LONG-HORIZON META-OBJECTIVES INDUCE SPECTRAL FLATTENING

Anonymous authors
Paper under double-blind review

ABSTRACT

Muon accelerates matrix optimization by flattening the singular spectrum of gradient updates. But why should spectral flattening be a good optimization strategy, and can it arise without being explicitly prescribed? We approach these questions from a meta-optimization perspective, where update transformations are evaluated by their consequences for future learning rather than immediate descent alone. In controlled matrix-factorization problems, we find that extending the optimization horizon systematically shifts the preferred update toward flatter spectra. This shift reflects a long-term trade-off: spectrum-aware updates may sacrifice short-term progress along dominant directions while improving the optimization of modes that gradient descent leaves behind. We connect this behavior to the evolution of error modes and show how spectral flattening creates more balanced learning dynamics. Our results interpret Muon-like flattening as an emergent strategy for improving the geometry of subsequent optimization. This perspective suggests a path toward learning spectrum-aware optimizers directly from long-horizon objectives.

1 INTRODUCTION

Muon is an optimizer for matrix parameters. Given a gradient matrix $G = U\Sigma V^\top$, Muon replaces it by the flat-spectrum update $\tilde{G} = UV^\top$ (Jordan et al., 2024). The update keeps the singular directions of G, but equalizes every singular value. Empirically, Muon has become a competitive optimizer for language-model training. In Jordan et al.'s NanoGPT speedrun report, replacing AdamW with Muon improved training speed by about $1.35\times$, and subsequent record-setting runs continued to use Muon (Jordan et al., 2024). At LLM scale, with weight decay and update-RMS scaling, Muon attains roughly $2\times$ the compute efficiency of AdamW in scaling-law experiments (Kimi Team, 2025). A common account is spectral-norm steepest descent (Bernstein & Newhouse, 2024; Kovalev, 2025):

$$\Delta^\star \in \arg\min_{\|\Delta\|_2 \le \eta} \langle G, \Delta \rangle. \tag{1}$$

The closed-form solution is $\Delta^\star = -\eta UV^\top$, which is exactly the Muon update. Other explanations emphasize different facets of the same rule: Frank–Wolfe on a spectral-norm ball (Sfyraki & Wang, 2025), better learning of rare directions (Vasudeva et al., 2025), and a smaller curvature penalty than Adam (Wang et al., 2026). These accounts characterize what the Muon step is, but leave the central question open: why should this definition yield this empirical value? Euclidean steepest descent is greedy, and there is broad intuition for why greedy descent works. There is no comparable intuition for why steepest descent in the spectral norm should beat steepest descent in the Frobenius norm. Moreover, any advantage that accrues over many steps is, by construction, outside what a one-step descent argument can capture.

We therefore reverse the question. The usual route goes from Muon's definition to Muon's value:

$$\text{Muon definition} \longrightarrow \text{Muon value}. \tag{2}$$

We instead look for a meta-objective under which a Muon-like optimizer emerges without being prescribed, and explain its value from that objective:

$$\text{meta-objective} \longrightarrow \text{Muon value}. \tag{3}$$

Under such an objective, the optimizer is free to discover its own spectral shape.

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.

Meta-optimization, or learning to optimize, treats the optimizer itself as the object of learning. An inner loop runs ordinary parameter updates $\theta_{t+1} = F(\theta_t; \phi)$ under an optimizer with meta-parameters ϕ. An outer loop chooses ϕ to improve a multi-step objective, for example terminal loss after H steps:

$$\min_{\phi} \mathbb{E}\big[\mathcal{L}(\theta_H(\phi))\big]. \tag{4}$$

Learned optimizers of this kind plan across steps rather than maximize immediate descent (Andrychowicz et al., 2016; Li & Malik, 2017), and the horizon shapes what they learn: short-horizon meta-objectives systematically prefer overly small step sizes (Wu et al., 2018). We ask whether the horizon shapes the update spectrum in the same way.

Our contribution is a meta-optimization account of Muon's spectral geometry. We do not take flat updates as given; we ask which matrix update maps a long-horizon terminal objective prefers. An update map is a rule that turns a matrix gradient into a matrix update. We score each candidate map by terminal loss after H steps of deep linear matrix factorization, and call H the planning horizon. A prior is how much spectral structure we hard-wire into the family of candidate maps before learning. Across three update families with progressively weaker priors, at every depth and at learning rates up to one eighth of the stability edge of SGD, the conclusion is the same: longer planning horizons prefer flatter matrix updates.

- Strong prior (Section 4.1): a single power controls the singular spectrum, and the selected power falls steadily as the horizon grows.
- Weak prior (Section 4.2): with freely learned gains for each singular rank, the same trend appears, and the learned map mainly suppresses the leading singular directions.
- No spectral prior (Section 4.3): a per-parameter MLP optimizer of the kind used in current learned optimization, which never computes singular values, still produces flatter updates at longer horizons.

The two learned families are trained at a single width and keep improving on SGD at the wider matrices they never saw. Section 4.4 keeps only the singular values of this optimizer's updates, placed on the gradient's singular vectors, and shows that they alone lower terminal loss, although most of the gain comes from its singular directions; it also shows that each learned optimizer is specialized to its planning horizon. Section 4.5 leaves deep linear models: on CIFAR-10, ResNet-18 and TinyViT select more Newton–Schulz iterations at long horizons, and a six-layer GPT on FineWeb using the same spectral-power map selects the Muon direction at intermediate horizons, also when every power gets its own tuned learning rate. Section 5 explains why longer horizons prefer flatter updates. In a solvable decoupled quadratic model we derive the horizon-optimal spectrum in closed form (Theorem 1): its effective spectral power falls strictly from gradient descent at short horizons to the Muon direction at long horizons when residuals are balanced, because a long horizon equalizes the decay rates of all modes. Differentiating the terminal loss exactly through the rollout, we find that longer horizons give more credit to the tail singular directions of the first update; boosting only these directions of the first SGD update, at fixed Frobenius norm, lowers terminal loss more at long horizons than at short ones.

2 Related Work

Spectrum-aware matrix optimization. Muon approximates the polar factor of a matrix update with a Newton–Schulz iteration (Jordan et al., 2024). Spectral-norm steepest descent and non-Euclidean trust-region analyses recover that factor as a locally optimal step under a fixed geometry (Bernstein & Newhouse, 2024; Kovalev, 2025). Other work reads Muon with weight decay as stochastic Frank–Wolfe (Sfyraki & Wang, 2025), studies rare-direction benefits (Vasudeva et al., 2025), or compares second-order curvature penalties (Wang et al., 2026). Regime-dependent Schatten geometry instead predicts that the preferred norm depends on oracle budget, effective dimension, and noise (Pethick, 2026). One-step descent analysis attributes equalized singular values to a larger stable step size (Nguyen et al., 2026). Complementary convergence analysis proves that on matrix factorization a simplified Muon converges at a rate independent of the target's condition number, unlike gradient descent (Ma et al., 2026). A separate line studies the implicit bias of a fixed spectral optimizer: on separable classification, spectral descent and Muon favor spectral-norm max-margin

solutions (Fan et al., 2025). These works study the consequences of a prescribed spectral update; we ask which update a multi-step objective selects.

Learning to optimize. Learned optimizers train recurrent or feed-forward update rules by differentiating through optimization (Andrychowicz et al., 2016; Li & Malik, 2017; Chen et al., 2022). Current learned optimizers replace the recurrent cell with a per-parameter MLP over momentum and second-moment features (Metz et al., 2022; Thérien et al., 2024; Moudgil et al., 2025), and the most recent one adds Muon's Newton–Schulz orthogonalization by hand and finds that it helps (Moudgil et al., 2026). Truncation horizon biases that outer loop; short horizons, for instance, favor overly small step sizes (Wu et al., 2018). That literature treats the horizon mainly as a source of meta-training bias and reports losses or step sizes, not the spectral geometry of matrix updates.

Deep linear models. Deep linear networks keep nontrivial factorization dynamics while admitting exact gradients and tractable local mode analyses (Saxe et al., 2014; Arora et al., 2019). Under small initialization, gradient flow on these models acquires large target singular components before small ones, passing near saddles of increasing rank (Saxe et al., 2014; Jacot et al., 2021). This ordering describes how a fixed update rule allocates progress across modes; we ask which allocation a terminal objective prefers.

3 Setting and Meta-Objective

3.1 Deep linear factorization

For a target $A \in \mathbb{R}^{d\times d}$, an L-layer model predicts

$$\widehat{A} = W^{(L)}W^{(L-1)}\cdots W^{(1)}, \qquad \mathcal{L}(W) = \frac{1}{2}\|\widehat{A} - A\|_F^2, \tag{5}$$

where $W = (W^{(1)}, \ldots, W^{(L)})$ collects all layers. Targets and initial weights have i.i.d. Gaussian entries with variance $1/d$. With W_t the weights after t steps and $G_t^{(\ell)} = \nabla_{W^{(\ell)}}\mathcal{L}(W_t)$, an update map D_ϕ produces

$$W_{t+1}^{(\ell)} = W_t^{(\ell)} - \eta D_\phi(G_t^{(\ell)}). \tag{6}$$

We write

$$\Pi_G(X) := X\frac{\|G\|_F}{\|X\|_F} \tag{7}$$

for rescaling X to the Frobenius norm of G. Every candidate applies Π_G last, so $\|D_\phi(G)\|_F = \|G\|_F$ and global update energy is removed as a confounder.

A task $\tau = (A, W_0)$ is a random target together with random initial weights for all L layers; we unroll a fixed map for H steps and optimize

$$J_H(\phi) = \mathbb{E}_\tau[\mathcal{L}(W_H(\tau;\phi))]\,. \tag{8}$$

Model selection uses validation mean terminal loss; independent paired test tasks report $\Delta\mathcal{L} = \mathcal{L}_H^{\text{candidate}} - \mathcal{L}_H^{\text{SGD}}$, so negative is better. Tasks are deterministic, so "SGD" denotes the raw-gradient map $D(G) = G$ at the same learning rate.

3.2 Three update families

Canonical spectral power. For $G = U\,\mathrm{diag}(\sigma_i)V^\top$, let $r_i = \sigma_i/\sigma_{\max}$ and define

$$D_p(G) = \Pi_G\big(U\,\mathrm{diag}\big(\mathbf{1}_{\{r_i>0\}}r_i^p\big)\,V^\top\big)\,. \tag{9}$$

Thus $D_1(G) = G$ is SGD and $D_0(G) = \Pi_G(UV^\top)$, with $UV^\top$ restricted to the support of G, is the Muon direction. We scan $p \in [0, 1.25]$ at spacing 0.05.

Learned rank-wise gains. Each singular value is rescaled by a learned gain $e^{g_i(\theta)}$, where g_i interpolates linearly between $K = 8$ learned values $\theta \in \mathbb{R}^K$ placed at evenly spaced ranks. For rank $i = 1, \ldots, d$, let

$$t_i = \frac{i-1}{d-1}(K-1), \quad g_i(\theta) = (1-\alpha_i)\theta_{\lfloor t_i \rfloor + 1} + \alpha_i \theta_{\lceil t_i \rceil + 1}, \quad \alpha_i = t_i - \lfloor t_i \rfloor. \tag{10}$$

The update is

$$D_\theta(G) = \Pi_G\Big(U \operatorname{diag}\Big(\sigma_i e^{g_i(\theta)}\Big) V^\top\Big). \tag{11}$$

We clamp g_i to $[-4, 4]$ before exponentiation. Frobenius matching absorbs a global shift in θ, so we impose the gauge $\mathbf{1}^\top \theta = 0$ after every outer step, and $D_{\mathbf{0}}(G) = G$ is SGD.

Per-parameter MLP optimizer. Following current learned optimizers (Metz et al., 2022; Moudgil et al., 2025), a shared MLP reads, for every matrix entry, RMS-normalized features of the gradient, of three momenta, of Adam- and Adafactor-style second moments, and of the step count, and outputs a correction O_ϕ:

$$D_\phi(G_t) = \Pi_G(G_t / \operatorname{rms}(G_t) + O_\phi). \tag{12}$$

Its last layer starts at zero, so training starts from SGD. Entry-wise corrections can change both the singular values and the singular vectors of the update, so this optimizer is free to reshape the spectrum without ever computing it.

3.3 Spectral diagnostic

For singular values $s_i(X)$, entropy effective rank is

$$r_{\text{eff}}(X) = \exp\left(-\sum_i \pi_i \log \pi_i\right), \qquad \pi_i = \frac{s_i(X)}{\sum_j s_j(X)}. \tag{13}$$

It satisfies $1 \le r_{\text{eff}}(X) \le \operatorname{rank}(X)$, with the upper bound attained exactly by a flat spectrum. For the learned optimizer we compare $r_{\text{eff}}(\Delta)$ with $r_{\text{eff}}(G)$. The diagnostic never enters the training objective.

4 Experiments

Every deep-linear experiment runs on one grid of depths and learning rates that is fixed before any candidate is evaluated. For each depth $L \in \{2, 3, 4\}$, the six learning rates are spaced by factors of two and end at the largest rate at which SGD neither diverges nor ends above its initial loss at any horizon and width on separate pilot tasks (Appendix E). We use horizons $H \in \{1, 2, 5, 10, 20\}$ and widths $d \in \{16, 32, 64\}$; the two learned families train at width 16 with three restarts per configuration and are evaluated at all three widths. Selection uses 48 validation tasks, and all comparisons are paired on 96 held-out tasks per depth that every configuration shares.

We also follow one representative setting, $L = 3$ and $\eta = 0.02$, across horizons (Figure 2); Sections 4.4 and 5 analyze this reference setting further.

4.1 Strong prior: a single spectral power

We first scan the power family of Equation 9, which keeps the singular vectors of G and leaves one scalar of spectral shape free, over $p \in [0, 1.25]$ at spacing 0.05. At the three lowest rates of every ladder, $p^\star$ falls monotonically with the horizon at every width (Figure 1, top); in the reference setting it falls from about 0.8 at $H = 1$ to about 0.3 at $H = 20$ (Figure 2). Doubling η while halving H changes $p^\star$ by 0.03 on average, so the horizon acts through the effective time ηH of Section 5. The selected power improves terminal loss over SGD in every configuration.

4.2 Weak prior: learned rank-wise gains

To test whether the power law itself drives this trend, we replace it with the learned rank-wise gains of Equation 11 (budgets in Appendix A). At the three lowest rates, the effective-rank ratio rises

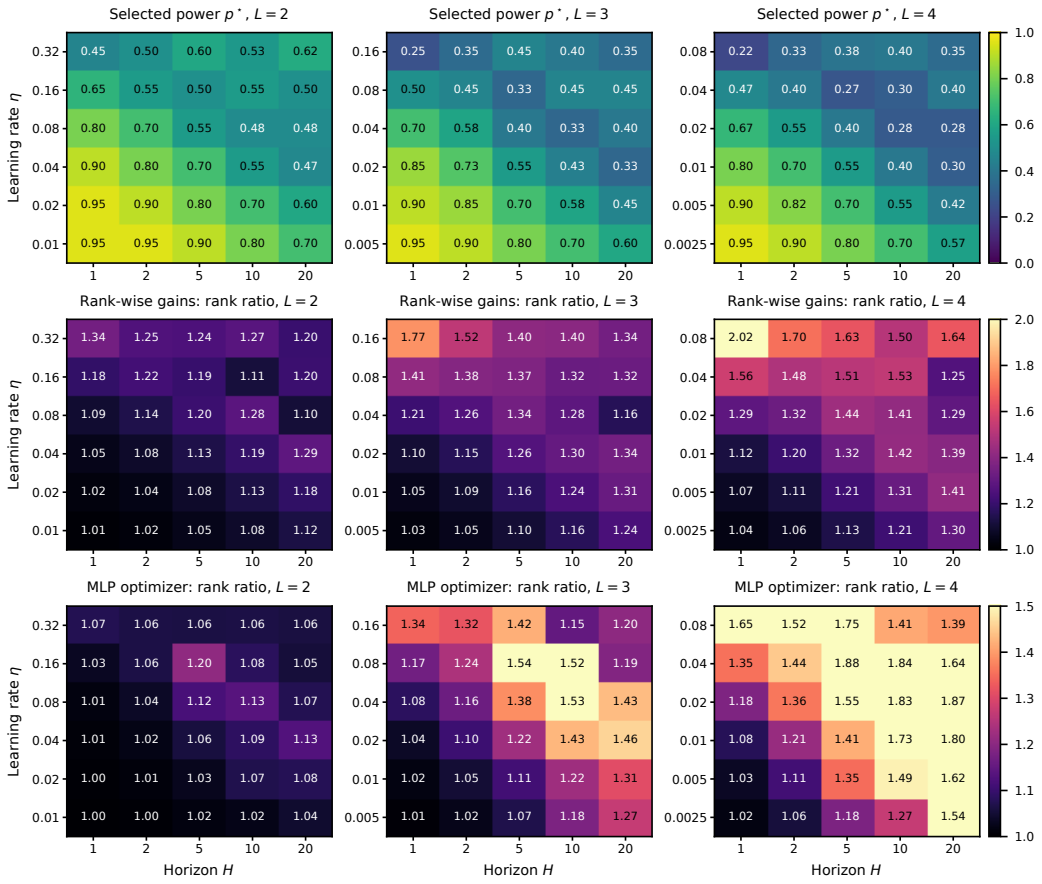


Figure 1: Depth and learning-rate grid. Each heatmap shows one depth over learning rate η (rows) and horizon H (columns), averaged over widths. Top: selected spectral power $p^\star$ (smaller is flatter). Middle and bottom: update effective rank relative to the gradient, $r_{\text{eff}}(\Delta)/r_{\text{eff}}(G)$, for the learned rank-wise gains and the MLP optimizer (larger is flatter). The top row of each heatmap is the stability edge of SGD.

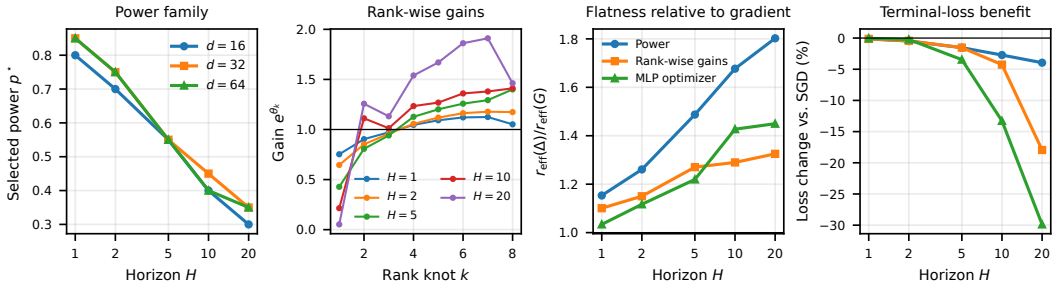


Figure 2: Reference setting $L = 3$, $\eta = 0.02$. From left: selected spectral power at each width; learned rank-wise gains; update effective rank relative to the gradient; and relative change of terminal loss against SGD, both averaged over widths.

monotonically with the horizon in 24 of 27 depth, rate, and width combinations, and the $H = 20$ update is flatter than the $H = 1$ update in all of them (Figure 1, middle). Instead of a uniform power law, the learned map clips the spectral head: in the reference setting, the gain on the leading singular

direction shrinks 14-fold between $H = 1$ and $H = 20$ (Figure 2, second panel). It is flatter than the gradient in 89 of 90 configurations and beats SGD in 89, including at widths 32 and 64, which it never saw in training.

4.3 No spectral prior: a learned per-parameter optimizer

Both families above still act in the singular basis of G. We finally remove that prior entirely and train the per-parameter MLP optimizer of Equation 12. Although the optimizer never computes a spectrum, at the three lowest rates its effective-rank ratio rises monotonically with the horizon in every combination (Figure 1, bottom). Its update is flatter than the gradient in every configuration, and it beats SGD in 80 of 90; all exceptions have $L \in \{3, 4\}$ and $H \geq 5$. In the reference setting it also beats Frobenius-matched heavy-ball momentum in every restart and width, and the trend survives dropping the Adafactor features (Appendix D).

Near the stability edge. The top of each ladder behaves differently. At the top rate, the power family and the rank-wise gains already select their flattest update at $H = 1$, and the MLP optimizer no longer flattens monotonically with the horizon. Flat updates then pay off through a larger stable step (Nguyen et al., 2026), and ηH is large enough to reach the saturation regime of Section 5. When SGD and each family choose their own rate from the grid, all of them choose the top rate at every depth, width, and horizon. There the power family beats SGD in 45 of 45 comparisons, the rank-wise gains in 42, and the MLP optimizer in 40 (Table 12). Rates above the edge of SGD were not tried, so these margins are conservative for the flatter families.

4.4 Is the learned flattening useful?

The MLP optimizer changes both the singular values and the singular vectors of the gradient, so its gains need not come from flattening. Writing $G = U_G \Sigma_G V_G^\top$ and the learned update as $\Delta = U_\Delta \Sigma_\Delta V_\Delta^\top$, we isolate the spectrum by keeping the learned singular values on the gradient's singular vectors, $\Delta_{\text{spec}} = \Pi_G(U_G \Sigma_\Delta V_G^\top)$, and, conversely, the learned singular vectors with the gradient's singular values, $\Delta_{\text{dir}} = \Pi_G(U_\Delta \Sigma_G V_\Delta^\top)$, at every inner step. The spectrum alone improves on SGD at every horizon and width (Figure 3, left). At $H = 20$ it recovers 10–11% of the full benefit, and the directions account for most of the rest (intervals in Table 13), so flattening is a real but secondary part of what the optimizer learns.

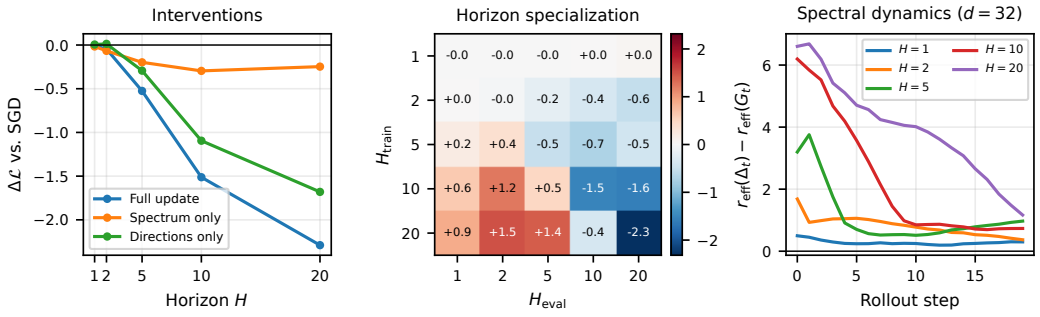


Figure 3: The learned optimizer's flattening, reference setting. Left: paired $\Delta\mathcal{L}$ versus SGD of the full learned update and of the spectrum-only and directions-only interventions, median across widths. Middle: median paired $\Delta\mathcal{L}$ versus SGD across widths for each training and evaluation horizon. Right: effective rank of the $H = 20$ optimizer's update minus that of the gradient along its own rollout.

Horizon specialization. The learned update is also specific to the horizon it was trained for. Each evaluation horizon is best served by the optimizer trained at that horizon (Figure 3, middle), and optimizers trained at $H \in \{10, 20\}$ are worse than SGD at every $H \leq 5$ and every width, so the long-horizon update is not a uniformly better step. Along its own rollout, the $H = 20$ optimizer is flattest in its first steps, where its effective rank exceeds the gradient's by 6.6 at $d = 32$, and the

excess shrinks to 1.2 by the last step (Figure 3, right); the first-update intervention of Section 5 targets this early phase.

4.5 BEYOND DEEP LINEAR MODELS

Deep linear factorization isolates the spectral question but lacks nonlinearity, real data, momentum, and the finite iterative map that Muon actually uses. Each experiment below selects a spectral setting on some runs, reports how the selection changes with the horizon, and compares it with the unflattened update on new runs (protocols in Appendix G).

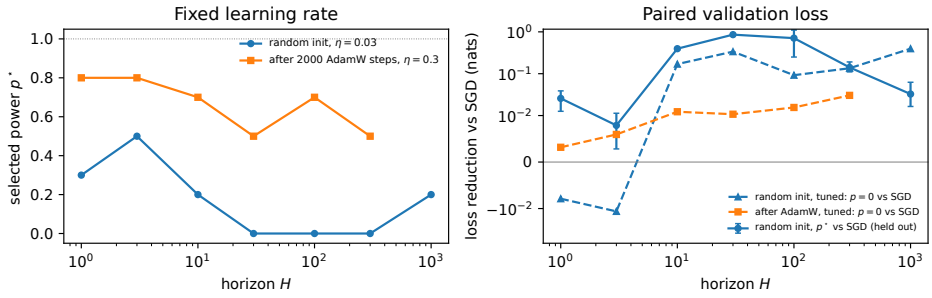


Figure 4: GPT on FineWeb. Left: selected power at a fixed learning rate. Right: validation-loss reduction relative to SGD (symmetric log scale). Solid: the selected power on held-out seeds, with bootstrap intervals; dashed: the Muon direction and SGD, each at its own best learning rate.

Finite Newton–Schulz and a nonlinear network. We add the missing ingredients one at a time, on grids built by the same rule (Figure 5). With Nesterov momentum and k Newton–Schulz iterations in deep factorization, the selected $k^\star$ rises monotonically with the horizon in every combination at the three lowest rates; wherever it selects $k^\star > 0$, it improves on momentum without Newton–Schulz in all but 5 of 205 combinations. On a two-layer ReLU MLP for Digits, the selected power falls with the horizon at the two lowest rates, from 0.95 to 0.6 at $\eta = 0.04$, and at the top two rates it is already smallest at $H = 1$, the same edge behavior as above. With momentum and Newton–Schulz, $k^\star$ rises from 0 to 1 with the horizon at the lowest rates and is 1 at every horizon from $\eta = 0.32$ upward.

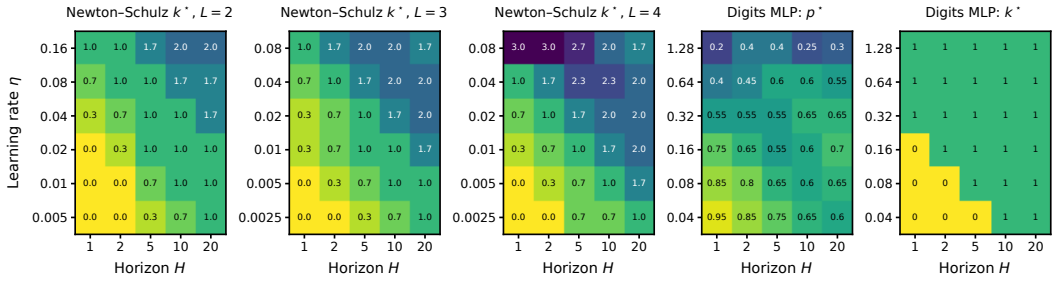


Figure 5: Grids beyond the Frobenius-matched deep-linear families. Selected Newton–Schulz depth $k^\star$ on Nesterov momentum in deep factorization (left three), and selected power $p^\star$ and Newton–Schulz depth on a Digits MLP (right two).

CIFAR-10. We train a ResNet-18 and a matrix-heavy TinyViT, applying Nesterov momentum and k Newton–Schulz iterations to hidden matrices and flattened convolution kernels. Two data splits select k and three new splits evaluate it. Both architectures select more iterations at $H = 400$ than at $H = 1$, though not always monotonically in between (Figure 9 in Appendix G), and at $H = 400$ the selected depth improves every evaluation run.

GPT on FineWeb. We apply the power map of Equation 9 to every attention and MLP matrix of a six-layer, width-384 GPT; embeddings and normalization gains share one AdamW rule. Three seeds select p and three new seeds evaluate it. From random initialization at $\eta = 0.03$, the selected power falls to the Muon direction at intermediate horizons and rises again at $H = 1000$, as large effective time predicts (Figure 4, left). It beats SGD on every held-out seed at every horizon, by up to 0.86 nats. With a learning rate tuned for each power, joint selection still picks a sharpened spectrum at $H \leq 3$ and the Muon direction at every $H \geq 10$. Starting instead from a checkpoint trained with AdamW gives the same direction with smaller effects (Appendix G, Table 14).

5 How Horizon Reweights Spectral Credit

The experiments establish an association between planning horizon and update-spectrum flatness. We now explain it. Differentiating the terminal objective through the rollout credits each update with its influence on the final loss after all later steps, so the question is how this credit is distributed across spectral directions, and how it moves as the horizon grows. We first solve a decoupled model exactly and then measure the prediction in a coupled one.

A solvable decoupled model. Let the loss be quadratic and diagonal in a fixed singular basis, with residuals $z_i = \langle X - X^\star, u_i v_i^\top \rangle$ and curvatures $\kappa_i > 0$, and let a spectral rule scale mode i by a gain q_i:

$$\mathcal{L}(X) = \tfrac{1}{2}\textstyle\sum_i \kappa_i z_i^2, \qquad \sigma_i = \kappa_i |z_i|, \qquad z_{i,t+1} = \rho_i z_{i,t}, \;\; \rho_i = 1 - \eta\kappa_i q_i, \qquad \delta_i = q_i \sigma_i. \tag{14}$$

For a stable mode, the terminal credit of q_i scales as ρ_i^{2H-1}, so the credit ratio of a persistent to a fast mode, $(\rho_s/\rho_f)^{2H-1}$, grows with H (Lemma 1 and Corollary 1, Appendix B). In the continuous-time limit $\eta \to 0$ at fixed $T = \eta H$, the Frobenius-matched problem of Equation 7 becomes

$$\min_q \; \mathcal{L}_T(q) = \tfrac{1}{2}\textstyle\sum_i \kappa_i z_i^2 e^{-2T\kappa_i q_i} \quad \text{s.t.} \quad \sum_i q_i^2 \sigma_i^2 = \sum_i \sigma_i^2. \tag{15}$$

For $\sigma_i > \sigma_j$ define the pairwise effective spectral power $p_{ij} = \log(\delta_i/\delta_j)/\log(\sigma_i/\sigma_j)$, which equals p for every pair exactly when $\delta \propto \sigma^p$.

Theorem 1 (Horizon-optimal spectrum)**.** *Assume all $z_i \neq 0$. For every $T > 0$, $\mathcal{L}_T$ has a unique minimizer under the Frobenius constraint,*

$$q_i^\star(T) = \frac{W\big(y(T)\,\kappa_i\big)}{2T\kappa_i}, \tag{16}$$

where W is the Lambert W function and $y(T)$ increases strictly from 0 to ∞. For any pair with $\kappa_i > \kappa_j$ and $\sigma_i > \sigma_j$, $p_{ij}(T)$ decreases strictly in T, with

$$\lim_{T\to 0} p_{ij}(T) = 1, \qquad \lim_{T\to\infty} p_{ij}(T) = \frac{\log(|z_i|/|z_j|)}{\log(\sigma_i/\sigma_j)}. \tag{17}$$

Consequently, the optimal first update is the gradient direction as $T \to 0$ and converges to the Muon direction $D_0(G)$ as $T \to \infty$ when residuals are equal; if $|z_i| \propto \kappa_i^\beta$ with $\beta \geq 0$, it converges to $D_{\beta/(1+\beta)}(G)$.

The proof (Appendix B) reduces the KKT condition to $q_i e^{2T\kappa_i q_i} = \text{const}$ and uses $\frac{\mathrm{d}}{\mathrm{d}y}\log\frac{W(ay)}{W(y)} < 0$ for $a > 1$. Since $W(ay)/W(y) \to 1$ as $y \to \infty$, long horizons equalize the contraction rates, and the optimal update follows the residual instead of the gradient:

$$\frac{\kappa_i q_i^\star}{\kappa_j q_j^\star} = \frac{W(y\kappa_i)}{W(y\kappa_j)} \xrightarrow[T\to\infty]{} 1, \qquad \delta_i^\star = q_i^\star \sigma_i \propto |z_i|. \tag{18}$$

All modes then decay at the same speed, as Ma et al. (2026) show for orthogonalized updates; with balanced residuals, $\delta^\star$ is flat, which is Muon.

The theorem also makes two predictions for deep factorization, and both hold (Appendix F). First, the horizon should matter only through the effective time ηH: at small learning rates, runs with the same ηH select nearly the same power. Second, $p^\star$ should saturate: with horizons up to 200, it levels off between 0.3 and 0.4 instead of reaching zero.

Spectral-coordinate measurement. The theorem is about dynamical modes, not gradient singular directions, so we measure the correspondence directly. We make the actual first-step displacement $S_0 = -\eta G_0$ an independent leaf and differentiate through the remaining rollout. For $G_0 = U\,\mathrm{diag}(\sigma_i)V^\top$, the loss reduction attributed to component i is $c_i(H) := \eta\sigma_i\langle \partial\mathcal{L}_H/\partial S_0, u_i v_i^\top\rangle$. In both deep factorization and the Digits MLP of Section 4.5, the credit of tail directions relative to head directions rises with the horizon, and within each width it rises as the selected power falls. Since both quantities vary with the horizon, we read this alignment as descriptive (Appendix C).

Causal first-update intervention. We shift part of the first update's spectral mass toward the tail at fixed Frobenius norm, choose the strength on calibration tasks, and run SGD afterwards. On held-out tasks, the tail boost lowers long-horizon terminal loss at every width and on every paired task (Figure 6, Table 1). Its gain at $H = 20$ is about twice its gain at $H = 1$, while boosting the head or suppressing the tail makes the loss worse.

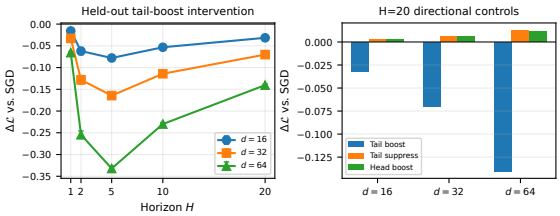


Figure 6: Held-out causal first-update intervention and directional controls.

6 Discussion

A meta-objective account of Muon-like geometry. Changing only the planning horizon changes the selected spectrum. Along the route meta-objective $\longrightarrow$ Muon value, flattening is selected for its effect on future loss instead of being prescribed. The same horizon dependence is the spectral counterpart of short-horizon bias, which for a scalar step size favors overly small rates (Wu et al., 2018).

Planning horizon is not cumulative training budget. Pethick (2026) predicts Schatten geometry from cumulative oracle budget, effective dimension, and noise; we vary the complementary axis of lookahead at a fixed training state.

Relation to one-step accounts. One-step accounts explain flattening through a larger stable learning rate (Nguyen et al., 2026). Near the stability edge, the power family and the rank-wise gains indeed select their flattest update already at $H = 1$. Below the edge the horizon is a second route: at a matched step size, the effective time ηH drives flattening.

Scope and limitations. The monotone trend holds at rates up to one eighth of the stability edge for depths 2 to 4; near the edge, and at large ηH, $p^\star$ can rise again (Appendices E and F). On CIFAR-10 we see the endpoint shift but not strict monotonicity, the modern-network evidence covers at most 1000 steps of CIFAR-10 and small-GPT training, and the theory assumes decoupled modes with gains tied across steps.

7 Conclusion

Across a single spectral power, learned rank-wise gains, and a learned per-parameter optimizer, longer planning horizons prefer flatter update spectra; the shift recurs on CIFAR-10 and in a GPT language model on FineWeb. A closed-form solution of a decoupled model shows why: the horizon-optimal spectrum moves strictly from the gradient toward the Muon direction as effective time grows, and exact terminal-loss gradients and a first-update intervention support this account. These results suggest learning spectrum-aware optimizers directly from long-horizon objectives instead of fixing the spectral identity in advance.

AI USE STATEMENT

We used generative AI coding assistants to write and debug experiment code, to launch and monitor experiment runs, to produce figures and summary statistics from raw results, and to draft and edit the text of this paper. Research questions, experimental protocols, and all claims were decided and checked by the authors; every reported number is generated by code from raw result files included in the supplementary material. We have reviewed all AI-assisted work and take responsibility for the final content of this paper, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 defines the tasks, update families, and meta-objective. Appendix A lists training budgets, Appendix E gives the learning-rate ladders and the significance rule of the grid, Appendix B proves Theorem 1 and the credit results and verifies them numerically, and Appendix G gives the selection and evaluation protocols for the experiments beyond deep linear models and for exact credit. All comparisons are paired and Frobenius-matched, selection follows a fixed rule, and evaluation uses runs disjoint from selection. The supplementary material contains code and raw results for every experiment, table, and figure, with all random seeds fixed.

REFERENCES

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, 2016.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, 2019.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. *arXiv preprint arXiv:2409.20325*, 2024.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 2022.

Chen Fan, Mark Schmidt, and Christos Thrampoulidis. Implicit bias of spectral descent and Muon on multiclass separable data. In *Advances in Neural Information Processing Systems*, 2025.

Arthur Jacot, François Ged, Berfin Simsek, Clément Hongler, and Franck Gabriel. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity. In *International Conference on Machine Learning*, 2021.

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

Kimi Team. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Dmitry Kovalev. Understanding gradient orthogonalization for deep learning via non-euclidean trust-region optimization. *arXiv preprint arXiv:2503.12645*, 2025.

Ke Li and Jitendra Malik. Learning to optimize neural nets. *arXiv preprint arXiv:1703.00441*, 2017.

Jianhao Ma, Yu Huang, Yuejie Chi, and Yuxin Chen. Preconditioning benefits of spectral orthogonalization in Muon. *arXiv preprint arXiv:2601.13474*, 2026.

Luke Metz, C. Daniel Freeman, James Harrison, Niru Maheswaranathan, and Jascha Sohl-Dickstein. Practical tradeoffs between memory, compute, and performance in learned optimizers. In *Conference on Lifelong Learning Agents*, 2022.

Abhinav Moudgil, Boris Knyazev, Guillaume Lajoie, and Eugene Belilovsky. Celo: Training versatile learned optimizers on a compute diet. *arXiv preprint arXiv:2501.12670*, 2025.

Abhinav Moudgil, Boris Knyazev, and Eugene Belilovsky. Celo2: Towards learned optimization free lunch. *arXiv preprint arXiv:2602.19142*, 2026.

Tien-Phat Nguyen, Truong Nguyen, Minh-Phuc Truong, Tuc Nguyen, James Bailey, and Trung Le. Spectral flattening is all Muon needs: How orthogonalization controls learning rate and convergence. *arXiv preprint arXiv:2605.13079*, 2026.

Thomas Pethick. When to use what Schatten-p norm in deep learning? *arXiv preprint arXiv:2606.15268*, 2026.

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.

Maria-Eleni Sfyraki and Jun-Kun Wang. Lions and muons: Optimization via stochastic frank–wolfe. *arXiv preprint arXiv:2506.04192*, 2025.

Benjamin Thérien, Charles-Étienne Joseph, Boris Knyazev, Edouard Oyallon, Irina Rish, and Eugene Belilovsky. μLO: Compute-efficient meta-generalization of learned optimizers. *arXiv preprint arXiv:2406.00153*, 2024.

Bhavya Vasudeva, Puneesh Deora, Yize Zhao, Vatsal Sharan, and Christos Thrampoulidis. How muon's spectral design benefits generalization: A study on imbalanced data. *arXiv preprint arXiv:2510.22980*, 2025.

Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Dirk Bergemann, and Zhuoran Yang. Why muon outperforms adam: A curvature perspective. *arXiv preprint arXiv:2606.04662*, 2026.

Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger Grosse. Understanding short-horizon bias in stochastic meta-optimization. In *International Conference on Learning Representations*, 2018.

A Experimental Budgets

Power family. We evaluate widths $\{16, 32, 64\}$ and horizons $\{1, 2, 5, 10, 20\}$. Each of the 26 powers uses 48 validation tasks; the selected power and SGD are evaluated on 96 independent paired tasks.

Rank response. For each horizon, the eight gains θ are trained at width 16 for 150 Adam iterations, with batch size 12, outer learning rate 0.04, and gradient clipping at 5. Evaluation uses 96 paired tasks at each width.

Per-parameter MLP optimizer. The MLP has two hidden layers of width 32. Its features are the gradient, momenta with decays 0.5, 0.9, and 0.99, these four signals divided by the square root of an Adam-style second moment (decay 0.999) and by the square root of its Adafactor row–column factorization, the inverse square root of that factorization, and $\tanh(t/\tau)$ for $\tau \in \{1, 3, 10, 30\}$; all moments are bias corrected, and every feature except time is divided by its RMS over the matrix. For each horizon and each of three restarts, it is trained for 600 outer Adam iterations with batch size 16, outer learning rate 0.01, gradient clipping at 1, and training width 16; we keep the iterate with the lowest training objective. We chose this outer setting once, from outer learning rates $\{0.001, 0.003, 0.01\}$ and $\{300, 600\}$ iterations, as the one with the lowest mean validation loss relative to SGD over the grid of Section 4. The coordinate-wise variant drops all Adafactor features. The momentum baseline is Π_G of the bias-corrected heavy-ball average with decay 0.9. Main evaluation uses 96 paired tasks per width; the interventions use 64 separate paired tasks.

Terminal-credit intervention. For each (d, H), intervention strength is selected on 48 calibration tasks and frozen before evaluation on 96 disjoint paired tasks. Only the first update spectrum changes; its layerwise Frobenius norm and all subsequent SGD updates are shared.

B Proofs for the Decoupled Mode Model

B.1 Terminal credit

Consider one mode with step-specific gain $q_{i,t}$, so $z_{i,t+1} = z_{i,t} - \eta q_{i,t}\kappa_i z_{i,t}$ and $\ell_{i,H} = \frac{1}{2}\kappa_i z_{i,H}^2$.

Lemma 1 (Terminal credit of a stable decoupled mode)**.** *At the shared baseline $q_{i,t} = q_i$ with $\rho_i = 1 - \eta\kappa_i q_i \in (0,1)$,*

$$\frac{\partial \ell_{i,H}}{\partial q_{i,t}} = -\eta\kappa_i^2 z_{i,0}^2 \rho_i^{2H-1}, \qquad \frac{\partial \ell_{i,H}}{\partial q_i} = -\eta H \kappa_i^2 z_{i,0}^2 \rho_i^{2H-1}, \tag{19}$$

for a gain used at one step and a gain tied across all H steps, respectively.

Corollary 1 (Horizon reweights relative credit toward persistent modes)**.** *For two stable modes with $\rho_s > \rho_f$, the ratio of their sensitivities is $R_H = (\rho_s/\rho_f)^{2H-1}$ for either perturbation, and it grows with H; the absolute sensitivity of either mode need not.*

Proof of Lemma 1. At the shared baseline, repeated substitution gives $z_{i,H} = \rho_i^H z_{i,0}$ and $\ell_{i,H} = \frac{1}{2}\kappa_i z_{i,0}^2 \rho_i^{2H}$. Perturbing the gain at a single step changes the terminal residual by

$$\frac{\partial z_{i,H}}{\partial q_{i,t}} = -\eta\kappa_i z_{i,0}\rho_i^{H-1}.$$

Multiplying by $\partial \ell_{i,H}/\partial z_{i,H} = \kappa_i z_{i,H}$ gives the first identity in Equation 19. When one parameter q_i is tied across all steps, the H equal stepwise sensitivities sum to the second. □

Proof of Corollary 1. Removing mode-specific constants leaves ρ_i^{2H-1} for a one-step perturbation and $H\rho_i^{2H-1}$ for a tied gain. The common factor H, when present, cancels between modes, so the ratio is $R_H = (\rho_s/\rho_f)^{2H-1}$ and $R_{H+1}/R_H = (\rho_s/\rho_f)^2 > 1$. □

B.2 Proof of Theorem 1

Existence and uniqueness. Write $S = \sum_i \sigma_i^2$ and $K = \{q : \sum_i q_i^2\sigma_i^2 \le S\}$. Each term $\frac{1}{2}\kappa_i z_i^2 e^{-2T\kappa_i q_i}$ is strictly convex and strictly decreasing in q_i, so $\mathcal{L}_T$ is strictly convex on the compact convex set K and has a unique minimizer there. If the minimizer were interior, increasing any coordinate would lower $\mathcal{L}_T$, so it lies on the boundary; hence it is also the unique minimizer on the constraint surface. Replacing a negative q_i by $|q_i|$ keeps the constraint and lowers the loss, so $q^\star \ge 0$, and in fact $q^\star > 0$ because the gradient of $\mathcal{L}_T$ at $q_i = 0$ is nonzero.

Closed form. Because K satisfies Slater's condition, the KKT conditions are necessary and sufficient: there is $\mu > 0$ with $T\kappa_i^2 z_i^2 e^{-2T\kappa_i q_i} = 2\mu q_i \sigma_i^2 = 2\mu q_i \kappa_i^2 z_i^2$. The mode-specific factor $\kappa_i^2 z_i^2$ cancels, giving $q_i e^{2T\kappa_i q_i} = c := T/(2\mu)$. With $x_i = 2T\kappa_i q_i$ this reads $x_i e^{x_i} = 2Tc\,\kappa_i$, so $x_i = W(y\kappa_i)$ with $y = 2Tc$, which is Equation 16. Substituting into the constraint,

$$\sum_i z_i^2\, W(y\kappa_i)^2 = 4T^2 S. \tag{20}$$

The left side is continuous and strictly increasing in $y \ge 0$, equals 0 at $y = 0$, and tends to ∞; hence $y(T)$ is unique, strictly increasing, and $y(T) \to 0$ as $T \to 0$ and $y(T) \to \infty$ as $T \to \infty$.

Monotonicity of the effective power. Since $\delta_i = q_i\sigma_i$,

$$p_{ij} = 1 + \frac{\log(q_i/q_j)}{\log(\sigma_i/\sigma_j)}, \qquad \log\frac{q_i}{q_j} = f(u) - \log a, \qquad f(u) := \log W(au) - \log W(u),$$

where $a = \kappa_i/\kappa_j > 1$ and $u = y\kappa_j$. Using $W'(u) = W(u)/(u(1+W(u)))$,

$$f'(u) = \frac{1}{u}\left(\frac{1}{1+W(au)} - \frac{1}{1+W(u)}\right) < 0,$$

because W is strictly increasing on $[0,\infty)$. As $u \to 0$, $W(u) \sim u$ gives $f \to \log a$; as $u \to \infty$, $W(u) \sim \log u$ gives $f \to 0$. Composing with the strictly increasing $u = y(T)\kappa_j$ and dividing by $\log(\sigma_i/\sigma_j) > 0$, $p_{ij}(T)$ decreases strictly from 1 to $1 - \log a/\log(\sigma_i/\sigma_j) = \log(|z_i|/|z_j|)/\log(\sigma_i/\sigma_j)$.

Spectral limits. As $T \to 0$, $W(y\kappa_i) \sim y\kappa_i$ gives $q_i^\star \to y/(2T)$, a common constant that the constraint fixes to 1: the gradient update. As $T \to \infty$, $W(y\kappa_i)/W(y\kappa_j) \to 1$, so $\kappa_i q_i^\star/(\kappa_j q_j^\star) \to 1$ and $\delta_i \propto q_i^\star \kappa_i |z_i| \to$ a common multiple of $|z_i|$. With equal residuals this is a uniform spectrum, i.e. $D_0(G)$ after Frobenius matching. If $|z_i| \propto \kappa_i^\beta$, then $\sigma_i \propto \kappa_i^{1+\beta}$ and $|z_i| \propto \sigma_i^{\beta/(1+\beta)}$, so the limit is $D_{\beta/(1+\beta)}(G)$. □

B.3 Numerical verification and discrete time

We draw 12 curvatures log-uniformly in $[0.05, 1]$ and solve the constrained problem both with Equation 16 and with a generic constrained optimizer. For $T \in [0.01, 1000]$ the two solutions agree to relative error below 10^{-8} for $T \geq 1$ and below 10^{-2} for $T < 1$, where the optimum is nearly isotropic and solver tolerance dominates. The least-squares power fitted to $(\log \sigma_i, \log \delta_i)$ falls from 1.00 to 0.00 with equal residuals and from 1.00 to 0.33 for $\beta = \frac{1}{2}$, matching $p_\infty = \frac{1}{3}$. The theorem is a continuous-time statement. Optimizing the discrete objective $\frac{1}{2}\sum_i \kappa_i z_i^2 (1 - \eta\kappa_i q_i)^{2H}$ with equal residuals gives the same picture: at $\eta = 0.02$ the fitted power falls from 0.99 at $H = 1$ to 0.05 at $H = 1000$, and at $\eta = 0.2$ it falls about ten times faster in H, since $T = \eta H$ controls the limit. The model assumes a fixed diagonal curvature in the gradient's singular basis and gains tied across steps; deep factorization violates both, so we use the theorem as a qualitative explanation of the horizon trend rather than a quantitative prediction of $p^\star$.

C Additional Mechanism Results

For exact spectral credit (Figure 7), mean tail-to-head credit rises from 1.1×10^{-4} at $H = 1$ to 3.4×10^{-3} at $H = 20$ in deep factorization, and from 0.0129 to 0.0256 in the nonlinear MLP. The outer rank quartiles define head and tail, and component credits are normalized by total absolute credit within each task and layer before aggregation. Within each width, tail-to-head credit rises as selected power falls; equivalently, head-to-tail credit and selected power have pooled descriptive Spearman correlation $\rho = 0.93$.

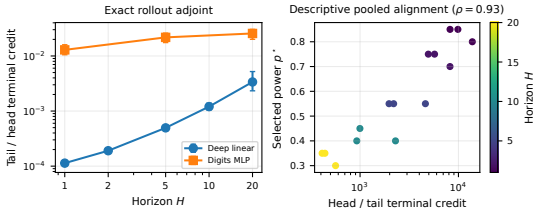


Figure 7: Exact terminal credit. Left: tail-to-head credit rises with horizon. Right: descriptive alignment with learned spectral rules.

Table 1: Held-out first-update intervention. Values are paired terminal-loss changes for calibration-selected tail boosting versus SGD; brackets are paired 95% bootstrap intervals.

Width	$H = 1$	$H = 20$	$H = 20$ win rate
16	-0.016	$-0.032\,[-0.034, -0.029]$	100%
32	-0.033	$-0.070\,[-0.073, -0.068]$	100%
64	-0.066	$-0.140\,[-0.143, -0.138]$	100%

At every width, the $H = 20$ improvement is larger than its $H = 1$ counterpart. Tail boosting also beats both head boosting and tail suppression at $H = 20$.

D REFERENCE-SETTING TABLES

The tables below report the reference setting $L = 3$, $\eta = 0.02$ from dedicated runs with their own task draws; the grid of Section 4 reproduces its selected powers within 0.05. At $H = 20$ every paired interval of these runs excludes zero, for the learned optimizer in each restart (Appendix G).

Table 2: Reference setting: validation-selected spectral power $p^\star(H)$.

d	H=1	H=2	H=5	H=10	H=20
16	0.80	0.70	0.55	0.40	0.30
32	0.85	0.75	0.55	0.45	0.35
64	0.85	0.75	0.55	0.40	0.35

Table 3: Power-family entropy effective rank.

d	H=1	H=2	H=5	H=10	H=20
16	8.66	9.49	10.90	12.48	13.60
32	16.70	18.23	21.82	23.88	26.12
64	33.87	37.01	44.25	50.50	52.65

Table 4: Power-family paired $\Delta\mathcal{L}$ versus SGD.

d	H=1	H=2	H=5	H=10	H=20
16	-0.016	-0.048	-0.113	-0.153	-0.157
32	-0.036	-0.106	-0.249	-0.323	-0.312
64	-0.068	-0.205	-0.482	-0.622	-0.601

Table 5: Learned rank-wise gains e^{θ_k}.

Knot	H=1	H=2	H=5	H=10	H=20
1	0.753	0.645	0.427	0.214	0.053
2	0.904	0.852	0.807	1.112	1.257
3	0.972	0.957	0.940	1.013	1.132
4	1.046	1.056	1.127	1.234	1.539
5	1.092	1.119	1.201	1.268	1.669
6	1.121	1.163	1.258	1.360	1.861
7	1.125	1.178	1.293	1.380	1.911
8	1.052	1.174	1.400	1.411	1.462

Table 6: Learned rank-wise gains: paired $\Delta\mathcal{L}$ versus SGD.

d	H=1	H=2	H=5	H=10	H=20
16	-0.016	-0.040	-0.114	-0.245	-0.659
32	-0.036	-0.100	-0.245	-0.475	-1.402
64	-0.073	-0.203	-0.482	-0.996	-2.939

Table 7: Per-parameter MLP optimizer: entropy effective rank.

d	H=1	H=2	H=5	H=10	H=20
16	7.58	8.27	9.25	10.53	10.74
32	15.23	16.29	17.89	20.79	21.39
64	30.34	32.24	35.15	41.48	42.47

Table 8: Per-parameter MLP optimizer: paired $\Delta\mathcal{L}$ versus SGD.

d	H=1	H=2	H=5	H=10	H=20
16	-0.008	-0.032	-0.295	-0.835	-1.250
32	-0.009	-0.049	-0.532	-1.506	-2.292
64	-0.007	-0.059	-0.968	-2.799	-4.262

Table 9: Per-parameter MLP optimizer: paired $\Delta\mathcal{L}$ versus Frobenius-matched momentum. Momentum itself is worse than SGD at every $H \geq 2$.

d	H=1	H=2	H=5	H=10	H=20
16	-0.008	-0.063	-0.475	-1.209	-1.806
32	-0.009	-0.111	-0.893	-2.272	-3.423
64	-0.007	-0.179	-1.704	-4.329	-6.545

Table 10: Coordinate-wise MLP variant without Adafactor features: entropy effective rank.

d	H=1	H=2	H=5	H=10	H=20
16	7.38	7.62	8.21	9.11	9.50
32	14.91	15.31	16.53	18.31	19.41
64	29.96	30.94	33.59	37.00	39.06

Table 11: Coordinate-wise MLP variant without Adafactor features: paired $\Delta\mathcal{L}$ versus SGD.

d	H=1	H=2	H=5	H=10	H=20
16	-0.001	-0.014	-0.232	-0.769	-1.124
32	-0.001	-0.023	-0.436	-1.466	-2.155
64	0.000	-0.033	-0.859	-2.781	-4.068

E Learning-Rate and Depth Grid

Ladders. For each depth, the ladder is a sequence of learning rates spaced by factors of two through 0.02. Its top is the largest such rate at which the baseline (SGD, or Nesterov momentum without Newton–Schulz for the finite map) neither diverges nor ends above its initial loss on any of 48 pilot tasks at any horizon and width; the ladder then extends five rates downward. Pilot tasks are disjoint from selection and test tasks, and the rule was fixed before any candidate was evaluated. The ladders are $[0.01, 0.32]$, $[0.005, 0.16]$, and $[0.0025, 0.08]$ at $L = 2, 3, 4$ for SGD, and $[0.005, 0.16]$, $[0.0025, 0.08]$, and $[0.0025, 0.08]$ for momentum; the Digits MLP uses $[0.04, 1.28]$ for both baselines.

Configurations. The fixed families select p or $k \in \{0, 1, 2, 3, 5\}$ by mean validation terminal loss. The learned families use the budgets of Appendix A at every (L, η, H) with three restarts; in the reference setting the MLP optimizer reuses the three restarts of Appendix D. A configuration counts as beating the baseline only if the paired 95% bootstrap interval (4000 resamples) excludes zero in every width and, for learned families, every restart. The rank-wise gains miss only at $(L, \eta, H) = (2, 0.32, 20)$. For the finite Newton–Schulz map, configurations that select $k^\star = 0$ coincide with the baseline, so Section 4.5 counts only combinations with $k^\star > 0$. The Digits grids and the finite Newton–Schulz grid use the protocols of Appendix G.

Rate chosen per method. For Table 12, SGD and each family choose their rate from the ladder by validation loss, separately for each width, horizon, and restart. All of them choose the top rate everywhere, so the table compares every family with SGD at the stability edge of SGD.

Table 12: Paired change of terminal loss at the top rate, relative to the baseline loss (SGD; momentum for Newton–Schulz) and averaged over widths. †: some width or restart has an interval that includes zero.

L	H	Power	Rank-wise gains	MLP optimizer	Newton–Schulz
2	1	-15.7%	-15.5%	-3.1%	-1.9%
2	2	-11.3%	-10.9%	-3.9%	-2.7%
2	5	-11.4%	-14.3%	-11.4%	-4.7%
2	10	-18.0%	-24.0%	-31.8%	$-4.8\%^\dagger$
2	20	-13.2%	$-15.2\%^\dagger$	-13.5%	-15.8%
3	1	-21.6%	-20.7%	-6.9%	-4.3%
3	2	-18.6%	-17.4%	-6.0%	-4.7%
3	5	-15.3%	-13.8%	$-10.4\%^\dagger$	-9.6%
3	10	-22.2%	-24.2%	-4.3%	-11.0%
3	20	-28.9%	-30.0%	-6.1%	-4.7%
4	1	-20.8%	-19.9%	-7.8%	-17.6%
4	2	-15.7%	-14.1%	-5.6%	-13.1%
4	5	-11.5%	-18.6%	$-4.5\%^\dagger$	-39.4%
4	10	-14.2%	-13.1%	-5.4%	-7.4%
4	20	-23.8%	-27.2%	$-4.1\%^\dagger$	-3.4%

F Effective Time and Long Horizons

Protocol. We use the deep factorization of Section 3 (depth 3, widths $\{16, 32, 64\}$) with tasks drawn independently of those behind Table 2, 48 validation and 96 disjoint paired test tasks, horizons $H \in \{1, 2, 5, 10, 20, 50, 100, 200\}$, and the 26 powers $p \in \{0, 0.05, \ldots, 1.25\}$. One rollout of 200 steps per (task set, p, η) records the terminal loss at every horizon. A candidate is ineligible if any validation task diverges (loss above 10^4 or non-finite); ties go to the smaller p. The learning rates are $\eta \in \{0.005, 0.01, 0.02, 0.03, 0.05, 0.08, 0.12, 0.2\}$.

Effective time and saturation. Figure 8 (left) plots $p^\star$ selected at each fixed η against ηH. For $\eta \leq 0.05$ the curves collapse: runs with equal ηH select the same power within 0.05–0.10; for instance, $(\eta, H) = (0.005, 20)$, $(0.01, 10)$, $(0.02, 5)$, and $(0.05, 2)$ all select 0.50–0.60. The collapsed curve falls from 0.95 to a plateau of 0.30–0.40 for $\eta H \in [0.4, 5]$ at every width. At $\eta = 0.02$ the new tasks reproduce Table 2 within 0.05, and $p^\star$ stays at 0.30–0.35 for $H = 50, 100, 200$. The right panel shows the corresponding closed-form curves of Theorem 1. Two regimes fall outside the theorem. At large η and $H = 1$, flat updates are selected because they tolerate a larger stable step ($p^\star = 0.10$ at $\eta = 0.2$). For $\eta H \gtrsim 5$, where terminal losses approach their floor, the width-averaged $p^\star$ rises again to 0.5–0.75; the theorem holds gains fixed while the gradient spectrum changes as modes converge, so it does not describe this late regime.

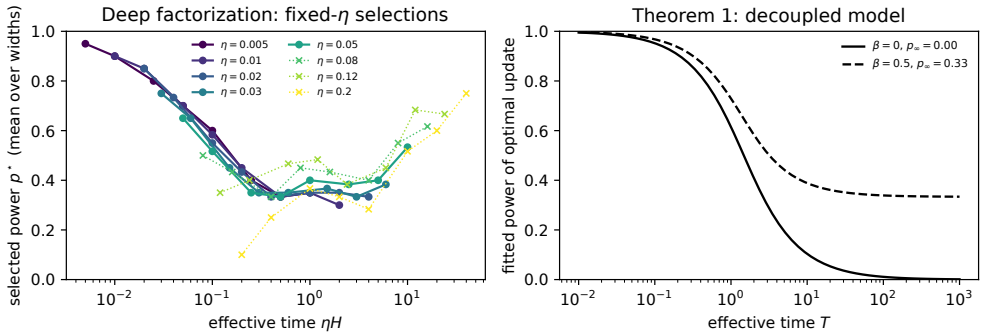


Figure 8: Left: fixed-η selected power in deep factorization against effective time ηH, averaged over widths; solid lines are $\eta \leq 0.05$. Right: fitted spectral power of the horizon-optimal update in the decoupled model of Theorem 1 for equal residuals ($\beta = 0$) and $|z_i| \propto \kappa_i^{1/2}$.

G Uncertainty and Protocols Beyond Deep Linear Models

Paired uncertainty. Table 13 reports paired task-bootstrap intervals at $H = 20$. For methods trained with three restarts, the point estimate is the median restart and the bracket is the envelope of the three per-restart 95% intervals. These envelopes summarize restart robustness; they are not hierarchical confidence intervals for the median-restart estimator. Every interval lies below zero. Median task-level win rate is 100% in every multi-restart configuration; the lowest single-run win rate in the table is 90.6%.

Table 13: Paired $\Delta\mathcal{L}$ versus SGD at $H = 20$; brackets give paired 95% bootstrap intervals, or their envelope across restarts.

Method	$d = 16$	$d = 32$	$d = 64$
Spectral power	$-0.157\,[-0.182, -0.133]$	$-0.312\,[-0.334, -0.289]$	$-0.601\,[-0.625, -0.577]$
Rank-wise gains	$-0.659\,[-0.731, -0.588]$	$-1.402\,[-1.478, -1.329]$	$-2.939\,[-3.015, -2.863]$
MLP optimizer	$-1.250\,[-1.323, -1.173]$	$-2.292\,[-2.366, -2.246]$	$-4.262\,[-4.329, -4.186]$
MLP spectrum only	$-0.140\,[-0.164, -0.115]$	$-0.247\,[-0.264, -0.230]$	$-0.409\,[-0.464, -0.398]$

All protocols below follow the same pattern: candidates share initialization and data order within a task, a fixed rule selects the candidate on selection tasks independently for each H, and disjoint evaluation tasks compare the frozen choice with a baseline in paired fashion.

Digits MLP. A width-64 two-layer ReLU MLP trains with full-batch cross-entropy on a fixed split of 1200 training and 500 held-out Digits examples, at each rate of the ladder in Appendix E. Both weight matrices use D_p with $p \in [0, 1.25]$ at spacing 0.05. Eight initializations select p by terminal training loss and sixteen new ones evaluate it against $p = 1$; intervals therefore reflect initialization, not data-split, variation.

Finite Newton–Schulz in deep factorization. The tasks, widths, and selection rule match the power family. Each matrix forms $M_t = 0.95M_{t-1} + 0.05G_t$ and $N_t = 0.05G_t + 0.95M_t$, normalizes N_t, applies k quintic iterations with the standard Muon coefficients $(3.4445, -4.7750, 2.0315)$, and matches the Frobenius norm of G_t. Validation tasks select $k \in \{0, 1, 2, 3, 5\}$ and new tasks evaluate it against $k = 0$.

Digits MLP with finite Newton–Schulz. The last grid combines the two above: the Digits MLP, split, and selection rule with the momentum and Newton–Schulz map; biases follow raw gradients.

CIFAR-10. ResNet-18 uses a CIFAR stem; TinyViT uses 4×4 patches, width 128, four transformer blocks, and four heads. Each task draws a random 45k/5k train/held-out split, initialization, and minibatch stream, with batch size 128, crop/flip augmentation, learning rate 0.05 (ResNet-18) or 0.03 (TinyViT), $\beta = 0.95$, and weight decay 5×10^{-4}. Hidden matrices and flattened convolution kernels use the finite Newton–Schulz map above; all other parameters use the Nesterov direction. Two splits with three streams each select k by terminal training loss; three new splits with four streams each evaluate it on the held-out 5k examples against $k = 0$. The official test set is not used. Intervals resample splits and then streams; with three top-level splits they are approximate. Over $H = (1, 5, 20, 100, 400)$, the selected paths are $(0, 2, 1, 2, 2)$ for ResNet-18 and $(0, 0, 2, 3, 3)$ for TinyViT. At $H = 400$, held-out loss changes by -0.205 $[-0.251, -0.174]$ and -0.167 $[-0.180, -0.157]$, and all twelve evaluation tasks improve for each architecture.

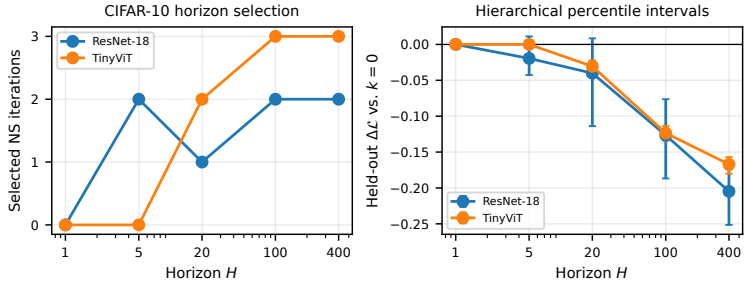


Figure 9: CIFAR-10. Left: selected Newton–Schulz depth. Right: paired held-out loss versus Nesterov momentum without Newton–Schulz ($k = 0$); intervals resample splits and streams hierarchically.

GPT on FineWeb. The model has six pre-norm decoder blocks of width 384 with six heads, RMSNorm, GELU MLPs, learned positions, context length 256, and a head tied to the GPT-2 token embedding (50,304 entries). Each step uses 32 sequences sampled from one FineWeb training shard; validation loss is measured on a fixed set of 256 sequences from the validation shard. All 24 attention and MLP matrices use D_p of Equation 9 on the raw gradient, without momentum, with a per-matrix Frobenius match. The exact map is computed from a batched fp64 eigendecomposition of the short-side Gram matrix, with singular values taken as row norms of $U^\top G$; it agrees with fp64 SVD to median relative error 10^{-7}. Embeddings and normalization gains use AdamW with learning rate 3×10^{-3} and $(\beta_1, \beta_2) = (0.9, 0.95)$ for every candidate. Within a seed, all candidates share the initialization and the batch stream. The fixed-rate grid uses $p \in \{0, 0.1, \ldots, 0.8, 1.0, 1.2\}$; three seeds select $p^\star$ by mean terminal validation loss and three disjoint seeds evaluate it against SGD.

The warm start first trains each seed for 2000 steps with AdamW on all parameters (matrix learning rate 10^{-3}), reaching validation loss near 5.65, and forks every candidate from that checkpoint with the AdamW state of the remaining parameters. The tuned-rate grids use $p \in \{0, 0.25, \ldots, 1.25\}$ with $\eta \in \{0.0003, 0.001, \ldots, 30\}$ from random initialization (four seeds), and $p \in \{0, 0.5, 1\}$ with $\eta \in \{0.1, 0.3, 1, 3\}$ after the warm start (three seeds); they report selection-set losses.

Table 14: GPT on FineWeb. Fixed-rate rows give $p^\star$ and the held-out paired change against SGD with a bootstrap interval. Tuned rows give the jointly selected (p, η) and the loss of the Muon direction minus that of SGD, each at its own best rate.

H	1	3	10	30	100	300	1000
Random initialization, $\eta = 0.03$							
$p^\star$	0.3	0.5	0.2	0	0	0	0.2
$\Delta\mathcal{L}$ vs. SGD	-0.026	-0.008	-0.40	-0.86	-0.71	-0.14	-0.033
Random initialization, tuned rates							
joint (p, η)	$(0.75, 0.01)$	$(0.5, 0.03)$	$(0, 10)$	$(0, 0.03)$	$(0, 0.03)$	$(0, 10)$	$(0, 10)$
Muon $-$ SGD	$+0.008$	$+0.012$	-0.17	-0.34	-0.09	-0.14	-0.40
After 2000 *AdamW steps,* $\eta = 0.3$							
$p^\star$	0.8	0.8	0.7	0.5	0.7	0.5	–
$\Delta\mathcal{L}$ vs. SGD	0.000	-0.004	-0.002	0.000	-0.001	-0.004	–
After 2000 *AdamW steps, tuned rates*							
Muon $-$ SGD	-0.003	-0.006	-0.012	-0.011	-0.016	-0.030	–

From random initialization (Table 14, top), every held-out seed improves at every horizon, and all intervals exclude zero (the widest is $[-1.17, -0.31]$ at $H = 100$). The warm fixed-rate gains are small, and three of the six intervals include zero. In the tuned grids the Muon direction prefers large rates ($\eta = 10$, interior to the grid, at four horizons from random initialization; $\eta = 3$, the largest value, after the warm start, where $\eta = 10$ is unstable in a pilot run). Tuned SGD instead prefers small rates, because larger SGD steps cause loss spikes early in training; at $H = 30$ its best rate is the smallest one tried, and lowering it from 10^{-3} to $3{\times}10^{-4}$ changes the gap only from -0.37 to -0.34.

Exact terminal credit. For each first-step matrix update, we retain S_0 as an independent differentiable leaf, run the remaining $H - 1$ SGD steps, and reverse-differentiate the summed terminal loss. We project the exact adjoint onto every initial-gradient singular dyad and normalize component credits by their total absolute value. Deep-linear results use 12 tasks at each $d \in \{16, 32, 64\}$ and $H \in \{1, 2, 5, 10, 20\}$; the nonlinear check uses four width-64 Digits MLP initializations, 512 examples, and $H \in \{1, 5, 20\}$. Task bootstrap intervals first average layers within each task. Alignment across the 15 (width, horizon) pairs is reported descriptively: the pairs are not treated as exchangeable because both variables share horizon, and the learned rank-wise gains repeat across widths.