of ORA-Muon trajectories satisfying the local conditions of Theorem 1. κ is the target condition number.

D Language Model Configurations and Additional Results

D.1 Experimental Setup

We use the 124M and 720M Llama-based models from the optimizer benchmark (Semenov et al., 2025) and its public codebase,[3] following AMUSE (Kim et al., 2026). Both models have 12 layers with RMSNorm, SwiGLU activations, and tied input and output embeddings. We train on FineWeb sample-100BT (Penedo et al., 2024), using the benchmark's data preparation with the GPT-2 tokenizer. Table 1 summarizes the model dimensions and batch sizes. Compared runs share initialization, data order, and validation data within each model size and context length.

Table 1: **Model dimensions and batch sizes.**

	124M	124M, larger batch	720M
Hidden dimension	768	768	2048
Attention heads	12	12	16
Context length	512	1024	512
Sequences per step	256	512	1984
Tokens per step	131,072	524,288	1,015,808

We report cross-entropy loss on a fixed subset of approximately 2.1M validation tokens, evaluating SF, ORA, and AMUSE at X. Training uses BF16 mixed precision with FP32 parameters, gradients, and optimizer states.

For the main Muon comparisons, the cosine baseline uses the configurations selected in the optimizer benchmark, also adopted by AMUSE (Semenov et al., 2025; Kim et al., 2026). Muon uses cosine warmup from $0.01\eta_{\max}$ to $\eta_{\max}$, followed by cosine decay back to $0.01\eta_{\max}$. Following the benchmark implementation, weight decay excludes embeddings and normalization parameters.

SF-Muon and ORA-Muon use AMUSE's 124M settings for gradient clipping, linear warmup, and weight decay on all parameters, with the fixed $\beta = 0.95$ selected in its SF-Muon comparison (Kim et al., 2026, Appendices E.2 and F.6). We use c_t defined by (7). Muon, SF-Muon, and ORA-Muon use Muon momentum of 0.95 with Nesterov acceleration and auxiliary AdamW $\epsilon = 10^{-8}$. The main 124M SF-Muon baseline uses $\eta_{\max} = 0.01$ and $\lambda_X = \lambda_P = 0.05$.

SF hyperparameters can involve tradeoffs between early and late performance (Sections 5.4 and 5.5). We focus on learning rate and weight decay in our main 124M ORA-Muon comparisons, keeping the other optimizer hyperparameters fixed. For 720M, we use the 124M ORA-Muon hyperparameters in Table 2 without further tuning, rescaling only the warmup length.

Table 2: **Muon and ORA-Muon hyperparameters for Figure 2.** ORA-Muon entries refer to the starred curves.

	124M	720M
Muon with cosine decay		
$\eta_{\max}$	0.002	0.001
Warmup steps	2000	2000
Auxiliary AdamW (β_1, β_2)	$(0.8, 0.999)$	$(0.9, 0.99)$
Weight decay	0.1	0.1
Gradient clipping	0.5	0.1
ORA-Muon		
$\eta_{\max}$	0.05	0.05
Warmup steps	6000	4000
Auxiliary AdamW (β_1, β_2)	$(0, 0.999)$	$(0, 0.999)$
β	0.95	0.95
(λ_X, λ_P)	$(0.05, 0.02625)$	$(0.05, 0.02625)$
Gradient clipping	0.5	0.5

For SF-Muon and ORA-Muon, we rescale the 6k linear warmup when changing batch or model size. The larger-batch 124M runs use 1500 steps to match the number of warmup tokens in the main 124M runs. The 720M run uses 4k steps to keep the number of warmup tokens per parameter

[3] https://github.com/epfml/llm-optimizer-benchmark

approximately the same as in the 124M runs. We use this rescaling for simplicity, without further tuning the warmup length.

D.2 Validation Loss and Token Savings

Tables 3 and 4 report numerical results for the 124M and 720M comparisons in Figure 2. We estimate the tokens needed to match a baseline's final validation loss by linear interpolation between the first pair of consecutive ORA evaluations that cross that loss. Token saving is the percentage reduction relative to the baseline's training tokens.

Table 3: **124M validation loss at selected training steps.**

Steps	Tokens (B)	Muon	SF-Muon	AMUSE	ORA-Muon		
	$\eta_{\max}$	0.002	0.01	0.01	0.01	0.05	0.05, λ_P=0.02625
32k	4.19	3.2867	3.2888	3.2791	3.2830	**3.2644**	3.2665
48k	6.29	3.2445	3.2574	3.2463	3.2435	3.2292	**3.2247**
64k	8.39	3.2193	3.2400	3.2271	3.2209	3.2088	**3.2004**
128k	16.78	3.1698	3.2105	3.1880	3.1791	3.1698	**3.1544**

Table 4: **720M validation loss and token savings.**

Steps	Tokens (B)	Muon	ORA-Muon	Token saving
8k	8.13	2.9906	**2.9762**	7.1%
16k	16.25	2.8959	**2.8625**	20.1%
32k	32.51	2.8214	**2.7854**	29.9%
48k	48.76	2.7849	**2.7515**	33.0%

D.3 ORA across Optimizers and Batch Sizes

To assess ORA's effectiveness across training settings and SF variants, we test a larger batch, combine ORA with AMUSE's varying β_t, and replace Muon with AdamW in the following 124M experiments. Each setting includes a comparison at the same $\eta_{\max}$, changing only the learning rate rule after warmup to isolate ORA's benefit. We also test larger values of $\eta_{\max}$ to illustrate further gains, without an extensive hyperparameter search.

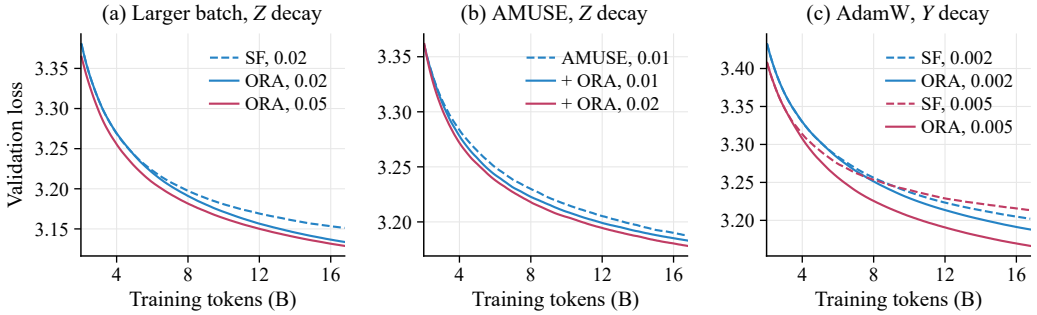

Figure 10: **Additional 124M comparisons.** Legend values denote $\eta_{\max}$. Dashed lines use fixed learning rates after warmup, and solid lines use ORA.

Larger batch. We use the larger-batch 124M setting in Table 1 and train for 32k steps, matching the total training tokens of the main runs. With Z decay and the same $\eta_{\max} = 0.02$, ORA-Muon improves late performance over SF-Muon. Increasing only ORA's $\eta_{\max}$ to 0.05 gives a further improvement (Figure 10(a)).

AMUSE. We follow AMUSE's update equations and 124M hyperparameters (Kim et al., 2026). Online learning rate control provides an additional benefit alongside its varying β_t, with a modest improvement at the same $\eta_{\max} = 0.01$ and a further gain at $\eta_{\max} = 0.02$ (Figure 10(b)).

AdamW. We start from the benchmark's SF-AdamW settings (Semenov et al., 2025). In addition, we use momentum in AdamW and apply Y decay to all parameters, following SF+ (Defazio, 2026). SF-AdamW and ORA-AdamW use a decay coefficient of 0.1, $\beta = 0.9$, Adam moments $(0.9, 0.9999)$, an 8k linear warmup, and c_t from (7). As with SF-Muon, fixed learning rates give an early–late tradeoff: $\eta_{\max} = 0.005$ gives lower loss than 0.002 early in training but is overtaken later (Figure 10(c)). ORA improves late performance at both values of $\eta_{\max}$, with $\eta_{\max} = 0.005$ giving the lowest final loss among these AdamW runs.

D.4 Comparisons with Polyak-Based Methods

SF+ (Defazio, 2026) and SF-Adam-SPSsafe (Oikonomou et al., 2026) also choose learning rates online, making them relevant alternatives to ORA. Both Polyak methods use Adam-based optimizers, following the original papers. We compare their performance and stability in the two 124M settings from Table 1. Runs use 128k steps with the main batch and 32k with the larger batch, matching total training tokens.

The Polyak learning rates follow (54) and (55), with $\rho = 1$, $f^\star = 0$, and the warmup lengths specified below. All Adam-based methods use $\beta = 0.9$ and apply decay to all parameters. The SF-AdamW and ORA-AdamW references use the settings in Appendix D.3, with the SF-AdamW warmup shortened to 2k steps in the larger-batch setting to preserve the number of warmup tokens.

SF+ Polyak. We base the optimizer setup on the 120M Polyak experiment in Defazio (2026, Figure 8 and Appendix C.10). We use Adam moments $(0.9, 0.95)$, gradient clipping at one, and an EMA coefficient of 0.9 for the gradient ℓ_1 norm. AdamC contributes $-5\eta_t^2 Y_t$ to the update of Z. Linear warmup lasts 3124 steps with the main batch and 781 with the larger batch, approximately preserving the original 0.4B warmup tokens. We compute c_t following Defazio (2026, Algorithm 1), with $c_t = 1$ during warmup.

SPSsafe. We take the optimizer hyperparameters from the 124M language model experiments in Oikonomou et al. (2026, Appendix C.3), using Adam without momentum, a second-moment coefficient of 0.98, safeguard $M = 10$, and no gradient clipping. We apply Y decay with displacement $-0.1\eta_t Y_t$. Linear warmup lasts 7500 steps with the main batch and 1875 with the larger batch, preserving the approximately 0.98B warmup tokens in those experiments. We use $c_t = 1/t$ from Oikonomou et al. (2026, Algorithm 3).

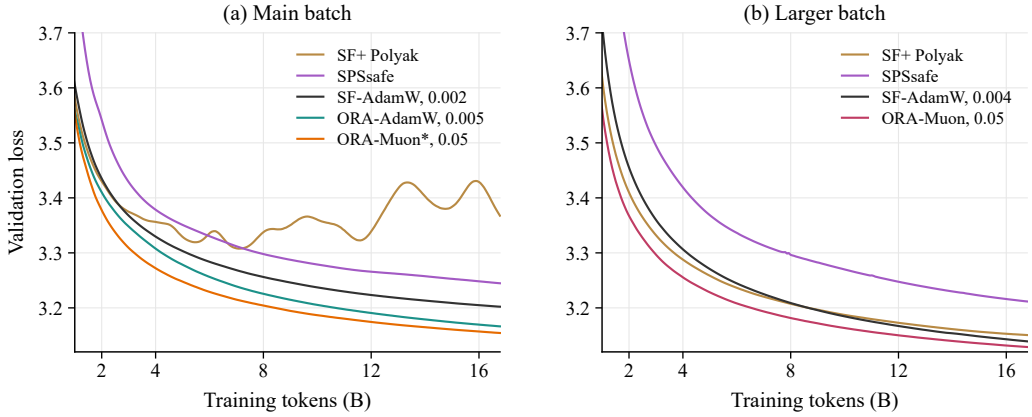

Figure 11: **Polyak-based methods in two 124M training settings.** SF-AdamW and ORA-AdamW use Y decay on all parameters. Legend values denote $\eta_{\max}$. ORA-Muon* is the starred 124M run in Figure 2, and the larger-batch ORA-Muon run is from Figure 10(a).

SF+'s validation loss rises and oscillates late in training with the main batch, whereas it continues to decrease with the larger batch (Figure 11). These results show that SF+ does not perform reliably across the two training settings. Both SF+ and SPSsafe finish with higher validation loss than SF-AdamW in both settings. ORA offers an effective alternative to these Adam-based Polyak methods and also works with Muon. The stronger performance of ORA-Muon highlights the practical value of combining online learning rates with a stronger base optimizer.

E Loss Scaling Laws and Extrapolation

E.1 Fitting and Extrapolation Protocol

We evaluate how well the loss model in (14) predicts validation loss later in training. This inverse-square-root function with a horizontal shift follows SF+ (Defazio, 2026, Section 9). We measure T, s in billions of tokens and estimate A, b, s by unweighted nonlinear least squares. All fits use the validation subset described in Appendix D.1. Muon fits use the final losses of cosine runs with different training lengths. SF-Muon and ORA-Muon fits use validation records from a single run.

To limit the influence of the transition to online control, ORA fitting begins at the first validation point after the learning rate has remained below $\eta_{\max}$ for 3k steps. The start then remains fixed as the fitting cutoff advances. Each SF comparison uses the same start as its paired ORA run. Each SF-Muon or ORA-Muon fit requires at least 12 observations, spanning at least 5.5k steps in the main 124M setting and 2.2k in the 720M setting. This minimum permits a fit but does not establish that the history is sufficient for accurate extrapolation. We assess the 3k choice by repeating the analysis with delays from 2k to 6k, rather than selecting a start by its prediction error.

We simulate predictions from shorter histories of the completed runs. At each cutoff, the fit uses only observations available by that step. Later observations are used solely to evaluate its predictions. A small error on the fitted interval need not imply a small error beyond it, so our primary measure is prediction error. Let C and T_{end} denote the token counts, in billions, at the cutoff and the end of the run. To focus on long-range prediction, we compute RMSE over validation observations in $[(C + T_{\text{end}})/2, T_{\text{end}}]$, the second half of the remaining interval, and also report the error at T_{end}. At a given cutoff, every method and fitting start in a comparison uses the same prediction targets.

E.2 Extrapolation at 124M and 720M

The Muon cosine runs provide a reference. At both model sizes, fits to the final losses of shorter runs closely predict those of longer runs (Table 5).

The fits for SF-Muon ($\eta_{\max} = 0.01$) and ORA-Muon ($\eta_{\max} = 0.05$) in Figure 3 both start at 10.5k steps. Figure 12 shows predicted minus observed validation loss. Predictions for SF-Muon increasingly underestimate loss farther beyond each cutoff, while those for ORA-Muon remain closer to subsequent observations. Across 39 cutoffs from 24k to 100k in 2k increments, ORA-Muon reduces prediction RMSE (Appendix E.1) by a median 60% relative to SF-Muon.

Both methods fit the observed losses closely, but this does not ensure accurate extrapolation (Table 6). At 24k, SF-Muon has a higher R^2 and lower fitting RMSE, yet its prediction RMSE is nearly six times that of ORA-Muon. We therefore place greater emphasis on extrapolation accuracy.

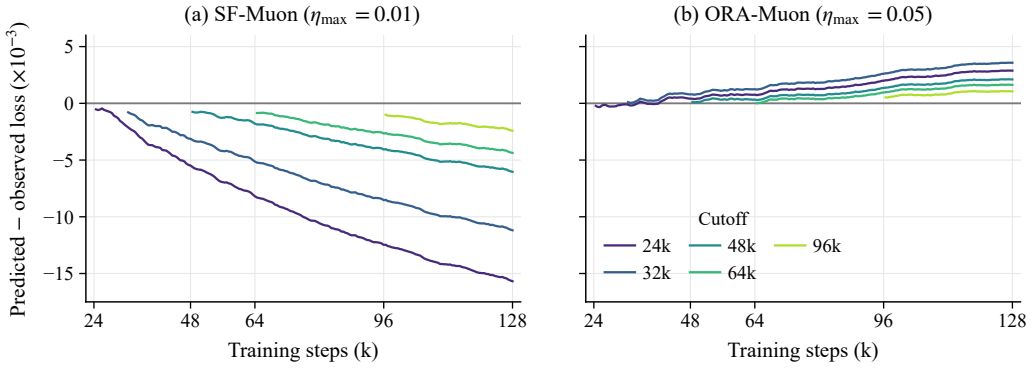

Figure 12: **Prediction errors for the fits in Figure 3.** Each curve shows predicted minus observed validation loss after its fitting cutoff. Both panels use the same scale. Negative values indicate overly optimistic predictions.

We also evaluate the starred ORA-Muon runs in Figure 2, which use reduced decay of the pending update. The fitting starts are 13k for 124M and 11.8k for 720M. Figure 13 shows their predictions against subsequent losses, and Table 5 reports errors at selected cutoffs. The starred 124M run has prediction errors similar to those of ORA with Z decay. At 720M, fits through 24k and 32k predict the 48k loss with errors equal to 5.6% and 4.6% of the subsequent loss decrease, respectively. These results also support accurate extrapolation at 720M.

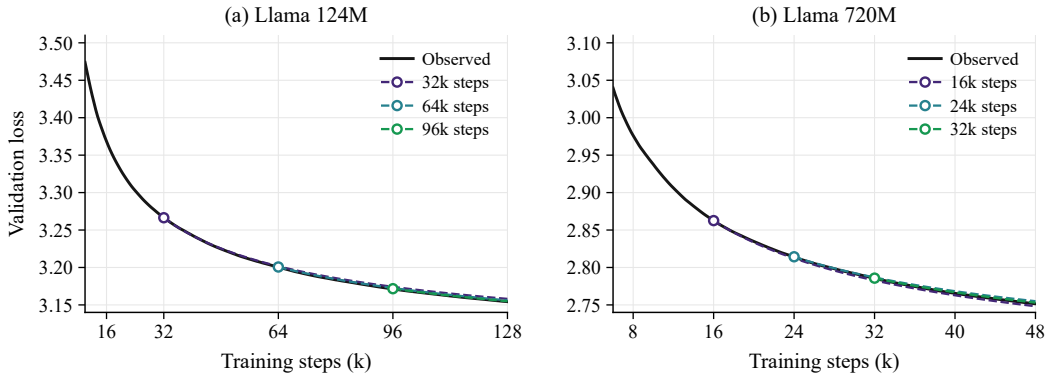

Figure 13: **Extrapolating the starred ORA-Muon runs in Figure 2.** Black curves show observed validation losses. Dashed curves show predictions beyond the cutoffs marked by circles, using fitting starts of 13k for 124M and 11.8k for 720M.

Table 5: **Prediction errors at selected cutoffs.** Steps are in thousands and errors in units of 10^{-3}. Endpoint error is predicted minus observed loss. RMSE is computed as in Appendix E.1. For Muon, only endpoint errors are reported. Unstarred SF-Muon and ORA-Muon use $\eta_{\max} = 0.01$ and $\eta_{\max} = 0.05$, respectively. The star denotes the runs in Figure 2.

Model	Method	Cutoff	Target	Endpoint error	RMSE
124M	Muon	48	128	−1.1	—
		64	128	0.2	—
	SF-Muon	32	128	−11.2	9.3
		64	128	−4.4	3.5
	ORA-Muon	32	128	3.6	2.9
		64	128	1.6	1.4
	ORA-Muon*	32	128	3.7	3.0
		64	128	1.8	1.5
720M	Muon	32	48	1.9	—
	ORA-Muon*	16	48	−3.3	2.8
		20	48	5.0	4.1
		24	48	3.5	3.0
		32	48	1.6	1.5

Table 6: **Fit quality and extrapolation accuracy.** SF and ORA denote the 124M runs in Figure 3. Cutoffs are in thousands of steps and RMSE values in units of 10^{-3}. Prediction RMSE is computed as in Appendix E.1, with a target of 128k steps.

Cutoff	Fitting R^2		Fitting RMSE		Prediction RMSE	
	SF	ORA	SF	ORA	SF	ORA
24	0.99998	0.99997	0.19	0.22	13.1	2.22
32	0.99995	0.99998	0.32	0.19	9.26	2.91
64	0.99988	0.99998	0.59	0.21	3.54	1.40

E.3 Sensitivity of Predictions and Fitted Parameters

We vary the fitting start to check whether the extrapolation results depend strongly on the 3k delay. Figure 14 compares delays from 2k to 6k using the same prediction targets at each cutoff. In the main 124M comparison, ORA retains smaller prediction errors than SF-Muon across these choices. The earliest 720M forecasts are more sensitive to the fitting start.

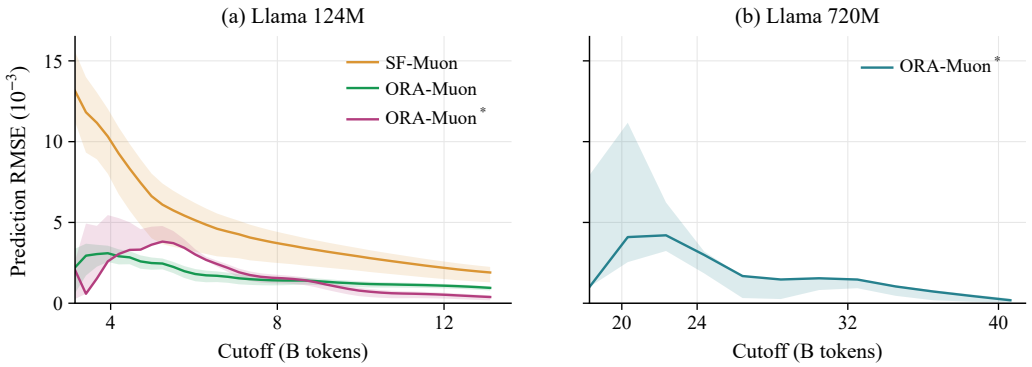

Figure 14: **Prediction error under different fitting starts.** Lines use the 3k rule in Appendix E.1. Shading gives the full range over delays of 2k, 3k, 4k, 5k, and 6k, not a confidence interval. Both panels use the same scale and begin where all five fits have at least 12 observations. SF-Muon uses $\eta_{\max} = 0.01$ and ORA-Muon uses $\eta_{\max} = 0.05$. Stars denote the runs in Figure 2.

Figure 15 complements Figure 4(b) by showing all three fitted parameters for the 124M runs. ORA-Muon ($\eta_{\max} = 0.05$) shows smaller parameter changes across cutoffs and fitting starts than SF-Muon ($\eta_{\max} = 0.01$). With shorter histories, the estimates remain sensitive to the fitting start, particularly for ORA-Muon ($\eta_{\max} = 0.01$). With longer histories, the four ORA runs have similar fitted b values, while A remains different. The shifts s are small relative to T later in training, consistent with the focus on A, b in Section 5.4.

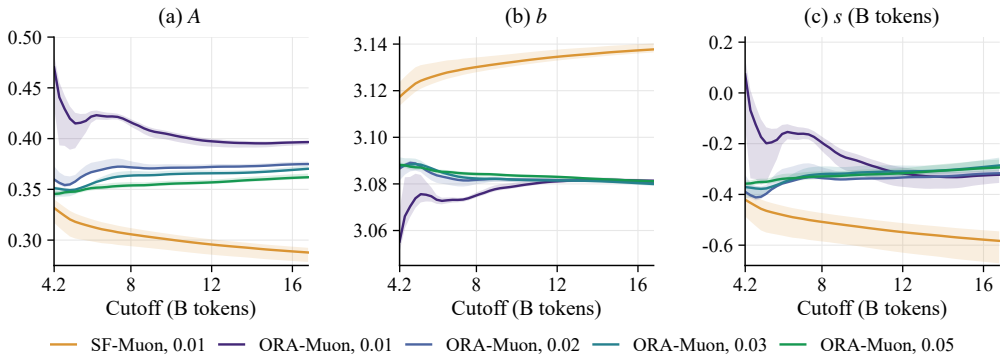

Figure 15: **Fitted parameters for the 124M runs in Figure 4(b).** Legend values denote $\eta_{\max}$. Lines use the 3k delay, and shading shows the range of fitted parameters over the 2k–6k delays. Each fit uses only observations up to its cutoff. Parameters use the billion-token convention in (14).

To quantify this similarity, we jointly fit the four ORA-Muon curves through 128k steps with a shared b. This increases pooled fitting RMSE by only 3.6% compared with separate fits. The increase remains small across fitting starts, supporting a common b as an approximate description of the four loss curves.

E.4 Predicting Token Savings in Longer Training

We use the fitted loss curves to predict whether the token savings in Figure 2 persist in longer training. The comparison uses the starred ORA-Muon runs and Muon with cosine decay. Table 7 reports their fits from the completed runs, with ORA fitting starts chosen as in Appendix E.1. For a Muon training length of T tokens, we find the number of tokens T_{ORA} at which the ORA fit reaches the same predicted loss. The predicted saving is $1 - T_{\mathrm{ORA}}/T$.

Table 7: **Fits using the completed runs.** Steps are in thousands. Parameters use the billion-token convention in (14).

Model	Method	Fitting data (k steps)	A	b	s
124M	Muon	16, 32, 48, 64, 128	0.5061	3.0472	0.2764
	ORA-Muon	13–128	0.4333	3.0480	−0.2667
720M	Muon	8, 16, 32, 48	1.1900	2.6184	2.1012
	ORA-Muon	11.8–48	1.0146	2.6050	−0.7991

Figure 16 extends the comparison to twice the longest completed Muon training length. At 124M, the predicted saving is about 24% at 256k steps, close to the observed 23% at 128k. At 720M, it is about 39% at 96k steps, compared with the observed 33% at 48k. These forecasts suggest that the token savings can persist over this extension, conditional on the fitted loss models continuing to describe longer training.

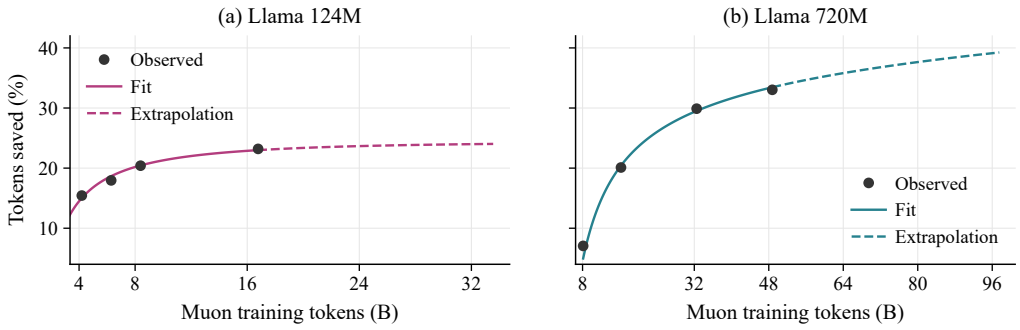

Figure 16: **Predicting token savings in longer training.** Dots show observed savings for the starred runs in Figure 2. Curves use the fits in Table 7 and become dashed after the longest completed Muon training length.

F Output Decay and Forgetting of Pending Updates

F.1 Forgetting Pending Updates

Writing the decay terms in S_t explicitly using (13), we can express iteration (2) as

$$\begin{aligned} X_{t+1} &= (1 - c_t\eta_t\lambda_X)X_t + c_t(1 - \eta_t\lambda_P)P_t - c_t\eta_t \operatorname{Opt}_t(Y_t), \\ P_{t+1} &= (1 - c_t)(1 - \eta_t\lambda_P)P_t - (1 - c_t)\eta_t\big(\operatorname{Opt}_t(Y_t) + \lambda_X X_t\big). \end{aligned} \tag{56}$$

Both decay terms are included in D_t when ORA determines the learning rate.

The choices of decay point in Section 5.2 are special cases of (56). Decay at Z gives $\lambda_P = \lambda_X$, while decay at Y gives $\lambda_P = (1 - \beta)\lambda_X$. Decay at $(Y + Z)/2$ gives $\lambda_P = (1 - \beta/2)\lambda_X$, the intermediate choice used by the starred curves in Figure 2.

For $0 \le \eta_t\lambda_P \le 1$, decay reduces the existing pending update to $(1 - \eta_t\lambda_P)P_t$. A fraction c_t of this quantity is applied to X_t, and the remaining fraction $1 - c_t$ stays in P_{t+1}. The portion removed by decay is neither applied to X_t nor retained in P_{t+1} for future steps. We therefore interpret λ_P as controlling forgetting of pending updates.

F.2 Forgetting and the Training Horizon

We examine whether the tradeoff in A, b shown in Figure 5 persists across fitting cutoffs and starts. The four 124M ORA-Muon runs differ only in λ_P, with $\lambda_X = 0.05$ and $\eta_{\max} = 0.05$. We use the fitting starts and prediction protocol from Appendix E.1. Table 8 reports the parameters fitted through 128k steps, together with prediction errors from separate fits using only data through 32k or 64k.

Table 8: **Fits and extrapolation for different λ_P.** Starts and cutoffs are in thousands of steps. Parameters use the billion-token convention in (14). The RMSE columns use fits through the indicated cutoffs and the prediction protocol in Appendix E.1, with a target of 128k and errors in units of 10^{-3}. The Muon reference uses the cosine endpoints in Table 7.

Method	λ_P	Start	Fit through 128k			Prediction RMSE	
			A	b	s	32k	64k
ORA-Muon	0.05	10.5	0.362	3.081	−0.294	2.9	1.4
	0.0375	11.5	0.390	3.066	−0.303	0.4	0.5
	0.02625	13	0.433	3.048	−0.267	3.0	1.5
	0.0125	16.5	0.524	3.021	−0.253	2.5	0.9
Muon	—	—	0.506	3.047	0.276	—	—

The small prediction errors in Table 8 support using the fitted curves to compare later performance. Smaller λ_P gives larger A and lower b in fits through 32k, 64k, and 128k steps, consistent with the observed crossings. This ordering also holds for the full-history fits with delays from 2k to 6k after sustained online control. Changes in b across these starts are below 0.002, compared with differences of at least 0.015 between adjacent decay choices. Unlike changes to $\eta_{\max}$ under ORA (Section 5.4), reducing λ_P lowers the fitted b while increasing A. Here b describes the fitted curves, rather than an observed limiting loss.

The intermediate choice $\lambda_P = 0.02625$ has b close to the Muon reference and a smaller A, while $\lambda_P = 0.0125$ lowers b further at the cost of a larger A. The latter run overtakes the former around 87k steps and remains ahead through 128k. The preferred λ_P therefore depends on the training horizon. Early in training, pending updates may become outdated more quickly as parameters change rapidly, favoring stronger forgetting. As training slows, weaker forgetting allows accumulated updates to be applied more fully to X.

The starred runs in Figure 2 use $\lambda_P = 0.02625$ as one possible compromise between early and late performance. Appendix E.4 examines their predicted token savings relative to Muon in longer training.

F.3 Decay of the Output Iterate

We compare decay of the output parameters with forgetting of pending updates using all 12 combinations of $\lambda_X \in \{0.025, 0.05, 0.1\}$ and $\lambda_P \in \{0.0125, 0.02625, 0.0375, 0.05\}$. All other settings match Appendix F.2. Fitting and prediction follow Appendix E.1.

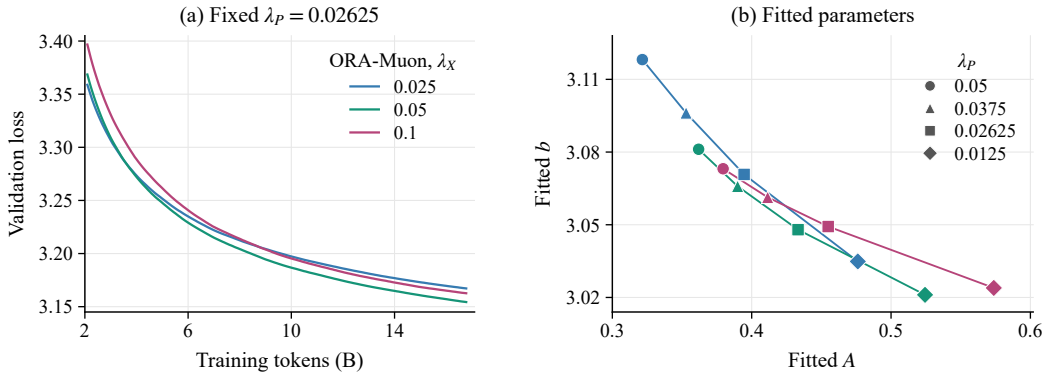

Figure 17: **Output decay and forgetting of pending updates.** Left: ORA-Muon loss at fixed $\lambda_P = 0.02625$. Right: fitted A, b through 128k for all 12 combinations. Colors denote λ_X in both panels, and marker shapes denote λ_P on the right. Lines connect points with the same λ_X. The squares correspond to the three runs in the left panel.

At fixed $\lambda_P = 0.02625$, $\lambda_X = 0.05$ gives lower late loss than either 0.025 or 0.1 (Figure 17(a)). The corresponding squares in Figure 17(b) show that increasing λ_X from 0.025 to 0.05 lowers b, whereas increasing it further to 0.1 mainly raises A, with little change in b.

Across the full grid, reducing λ_P increases A and lowers b at every tested λ_X, preserving the early–late tradeoff in Appendix F.2. Increasing λ_X from 0.025 to 0.05 lowers b throughout this grid, but a further increase to 0.1 can either lower or raise b, depending on λ_P. Larger λ_X also gives flatter curves in Figure 17(b): reducing λ_P yields a smaller reduction in b relative to the increase in A. Fits through 64k predict later losses through 128k with RMSE below 0.003 for all 12 runs.

The two coefficients thus have distinct effects on performance: reducing λ_P consistently trades early progress for lower b, while increasing λ_X can lower or raise b and alter this tradeoff. This supports treating decay of the output parameters and forgetting of pending updates separately. Adapting λ_P online to make forgetting of pending updates horizon-free is a natural direction for future research.