MUON IS NOT THAT SPECIAL: RANDOM OR INVERTED SPECTRA WORK JUST AS WELL

Anonymous authors
Paper under double-blind review

ABSTRACT

The recent empirical success of the `Muon` optimizer has renewed interest in non-Euclidean optimization, typically justified by similarities with second-order methods, and linear minimization oracle (LMO) theory. In this paper, we challenge this geometric narrative through three contributions, demonstrating that precise geometric structure is not the key factor affecting optimization performance. First, we introduce `Freon`, a family of optimizers based on Schatten (quasi-)norms, powered by a novel, provably optimal QDWH-based iterative approximation. `Freon` naturally interpolates between `SGD` and `Muon`, while smoothly extrapolating into the quasi-norm regime. Empirically, the best-performing Schatten parameters for GPT-2 lie strictly within the quasi-norm regime, and thus *cannot* be represented by any unitarily invariant LMO. Second, noting that `Freon` performs well across a wide range of exponents, we introduce `Kaon`, an absurd optimizer that replaces singular values with random noise. Despite lacking any coherent geometric structure, `Kaon` matches `Muon`'s performance and retains classical convergence guarantees, proving that strict adherence to a precise geometry is practically irrelevant. Third, having shown that geometry is not the primary driver of performance, we demonstrate it is instead controlled by two local quantities: *alignment* and *descent potential*. Ultimately, each optimizer must tune its step size around these two quantities. While their dynamics are difficult to predict a priori, evaluating them within a stochastic random feature model yields a precise insight: `Muon` succeeds not by tracking an ideal global geometry, but by guaranteeing step-size optimality.

1 INTRODUCTION

First-order optimization algorithms have become increasingly central to modern machine learning, fueled by their routine use in training models with billions of parameters on trillions of tokens. While adaptive gradient methods like `AdamW` (Kingma & Ba, 2017; Duchi et al., 2011; Loshchilov & Hutter, 2019) have long served as the dominant baselines, recent years have seen a surge of interest in matrix-based and spectral optimizers (Gupta et al., 2018; Vyas et al.; Jordan et al., 2024).

Algorithms such as `Shampoo` and `Muon` have demonstrated remarkable success at scale (Liu et al., 2025; Team et al., 2025a;b; DeepSeek-AI, 2026), sparking a renaissance in the study of non-Euclidean descent methods (Bernstein & Newhouse) and an increasing suite of proposed modifications (Lau et al., 2026; Du & Su, 2026; Ahn et al., 2025; Riabinin et al.; Si et al., 2025; Amsel et al., 2026; Gong et al., 2026). Theoretically, this success is almost universally explained through its spectral structure: as steepest descent, trust-region or Frank–Wolfe/Linear Minimization Oracle (LMO) optimization under spectral-norm geometry (Bernstein & Newhouse; Bernstein, 2025; Li & Hong, 2025; Pethick et al., 2025; Kovalev, 2025; Riabinin et al.; Fan et al., 2025; Sfyraki & Wang, 2026; Shulgin et al., 2025), or through the preconditioning and stability effects of spectral orthogonalization itself (Su, 2025; Davis & Drusvyatskiy, 2026; Wang et al., 2025; Kim & hwan Oh, 2026; Nguyen et al., 2026). Analysis of simple models (Kim et al., 2026; Ma et al., 2026) commonly confirms this narrative, especially under distributional heavy tails (Yu et al., 2026; Wang et al., 2025).

However, recent benchmarking studies have begun to reveal cracks in this geometric facade, noting that advantages of spectral optimizers diminish under proper baseline tuning and are highly sensitive to batch size (Wen et al., 2025; Semenov et al., 2025). Concurrently, Su (2025) showed with an isotropic curvature model that optimal updates need only preserve the ordering of singular values,

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.

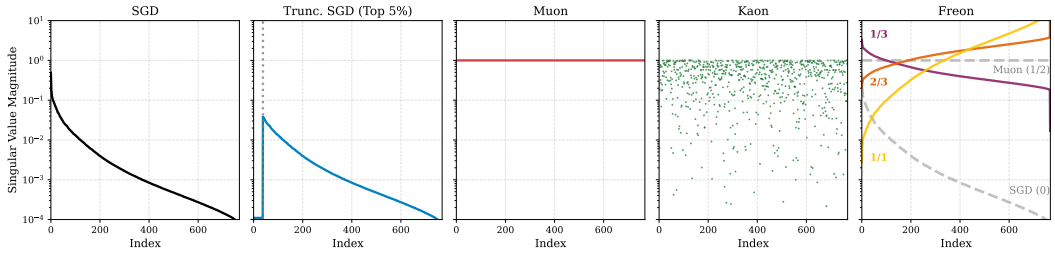


Figure 1: Transformations of the gradient singular value spectrum under various spectral optimizers for a GPT-2 layer. From left to right: **SGD** shows standard gradient spectrum. **Truncated SGD** (Appendix D.1) explicitly zeros out the largest 5% of singular values. Exact **Muon** maps all singular values uniformly to 1.0. **Kaon** sets the values randomly. Finally, **Freon** applies normalized updates of the form $(GG^\top)^{-c}G$, smoothly interpolating between standard SGD ($c = 0$), Muon ($c = 1/2$), and the pseudoinverse-like endpoint ($c = 1$). *Note the logarithmic scale on the y-axis.*

whereas `Muon`'s whitening imposes much stronger constraints. Motivated by this, we initially set off to determine an optimal spectral geometry[1]. Yet, much like analyses of `Muon` on quadratics (Gonon et al., 2026), we ultimately find no simple geometric recipe reliably improves performance. Instead, we are forced to ask: is adherence to a particular norm-ball structure necessary for Muon-like optimization performance, or is the LMO framework masking a more fundamental driver of optimization success?

In this work, we argue that while viewing optimization through the lens of non-Euclidean geometry provides an elegant theoretical framework, it fails to capture the core phenomena underlying the performance of spectral optimizers in deep learning. By systematically relaxing the assumptions of geometric preconditioning, we make the following specific contributions:

1. To explore the broader space of order-preserving updates, we introduce `Freon`, a family of Schatten (quasi-)norm updates of the form $(GG^\top)^{-c}G$. `Freon` naturally interpolates between SGD ($c = 0$) and Muon ($c = 1/2$). Through a symmetry analysis analogous to Yen et al. (2025), we argue that extrapolation into the $c \geq 1/2$ regime is of immense theoretical interest (Theorem 2.6), despite plunging into a quasi-norm regime that fundamentally breaks standard LMO theory (Theorem 2.1). To compute these updates stably, we develop an optimal (Theorems 2.7 and F.5) QDWH-based iteration utilizing rational approximations (Algorithm 3) similar to Nakatsukasa et al. (2010); Nakatsukasa & Freund (2016), extending the polar express theory of Amsel et al. (2026). Empirically, unlike simple quadratics, we find that optimal exponents for a GPT-2 model are in the quasi-norm region $c \in [1/2, 1]$ (Figures 6 and J.7), but vary significantly between batches. As this is strictly outside the range of any unitarily invariant norm, it demonstrates conclusively that standard LMO theory cannot be enough to explain performance gains.
2. Our empirical sweeps in Figures 2 and 4 revealed a profound conundrum: almost all `Freon` updates with $c > 0$ performed remarkably similarly, provided large singular values were sufficiently suppressed. This led us to hypothesize that the exact geometric structure of the update is largely irrelevant. To test this, we introduce `Kaon`, an optimizer that computes no LMO, targets no specific Schatten norm, and simply replaces the gradient's singular values with noise (Algorithm 2). Despite possessing zero coherent geometry, `Kaon` closely matches `Muon`'s performance at scale (Figures 2 and 3) and curiously retains classical convergence guarantees (Theorem 2.5). This shows that precise target spectra are largely irrelevant for optimization performance.
3. If LMOs are discarded, how can we mathematically understand the gap between all these updates? Extending the framework of Davis & Drusvyatskiy (2026), we decompose the exact local Taylor expansion into two structurally revealing quantities: **batch gradient alignment** (γ_k) and **directional descent potential** (Φ_k). While this is only mechanistic in quadratic models (Section 3.2), it cleanly exposes the fundamental tradeoff (Section 3.3): different optimizers implicitly choose how much alignment to sacrifice for increased descent potential, but actually realizing the benefit of a large Φ_k hinges on using step sizes that are tuned to exploit it.

[1]Throughout, *geometry* refers specifically to the weight-space norm geometry encoded by an LMO; all methods studied here are spectral, and we make no claim that spectral equivariance or loss-landscape curvature are irrelevant.

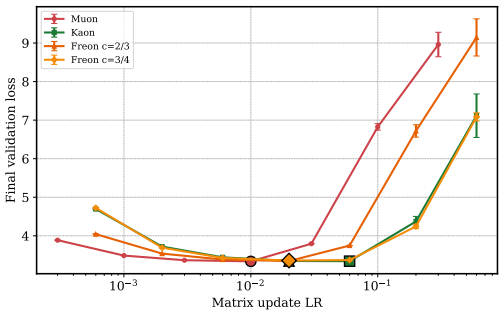


Figure 2: **NanoGPT learning rate sensitivity.** Final validation loss as the tuned learning rates are jointly scaled, averaged over three seeds with ± 2 std error bars. Black outlines mark the best learning rate per optimiser.

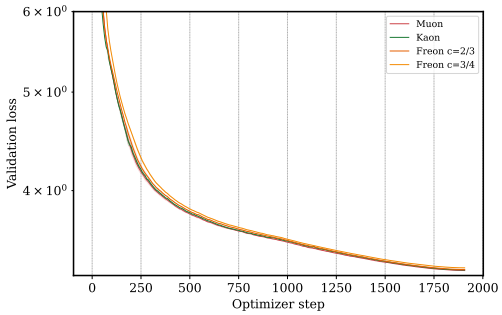


Figure 3: **NanoGPT validation curves.** Validation-loss curves for Muon, Kaon, and Freon at $c = 2/3$ and $c = 3/4$. Lines are averages over three seeds, with ± 2 std bands. The y-axis is clipped to $[3.3, 6]$ for legibility.

Algorithm 1: `Muon`

Set $X_0 = G/\|G\|_F$

for $t = 1, 2, \ldots, T$ **do**
  $A_t = X_{t-1}X_{t-1}^\top$
  $B_t = bA_t + cA_t^2$
  $X_t = aX_{t-1} + B_tX_{t-1}$
end

return X_T

Algorithm 2: `Kaon`

Set $X_0 = G/\|G\|_F$

for $t = 1, 2, \ldots, T$ **do**
  $A_t = X_{t-1}X_{t-1}^\top$
  $B_t = (I - A_t)^2$
  $X_t = 4.1 \cdot B_tX_{t-1}$
end

return $X_T/1.175$

Algorithm 3: `Freon`

Set $X_0 = G/\|G\|_F$
Set $A_0 = X_0X_0^\top$
for $t = 1, 2, \ldots, T$ **do**
  $B = R_t(A_{t-1})$
  $X_t = B^a X_{t-1}$
  $A_t = B^b A_{t-1}$
end
$\mu = \left(\frac{1}{n}\langle X_T, G\rangle\right)^{(a+2b)/(2a-2b)}$
return X_T/μ

Algorithms: Simplified pseudocode illustrating the main differences between the optimizers considered. Full algorithm for Freon is shown in Algorithm 4, utilizing block-QR for the rational map.

2 Background and Methods

We begin by considering the regularized steepest descent formulation of `Muon`. Given a norm $\|\cdot\|$ on an inner product space (spectral norm in the case of `Muon`), we define the Linear Minimization Oracle (LMO) and the corresponding dual norm as:

$$\mathrm{lmo}_{\|\cdot\|}(V) = \arg\max_{\|U\|\le 1} \langle U, V\rangle \qquad \|V\|_* = \max_{\|U\|=1} \langle U, V\rangle = \langle V, \mathrm{lmo}_{\|\cdot\|}(V)\rangle. \tag{1}$$

Then, for a differentiable loss function $f : \mathbb{R}^d \to \mathbb{R}$, with $G_k = \nabla f(X_k)$, regularized steepest descent updates by minimizing a first-order approximation of the loss under a quadratic penalty:

$$X_{k+1} = \arg\min_X \left\{ f(X_k) + \langle G_k, X - X_k\rangle + \frac{1}{2\eta}\|X - X_k\|^2 \right\} = X_k - \eta \|G_k\|_* \,\mathrm{lmo}_{\|\cdot\|}(G_k).$$

For example, for `Muon`, $\mathrm{lmo}_{\|\cdot\|}(W) = (WW^\top)^{-1/2}W$, where the inverse is taken in the sense of a Moore-Penrose pseudo-inverse. Unless otherwise specified, the proofs of theorems in this section can be found in Appendix E.

2.1 Limitations of the LMO Framework

For simplicity, in this section we restrict our attention to asingle layer, with a matrix domain $\mathbb{R}^{m\times n}$. We consider a differentiable loss function $f : \mathbb{R}^{m\times n} \to \mathbb{R}$. Let $r = \min(m, n)$ denote the maximal rank, and write the singular value decomposition $G_k = \nabla f(X_k) = U_k \,\mathrm{diag}(\sigma_k) V_k^\top$ with singular values $\sigma_k \in \mathbb{R}^r_{\ge 0}$. We define Preconditioned Spectral Descent updates as updates of the form $X_{k+1} = X_k - \alpha_k \langle G_k, D_k\rangle D_k$ with $D_k = U_k \,\mathrm{diag}(p_k(\sigma_k)) V_k^\top$ for some sequence of vector valued mappings $p_k : \mathbb{R}^r_{\ge 0} \to \mathbb{R}^r_{\ge 0}$ with $p_k(0) = 0$.

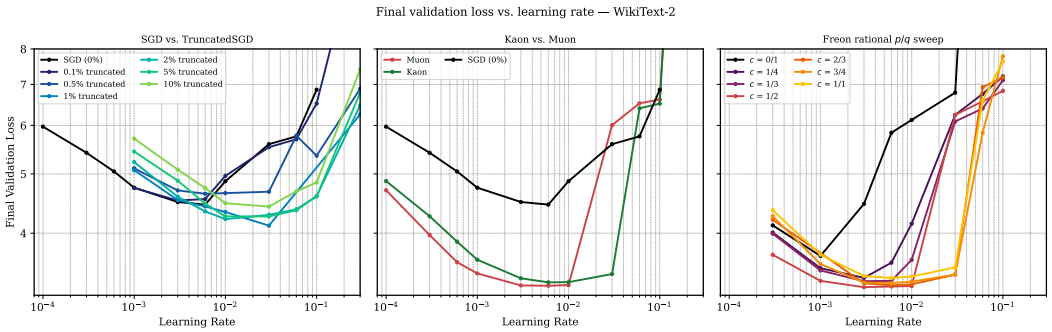


Figure 4: **Final validation loss versus learning rate for all optimizer families on WikiText-2 (118M tokens).** **(a)** `TruncatedSGD` improves over `SGD` but fails to close the gap to `Muon` – suppressing large singular values is necessary but not sufficient. **(b)** `Kaon` closely matches `Muon` despite replacing singular values with random noise. **(c)** `Freon` with $c \approx 2/3$ closely matches `Muon`; the optimal c lies strictly outside the range of any proper Schatten norm ($c > 1/2$).

Two simple members of this family are gradient descent (GD) and `Muon` (spectral descent). GD sets $p_k(S) = S/\|S\|_2$, while `Muon` sets $p_k(S) = 1$ (for positive S), practically achieved using Newton–Schulz iterations (Algorithm 1). However, not all preconditioned spectral descent updates are steepest descent updates: `Freon` for $c > 1/2$, `TruncatedSGD` and `Kaon` are not:

Theorem 2.1 (Preconditioned Spectral Descent vs. Regularized Steepest Descent)**.** *Fix the function f and the point X_k. A regularized steepest descent update w.r.t. a unitarily invariant matrix norm from X_k is a preconditioned spectral descent update. On the other hand, a preconditioned spectral descent update from X_k is a steepest descent update w.r.t. a unitarily invariant matrix norm if and only if p_k preserves the order $\leq$ of the entries of σ_k.*

Note 2.2. This theorem exposes a fundamental limitation of LMOs; maximizing descent cannot remove "bad" large or intermediate singular directions from a batched gradient.

But, as we show, LMOs are not necessary for convergence of preconditioned spectral descent.

Theorem 2.3 (Convergence of Preconditioned Spectral Descent)**.** *Let $\|\cdot\|$ be a unitarily invariant matrix norm, and let $\|\cdot\|_*$ be its dual norm. Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be continuously differentiable, bounded below by $f_{\min}$, with $L_{\mathcal{X}}$-Lipschitz continuous gradients with respect to $\|\cdot\|$.*

Consider preconditioned spectral descent updates as in Section 2.1, and assume there exist constants $m_k \geq 0$ and $M_k > 0$ satisfying the divergence condition $\sum_{k=0}^{\infty} (m_k/M_k)^2 = \infty$, such that for all $\sigma \in \mathbb{R}^r_{>0}$, the mapping p_k satisfies the following sufficient descent and boundedness conditions:

$$\langle \sigma, p_k(\sigma)\rangle \geq m_k\|\sigma\|_* \qquad \text{and} \qquad \|p_k(\sigma)\| \leq M_k.$$

Then, choosing $\alpha_k = \eta\frac{1}{L_{\mathcal{X}} M_k^2}$ for any $\eta \in (0, 2)$, the following convergence rate holds for $K \geq 1$:

$$\min_{0\leq k<K} \|G_k\|_*^2 \leq \frac{L_{\mathcal{X}}(f(X_0) - f_{\min})}{\eta(1-\frac{\eta}{2})\sum_{k=0}^{K-1}\left(\frac{m_k}{M_k}\right)^2} \qquad \text{and} \qquad \liminf_{k\to\infty} \|G_k\|_* = 0.$$

Note 2.4. The proof relies on standard convergence arguments, requiring that updates do not become too small, do not blow up and are mostly in the right direction. When p_k is derived from an $\mathrm{lmo}_{\|\cdot\|}$, these constants evaluate to $m_k = M_k = 1$. Crucially, this theorem can accommodate non-norm-descent methods like `Kaon` (Theorem 2.5) and `Freon` under regularization (Theorem E.2). While this does not yield a tight convergence rate, it does provide an essential sanity check.

2.2 KAON

Motivated by TruncatedSGD's gains over SGD, the broad insensitivity across `Freon`, and the observation that original `Muon` itself deliberately produces a broad, inexact polar spectrum rather than unit singular values (Jordan et al., 2024), we ask: does the target geometry matter at all? To test

this, we replaced the gradient's singular values with uniform random noise. This stochastic spectrum redistribution still performed surprisingly well, closely matching `Muon`.

To make this "absurd optimizer" efficient, we draw on ideas for pseudorandom number generation using chaotic iterations (Phatak & Rao, 1995). We use the generalized third-order logistic recurrence $x_{t+1} = \lambda x_t(1 - x_t^2)^2$, lifted to matrix iterations (Algorithm 2). Setting $\lambda = 4.1$ pushes the recurrence deep into the chaotic regime. We visualize an example in Figure 1, and illustrate the stationary distribution of these iterations in Appendix G.2, Figure G.1a. The phenomenon is not specific to the chaotic recurrence above. As a control, we also compute exact SVD updates from qualitatively different distributions. These perform similarly to the iterative `Kaon` (Appendix G.1).

We emphasize that `Kaon` is primarily a pedagogical construction. Its strong performance (Figure 4) is a direct counter-argument to LMO-based explanations of spectral optimizers: state-of-the-art performance does not require adherence to a precise target geometry, and can even be achieved with random spectral reshaping, while retaining standard convergence guarantees:

Theorem 2.5 (Convergence of Random Spectral Descent)**.** *Let the setting be identical to Theorem 2.3. Suppose at each iteration k, the stochastic mapping is defined specifically as $p_k(\sigma) = \frac{E_k}{\|E_k\|}$, where the entries of E_k are i.i.d. samples of a probability distribution on $\mathbb{R}_{>0}$ independent of k.*

If $\alpha_k = \frac{\eta}{L_{\mathcal{X}}}$ for any constant $\eta \in (0, 2)$, then almost surely, the sequence $\min_{0\le k<K} \|G_k\|_*^2$ achieves a convergence rate of $\mathcal{O}(1/K)$.

2.3 Freon

Following the setup of Section 2, we consider steepest descent in Schatten p-norms, defined as $\|X\|_p = (\sum \sigma_i\{X\}^p)^{1/p}$. By Hölder's inequality, the dual of the p-Schatten norm is the q-Schatten norm (Bhatia, 2013), where $\frac{1}{p} + \frac{1}{q} = 1$ for $p \in [1, \infty]$. As before, let the singular value decomposition of the gradient be $G = U\,\mathrm{diag}(\sigma)V^\top$. The steepest descent direction for $p \in [1, \infty]$ is:

$$\mathrm{lmo}_{\|\cdot\|_p}(G) = U\,\mathrm{diag}\left(\left(\frac{\sigma}{\|G\|_q}\right)^{q-1}\right)V^\top \quad \text{or equivalently} \quad D = \left(G_n G_n^\top\right)^{-c} G_n, \tag{2}$$

where we denote the dual-normalized gradient as $G_n = G/\|G\|_q$ and $c = 1 - q/2$. This update structure, parametrized by the dual exponent q, forms the basis of the general `Freon` optimizer.

A note on $p \in (0, 1)$: While the derived update rule extends analytically to $p \in (0, 1)$ (where $q < 0$), the p-Schatten ceases to be a proper norm in this regime, becoming a *quasi-norm* due to the violation of the triangle inequality. Strictly speaking, the dual of a p-quasi-norm for $p < 1$ is the ∞-norm (spectral norm). Consequently, exact steepest descent under a true p-quasi-norm constraint does not yield the continuous deformation above (Theorem 2.1 prevents that); rather, the LMO collapses, activating only the singular vector of the largest singular value. Therefore, the general `Freon` update with $q < 0$ does not correspond to a steepest descent under the quasi-norm either. It is better viewed as a preconditioned spectral descent (c.f. Section 2.1).

2.3.1 Norm Hyperparameter Transfer

If one were to execute the exact update derived above with the same step size across different values of p, the optimization would fail dramatically. This is because restricting the update to the standard unit p-ball ($\|\Delta W\|_p \le 1$) causes the maximum achievable step size in the spectral space to shrink at a rate dependent on p, as the rank $r = \min(m, n)$ increases. To anchor the updates to a consistent scale regardless of p, we must tie the updates to a single overarching invariant ball: the spectral ball. We achieve this by scaling the update constraint precisely by $r^{1/p}$, enforcing $\|\Delta W\|_p = r^{1/p}$. This means that for any $p > 0$, $r^{-1/p}\|\Delta W\|_p \le \|\Delta W\|_\infty$. By shifting to this mean Schatten norm, defined by $\|\cdot\|_{p,\mathtt{mean}} = \frac{1}{r^{1/p}}\|\cdot\|_p$, we geometrically inflate the p-Schatten ball, guaranteeing that the baseline maximum step size is always $\mathcal{O}(1)$ independent of p, as domain of the $\mathrm{lmo}_{\|\cdot\|}$: $\{\|D\|_\infty \le 1\} \subseteq \{\|D\|_{p,\mathtt{mean}} \le 1\}$ for all $p \ge 0$. We also note, that concurrently with our work, a similar normalization was introduced by Xu et al. (2026), restricted to the context of vector-induced row-wise and column-wise p, q norms, based on arguments of dimension independence.

This scaling is also consistent with the spectral feature-learning condition of Yang et al. (2024). Since $r^{-1/p}\|\Delta W\|_p \leq \|\Delta W\|_\infty$, enforcing $\|\Delta W\|_p = r^{1/p}$ prevents the spectral scale of the update from vanishing with rank. However, this applies throughout the normalized family and therefore does not by itself explain the empirical separation between the norm-representable and $c > 1/2$ regimes.

2.3.2 Particular case of $p = 0$ ($c = 1$)

The mean-norm scaling introduced in Section 2.3.1 does more than just stabilize hyperparameters; it mathematically enables us to explore the limiting case where $p \to 0$ (and $q \to 0$). When combined with the normalization from Section 2.3.1 the scaling in Equation (2) converges: $\|G\|_q / r^{1/q} \to \det(GG^\top)^{1/2n}$ as $q \to 0$. The beauty of the $p = 0$ limit extends far beyond this simple closed form; up to scaling, this specific update turns out to be invariant under layerwise symmetries.

Theorem 2.6 (Equivariance of `Freon` ($c = 1$). Informal, formal statement in Theorem E.4)**.** *Consider an arbitrary neural network ϕ_W with layered parameters $W = (W_1, \ldots, W_L)$. We define a symmetry of ϕ_W to be any transformation $W \mapsto g \cdot W$ such that $\phi_W = \phi_{g \cdot W}$. Then, under rank conditions of Theorem E.4, updates of the form $W_l - \eta(G_l G_l^\top)^{-1} G_l$ for each layer W_l are equivariant under arbitrary invertible linear layerwise symmetries of ϕ_W, i.e. $g \cdot W^k = [g \cdot W]^k$, where W^k represents the parameters after k updates. Therefore, the optimization trajectory of the neural network ϕ_W is invariant, i.e. $\phi_{[g \cdot W]^k} = \phi_{W^k}$.*

2.3.3 Efficient Computation: From Polar to Hölder Express

An exact SVD is sufficient for studying the family conceptually. To implement `Freon` practically, we require an efficient method to compute terms of the form $(GG^\top)^{-\frac{a}{b}} G$ from Equation (2), where $q = 2(1 - \frac{a}{b})$ and $\frac{a}{b} \in \mathbb{Q}$. This section realizes this using matrix-multiplication and QR primitives analogous to practical `Muon`, leading to significant and consistent speed-ups across scales (Table F.1).

On polynomial iterations: A natural approach is a polynomial scheme akin to (Jordan et al., 2024; Amsel et al., 2026). Extending the Polar Express theory (Theorem F.5), we find optimal polynomials P_k such that $xP_k(x^b) \to 1$ under composition (Appendix F.1, Remark F.6), which can be applied iteratively via updates of the form $O_{k+1} = P_k\left((O_k O_k^\top)^{b/2} (GG^\top)^{a-b/2}\right) O_k$. While theoretically convergent (Appendix F.1, Remark F.9), these iterations are highly unstable in practice for $a/b \neq 1/2$ (Appendix F.3). Requiring a computationally prohibitive $\sim \mathcal{O}(b \log \sigma_{\min})$ steps, they diverge rapidly, especially in lower precision. This instability is well-known for polynomial iterations outside of $a/b = 1/2$ (Higham, 2008). In line with Zhang et al. (2026), we similarly identify two primary causes: the explicit formation of poorly conditioned matrices (which introduces spurious negative eigenvalues) and the rapid accumulation of floating-point errors. A robust method must therefore avoid explicitly forming such matrices and converge in as few steps as possible.

On rational iterations: Generalized rational functions, rooted in Zolotarev's best rational approximants to the sign function (Achieser, 1992; Nakatsukasa & Freund, 2016), offer significantly greater stability. For instance, the QR-based Dynamically Weighted Halley (QDWH) method (Nakatsukasa et al., 2010; Nakatsukasa & Freund, 2016) uses these to approximate the polar map. For the polar case, Lau et al. (2026) similarly use QDWH and as a numerically robust alternative for computing the polar underlying `Muon`. Our goal here is different: we extend this QDWH-style rational machinery from the polar map to general inverse powers, establishing optimality of the resulting composed approximation. Drawing on this classical theory (Achieser, 1992; Cheney & Loeb, 1964) and use in matrix computations (Nakatsukasa & Freund, 2016; Gawlik & Nakatsukasa, 2021), we use the Remez algorithm (Filip et al., 2018) to iteratively approximate the sign function using rational functions of the form $xR(x^{2b}) = x\frac{a+bx^{2b}}{1+cx^{2b}}$ akin to Amsel et al. (2026), establishing optimality of this approach:

Theorem 2.7 (Proof in Appendix F.1)**.** *Let $a/b \in \mathbb{Q}$, $l > 0$ and suppose G satisfies $\sigma_{\min}(G) \geq l$. Moreover, set $\mathcal{G} := (GG^\top)^{-\frac{a}{b}} G$, $\mathcal{G}^\dagger := G^\top (GG^\top)^{\frac{a}{b}-1}$ and $\mathfrak{R}_{n,d}\left(\frac{b}{2}, \frac{b}{2}\right)$ as in equation 15 with $n, d \in \mathbb{N}$ such that $\frac{n+1}{d+1} \in \mathbb{N}$. Then, there exists $\{R_t\}_{t \in \mathbb{N}} \subset \mathfrak{R}_{n,d}\left(\frac{b}{2}, \frac{b}{2}\right)$ such that, setting $O_T := \widetilde{R}_T \circ \cdots \circ \widetilde{R}_1$, with $\widetilde{R}_t(X) = R_t(X\mathcal{G}^\dagger)\mathcal{G}$, $O_T(G)$ optimally approximates $\mathcal{G}$ for every $T \in \mathbb{N}$. Moreover, there exists $C = C(a, b, l) > 0$ such that the following doubly exponential convergence rate holds:*

$$\|\mathcal{G} - O_T(G)\|_2 \leq C|1 - l^b|^{(n+d+1)^T}.$$

In practice, O_T is computed using the coupled system presented in Algorithm 3, with the complete algorithm summarized in Algorithm 4. Unlike prior work (Gawlik & Nakatsukasa, 2021), wherein the considered class of rational functions does not contain the best approximant for $p \geq 3$, our recursion guarantees the existence of an optimal approximant at every step. Moreover, the condition on n, d permits choosing higher-degree polynomials in the numerator without increasing the denominator's degree, and thus allowing block-QR tricks.

3 Theoretical and Empirical Observations

Section 2 asks a basic legitimacy question: can spectral updates that are not LMOs still admit standard descent or convergence guarantees? The answer is yes. Those guarantees, however, are deliberately too coarse to explain the empirical differences between optimizers. We therefore turn to a more suitable question – what does? We build upon Davis & Drusvyatskiy (2026), which provides a foundational theoretical lens for understanding spectral descent. To properly diagnose the gap between optimizers, we must abandon global bounds (as in Cohen et al. (2021); Islamov et al. (2026)) and examine the exact local expansion of the loss.

3.1 The Two Quantities

Let $f : \mathbb{R}^d \to \mathbb{R}$ be twice differentiable, and suppose we update $X_{k+1} = X_k - \alpha_k \Delta_k$, with $\Delta_k = \langle \widetilde{G}_k, D_k \rangle D_k$, $\widetilde{G}_k$ is a stochastic gradient of f, and D_k is the direction given by the optimizer, defined for example through preconditioned spectral descent (Section 2.1). By Taylor's theorem, $f(X_{k+1}) = f(X_k) - \alpha_k \langle G_k, \Delta_k \rangle + \frac{\alpha_k^2}{2} \langle \Delta_k, \nabla^2 f(Z_k)[\Delta_k] \rangle$, with Z_k on the line segment between X_k and X_{k+1}. Plugging in our update $\Delta_k = \langle \widetilde{G}_k, D_k \rangle D_k$, and factoring inherently isolates two fundamental quantities that govern the trajectory of the optimizer:

$$\underbrace{\gamma_k := \frac{\langle G_k, D_k \rangle}{\langle \widetilde{G}_k, D_k \rangle}}_{\text{Batch Gradient Alignment}}, \qquad \underbrace{\Phi_k := \frac{\langle \widetilde{G}_k, D_k \rangle^2}{\langle D_k, \nabla^2 f(Z_k)[D_k] \rangle}}_{\text{Local Directional Descent Potential}}. \tag{3}$$

Define in addition $\lambda_k = \langle D_k, \nabla^2 f(Z_k)[D_k] \rangle$, a quantity which will only appear through multiplication with the learning rate α_k. Substituting these into the above yields the exact descent condition:

$$f(X_{k+1}) - f(X_k) = -\Phi_k \left(\gamma_k - \frac{[\alpha_k \lambda_k]}{2} \right) [\alpha_k \lambda_k]. \tag{4}$$

In particular, for quadratic objectives λ_k is constant, so Equation (4) is quadratic in α_k minimised at $\alpha_k^* = \frac{\gamma_k}{\lambda_k}$. More generally, the same expression gives the local quadratic approximation. At this optimum, Equation (4) depends on the two quantities above, with $\Delta f = -\frac{1}{2} \Phi_k \gamma_k^2$. There are a number of observations we should make about this:

1. The derivation of Equation (4) does not utilize anything apart from smoothness. Crucially, neither quantity invokes an LMO or places any restriction on how D_k is constructed. While we cannot directly prove convergence without further assumptions (like Theorem 2.3), it still remains useful for explaining multiple phenomena, as explained below.
2. This directly recovers the formula of Davis & Drusvyatskiy (2026) as a special case, by bounding λ_k with a Lipschitz constant, setting $\widetilde{G}_k = G_k$ and $D_k = \mathrm{lmo}_{\|\cdot\|}(G_k)$, gives $\gamma_k = 1$ with bound $-\|G_k\|_*^2 / 2L_{\|\cdot\|}$. But, compared with (Davis & Drusvyatskiy, 2026), this includes three important generalizations: a non-fixed step-size, non-LMO-based updates, and stochasticity, and momentum through the choice of D_k.
3. While for any fixed norm, the D_k maximizing $\Phi_k \lambda_k$ is $\mathrm{lmo}_{\|\cdot\|}(\widetilde{G}_k)$, this may not be the direction that maximizes γ_k or minimizes λ_k.
4. Because the intermediate point Z_k is unknown a priori, λ_k cannot be calculated during practical training. Its exact analytical use requires quadratic models (as explored in Section 3.2).

Ultimately, optimization performance is driven by the interplay of these quantities (γ_k, Φ_k, and the learning rate $\alpha_k \lambda_k$), rather than the strict form of the update direction itself. In the remainder of this paper, we analyze these quantities exactly in the RF model and empirically for a GPT2 training run.

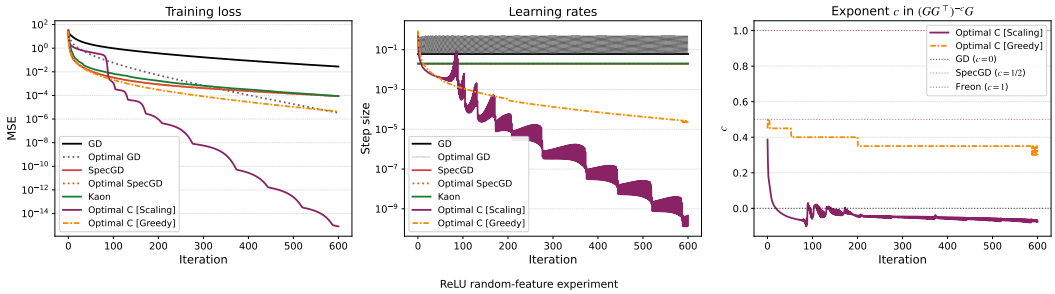


Figure 5: **Random-feature regression with ReLU activations.** We consider the problem in Equation (5), using the setup of Davis & Drusvyatskiy (2026). **Left:** training loss for GD, SpecGD, Kaon, their optimal-step variants, and two adaptive-exponent methods. **Centre:** effective step sizes; the optimal-GD step size highly oscillates, while optimal-specGD step size is constant, being equal to specGD. **Right:** the exponent c in the update direction $(GG^\top)^{-c}G$ as tracked by the two Optimal C methods during training. Further details can be found in Appendix I.1.

3.2 Exact Asymptotics in a Random Feature Model

We take the well-specified random feature regression model as in Davis & Drusvyatskiy (2026) with W the weights and $A \in \mathbb{R}^{n\times b}$ batched post-activations of the previous layer:

$$\min_{W\in\mathbb{R}^{o\times n}} f(W) = \frac{1}{2b}\|WA - Y'A\|_F^2. \tag{5}$$

We will consider the proportional asymptotic limit ($n, b \to \infty$ with $n/b \to \delta$) and assume the batched post-activations are mean-zero Gaussians, $a_i \sim \mathcal{N}(0, C)$, with a well-defined limit for C. Let $G = Y' - W$ be the oracle gradient. Using standard random matrix theory, one can show almost sure convergence of both Φ and γ, and we give full details of this in Appendix H. We have the following results.

Proposition 3.1. *If $C = I$ then $\Phi_{\mathrm{SGD}} \geq \Phi_{\mathrm{Muon}}$ and therefore $\gamma^2_{\mathrm{SGD}}\Phi_{\mathrm{SGD}} \geq \gamma^2_{\mathrm{Muon}}\Phi_{\mathrm{Muon}}$*

Theorem 3.2. *In the undersampled case ($\delta \to \infty$) we have $\lim_{\delta\to\infty} \frac{\gamma^2_{\mathrm{SGD}}\Phi_{\mathrm{SGD}}}{\gamma^2_{\mathrm{Muon}}\Phi_{\mathrm{Muon}}} \geq 1$*

We state the next result in the aligned case for simplicity of presentation (see Proposition H.4).

Proposition 3.3. *Assume that $CV = V\Lambda$ for a diagonal matrix Λ. If the entries of Λ are in non-increasing order then $\gamma_{\mathrm{SGD}} \leq \gamma_{\mathrm{Muon}}$ with equality if and only if $\Lambda = cI$.*

3.3 Empirical Observations

The RF optimal step sizes: The asymptotic results in Section 3.2 compare update directions through local descent quantities, but they abstract away a key practical issue of step size stability. Specifically, if we evaluate the optimal per-step step size (i.e., exact line search), the conclusions of Davis & Drusvyatskiy (2026) completely reverse: standard GD achieves greater loss reduction than spectral descent, a phenomenon also noted by Gonon et al. (2026). However, optimal GD requires a highly oscillatory and practically unrealizable step-size schedule (Figure 5). Conversely, this exact line-search model reveals the core mechanistic advantage of spectral descent: its optimal step size is constant. Because this step size is proportional to $1/\|DA\|_F^2$, where D is the polar factor of G, the value remains practically constant throughout training.

The RF optimal norms: Beyond step-size stability, the RF model can be pushed further by dynamically selecting the Schatten norm exponent c that maximizes the local directional potential. We observe, however, that greedily optimizing c at every step yields rapid initial descent but degrades long-term dynamics, even when paired with optimal step sizes (Figure 5). Instead, by assuming a certain power-law decay, we derive a simple, fixed-exponent rule (Appendix I.2). This approximate choice significantly improves overall performance, overcoming the limitations of spectral descent for mean-zero activations identified by Davis & Drusvyatskiy (2026), as demonstrated in Figure I.1.

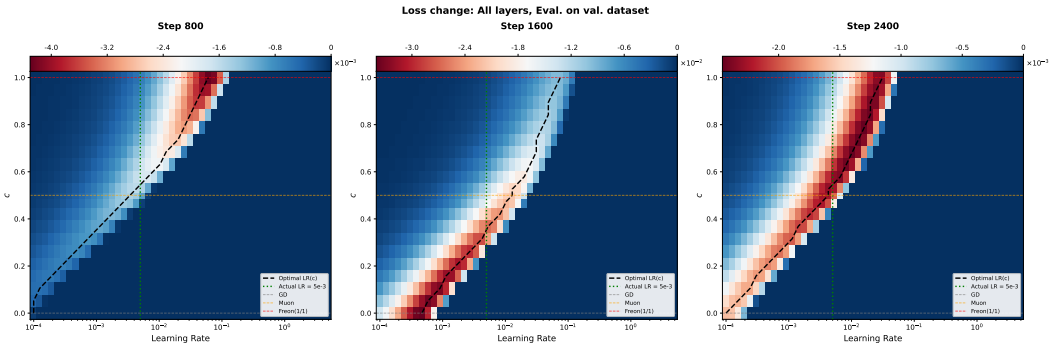


Figure 6: Loss change Δf over the joint (c, η) grid, evaluated on the full validation set. Each panel shows a different training point used for the calculation of the gradient. The dashed black curve marks the optimal learning rate per c; the green dotted line marks the actual training learning rate.

The naive RF plug-in for GPT-2: Attempting to map the RF model directly onto a GPT-2 architecture exposes distinct limitations, as illustrated by the optimal exponent heatmaps in Figure J.5 (Left) and Figure J.4. The RF model fails to provide an accurate quadratic approximation of the layer. Specifically, optimal exponents frequently fall into the strict non-norm regime ($c > 1/2$), indicating that such updates fundamentally deviate from typical RF quadratic landscapes (Figure J.3).

Single batch vs. Full dataset geometry in GPT-2: Given the inapplicability of the RF model, we evaluate both the Generalized Gauss-Newton (GGN) approximation and the exact directional landscape, which qualitatively agree (Figure J.5(ab)). For a single batch, evaluating the per-step decrease confirms that `Freon` at $c = 1$ yields the optimal local decrease (Figure J.5(c)), consistently across all layers (Figure J.11). This shows that the local geometry is severely distorted: favoring $c = 1$ is impossible in standard RF models (Figure J.3). Because local geometry poorly predicts overall performance, we evaluate the global dynamics across the full dataset, revealing a highly stochastic landscape: rather than converging, the optimal exponents fluctuate dynamically over batches during training (Figure 6, Figure J.7). Thus, the actual global landscape is significantly more complex than local, single-batch models suggest.

Φ and γ dynamics in GPT-2: Tracking Φ and γ throughout training reveals a shifting optimization hierarchy (Figure J.1). Initially, methods like `TruncatedSGD` exhibit strong local decrease (high Φ), with most `Freon` variants similar. At the next snapshot, this inverts: `TruncatedSGD` and `SGD` become heavily suboptimal with larger variance. This mirrors the validation loss trajectories Figure J.2, where the rate of decrease slows down for `SGD` variants, but remains good for `Freon`.

This reveals an interesting empirical observation: gradients tend to agree most in the middle of the singular value spectrum (in the sense of signal-to-noise ratio), rather than at the extremes ($c = 0$ or $c = 1$). Increasing c to target this high-SNR mid-range (e.g., $c \approx 3/4$) naturally suppresses the mean and variance of γ, as updates align with these mid-tier rather than maximal singular values.

Overall, these empirical observations clarify the mechanics of Equation (4): changing directions of descent from the gradient naturally suppresses γ, yet this deliberate sacrifice of γ enables a massive amplification of Φ – the primary driver of loss reduction. Trading γ for a boosted Φ fundamentally explains why the non-standard $c > 1/2$ regime ultimately dominates the optimization landscape.

4 Conclusion

This work challenges the prevailing geometric narrative of spectral optimizers like `Muon`. `Freon` reveals optimal updates often violate proper norm regimes, while `Kaon` proves randomized spectra perform just as well. What actually drives performance is the interplay between batch gradient alignment and directional descent potential, combined with appropriate choices of step-sizes. We discuss limitations in Appendix C, leaving a final question: if geometry is not necessary for performance, should we expect theories stemming from it to yield practically useful insights?

AI USE STATEMENT

In this work, we used generative AI tools for help in developing theoretical models or conceptual frameworks, provide feedback on research methodology or experiments, help implement methods. And the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for drafting parts of a research paper, discovering research topics or identify gaps, brainstorming, editing to improve readability, propose title and keywords for the paper. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

REPRODUCIBILITY STATEMENT

All datasets, model architectures, and dependencies used in this work are publicly available from existing open-source repositories. We provide experimental details, hyperparameters, algorithms, and proofs throughout the paper and appendices. Code for reproducing the experiments will be released upon publication.

REFERENCES

Milton Abramowitz and Irene A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. National Bureau of Standards / Dover, 1964. Section 15.1.

N. I. Achieser. *Theory of approximation*. Dover Publications, Inc., New York, 1992. ISBN 0-486-67129-1. Translated from the Russian and with a preface by Charles J. Hyman, Reprint of the 1956 English translation.

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates. *arXiv preprint arXiv:2504.05295*, 2025.

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=yRtgZ1K8hO.

Jeremy Bernstein. Deriving muon, 2025. URL https://jeremybernste.in/writing/deriving-muon.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology. In *OPT 2024: Optimization for Machine Learning*.

Rajendra Bhatia. *Matrix analysis*. Springer Science & Business Media, 2013.

Elliott Ward Cheney and Henry L Loeb. Generalized rational approximation. *SIAM Journal on Numerical Analysis, Ser. B*, 1(1):11–25, 1964.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Romain Couillet and Zhenyu Liao. *Random Matrix Theory*, pp. 35–154. Cambridge University Press, 2022.

Michael Crawshaw, Chirag Modi, Mingrui Liu, and Robert M. Gower. An exploration of non-euclidean gradient descent: Muon and its many variants, 2025. URL https://arxiv.org/abs/2510.09827.

Damek Davis and Dmitriy Drusvyatskiy. When do spectral gradient updates help in deep learning?, 2026. URL https://arxiv.org/abs/2512.04299.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Zhehang Du and Weijie Su. The newton-muon optimizer, 2026. URL https://arxiv.org/abs/2604.01472.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Chen Fan, Mark Schmidt, and Christos Thrampoulidis. Implicit bias of spectral descent and muon on multiclass separable data. *arXiv preprint arXiv:2502.04664*, 2025.

Silviu-Ioan Filip, Yuji Nakatsukasa, Lloyd N. Trefethen, and Bernhard Beckermann. Rational minimax approximation via adaptive barycentric representations. *SIAM Journal on Scientific Computing*, 40(4):A2427–A2455, 2018. doi: 10.1137/17M1132409. URL https://doi.org/10.1137/17M1132409.

Evan S Gawlik and Yuji Nakatsukasa. Approximating the pth root by composite rational functions. *Journal of Approximation Theory*, 266:105577, 2021.

Wenbo Gong, Javier Zazo, Qijun Luo, Puqian Wang, James Hensman, and Chao Ma. Aro: A new lens on matrix optimization for large models, 2026. URL https://arxiv.org/abs/2602.09006.

Antoine Gonon, Andreea-Alexandra Muşat, and Nicolas Boumal. Insights on muon from simple quadratics, 2026. URL https://arxiv.org/abs/2602.11948.

W. B. Gragg. The padé table and its relation to certain algorithms of numerical analysis. *SIAM Review*, 14(1):1–62, 1972. ISSN 00361445, 10957200. URL http://www.jstor.org/stable/2028911.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, pp. 1842–1850. PMLR, 2018.

Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

Rustem Islamov, Michael Crawshaw, Jeremy Cohen, and Robert Gower. Non-euclidean gradient descent operates at the edge of stability. *arXiv preprint arXiv:2603.05002*, 2026.

Xiaowen Jiang, Andrei Semenov, and Sebastian U. Stich. Enhancing llm training via spectral clipping, 2026. URL https://arxiv.org/abs/2603.14315.

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

Charles Kenney and Alan J. Laub. Rational iterative methods for the matrix sign function. *SIAM Journal on Matrix Analysis and Applications*, 12(2):273–291, 1991. doi: 10.1137/0612020. URL https://doi.org/10.1137/0612020.

Gyu Yeol Kim and Min hwan Oh. Convergence of muon with newton-schulz, 2026. URL https://arxiv.org/abs/2601.19156.

Juno Kim, Eshaan Nichani, Denny Wu, Alberto Bietti, and Jason D. Lee. Sharp capacity scaling of spectral optimizers in learning associative memory, 2026. URL https://arxiv.org/abs/2603.26554.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

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

Frederik Kunstner, Philipp Hennig, and Lukas Balles. Limitations of the empirical Fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/46a558d97954d0692411c861cf78ef79-Abstract.html.

Tim Tsz-Kit Lau, Qi Long, and Weijie Su. Polargrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective, 2026. URL https://arxiv.org/abs/2505.21799.

Jiaxiang Li and Mingyi Hong. A note on the convergence of muon, 2025. URL https://arxiv.org/abs/2502.02900.

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

H. L. Loeb. Approximation by generalized rationals. *SIAM Journal on Numerical Analysis*, 3(1): 34–55, 1966. doi: 10.1137/0703003.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. URL https://arxiv.org/abs/1711.05101.

Jianhao Ma, Yu Huang, Yuejie Chi, and Yuxin Chen. Preconditioning benefits of spectral orthogonalization in muon, 2026. URL https://arxiv.org/abs/2601.13474.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. The WikiText long term dependency language modeling dataset. 2016.

Yuji Nakatsukasa and Roland W. Freund. Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of zolotarev's functions. *SIAM Review*, 58(3):461–493, 2016. doi: 10.1137/140990334. URL https://doi.org/10.1137/140990334.

Yuji Nakatsukasa, Zhaojun Bai, and François Gygi. Optimizing halley's iteration for computing the matrix polar decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2700–2720, 2010.

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, 2026. URL https://arxiv.org/abs/2605.13079.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained lmos. *arXiv preprint arXiv:2502.07529*, 2025.

Shashikant C Phatak and S Suresh Rao. Logistic map: A possible random-number generator. *Physical review E*, 51(4):3670, 1995.

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making muon & scion great again!(bridging theory and practice of lmo-based optimizers for llms). In *High-dimensional Learning Dynamics 2025*.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining, 2025. URL https://arxiv.org/abs/2509.01440.

Maria-Eleni Sfyraki and Jun-Kun Wang. Lions and muons: Optimization via stochastic frank-wolfe under heavy-tailed noise, 2026. URL https://arxiv.org/abs/2506.04192.

Egor Shulgin, Sultan AlRashed, Francesco Orabona, and Peter Richtárik. Beyond the ideal: Analyzing the inexact muon update, 2025. URL https://arxiv.org/abs/2510.19933.

Chongjie Si, Debing Zhang, and Wei Shen. Adamuon: Adaptive muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025.

Weijie Su. Isotropic curvature model for understanding deep learning optimization: Is gradient orthogonalization optimal?, 2025. URL https://arxiv.org/abs/2511.00674.

5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, Yean Cheng, Yifan An, Yilin Niu, Yuanhao Wen, Yushi Bai, Zhengxiao Du, Zihan Wang, Zilin Zhu, Bohan Zhang, Bosi Wen, Bowen Wu, Bowen Xu, Can Huang, Casey Zhao, Changpeng Cai, Chao Yu, Chen Li, Chendi Ge, Chenghua Huang, Chenhui Zhang, Chenxi Xu, Chenzheng Zhu, Chuang Li, Congfeng Yin, Daoyan Lin, Dayong Yang, Dazhi Jiang, Ding Ai, Erle Zhu, Fei Wang, Gengzheng Pan, Guo Wang, Hailong Sun, Haitao Li, Haiyang Li, Haiyi Hu, Hanyu Zhang, Hao Peng, Hao Tai, Haoke Zhang, Haoran Wang, Haoyu Yang, He Liu, He Zhao, Hongwei Liu, Hongxi Yan, Huan Liu, Huilong Chen, Ji Li, Jiajing Zhao, Jiamin Ren, Jian Jiao, Jiani Zhao, Jianyang Yan, Jiaqi Wang, Jiayi Gui, Jiayue Zhao, Jie Liu, Jijie Li, Jing Li, Jing Lu, Jingsen Wang, Jingwei Yuan, Jingxuan Li, Jingzhao Du, Jinhua Du, Jinxin Liu, Junkai Zhi, Junli Gao, Ke Wang, Lekang Yang, Liang Xu, Lin Fan, Lindong Wu, Lintao Ding, Lu Wang, Man Zhang, Minghao Li, Minghuan Xu, Mingming Zhao, Mingshu Zhai, Pengfan Du, Qian Dong, Shangde Lei, Shangqing Tu, Shangtong Yang, Shaoyou Lu, Shijie Li, Shuang Li, Shuang-Li, Shuxun Yang, Sibo Yi, Tianshu Yu, Wei Tian, Weihan Wang, Wenbo Yu, Weng Lam Tam, Wenjie Liang, Wentao Liu, Xiao Wang, Xiaohan Jia, Xiaotao Gu, Xiaoying Ling, Xin Wang, Xing Fan, Xingru Pan, Xinyuan Zhang, Xinze Zhang, Xiuqing Fu, Xunkai Zhang, Yabo Xu, Yandong Wu, Yida Lu, Yidong Wang, Yilin Zhou, Yiming Pan, Ying Zhang, Yingli Wang, Yingru Li, Yinpei Su, Yipeng Geng, Yitong Zhu, Yongkun Yang, Yuhang Li, Yuhao Wu, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yuxuan Zhang, Zezhen Liu, Zhen Yang, Zhengda Zhou, Zhongpei Qiao, Zhuoer Feng, Zhuorui Liu, Zichen Zhang, Zihan Wang, Zijun Yao, Zikang Wang, Ziqiang Liu, Ziwei Chai, Zixuan Li, Zuodong Zhao, Wenguang Chen, Jidong Zhai, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025a. URL https://arxiv.org/abs/2508.06471.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jing Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Junjie Yan, Yuzi Yan, Xiaofei Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yangkun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Huabin Zheng, Shaojie Zheng, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi k2: Open agentic intelligence, 2025b. URL https://arxiv.org/abs/2507.20534.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M Kakade. Soap: Improving and stabilizing shampoo using adam for language modeling. In *The Thirteenth International Conference on Learning Representations*.

Hubert Stanley Wall. *Analytic Theory of Continued Fractions*. D. Van Nostrand Company, Inc., New York, 1948. Reprinted by Chelsea Publishing Company, Bronx, N.Y., 1967.

Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Cunxiao Du, Chao Du, Tianyu Pang, Zhuoran Yang, Mingyi Hong, and Vincent Y. F. Tan. Muon outperforms adam in tail-end associative memory learning, 2025. URL https://arxiv.org/abs/2509.26030.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them, 2025. URL https://arxiv.org/abs/2509.02046.

Ruihan Xu, Jiajin Li, and Yiping Lu. On the width scaling of neural optimizers under matrix operator norms i: Row/column normalization and hyperparameter transfer, 2026. URL https://arxiv.org/abs/2603.09952.

Greg Yang, James B. Simon, and Jeremy Bernstein. A spectral condition for feature learning, 2024. URL https://arxiv.org/abs/2310.17813.

Jui-Nan Yen, Si Si, Zhao Meng, Felix Yu, Sai Surya Duvvuri, Inderjit S Dhillon, Cho-Jui Hsieh, and Sanjiv Kumar. LoRA done RITE: Robust invariant transformation equilibration for loRA optimization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=VpWki1v2P8.

Dingzhi Yu, Hongyi Tao, Yuanyu Wan, Luo Luo, and Lijun Zhang. Sign-based optimizers are effective under heavy-tailed noise, 2026. URL https://arxiv.org/abs/2602.07425.

Jack Zhang, Noah Amsel, Berlin Chen, and Tri Dao. Gram newton-schulz, 2026. URL https://dao-ailab.github.io/blog/2026/gram-newton-schulz/.

A NanoGPT Tuning Details for Figures 1 and 2 **16**

B GPT2 Training Details **16**

C Limitations **17**

D Methods Extras **17**

D.1 TruncatedSGD 17

E Methods Proofs **17**

E.1 Preconditioned Spectral Descent 17

E.2 Convergence of `Freon` in the quasi-norm regime 21

E.3 Equivariance of `Freon` ($c = 1$) 23

F From Polar to Hölder Express **24**

F.1 Fundamental Properties of the a/b-Method 24

F.2 Accuracy of approximations 29

F.3 Instability of Polynomial Iterations 29

F.4 Iterative Methods for $(GG^\top)^{-a/b}\,G$ 29

F.5 Picking Parameters for Freon Iterations: Cushion and l 34

F.6 Freon Rational Speed Ablations 34

G Further Optimizer Details **34**

G.1 Kaon with SVD 34

G.2 Kaon 36

H Random Feature Asymptotics **38**

I Random Feature Experiments **43**

I.1 Detailed Setup 43

I.2 Exponent Approximation for Optimal Schatten Norm 44

I.3 Optimal C methods 44

J GPT2 Additional Experiments **44**

Table A.1: Learning rates selected by the seed-42 tuning sweep and used for Figures 2 and 3.

Optimizer	η_{matrix}	η_{base}	Extra setting
`Muon`	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	Polar update
`Kaon`	$2 \cdot 10^{-2}$	$8 \cdot 10^{-3}$	5 matrix iterations
`Freon` $c = 2/3$	$2 \cdot 10^{-2}$	$2 \cdot 10^{-2}$	5 matrix iterations
`Freon` $c = 3/4$	$2 \cdot 10^{-2}$	$8 \cdot 10^{-3}$	5 matrix iterations

A NanoGPT Tuning Details for Figures 2 and 3

The NanoGPT results in Figures 2 and 3 were run through the same training pipeline on NanoGPT. Every tuning run and learning rate sweep used 8 NVIDIA H100 80GB GPUs.

For each optimizer, we tuned a matrix-update learning rate η_{matrix} and a base learning rate η_{base}. The recorded tuning grid was

```
(5e-3, 1e-3)  (1e-2, 1e-3)  (2e-2, 1e-3)  (1e-2, 5e-4)
(5e-3, 2e-3)  (5e-3, 4e-3)  (5e-3, 8e-3)  (5e-3, 2e-2)
(1e-2, 2e-3)  (1e-2, 4e-3)  (1e-2, 8e-3)  (1e-2, 2e-2)
(2e-2, 2e-3)  (2e-2, 4e-3)  (2e-2, 8e-3)  (2e-2, 2e-2)
```

for (`matrix_lr`, `base_lr`). The selected learning rates used for Figures 2 and 3 are listed in Table A.1. `Kaon` and both `Freon` variants used 5 Newton–Schulz/QDWH-style matrix iterations.

Figure 2 then jointly scales the tuned pair as $(\eta_{\text{matrix}}, \eta_{\text{base}}) = \rho(\eta^\star_{\text{matrix}}, \eta^\star_{\text{base}})$. The full sensitivity sweep ran 4 optimizers, 7 scale factors, and 3 seeds, for 84 runs in total. The scale factors were $\rho \in \{0.03, 0.1, 0.3, 1, 3, 10, 30, 100\}$ and the seeds were $42, 43, 44$. The plotted version omits $\rho = 100$, so the x-axis in Figure 2 shows the matrix-update learning rates induced by $\rho \in \{0.03, 0.1, 0.3, 1, 3, 10, 30\}$. Error bars show ± 2 std over the three seeds.

Figure 3 reruns the selected curve configurations with denser validation logging. These dense-validation jobs used the same tuned learning rates and the same seeds $42, 43, 44$, with validation every 8 optimizer steps. The dense runs generated Muon at $\rho = 1$, `Kaon` at $\rho = 1$ and $\rho = 3$, `Freon` $c = 2/3$ at $\rho = 1$, and `Freon` $c = 3/4$ at $\rho = 1$. The plotted dense-validation figure uses Muon at $\rho = 1$, `Kaon` at $\rho = 3$, and both `Freon` variants at $\rho = 1$. The plotted lines show averages across three seeds and shaded regions showing ± 2 std.

B GPT2 Training Details

We utilize the existing codebase from (Amsel et al., 2026; Crawshaw et al., 2025). All experiments train a GPT-2 model (12 layers, 12 attention heads, embedding dimension 768, 124M parameters) with GELU activations and FlashAttention on the WikiText-2 dataset (Merity et al., 2016). Each run uses a context length of 512 tokens, batch size 8, gradient norm clipping at 1.0, and a constant-then-linear decay schedule with a 5% warm-up. The combined figure Figure 4 reports final validation loss as a function of learning rate, sweeping 8–10 learning rates per optimizer configuration.

SGD and TruncatedSGD. Runs use SGD with momentum 0.9. Truncation levels of $0, 0.1, 0.5, 1, 2, 5, 10\%$ are swept, where 0% recovers vanilla SGD. Runs with non-zero truncation rescale the update 2-norm to match that of the untruncated gradient.

Muon. Standard Muon with momentum 0.95 and 5 Newton–Schulz iterations ($p/q = 1/2$).

Kaon. Same Newton–Schulz backbone as Muon (5 steps, momentum 0.95) but with the chaos iterations.

Freon. FreonRational with momentum 0.95, 5 Newton–Schulz steps, and the exponent $c = a/b$ swept over $0, 1/4, 1/3, 1/2, 2/3, 3/4, 1$, where $c = 0$ recovers gradient descent and $c = 1/2$ recovers Muon.

C LIMITATIONS

While our analysis provides novel mechanistic insights into the behavior of spectral optimizers, there is a number of theoretical and practical limitations:

1. **Scope of Empirics and Theory:** Our empirical validation is currently restricted to a random feature model and a single architecture modality (a GPT-2 language model). Furthermore, our rigorous analysis of the two core optimization quantities, batch gradient alignment and directional descent potential, is limited to the RF setting. It remains an open question how precisely these dynamics map to other modalities (e.g., vision) or more theoretically amenable non-quadratic loss landscapes.
2. **Theoretical Assumptions:** The exact asymptotic results of Section 3.2 rely strictly on the assumption of mean-zero activations, excluding activations such as ReLU.
3. **Lack of Predictive Power:** Our framework primarily serves as a diagnostic lens. The quantities governing the optimization trajectory are highly stochastic, meaning they can only be reliably evaluated post-factum rather than predicted a priori.
4. **Optimal c:** Because the optimal local geometry is heavily distorted and highly sensitive to individual batch dynamics, our attempts to dynamically track and utilize the optimal Schatten parameters (p, q) step-by-step ultimately failed. The optimal exponents fluctuate too wildly across batches to formulate a stable, greedily adaptive schedule.
5. **QDWH Coupled Cholesky** The current implementation of the `Freon` algorithm serves primarily to demonstrate the theoretical viability of quasi-norm updates. While the algorithm could certainly be engineered for greater computational efficiency, it is currently unclear whether the extensive engineering effort is even necessary for `Freon`, given the similar behaviour across all the optimizers.

D METHODS EXTRAS

D.1 TRUNCATEDSGD

As a first step in relaxing the strict geometric assumptions of the LMO framework, we isolate the role of the largest singular values in optimization stability. We study `TruncatedSGD`, an optimizer that zeros out the top $p\%$ of singular values while leaving the rest unchanged. Although a full SVD is impractical, `TruncatedSGD` serves a pedagogical purpose: it tests whether standard `SGD` fails primarily due to instability induced by excessively large singular values.

Concurrently with our work, Jiang et al. (2026) propose a Newton-Schulz based approximation to fixed-value clipping, similarly targeting large singular values under the assumption that batch noise concentrates in this part of the spectrum. In line with this motivation, we observe in Figure 4 that `TruncatedSGD` improves over `SGD`, confirming that suppressing large singular values is *necessary* for stability. However, it still falls short of `Muon`, showing that suppression alone is *insufficient*. The view that clipping primarily removes noise implicitly treats the underlying gradient direction as intrinsically meaningful; our results in Section 3.3 ultimately challenge this premise, suggesting instead that the gradient direction itself can be poorly aligned with useful progress.

E METHODS PROOFS

E.1 PRECONDITIONED SPECTRAL DESCENT

Theorem 2.1 (Preconditioned Spectral Descent vs. Regularized Steepest Descent)**.** *Fix the function f and the point X_k. A regularized steepest descent update w.r.t. a unitarily invariant matrix norm from X_k is a preconditioned spectral descent update. On the other hand, a preconditioned spectral descent update from X_k is a steepest descent update w.r.t. a unitarily invariant matrix norm if and only if p_k preserves the order $\leq$ of the entries of σ_k.*

Proof. Let $\|\cdot\|$ be a unitarily invariant matrix norm and $\|\cdot\|_*$ its dual norm. Write the SVD $G_k = U_k \operatorname{diag}(\sigma_k) V_k^\top$. Then since $\|\cdot\|$ is unitarily invariant, there is a norm $\|\cdot\|_\sigma$ on $\mathbb{R}^r$ symmetric in the coordinates such that $\|G_k\| = \|\sigma_k\|_\sigma$. The first part of the theorem then follows from

formulating steepest descent as

$$\begin{aligned} X_{k+1} &= X_k - \eta \|G_k\|_* \operatorname{lmo}_{\|\cdot\|}(G_k) \\ &= X_k - \alpha_k \langle G_k, D_k \rangle D_k \end{aligned}$$

with $\alpha_k = \eta$ and $D_k = \operatorname{lmo}_{\|\cdot\|}(G_k) = U_k \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k) V_k^\top$.

Now consider a preconditioned spectral descent update. Then if it is a steepest descent update, we have as in the above argument that

$$p_k(\sigma_k) = \frac{\eta \|G_k\|_*}{\alpha_k \langle G_k, D_k \rangle} \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k) \propto \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k). \tag{6}$$

Write $x = \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k)$. Suppose $[\sigma_k]_i \leq [\sigma_k]_j$ and $x_i > x_j$. Define $\tilde{x} \in \mathbb{R}^r$ such that $\tilde{x}_i = x_j$, $\tilde{x}_j = x_i$ and $\tilde{x}_{\tilde{i}} = x_{\tilde{i}}$ for all $\tilde{i} \neq i, j$. Then since $\|\cdot\|_\sigma$ is symmetric in the coordinates, $\|\tilde{x}\|_\sigma = \|x\|_\sigma = 1$. But we see that $\langle \tilde{x}, \sigma_k \rangle > \langle x, \sigma_k \rangle$. This contradicts $x = \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k)$. Thus $\operatorname{lmo}_{\|\cdot\|_\sigma}$ preserves the order $\leq$ of the entries of σ_k, hence so does p_k.

Conversely, suppose p_k preserves the order $\leq$ of the entries of σ_k. For $\tau \in \operatorname{Sym}(r)$ a permutation, let $P^\tau \in GL_r(\mathbb{R})$ be the corresponding permutation matrix. For $w \in \{-1, +1\}^r$, let $Q^w := \operatorname{Diag}(w)$. Then define

$$\mathcal{V} := \{Q^w P^\tau p_k(\sigma_k) : \tau \in \operatorname{Sym}(r), w \in \{-1, +1\}^r\} \qquad \text{and} \qquad \mathcal{B} = \operatorname{conv}(\mathcal{V}).$$

$\mathcal{B}$ is a bounded, convex, balanced neighbourhood of 0. So $\|\cdot\|_\sigma \colon \mathbb{R}^r \to \mathbb{R}_{\geq 0}$ given by

$$\|x\|_\sigma := \inf\{t \in \mathbb{R}_{>0} : t^{-1} x \in \mathcal{B}\}$$

defines a norm. $\|\cdot\|_\sigma$ is symmetric in the coordinates since $\mathcal{V}$, and hence $\mathcal{B}$, are. $\mathcal{B}$ is the closed unit ball of $\|\cdot\|_\sigma$. Now $\operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k)$ can always be chosen to be in $\mathcal{V}$. Indeed, writing the convex combination $\operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k) = \sum_{v \in \mathcal{V}} a_v v$, we have $\langle \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k), \sigma_k \rangle = \sum_{v \in \mathcal{V}} a_v \langle v, \sigma_k \rangle$, so there is $v \in \mathcal{V}$ such that $\langle v, \sigma_k \rangle \geq \langle \operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k), \sigma_k \rangle$, so we could have chosen $\operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k) = v$. Moreover, by order preservation of p_k, we see that $\langle p_k(\sigma_k), \sigma_k \rangle \geq \langle v, \sigma_k \rangle$ for any $v \in \mathcal{V}$. So we may choose $\operatorname{lmo}_{\|\cdot\|_\sigma}(\sigma_k) = p_k(\sigma_k)$. Finally, by applying $\|\cdot\|_\sigma$ to singular values, it defines a unitarily invariant matrix norm $\|\cdot\|$. Choosing $\eta = \frac{\alpha_k \langle G_k, D_k \rangle}{\|G_k\|_*}$ yields the desired eq. (6) and concludes the proof. □

Theorem 2.3 (Convergence of Preconditioned Spectral Descent)**.** *Let $\|\cdot\|$ be a unitarily invariant matrix norm, and let $\|\cdot\|_*$ be its dual norm. Let $f : \mathbb{R}^{m \times n} \to \mathbb{R}$ be continuously differentiable, bounded below by $f_{\min}$, with $L_{\mathcal{X}}$-Lipschitz continuous gradients with respect to $\|\cdot\|$.*

Consider preconditioned spectral descent updates as in Section 2.1, and assume there exist constants $m_k \geq 0$ and $M_k > 0$ satisfying the divergence condition $\sum_{k=0}^{\infty} (m_k / M_k)^2 = \infty$, such that for all $\sigma \in \mathbb{R}^r_{>0}$, the mapping p_k satisfies the following sufficient descent and boundedness conditions:

$$\langle \sigma, p_k(\sigma) \rangle \geq m_k \|\sigma\|_* \qquad \textit{and} \qquad \|p_k(\sigma)\| \leq M_k.$$

Then, choosing $\alpha_k = \eta \frac{1}{L_{\mathcal{X}} M_k^2}$ for any $\eta \in (0, 2)$, the following convergence rate holds for $K \geq 1$:

$$\min_{0 \leq k < K} \|G_k\|_*^2 \leq \frac{L_{\mathcal{X}}(f(X_0) - f_{\min})}{\eta(1 - \frac{\eta}{2}) \sum_{k=0}^{K-1} \left(\frac{m_k}{M_k}\right)^2} \qquad \textit{and} \qquad \liminf_{k \to \infty} \|G_k\|_* = 0.$$

Proof. As f has $L_{\mathcal{X}}$-Lipschitz continuous gradients with respect to $\|\cdot\|$, we can use the generalized descent lemma, which for the total update $\Delta_k = \langle G_k, D_k \rangle D_k$ yields:

$$\begin{aligned} f(X_{k+1}) &\leq f(X_k) - \alpha_k \langle G_k, \Delta_k \rangle + \frac{L_{\mathcal{X}}}{2} \alpha_k^2 \|\Delta_k\|^2 \\ &= f(X_k) - \alpha_k \langle G_k, D_k \rangle^2 + \frac{L_{\mathcal{X}}}{2} \alpha_k^2 \langle G_k, D_k \rangle^2 \|D_k\|^2 \end{aligned}$$

Factoring out the squared inner product isolates the step parameter logic:

$$f(X_{k+1}) \leq f(X_k) - \langle G_k, D_k \rangle^2 \left(\alpha_k - \frac{L_{\mathcal{X}}}{2} \alpha_k^2 \|D_k\|^2 \right)$$

By the absolute upper bound (Condition 2) and unitary invariance, $\|D_k\| = \|p_k(\sigma_k)\| \leq M_k$. Substituting this guarantees the worst-case penalty:

$$f(X_{k+1}) \leq f(X_k) - \langle G_k, D_k \rangle^2 \left(\alpha_k - \frac{L_{\mathcal{X}}}{2} \alpha_k^2 M_k^2 \right)$$

Substituting the generalized step size $\alpha_k = \frac{\eta}{L_{\mathcal{X}} M_k^2}$:

$$\begin{aligned} f(X_{k+1}) &\leq f(X_k) - \langle G_k, D_k \rangle^2 \left(\frac{\eta}{L_{\mathcal{X}} M_k^2} - \frac{L_{\mathcal{X}}}{2} \frac{\eta^2}{L_{\mathcal{X}}^2 M_k^4} M_k^2 \right) \\ &= f(X_k) - \frac{\eta(1 - \frac{\eta}{2})}{L_{\mathcal{X}} M_k^2} \langle G_k, D_k \rangle^2 \end{aligned}$$

By the scale-invariant lower bound (Condition 1), the inner product is bounded below by the scaled dual norm of the gradient: $\langle G_k, D_k \rangle = \langle \sigma_k, p_k(\sigma_k) \rangle \geq m_k \|\sigma_k\|_* = m_k \|G_k\|_*$. Squaring this and substituting yields:

$$f(X_{k+1}) \leq f(X_k) - \frac{\eta(1 - \frac{\eta}{2}) m_k^2}{L_{\mathcal{X}} M_k^2} \|G_k\|_*^2$$

Rearranging and summing over K iterations gives a telescoping sum:

$$\frac{\eta(1 - \frac{\eta}{2})}{L_{\mathcal{X}}} \sum_{k=0}^{K-1} \left(\frac{m_k}{M_k} \right)^2 \|G_k\|_*^2 \leq f(X_0) - f(X_K) \leq f(X_0) - f_{\min}$$

We can lower-bound the sum on the left by replacing each gradient norm with the minimum gradient norm encountered over the K iterations:

$$\left[\min_{0 \leq k < K} \|G_k\|_*^2 \right] \sum_{k=0}^{K-1} \left(\frac{m_k}{M_k} \right)^2 \leq \sum_{k=0}^{K-1} \left(\frac{m_k}{M_k} \right)^2 \|G_k\|_*^2$$

Substituting this into the inequality and isolating the minimum gradient norm perfectly recovers the finite-time rate:

$$\min_{0 \leq k < K} \|G_k\|_*^2 \leq \frac{L_{\mathcal{X}}(f(X_0) - f_{\min})}{\eta(1 - \frac{\eta}{2}) \sum_{k=0}^{K-1} \left(\frac{m_k}{M_k} \right)^2}$$

Because the series $\sum (m_k/M_k)^2$ diverges to infinity as $K \to \infty$, the right-hand side converges to exactly zero. Therefore, $\liminf_{k \to \infty} \|G_k\|_* = 0$. □

Theorem E.1 (Convergence of Stochastic Spectral Descent)**.** *Let the setting be identical to Theorem 2.3. Suppose $p_k(\sigma)$ is generated by a stochastic process over $(k, \sigma) \in \mathbb{N} \times \mathbb{R}^r_{>0}$, and denote by $\mathbb{E}_{k-1}[\cdot]$ the conditional expectation given the history up to step $k-1$. Assume that the random variables m_k, M_k from Theorem 2.3 satisfy the following bounds almost surely, $m_k \geq 0$ and $M_k \in (0, \infty)$.*

Let $\mu_k = \mathbb{E}_{k-1}[(m_k/M_k)^2]$, and assume $\sum_{k=0}^{\infty} \mu_k = \infty$ almost surely. Then, for step size chosen as $\alpha_k = \frac{\eta}{L_{\mathcal{X}} M_k^2}$ for $\eta \in (0, 2)$, almost surely, the sequence achieves the finite-time convergence rate:

$$\min_{0 \leq k < K} \|G_k\|_*^2 = \mathcal{O}\left(\frac{1}{\sum_{k=0}^{K-1} \mu_k} \right) \qquad \textit{and} \qquad \liminf_{k \to \infty} \|G_k\|_* = 0.$$

Proof. First, using Cauchy-Schwarz, we have $\langle \sigma, p_k(\sigma) \rangle \leq \|\sigma\|_* \|p_k(\sigma)\|$. Substituting our assumed almost-sure bounds into this inequality yields:

$$m_k \|\sigma\|_* \leq \langle \sigma, p_k(\sigma) \rangle \leq \|\sigma\|_* \|p_k(\sigma)\| \leq M_k \|\sigma\|_*$$

Dividing by $\|\sigma\|_*$ reveals that $m_k \leq M_k$ almost surely. Therefore, the sequence of random variables $Z_k = (m_k/M_k)^2$ is non-negative and uniformly bounded: $Z_k \in [0, 1]$.

We now invoke Lévy's Extended Borel-Cantelli Lemma for conditional expectations. For a sequence of uniformly bounded, non-negative random variables Z_k, the lemma states that the sum diverges almost surely and that their asymptotic ratio converges to exactly 1 almost surely if and only if the sum of their conditional expectations diverges:

$$\sum_{k=0}^{\infty} Z_k = \infty \text{ and } \lim_{K\to\infty} \frac{\sum_{k=0}^{K-1} Z_k}{\sum_{k=0}^{K-1} \mathbb{E}_{k-1}[Z_k]} = 1 \text{ a.s.} \iff \sum_{k=0}^{\infty} \mathbb{E}_{k-1}[Z_k] = \infty$$

By assumption, the sum of the conditional expectations diverges: $\sum_{k=0}^{\infty} \mu_k = \infty$. By the ratio limit above, the actual observed sum of the random variables grows asymptotically at the exact same rate:

$$\sum_{k=0}^{K-1} \left(\frac{m_k}{M_k}\right)^2 = \Omega\left(\sum_{k=0}^{K-1} \mu_k\right) \quad \text{a.s.}$$

Since the random mappings $p_k(\sigma)$ almost surely satisfy the deterministic bounds of Theorem 2.3 at every step, we can directly substitute this equivalent asymptotic growth into the denominator of the finite-time rate derived in Theorem 2.3:

$$\min_{0\le k<K} \|G_k\|_*^2 \le \frac{L_{\mathcal{X}}(f(X_0) - f_{\min})}{\eta(1-\frac{\eta}{2})\Omega\left(\sum_{k=0}^{K-1} \mu_k\right)} = \mathcal{O}\left(\frac{1}{\sum_{k=0}^{K-1} \mu_k}\right)$$

As $K \to \infty$, the diverging sum in the denominator guarantees that this bound converges to zero, ensuring asymptotic convergence with probability 1: $\liminf_{k\to\infty} \|G_k\|_* = 0$. □

Theorem 2.5 (Convergence of Random Spectral Descent)**.** *Let the setting be identical to Theorem 2.3. Suppose at each iteration k, the stochastic mapping is defined specifically as $p_k(\sigma) = \frac{E_k}{\|E_k\|}$, where the entries of E_k are i.i.d. samples of a probability distribution on $\mathbb{R}_{>0}$ independent of k.*

If $\alpha_k = \frac{\eta}{L_{\mathcal{X}}}$ for any constant $\eta \in (0, 2)$, then almost surely, the sequence $\min_{0\le k<K} \|G_k\|_*^2$ achieves a convergence rate of $\mathcal{O}(1/K)$.

Proof. Because all matrix norms are equivalent in finite dimensions, there exists a universal constant $c > 0$ such that for any vector $v \in \mathbb{R}^r$, $\|v\|_1 \ge c\|v\|_*$. The upper bound is trivially constant from normalization: $M_k = 1$. Next, we verify the scale-invariant lower bound. The inner product is:

$$\langle \sigma_k, p_k(\sigma_k)\rangle = \sum_{i=1}^{r} [\sigma_k]_i \frac{[E_k]_i}{\|E_k\|} \ge \frac{[E_k]_{\min}}{\|E_k\|} \|\sigma_k\|_1$$

where $[E_k]_{\min} = \min_i [E_k]_i$. Applying the norm equivalence $\|\sigma_k\|_1 \ge c\|\sigma_k\|_*$ gives:

$$\langle \sigma_k, p_k(\sigma_k)\rangle \ge \left(\frac{c[E_k]_{\min}}{\|E_k\|}\right) \|\sigma_k\|_*$$

We set $m_k = \frac{c[E_k]_{\min}}{\|E_k\|}$. Because the vectors E_k are drawn entirely independently of the algorithmic history up to step $k-1$, the conditional expectation equals the unconditional expectation.

Because $[E_k]_{\min}$ is the minimum of a finite number r of positive random variables, $m_k > 0$. Furthermore, since m_k/M_k is bounded, its expectation exists and evaluates to a strictly positive constant μ:

$$\mu_k = \mathbb{E}_{k-1}\left[\left(\frac{m_k}{M_k}\right)^2\right] = \mathbb{E}\left[\left(\frac{c[E_k]_{\min}}{\|E_k\|}\right)^2\right] = \mu > 0$$

Because $\mu_k = \mu$ is constant, the sum of expectations evaluates to $\sum_{k=0}^{K-1} \mu_k = K\mu$, which diverges as $K \to \infty$. The stochastic mapping thus satisfies all conditions of Theorem E.1. Substituting $\sum \mu_k = K\mu$ into the generalized rate yields exactly:

$$\mathcal{O}\left(\frac{1}{\sum_{k=0}^{K-1} \mu_k}\right) = \mathcal{O}\left(\frac{1}{K\mu}\right) = \mathcal{O}\left(\frac{1}{K}\right)$$

Therefore, setting the step size $\alpha_k = \frac{\eta}{L_{\mathcal{X}}}$ almost surely guarantees the $\mathcal{O}(1/K)$ finite-time rate and asymptotic convergence. □

E.2 CONVERGENCE OF FREON IN THE QUASI-NORM REGIME

The convergence result of Theorem 2.3 does not apply directly to unregularized Freon in the spectrum-inverting regime $c > 1/2$. For singular values, the Freon map is

$$h(t) = t^{1-2c},$$

which diverges as $t \to 0$. As a result, the update direction is not uniformly bounded: arbitrarily small singular values can be mapped to arbitrarily large values, so the boundedness condition required by Theorem 2.3 cannot hold in general.

A standard way to remove this singularity is to introduce a Levenberg–Marquardt-style regularization inside the inverse power,

$$(G_k G_k^\top)^{-c} G_k \quad \longrightarrow \quad (G_k G_k^\top + \epsilon_k I)^{-c} G_k.$$

This is distinct from weight decay: the regularizer acts on the matrix being inverted rather than directly on the parameters. For singular values, the corresponding map is

$$h_k(t) = \frac{t}{(t^2 + \epsilon_k)^c},$$

which is bounded for every $\epsilon_k > 0$. A fixed ϵ is sufficient to remove the singularity, but the resulting convergence constants can depend on the scale of the gradients encountered along the trajectory. Instead, choosing ϵ_k proportional to $\|G_k\|_{\mathrm{op}}^2$ makes the map scale-free and gives a uniform lower bound on the ratio m_k/M_k appearing in Theorem 2.3.

Theorem E.2 (Convergence of regularized Freon)**.** *Let the assumptions of Theorem 2.3 hold with operator norm, let $c > 1/2$ and $\delta > 0$. For every iteration with $G_k \neq 0$, define $\epsilon_k = \delta \|G_k\|_{\mathrm{op}}^2$ and let the regularized* `Freon` *direction be induced by the singular value map*

$$p_k(\sigma) = \frac{\sigma}{(\sigma^2 + \epsilon_k)^c}.$$

Then the constants m_k, M_k of Theorem 2.3 may be chosen so that

$$\frac{m_k}{M_k} \geq \gamma(c, \delta, r) := \frac{(2c)^c}{r(2c-1)^{c-\frac{1}{2}}} \frac{\delta^{\,c-\frac{1}{2}}}{(1+\delta)^c} > 0,$$

where $r = \min(m, n)$ is the maximal rank. In particular,

$$\sum_{k=0}^{\infty} \left(\frac{m_k}{M_k} \right)^2 = \infty.$$

And thus, results of Theorem 2.3 apply.

Proof. Let singular values of G_k be σ_i. Since the primal norm is the operator norm, the norm of the spectral direction is its largest transformed singular value. Thus we may take

$$M_k = \max_i p_k(\sigma_i).$$

Moreover,

$$\langle \sigma, p_k(\sigma) \rangle = \sum_{i=1}^{r} \sigma_i p_k(\sigma_i).$$

Since the dual of the operator norm is the nuclear norm, $\|\sigma\|_1 = \|G_k\|_{\mathrm{nuc}}$, so the tightest admissible lower constant is

$$m_k = \frac{\langle \sigma, p_k(\sigma) \rangle}{\|\sigma\|_1}.$$

Therefore

$$\frac{m_k}{M_k} = \frac{\langle \sigma, p_k(\sigma) \rangle}{\|\sigma\|_1 \max_i p_k(\sigma_i)}. \tag{*}$$

We now lower-bound this ratio independently of the spectrum. Since all terms are nonnegative,

$$\langle \sigma, p_k(\sigma) \rangle \geq \sigma_1 p_k(\sigma_1),$$

while

$$\|\sigma\|_1 \le r\sigma_1.$$

Furthermore,

$$\max_i p_k(\sigma_i) \le \sup_{t>0} p_k(t).$$

Combining these inequalities with $(*)$ gives

$$\frac{m_k}{M_k} \ge \frac{1}{r}\frac{p_k(\sigma_1)}{\sup_{t>0} p_k(t)}. \tag{**}$$

For $c > 1/2$, the function

$$p_k(t) = t(t^2+\epsilon_k)^{-c}$$

has its unique positive maximum at

$$t_k^\star = \sqrt{\frac{\epsilon_k}{2c-1}},$$

since

$$p_k'(t) = (t^2+\epsilon_k)^{-c-1}\big(\epsilon_k + (1-2c)t^2\big).$$

Hence

$$\sup_{t>0} p_k(t) = p_k(t_k^\star).$$

Now use the gradient-scaled regularization

$$\epsilon_k = \delta\sigma_1^2.$$

At the largest singular value,

$$p_k(\sigma_1) = \frac{\sigma_1}{\big((1+\delta)\sigma_1^2\big)^c} = \sigma_1^{1-2c}(1+\delta)^{-c}.$$

On the other hand,

$$t_k^\star = \sigma_1\sqrt{\frac{\delta}{2c-1}},$$

and therefore

$$\begin{aligned} p_k(t_k^\star) &= \frac{\sqrt{\epsilon_k/(2c-1)}}{\big(\epsilon_k/(2c-1)+\epsilon_k\big)^c} \\ &= \epsilon_k^{\frac{1}{2}-c}\frac{(2c-1)^{c-\frac{1}{2}}}{(2c)^c} \\ &= \sigma_1^{1-2c}\delta^{\frac{1}{2}-c}\frac{(2c-1)^{c-\frac{1}{2}}}{(2c)^c}. \end{aligned}$$

Consequently,

$$\frac{p_k(\sigma_1)}{p_k(t_k^\star)} = \frac{(2c)^c}{(2c-1)^{c-\frac{1}{2}}}\frac{\delta^{\,c-\frac{1}{2}}}{(1+\delta)^c}.$$

Substitution into $(**)$ yields

$$\frac{m_k}{M_k} \ge \frac{(2c)^c}{r(2c-1)^{c-\frac{1}{2}}}\frac{\delta^{\,c-\frac{1}{2}}}{(1+\delta)^c} = \gamma(c,\delta,r) > 0.$$

Crucially, this constant depends only on (c,δ,r) and not on k, the scale of G_k, or the remainder of its singular spectrum. Thus

$$\sum_{k=0}^{\infty}\left(\frac{m_k}{M_k}\right)^2 \ge \sum_{k=0}^{\infty}\gamma(c,\delta,r)^2 = \infty.$$

□

Remark E.3. The gradient-scaled regularization is important only for obtaining a trajectory-independent constant. For fixed $\epsilon > 0$, $p(t) = t/(t^2+\epsilon)^c$ is already uniformly bounded, since $\sup_{t>0} p(t) < \infty$. A convergence argument can then also be obtained if the gradients remain uniformly bounded, for example when f is coercive and the iterates remain in the initial sublevel set. Choosing $\epsilon_k = \delta\|G_k\|_{\mathrm{op}}^2$ avoids this additional argument: the regularized map becomes scale-free and the lower bound $\gamma(c,\delta,r)$ holds independently of the trajectory.

E.3 EQUIVARIANCE OF FREON ($c = 1$)

Theorem E.4 (Equivariance of Freon ($c = 1$))**.** *For $W \in \mathbb{R}^d$, let ϕ_W be an arbitrary function (the neural network) with a layered structure: $W = (W_1, \dots, W_L)$ with $W_l \in \mathbb{R}^{n_l \times m_l}$. Let $f : \mathbb{R}^d \to \mathbb{R}$ be the differentiable loss function which depends on W only through the neural network ϕ_W, and let $G_l \in \mathbb{R}^{n_l \times m_l}$ be the gradient of f for layer l. Then updates of the form $W_l - \eta(G_l G_l^\top)^{-1} G_l$ are equivariant under symmetries of ϕ_W which act on layer matrices by*

$$W_l \mapsto L_l W_l R_l \quad \textit{for} \quad L_l \in \mathbb{R}^* O(n_l),\ R_l \in \mathbb{R}^* O(m_l)$$

where $\mathbb{R}^* O(m_l) = \{\lambda Q : \lambda \neq 0, Q \in O(m_l)\}$. Therefore, the optimization trajectory of ϕ_W is invariant under such symmetries.

If G_l has full row (column) rank, the above holds more generally for symmetries with $L_l \in GL_{n_l}(\mathbb{R})$ ($R_l \in GL_{m_l}(\mathbb{R})$ respectively).

Proof. Write such a symmetry g. We have

$$f(g \cdot W) = f(W) \quad \text{and} \quad g \cdot (W_1, \dots, W_L) = (L_1 W_1 R_1, \dots, L_L W_L R_L)$$

where $W = (W_1, \dots, W_L)$ and some $L_l \in GL_{n_l}(\mathbb{R})$ and $R_l \in GL_{m_l}(\mathbb{R})$, and we use colours to denote the matrix symmetries.

For $1 \leq l \leq L$, write $G_l[W] := \nabla_l f(W)$ where ∇_l denotes the gradient in the lth coordinate. By the chain rule we have

$$G_l[W] = \nabla_l f(W) = \nabla_{W_l} f(W) = \nabla_{W_l} f(g \cdot W) = L_l^\top \nabla_l f(g \cdot W) R_l^\top = L_l^\top G_l[g \cdot W] R_l^\top.$$

If $L_l \in O(n_l)$ and $R_l \in O(m_l)$, note that

$$\begin{aligned}(G_l[g \cdot W] G_l[g \cdot W]^\top)^{-1} G_l[g \cdot W] &= \left(L_l^{-\top} G_l[W] R_l^{-\top} R_l^{-1} G_l[W]^\top L_l^{-1}\right)^{-1} L_l^{-\top} G_l[W] R_l^{-\top} \\ &= L_l (G_l[W] G_l[W]^\top)^{-1} G_l[W] R_l \end{aligned} \tag{7}$$

where pseudoinverses are understood wherever necessary. The same holds more generally for $L_l \in \mathbb{R}^* O(n_l)$ and $R_l \in \mathbb{R}^* O(m_l)$, by cancelling out the scalar multiples which commute around all matrices.

If $G_l[W]$, or equivalently $G_l[g \cdot W]$, has full row rank, then we have a genuine inverse in equation 7, so we can take L_l out of the inverse for any $L_l \in GL_{n_l}(\mathbb{R})$, and obtain the same result for such L_l.

If $G_l[W]$ has full column rank, we can apply the same trick on the other side for $R_l \in GL_{m_l}(\mathbb{R})$ by first using the identity

$$(G_l[g \cdot W] G_l[g \cdot W]^\top)^{-1} G_l[g \cdot W] = G_l[g \cdot W] (G_l[g \cdot W]^\top G_l[g \cdot W])^{-1}.$$

Thus, under such symmetries, the updates are equivariant in weight space, and hence they are invariant in output space. □

Note E.5. These encapsulate ReLU scaling symmetries, node permutation symmetries, attention linear symmetries, etc, and most particularly transformations that are symmetries only on sampled data, even if the parameterization itself does not admit it globally.

Comparison with Newton's method. The preconditioner $GG^\top$ to the update step in Freon ($1/1$) can seem reminiscent to the Fisher information matrix used as preconditioner in Newton's method. It is well-known that Newton's steps are equivariant under *all* linear symmetries (not just layerwise ones). Here, we briefly clarify the connection and distinction between the two optimizers.

Consider a general deep network $\phi_W : \mathbb{R}^{n_{\text{in}}} \to \mathbb{R}^{n_{\text{out}}}$, with parameters W. As before, $f : \mathbb{R}^d \to \mathbb{R}$ is the loss function.

In the following we abstract away the multivalued data and consider ϕ_W implicitly evaluated solely at one input $x \in \mathbb{R}^{n_{\text{in}}}$, the general case follows by summing or averaging over such inputs. The loss f is viewed as a function of ϕ_W.

Define the quantities $g_\phi := \nabla_W \phi_W \in \mathbb{R}^{n_{\text{out}}}$, $h_f := \nabla_{\phi_W} \nabla^\top_{\phi_W} f \in \mathbb{R}^{n_{\text{out}} \times n_{\text{out}}}$, $g_f := \nabla_W f \in \mathbb{R}^d$, $J_\phi := \nabla_W \phi_W \in \mathbb{R}^{n_{\text{out}} \times d}$, $G_l := \nabla_{W_l} f \in \mathbb{R}^{n_l \times m_l}$, $J_l := \nabla_{W_l} \phi_W \in \mathbb{R}^{n_{\text{out}} \times (n_l \times m_l)}$.

By the chain rule $g_\phi = J_\phi^\top g_f$. For the usual l_2 or softmax-cross entropy losses, we have the probabilistic interpretation $f(W) = -\log p(y^* \mid x, W)$ where $y^* \in \mathbb{R}^{n_{\text{out}}}$ is the label associated to x. The Fisher information matrix (or GGN) is given by

$$\mathbb{E}_{p(y|x,W)}[-\nabla_W \nabla^\top_W \log p(y \mid x, W)] = J_\phi^\top h_\phi J_\phi.$$

A common approximation is the empirical Fisher (Kunstner et al., 2019):

$$\begin{aligned} \mathbb{E}_{p(y|x,W)}[-\nabla_W \nabla^\top_W \log p(y \mid x, W)] &= \mathbb{E}_{p(y|x,W)}[\nabla_W \log p(y \mid x, W) \nabla_W \log p(y \mid x, W)^\top] \\ &\approx \nabla_W \log p(y^* \mid x, W) \nabla_W \log p(y^* \mid x, W)^\top \\ &= g_f g_f^\top \\ &= J_\phi^\top g_f g_f^\top J_\phi \end{aligned} \tag{8}$$

This is the preconditioner used in Newton's method.

Now by the chain rule $G_l = \text{Mat}(J_l^\top g_f)$, where the Mat operator makes the output a matrix of the appropriate shape. The preconditioner in `Freon` $(1/1)$ is

$$G_l G_l^\top = \text{Mat}(J_l^\top g_f) \, \text{Mat}(g_f^\top J_l). \tag{9}$$

Compare equation 9 with equation 8.

F From Polar to Hölder Express

F.1 Fundamental Properties of the a/b-Method

We aim in this section to generalise the results in Amsel et al. (2026) to our setup. To this end, let $r, p \in \mathbb{N}$ and consider the following subsets of the polynomial ring $\mathbb{P}$

$$\mathbb{P}^{\text{odd}}(r) := \text{span}\{x^{2ir+1} : i \in \mathbb{N}\}, \tag{10}$$

$$\mathbb{P}^{\text{even}}(r) := \text{span}\{x^{2ir} : i \in \mathbb{N}\}, \tag{11}$$

$$\mathbb{P}^{\text{odd}}_p(r) := \{P \in \mathbb{P}^{\text{odd}}(r) : \deg(P) \leq 2rp + 1\}, \tag{12}$$

$$\mathbb{P}^{\text{even}}_p(r) := \{P \in \mathbb{P}^{\text{even}}(r) : \deg(P) \leq 2rp\}. \tag{13}$$

Moreover, for any $n, d \in \mathbb{N}$, we also define the following sets of rational functions

$$\mathfrak{R}(r, s) := \left\{ \frac{N}{D} : N \in \mathbb{P}^{\text{odd}}(r), D \in \mathbb{P}^{\text{even}}(s),\ D > 0 \right\}, \tag{14}$$

$$\mathfrak{R}_{n,d}(r, s) := \left\{ \frac{N}{D} : N \in \mathbb{P}^{odd}_n(r), D \in \mathbb{P}^{even}_d(s),\ D > 0 \right\}. \tag{15}$$

F.1.1 Existence and Characterization of the Optimal Approximants.

The set $\mathfrak{R}(r, s)$ considered above can be understood as a particular subset of generalized rational functions of the form

$$R(x) = \frac{\sum_{i=0}^n a_i g_i(x)}{\sum_{j=0}^d b_j h_j(x)}, \qquad \sum_{j=0}^d b_j h_j(x) > 0,$$

where $g_i(x) = x^{2ri+1}$ and $h_j(x) = x^{2sj}$. For these objects, unlike the classical polynomial case, the existence of an optimal approximant is not guaranteed in general. Furthermore, additional issues arise from the fact that the numerator and denominator of the approximants R can have a common factor, which leads to simplifications that might break the constraint $R \in \mathfrak{R}_{n,d}(r)$. This phenomenon is studied in the literature through the so called deficiency index $\mathfrak{d}$ which strictly depends on the class of functions we want to approximate. In this regards, let us define the following class of functions

Definition F.1. Let $a, b \in \mathbb{R}$. We define the set of 0-deficiency continuous functions $C^0_{\mathfrak{d}=0}([a,b]) \subset C^0([a,b])$ as

$$C^0_{\mathfrak{d}=0}([a,b]) := \{f \in C^0([a,b]) : \mathfrak{d}_f = 0\}.$$

We can prove the following result.

Lemma F.2. *Let $r \in \mathbb{N}$ and consider $\mathfrak{R}(r,r)$ as in equation 15. Moreover, let $0 < a \leq b \in \mathbb{R}$ and define*

$$\mathscr{C}^0([a,b]) := \left\{ f \in C^0([a,b]) : \frac{f(x^{\frac{1}{2r}})}{x^{\frac{1}{2r}}} \in C^0_{\mathfrak{d}=0}([a,b]) \right\}.$$

Then, for every $f \in \mathscr{C}^0([a,b])$, there exists $R^* \in \mathfrak{R}_{n,d}(r,r)$ such that

$$\|f - R^*\|_{C^0([a,b])} \leq \|f - R\|_{C^0([a,b])}, \qquad \forall R \in \mathfrak{R}_{n,d}(r).$$

Proof. Let us observe that any $R \in \mathfrak{R}_{n,d}(r,r)$ can be rewritten as

$$R(x) = \frac{\sum_{i=0}^n a_i x^{2ri+1}}{\sum_{j=0}^d b_j x^{2rj}} = x \frac{\sum_{i=0}^n a_i x^{2ri}}{\sum_{j=0}^d b_j x^{2rj}} =: x \frac{P(x^{2r})}{Q(x^{2r})} \tag{16}$$

where $P \in \mathbb{P}_n$ and $Q \in \mathbb{P}_d$. By Achieser (1992), we know that for every $g \in C^0([a,b])$, there always exist $P^* \in \mathbb{P}_n$, $Q^* \in \mathbb{P}_d$ such that $\frac{P^*}{Q^*}$ best approximates g. Then, since $0 < a \leq b$, considering $f \in C^0([a,b])$ and setting

$$\widetilde{g}(x) = \frac{f\left(x^{\frac{1}{2r}}\right)}{x^{\frac{1}{2r}}}, \tag{17}$$

we get a best approximation candidate $R^*(x) = x\frac{P^*(x)}{Q^*(x)}$ for f.

To conclude, by Definition F.1, when $f \in \mathscr{C}^0([a,b])$, we have $\mathfrak{d}_{\widetilde{g}} = 0$ and we can ensure $R^* \in \mathfrak{R}_{n,d}(r,r)$. □

Finally, mimicking the polynomial case, we are interested in providing a characterization of the optimal approximants. The following holds.

Lemma F.3. *Let us assume the same setup as in Lemma F.2. Then, for every $f \in \mathscr{C}^0([a,b])$, any $R^* \in \mathfrak{R}_{n,d}(r,r)$ such that*

$$\#\{x \in [a,b] : |f(x) - R^*(x)| = \|f - R^*\|_{C^0}\} \geq n + d + 2 \tag{18}$$

is an optimal approximant of f.

Proof. Given $f \in \mathscr{C}^0([a,b])$, exploiting equation 16, any optimal $R^* \in \mathfrak{R}_{n,d}(r,r)$ fulfils the same properties that P^*/Q^* satisfies approximating $\widetilde{g}$. Then, by (Loeb, 1966, Theorem 1), we have the thesis. □

F.1.2 Optimality

We will now address the problem of the optimality of the iterations. To start with, the following closure property holds.

Lemma F.4. *Let $r \in \mathbb{N}$. Then, $\mathbb{P}^{\mathrm{odd}}(r)$, $\mathbb{P}^{\mathrm{even}}(r)$ and $\mathfrak{R}(r,r)$ are closed with respect to composition.*

Proof. Let us consider $\mathbb{P}^{\mathrm{odd}}(r)$. By induction on the monomials, it suffices to show that

$$p = p_1 \circ p_2 \in \mathbb{P}^{\mathrm{odd}}(r)$$

for $p_1 = x^{2lr+1}$, $p_2 = x^{2nr+1} + x^{2mr+1}$. Recalling now Newton's binomial formula, it holds

$$\begin{aligned} p(x) &= \sum_{i=0}^{2lr+1} \left(x^{2nr+1}\right)^i \left(x^{2mb+1}\right)^{2lb+1-i} \\ &= \sum_{i=0}^{2lr+1} x^{2nri+i+2m(2lr+1-i)r+k-i} \\ &= \sum_{i=0}^{2lr+1} x^{2[nri+m(2lr+1-i)+2l]r+1}. \end{aligned}$$

Then, we conclude by observing that, since $nri + m(2lr + 1 - i) + 2l \in \mathbb{N}$, every term in the above expansion is in $\mathbb{P}^{\mathrm{odd}}(r)$. Exploiting an analogous argument as above, we can also prove the following

1. if $p_1, p_2 \in \mathbb{P}^{\mathrm{even}}(r)$, then $p \in \mathbb{P}^{\mathrm{even}}(r)$ (which implies the closure of $\mathbb{P}^{\mathrm{even}}(r)$ as claimed);
2. if $p_1 \in \mathbb{P}^{\mathrm{odd}}(r)$, $p_2 \in \mathbb{P}^{\mathrm{even}}(r)$ (or vice versa), then $p \in \mathbb{P}^{\mathrm{even}}(r)$.

Let us consider q_1, $q_2 \in \mathfrak{R}_{n,d}(r, r)$. We have

$$\begin{aligned} D_1^{2nr+1}(x)(N_2 \circ q_1)(x) &= \sum_{i=1}^{n} c_i N_1^{2ir+1}(x) D_1^{2r(n-i)}(x), \\ D_1^{2dr}(x)(D_2 \circ q_1)(x) &= \sum_{i=1}^{d} e_i N_1^{2ir}(x) D_1^{2r(d-i)}(x). \end{aligned}$$

The arguments above implies then that $N_1^{2ir+1} \in \mathbb{P}^{\mathrm{odd}}(r)$ while $D_1^{2r(n-i)}$, N_1^{2ir}, $D_1^{2r(d-i)} \in \mathbb{P}^{\mathrm{even}}(r)$. Finally, recalling that the product of even polynomials is even and the product of an even polynomial and an odd one is odd, since only r-powers monomial appears, we immediately get $N_2 \circ q_1 \in \mathfrak{R}(r, r)$ and $D_1^{2dr}(D_2 \circ q_1) \in \mathbb{P}^{\mathrm{even}}(r)$, so that

$$(q_1 \circ q_2)(x) = \frac{(N_2 \circ q_1)(x)}{D_1^{2(d-n)r-1}(x)(D_2 \circ q_1)(x)} \in \mathfrak{R}_{n,d}(r, r),$$

for all n, $d \in \mathbb{N}$. □

We can then generalise (Amsel et al., 2026, Theorem 3.1) with the following optimality result.

Theorem F.5. *Let $l_1 = l, u_1 = u \in (0, 1)$, $r, T \in \mathbb{N}$, $n, d \in \mathbb{N}$ and consider*

$$R_{t+1} = \operatorname*{argmin}_{R \in \mathfrak{R}_{n,d}(r,r)} \max_{x \in [l_t, u_t]} |1 - R(x)|, \qquad \forall t \in [1, T],$$

with $l_{t+1} = \min_{x \in [l_t, u_t]} R_t(x)$, $u_{t+1} = \max_{x \in [l_t, u_t]} R_t(x)$. Then, $R^* := R_T \circ R_{T-1} \circ \cdots \circ R_1$ is optimal and

$$\max_{x \in [l_T, u_T]} |1 - R^*(x)| = 1 - l_T.$$

Moreover,

$$\begin{cases} l_{t+1} = R_t(l_t), \\ u_{t+1} = 2 - R_t(l_t), \\ \max\limits_{x \in [l_t, u_t]} |1 - R_t(x)| = 1 - l_t, \end{cases} \qquad \forall t \in [1, T]. \tag{19}$$

Proof. Let us first observe that, by Wall (1948), we have that the Hankel determinants of $x \mapsto x^\alpha$, $\alpha \notin \mathbb{N}$, are all non-zero. Thus, by (Gragg, 1972, Corollary 2), $f(x) \equiv 1 \in \mathscr{C}^0([a, b])$, so that, thanks to Lemma F.2, R_t (and consequently R^*) is well defined.

We will now adapt the argument used in (Amsel et al., 2026, Theorem 3.1), and split our discussion into two parts: first, we prove equation 19 by showing that l_t is a minimizer of R_t; second, we argue on the optimality of R^*.

l_t **is a minimizer of** R_t**.** By direct computation, we have

$$\begin{aligned} R_t'(x) &= \frac{N'(x)D(x) - D'(x)P(x)}{D^2(x)} \\ &= \frac{\left(\sum_{i=0}^n a_i(2ri+1)x^{2ri}\right)\left(\sum_{j=0}^d b_j x^{2rj}\right) - \left(\sum_{j=0}^d 2rjb_j x^{2rj-1}\right)\left(\sum_{i=0}^n a_i x^{2ri+1}\right)}{D^2(x)} \\ &= \frac{\sum_{i=0}^n \sum_{j=0}^d (2r(i-j)+1)a_i b_j x^{2r(i+j)}}{D^2(x)} = \frac{\widetilde{N}(x^{2r})}{\widetilde{D}(x^{2r})} \end{aligned}$$

where $\widetilde{N} \in \mathbb{P}_n$, $\widetilde{D} \in \mathbb{P}_d$. Thus, R_t has at most $2(n+d)$ extremal points, of which, due to the symmetry of $\widetilde{N}(x^{2r})$, at most $n+d$ can be contained in $[l_t, u_t] \subset [0,1]$. Exploiting now Lemma F.3, we know that, being R_t an optimal approximant, there exist at least $n+d+2$ extremal points for $|1-R_t|$ on $[l_t, u_t]$. In particular, we can conclude that the limiting points l_t and u_t need to be extremal points.

Since $R_t(0) = 0$, being R_t optimal, we have $R_t > 0$ on $[l_t, u_t]$ otherwise, $R_t = 0$ would be a better approximant. In particular, there exists $\hat{x} \in [0, l_t]$ such that $R_t'(\hat{x}) > 0$. If we suppose now that l_t is a maximum of R_t, then $R_t'(l_t) \leq 0$ and, by the intermediate value theorem, there must exist $\overline{x} \in [\hat{x}, l_t]$ such that $R_t(\overline{x}) = 0$. This is, however, a contradiction since all the nonnegative extremal points of R_t are in $[l_t, u_t]$ so that l_t must be a minimum of R_t. Finally, by direct computation, equation 19 immediately follows.

Optimality of R^***.** We will argue by induction. Let us observe that Lemma F.2 and the previous paragraph ensure the base step. Then, let us assume that the thesis holds for $T-1$ and consider $f : [l,u] \to \mathbb{R}$ given by $x \mapsto \overline{R}_{T-1} \circ \cdots \circ \overline{R}_1(x)$, for some $\overline{R}_1, ...\overline{R}_{T-1} \in \mathfrak{R}_{n,d}(r,r)$. Moreover, letting $[a,b]$ be the image of $[l,u]$ through f, for any constant c we immediately have

$$\max_{x\in[l,u]} |1 - cf(x)| = \max\{1 - ca, cb - 1\}. \tag{20}$$

Let us suppose now that $\frac{a}{b} > \frac{l_t}{u_t}$ then the following holds

$$\frac{a}{b} > \frac{l_t}{2 - l_t} \quad \Leftrightarrow \quad 1 - l_t > \frac{b-a}{a+b}.$$

Then, since by the previous discussion $l_t = \min_{x\in[l,u]} R_{T-1} \circ \cdots \circ R_1(x)$, by choosing $\overline{c} = \frac{2}{a+b}$ in equation 20, the above implies

$$\max_{x\in[l,u]} |1 - R_{T-1} \circ \cdots \circ R_1(x)| > \max_{x\in[l,u]} |1 - \overline{c} f(x)|$$

contradicting the inductive hypothesis. Finally, since $\frac{a}{b} < \frac{l_t}{u_t}$, for any $\overline{R}_T \in \mathfrak{R}_{n,d}(r,r)$, we have

$$\begin{aligned} \max_{x\in[l,u]} |1 - \overline{R}_T(f(x))| &\geq \min_{R\in\mathfrak{R}_{n,d}(r,r)} \max_{x\in[a,b]} |1 - R(x)| \\ &\overset{(*)}{=} \min_{R\in\mathfrak{R}_{n,d}(r,r)} \max_{x\in[a/b,1]} |1 - R(x)| \\ &\geq \min_{R\in\mathfrak{R}_{n,d}(r,r)} \max_{x\in[l_t/u_t,1]} |1 - R(x)| \\ &= \min_{R\in\mathfrak{R}_{n,d}(r,r)} \max_{x\in[l_t,u_t]} |1 - R(x)| \\ &= \min_{R\in\mathfrak{R}_{n,d}(r,r)} \max_{x\in[l,u]} |1 - R(R_{T-1} \circ \cdots \circ R_1(x))| \\ &= \max_{x\in[l,u]} |1 - R^*(x)| \end{aligned}$$

where $(*)$ is justified by the invariance of $\mathfrak{R}_{n,d}(r,r)$ under rescaling. □

Remark F.6. We emphasize that, since $\mathbb{P}_n(r) \cong \mathfrak{R}_{n,0}(r,r)$, the same conclusions of Theorem F.5 holds if we replace $\mathfrak{R}_{n,d}(r,r)$ with $\mathbb{P}_n(r)$.

Finally, we are ready for the following

Proof of Theorem 2.7 - Optimality. Let us consider $R_t \in \mathfrak{R}(r,r)$ and define $\widetilde{R}_t(X) := R_t(X\mathcal{G}^\dagger)\mathcal{G}$. It is immediate to verify that, if R_t is an optimal approximant of $f \equiv \mathrm{Id}$ according to Theorem F.5, then $\widetilde{R}_t$ is a best approximant of $\mathcal{G}$. More generally, considering R^* as in Theorem F.5, let us observe that

$$R^*(x) = \widetilde{R}_T(\widetilde{R}_{T-1}(\cdots \widetilde{R}_1(x)))\mathcal{G}^\dagger.$$

Then, it holds $O_T(X) = R^*(X\mathcal{G}^\dagger)\mathcal{G}$, which implies that $O_T(G)$ is the best approximant of $\mathcal{G}$ for every $T \in \mathbb{N}$. □

F.1.3 Approximation Errors

We will now provide an error estimate for the proposed method.

To start with, let us point out that, calling ${}_2F_1$ the generalized hypergeometric function given by

$${}_2F_1(\alpha,\beta,\gamma,\xi) = \sum_{n=0}^{\infty} \frac{(\alpha)_n(\beta)_n}{n!(\gamma)_n}\xi^n,$$

with $(\alpha)_n := \alpha(\alpha+1)\cdots(\alpha+n-1)$, by (Abramowitz & Stegun, 1964, Chapter 15, 15.1.8), it holds

$$(1-x)^{-\frac{1}{2r}} = {}_2F_1\left(\frac{1}{2r}, 1, 1, x\right).$$

Then, we immediately get the following generalisation of (Kenney & Laub, 1991, Theorem 3).

Lemma F.7. *Let $r \in \mathbb{N}$ and consider $P(x) \in \mathbb{P}_n^{even}(r)$, $Q(x) \in \mathbb{P}_d^{even}(r)$ such that $h(x) := \frac{P(x)}{Q(x)}$ is the $[n/d]$ Padé-approximant of $(1-x)^{-\frac{1}{2r}}$. Then, for every $t \in \mathbb{N}$, setting*

$$x_{t+1} = x_t h(1 - x_t^{2r}),$$

we have

$$|1 - x_t^{2r}| \leq |1 - x_0^{2r}|^{(n+d+1)^\top}.$$

Finally, the following approximation result holds (concluding the proof of Theorem 2.7).

Theorem F.8. *Let $T \geq 0$, $l \in (0,1)$, $\sigma_{\min}(G) \geq l$ and consider R^* as in Theorem 2.7. Then, X_T given by equation 23 satisfies the following estimate bound*

$$\|(GG^\top)^{-\frac{a}{b}}G - O_T\|_2 \leq \frac{|1-l^b|^{(n+d+1)^T}}{l^{\frac{2a-b}{b}}}.$$

Proof. First, let us observe that, setting $Z_t := (GG^\top)^{\frac{a}{b}-\frac{1}{2}}O_t$, $t \in [1,T]$, it holds

$$\begin{aligned} l^{\frac{2a-b}{b}} \left\|(GG^\top)^{-\frac{a}{b}}G - O_T\right\|_2 &\leq \sigma_{min}(G)^{\frac{2a-b}{b}} \left\|(GG^\top)^{-\frac{a}{b}}G - O_T\right\|_2 \\ &\leq \left\|(GG^\top)^{\frac{a}{b}-\frac{1}{2}}\left((GG^\top)^{-\frac{a}{b}}G - O_T\right)\right\|_2 \\ &= \|\mathrm{sign}(G) - Z_t\|_2 \end{aligned}$$

It is fundamental to observe that, by direct computation, we have $X_t \in \mathfrak{R}\left(\frac{b}{2},\frac{b}{2}\right)$.

Let now h be the $[n/d]$ Padé-approximant of $(1-x)^{-\frac{1}{b}}$ and $\widetilde{R}(x) = xh(1-x^b)$. Moreover, by direct computation, we can immediately check that $h(1-x^{2r}) = \frac{P(x^b)}{Q(x^b)}$ for some $P \in \mathbb{P}_n$, $Q \in \mathbb{P}_d$. Thus, $\widetilde{R} \in \mathfrak{R}_{n,d}(\frac{b}{2},\frac{b}{2})$, and, by Lemma F.4, it is well defined

$$f := \widetilde{R} \underbrace{\circ \cdots \circ}_{T \text{ times}} \widetilde{R} \in \mathfrak{R}(\frac{b}{2},\frac{b}{2}).$$

Finally, considering R^* as in Theorem F.5, by Lemma F.7, we have

$$\begin{aligned}\|\mathrm{sign}(G) - Z_T\|_2 &\leq \max_{x\in[l,1]} |1 - R^\star(x)| \leq \max_{x\in[l,1]} |1 - f(x)| \\ &\leq \max_{x\in[l,1]} \frac{|1 - f(x)^b|^{(n+1)^T}}{\sum_{i=0}^{b-1} f(x)^i} \leq \max_{x\in[l,1]} \frac{|1 - x^b|^{(n+d+1)^T}}{\sum_{i=0}^{b-1} f(x)^i} \\ &\leq |1 - l^b|^{(n+d+1)^T},\end{aligned}$$

and we can conclude. □

Remark F.9. Let us point out that using a polynomial approximant the above estimate reduces to

$$\|(GG^\top)^{-\frac{a}{b}} G - O_T\|_2 \leq \frac{|1 - l^b|^{(n+1)^T}}{l^{\frac{2a-b}{b}}}.$$

F.2 Accuracy of approximations

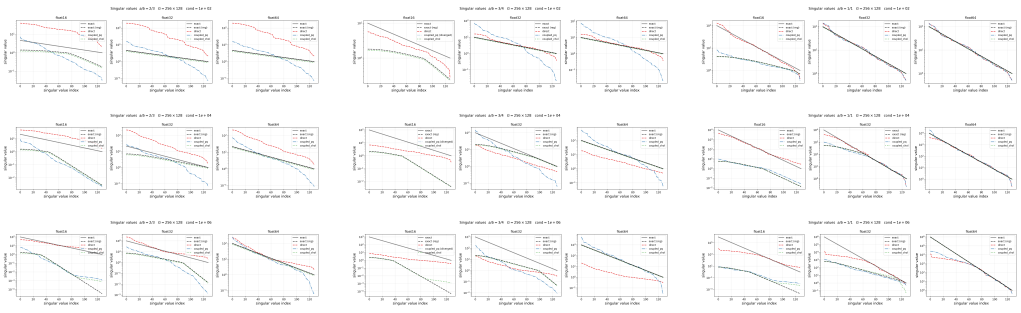


Figure F.1: Output singular values of three methods for computing $(GG^\top)^{-a/b}G$ on a synthetic 256×128 matrix $G = U\Sigma V^\top$, where the singular values of Σ are spaced logarithmically between 1 and κ^{-1} to give condition number κ. Columns correspond to exponents $a/b \in \{2/3,\ 3/4,\ 1\}$; rows to condition numbers $\kappa \in \{10^2,\ 10^4,\ 10^6\}$. Each panel shows three sub-panels for float16, float32, and float64 (left to right). Within each sub-panel the solid black line is the exact target $\sigma_i(G)^{1-2a/b}$ (sorted descending by index), the dashed black line is the regularized target $\sigma_i(G)\,(\sigma_i(G)^2 + \varepsilon)^{-a/b}$ where ε is the dtype-dependent regularization threshold scaled by the spectral norm of G, and the coloured curves are the output singular values of `direct` (red, polynomial Newton–Schulz applied directly to G), `coupled_pq` (blue, coupled polynomial iteration tracking G and its Gram matrix), and `coupled_chol` (green, rational Newton–Schulz via Cholesky tracking). All methods use 10 iterations. A method that diverges (NaN/Inf output) is shown as a flat phantom line in the legend only. `coupled_chol` remains stable across all conditions and precisions, while `direct` and `coupled_pq` degrade or diverge at high condition numbers in low precision.

F.3 Instability of Polynomial Iterations

Figure F.2 compares three iterative methods for computing $(GG^\top)^{-a/b}G$ across matrix sizes, condition numbers, and floating-point precisions. Grey cells indicate divergence (NaN or Inf output). Both `direct` and `coupled_pq` exhibit systematic divergence at high condition numbers, particularly in `float16` and `bfloat16`, with divergence spreading to `float32` at extreme conditioning. Regularization delays but does not eliminate this behaviour. In contrast, `coupled_chol`, remains numerically stable across all tested conditions and dtypes, achieving near-machine-precision accuracy throughout. This stability comes without any regularization: the Cholesky structure enforces positive definiteness at every step, preventing the eigenvalue collapse that causes the other methods to diverge.

F.4 Iterative Methods for $(GG^\top)^{-a/b}\,G$

Throughout this section, let $G \in \mathbb{R}^{m\times n}$ $(m \leq n)$ denote the (Frobenius-normalised) gradient matrix. Let P_t be a step-varying degree-2 polynomial with Remez-approximated coefficients, and

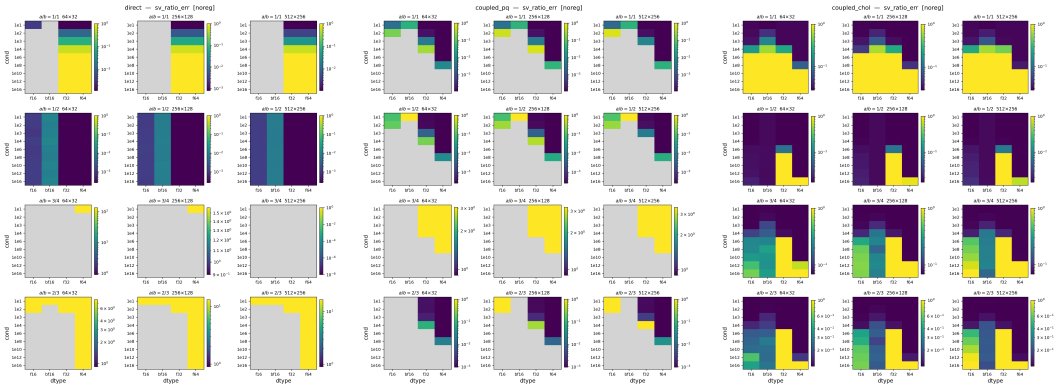


Figure F.2: Singular-value ratio error $\varepsilon_{\mathrm{sv}} = \max_i |\sigma_i(\hat{Z}) \cdot \sigma_i(G)^{(2a-b)/b} - 1|$ without regularization for three iterative methods computing $(GG^\top)^{-a/b}G$: `direct` (left), `coupled_pq` (centre), and `coupled_chol` (right). Each panel is a 4×3 grid of heatmaps indexed by exponent $a/b \in 1/1, 1/2, 3/4, 2/3$ (rows) and matrix size $(m, n) \in 64 \times 32, 256 \times 128, 512 \times 256$ (columns); within each cell the y-axis encodes condition number $\kappa \in 10^1, 10^2, \ldots, 10^{12}, 10^{16}$ (increasing downward) and the x-axis encodes floating-point type (f16, bf16, f32, f64, increasing precision left to right). Test matrices are constructed as $G = U\Sigma V^\top$, where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are independent Haar-distributed orthogonal matrices obtained by QR-decomposing Gaussian random matrices in `float64`, and $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$ with $k = \min(m, n)$ singular values spaced logarithmically from 1 to κ^{-1}; each matrix is then cast to the target dtype before the iterative method is applied. The error metric equals zero for a perfect computation and is invariant to the overall scale of G; no regularization is applied, so grey cells indicate that the method produced non-finite values for that configuration. `direct` and `coupled_pq` each perform 40 steps of a degree-(p, q) polynomial iteration whose coefficients are precomputed by Zolotarev-optimal minimax approximation on $[\ell, 1]$ with $\ell = 10^{-16}$, run entirely in the target dtype. `coupled_chol` performs 25 steps of a rational Remez iteration with Cholesky-based stabilization. Colour encodes error on a logarithmic scale (viridis); darker shades indicate smaller error. Error grows with condition number and worsens at reduced precision, while `coupled_pq` achieves consistently lower error than `direct` across all settings. Most notably `coupled_chol` remains stable at all condition numbers and precisions.

let $k \in \mathbb{Z}^+$ satisfy $P(x)^k x^{ka/b} \approx 1$ on the fitting interval. All methods target the fixed point $X_\infty = (GG^\top)^{-a/b} G$.

A critical note on numerical stability: It is important to state upfront that, as identified by Higham Higham (2008), and numerically illustrated in Appendix F.3, most of the polynomial and coupled iterations detailed below fail to converge in finite-precision arithmetic due to severe numerical stability issues (e.g., condition-number squaring and diverging coupled sequences). In practice, *only the final method presented* (the **Coupled-Chol Iteration**) exhibits the forward stability required to reliably converge. The preceding methods are documented primarily for their theoretical connections and historical context.

For cost accounting, we distinguish ***G-MMs*** (full $m \times n$ gradient products, $O(m^2 n)$) from ***S-MMs*** ($m \times m$ square products, $O(m^3)$) and ***QRs*** (thin QR of a $2m \times m$ block, $O(m^3)$). Integer powers A^k require $\lceil \log_2 k \rceil$ S-MMs via binary exponentiation.

F.4.1 Direct Iteration

The direct method iterates O_t without maintaining separate state for the polynomial argument A_t. Instead, A_t is recomputed at each step from the current iterate. Three cases arise depending on a/b:

General case ($a/b \notin \{1/2, 1\}$). A frozen Gram power $(GG^\top)^{k(a/b-1/2)}$ is precomputed once; each step forms:

$$\begin{cases} O_{t+1} = R_t(A_t)\, O_t, & O_0 = G, \\ A_t = (O_t O_t^\top)^{k/2}\, (GG^\top)^{k(a/b\,-\,1/2)}. \end{cases} \tag{21}$$

Polar case ($a/b = 1/2$). The frozen factor vanishes ($k(a/b - 1/2) = 0$) and $k/2 = 1$, simplifying the argument to the plain Gram matrix of the current iterate:

$$\begin{cases} O_{t+1} = R_t(A_t)\, O_t, \quad O_0 = G, \\ A_t = O_t O_t^\top. \end{cases} \tag{22}$$

Freon case ($a/b = 1$, $k = 1$). The general formula equation 21 would require an expensive matrix square root $(X_t X_t^\top)^{1/2}$ at every step. Instead, A_t is defined via a mixed product with the *frozen* initial transpose $G^\top$:

$$\begin{cases} O_{t+1} = R_t(A_t)\, O_t, \quad O_0 = G, \\ A_t = O_t\, G_0^\top. \end{cases} \tag{23}$$

Using $G_0^\top$ avoids both the matrix square root and condition-number squaring, since $\kappa(O_t G^\top) \leq \kappa(O_t)\kappa(G)$ compared to $\kappa(O_t O_t^\top) = \kappa(O_t)^2$.

In all three cases, convergence to $O_\infty O_\infty^\top = (GG^\top)^{1-2a/b}$ ensures $A_\infty = I$ (the fixed point of P).

Classical connection. For $a/b = 1/2$ and degree-1 P_t, the update reduces to the Newton–Schulz iteration for the polar decomposition Higham (2008). The higher power inside the polynomial allows for general $a/b \neq 1/2$.

Complexity.

- *Init:* 1 G-MM; $\lceil \log_2(k(a/b - 1/2)) \rceil$ S-MMs (zero if $a/b = 1/2$).
- *Per step:* 2 G-MMs; $(1 + \lceil \log_2(k/2) \rceil + \mathbf{1}_{[a/b \neq 1/2]})$ S-MMs.

F.4.2 M-Accumulator Iteration

Instead of iterating O_t directly, this method maintains a preconditioner $M_t \in \mathbb{R}^{m \times m}$, applied to the frozen gradient G only upon convergence:

$$\begin{cases} M_{t+1} = R_t(A_t)\, M_t, \qquad M_0 = I, \\ A_t = M_t^k\, (GG^\top)^{ka/b}. \end{cases} \tag{24}$$

The final result is $O_T = M_T G$. At convergence, $M_\infty = (GG^\top)^{-a/b}$ and $A_\infty = I$.

Classical connection. For $a/b = 1/2$, this is the uncoupled polynomial analogue of the Denman–Beavers iteration for the matrix square root Higham (2008). This is also the method of (Zhang et al., 2026).

Complexity.

- *Init:* 1 G-MM; $\lceil \log_2(ka/b) \rceil$ S-MMs.
- *Per step:* 0 G-MMs; $(\lceil \log_2 k \rceil + 3)$ S-MMs.
- *Final:* 1 G-MM ($M_T G_0$).

M-accumulator is the cheapest per-step method when $n \gg m$, deferring the large-matrix product to the end.

F.4.3 COUPLED ITERATION

The coupled method eliminates the per-step recomputation of A_t by tracking it as its own state. Because $P_t(A_t)$ commutes with A_t, the system closes exactly:

$$\begin{cases} O_{t+1} = R_t(A_t)\, O_t, & O_0 = G, \\ A_{t+1} = R_t(A_t)^k\, A_t, & A_0 = (GG^\top)^{ka/b}. \end{cases} \tag{25}$$

Classical connection. Tracking the argument matrix alongside the iterand echoes Schulz-type coupled iterations for matrix inversion Higham (2008).

Complexity.

- *Init:* 1 G-MM; $\lceil \log_2(ka/b) \rceil$ S-MMs.
- *Per step:* 1 G-MM; $(\lceil \log_2 k \rceil + 2)$ S-MMs.

Compared to M-accumulator, this adds one G-MM per step but saves one S-MM, making it favorable only when $m \approx n$.

F.4.4 COUPLED-ab ITERATION

A variant of coupled iteration where A_t initializes at the raw Gram matrix $GG^\top$, avoiding the fractional matrix power during initialization:

$$\begin{cases} O_{t+1} = R_t(A_t)^a\, O_t, & O_0 = G, \\ A_{t+1} = R_t(A_t)^b\, A_t, & A_0 = GG^\top. \end{cases} \tag{26}$$

The invariant $R(x)^b x \approx 1$ ensures $A_t \to I$ while $O_t \to (GG^\top)^{-a/b} G$.

Complexity.

- *Init:* 1 G-MM; 0 S-MMs.
- *Per step:* 1 G-MM; $(2 + \lceil \log_2 a \rceil + \lceil \log_2 b \rceil)$ S-MMs.

F.4.5 DUAL-ab ITERATION

This method maintains a factored representation $A_t = X_t Y_t$ to avoid forming $GG^\top$ explicitly at each step:

$$\begin{cases} A_t = O_t Y_t, \quad R_t = R_t(A_t), & \\ O_{t+1} = R_t^a\, O_t, & O_0 = G, \\ Y_{t+1} = Y_t\, R_t^{b-a}, & Y_0 = G^\top. \end{cases} \tag{27}$$

A balancing step $O_t \leftarrow \gamma_t O_t$, $Y_t \leftarrow Y_t/\gamma_t$ with $\gamma_t = \sqrt{\|Y_t\|_F / \|X_t\|_F}$ prevents mismatched sequence growth.

Classical connection. The factored form averts condition-number squaring, mirroring product formulas for evaluating matrix functions of products Higham (2008).

Complexity.

- *Init:* 0 G-MMs (transpose only).
- *Per step:* 3 G-MMs; $(1 + \lceil \log_2 a \rceil + \lceil \log_2(b-a) \rceil)$ S-MMs.

F.4.6 COUPLED-DUAL ITERATION

This method runs two coupled masters in parallel for exponents a/b and $1 - a/b$. A cross-stabilizer $E_t = I - X_t Y_t \to 0$ corrects the slave iterates directly, avoiding fractional matrix powers inside the loop:

$$\begin{cases} R_A = R_t(A_t), \quad R_B = R_t(B_t), \quad E_t = I - O_t Y_t, & \\ O_{t+1} = R_A\, O_t + \gamma\, E_t O_t, & O_0 = G, \\ Y_{t+1} = Y_t\, R_B + \gamma\, Y_t E_t, & Y_0 = G^\top, \\ A_{t+1} = R_A^k\, A_t, & A_0 = (GG^\top)^{ka/b}, \\ B_{t+1} = R_B^k\, B_t, & B_0 = (GG^\top)^{k(b-a)/b}, \end{cases} \tag{28}$$

where $\gamma > 0$ is a scalar hyperparameter.

Complexity.

- *Init:* 1 G-MM; $\lceil \log_2(ka/b) \rceil + \lceil \log_2(k(b-a)/b) \rceil$ S-MMs.
- *Per step:* 3 G-MMs; $(4 + 2\lceil \log_2 k \rceil)$ S-MMs.

F.4.7 RATIONAL-CHOL (QDWH-STYLE) ITERATION

Instead of polynomials, this method applies a rational function $R_t(x) = (\alpha_t + \beta_t x)/(1 + \delta_t x)$ targeting $x^{-1/b}$. It maintains a preconditioner $M_t \to (GG^\top)^{-1/b}$:

$$M_{t+1} = R_t(M_t^b\, GG^\top)\, M_t, \quad M_0 = I.$$

Writing $b = 2r$ and taking a thin QR $LL^\top = GG^\top$, we define $Y_t = M_t^r L$. A block QR avoids forming $(I + \delta_t Y_t Y_t^\top)^{-1}$:

$$\begin{cases} K_t = \begin{bmatrix} \sqrt{\delta_t}\, Y_t^\top \\ I_m \end{bmatrix}, \quad [Q_1;\, Q_2]\, R = K_t \text{ (thin QR)}, \\ M_{t+1} = \gamma_t\, M_t + (\alpha_t - \gamma_t)\, Q_2\, Q_2^\top M_t, \end{cases} \tag{29}$$

where $\gamma_t = \beta_t/\delta_t$. The final result is $M_T^a\, G$.

Classical connection. For $a/b = 1/2$, this is the QDWH algorithm for polar decomposition, using Zolotarev approximants via block QRs to avoid condition-number squaring Nakatsukasa & Freund (2016).

Complexity.

- *Init:* 1 thin QR of $G^\top$ (equivalent to 1 G-MM).
- *Per step:* 1 QR; $(3 + \lceil \log_2 r \rceil)$ S-MMs.
- *Final:* $\lceil \log_2 a \rceil$ S-MMs; 1 G-MM ($M_T^a G$).

F.4.8 COUPLED-CHOL ITERATION

This method merges Cholesky-factor tracking (from Coupled-ab) with the block QR technique (from Rational-Chol). It tracks a Cholesky-like factor $L_t \in \mathbb{R}^{m \times m}$ ($L_t L_t^\top \to I$) directly alongside an

accumulator $C_t \in \mathbb{R}^{m\times m}$:

$$\begin{cases} W_t = R_t(L_t L_t^\top) & \text{(via block QR on } L_t\text{)}, \\ L_{t+1} = W_t^r\, L_t, & L_0 = L \text{ (thin QR factor of } G^\top\text{)}, \\ C_{t+1} = W_t\, C_t, & C_0 = I. \end{cases} \tag{30}$$

The final result is $C_T^a\, G$.

Classical connection. This directly generalizes QDWH. Instead of exponentiating the accumulated preconditioner M_t^r, it exponents only the well-conditioned, single-step update W_t^r, improving forward-stability.

Complexity.

- *Init:* 1 thin QR of $G^\top$.
- *Per step:* 1 QR; $(3 + \lceil \log_2 r \rceil)$ S-MMs.
- *Final:* $\lceil \log_2 a \rceil$ S-MMs; 1 G-MM ($C_T^a G$).

Costs match Rational-Chol, but the numerical stability is superior because $W_t \approx I$.

F.4.9 Remez Visualisations

F.5 Picking Parameters for Freon Iterations: Cushion and l

The rational Newton–Schulz iteration requires two hyperparameters: a cushion factor δ that stabilises the Remez denominator coefficients, and a lower spectral bound l that determines how wide a range the iteration must cover. Both must be set as functions of b to avoid either numerical blow-up or wasted iterations. Given the iterations wildly oscillate until l gets mapped close to 1, which are then stabilized l is chosen such that the 5th iteration stabilizes. Cushion primarily controls the size of the resulting coefficients out of the fitting, with smaller cushions resulting in ever larger values of γ. For this reason we introduce an artificial numerics based bound of 10^5. From empirical sweeps (Figure F.4) we derive closed-form schedules $\delta(b) = (1.84 \times 10^{-8})^{1/b}$ and $l(b) = (10^{-11})^{2/b}$, which we use throughout without further tuning.

F.6 Freon Rational Speed Ablations

We benchmark the orthogonalisation map that dominates the Freon update, comparing an exact SVD-based factorisation against the coupled-Cholesky rational iteration for two exponents, $a/b = 1/2$ and $a/b = 1/4$, in `float32` on a single GPU (Table F.1). The iterative scheme is faster than SVD at every size and for both exponents, and the margin widens with the matrix dimension: it runs 1.4–1.8× faster on the weight shapes that occur in GPT-2 and GPT-2-Large (768–1280), rising to roughly 2.2× at 2048 and 3.8–4.6× at 4096, since SVD scales as $\mathcal{O}(mn\min(m,n))$ with a large solver constant whereas each Cholesky step reduces to a handful of matrix multiplications and a small QR. The two exponents give nearly identical timings – the SVD reference is independent of a/b, and the per-step matmul count differs only slightly – so the speedup is a property of the iteration rather than the chosen power. Five steps already match the exact direction to a relative error of $\sim 10^{-3}$; the approximation tightens further with more steps ($\sim 5 \times 10^{-5}$ at eight), but the runtime reaches parity with SVD there and exceeds it beyond, so 3–5 steps give the best accuracy–cost trade-off. (The larger errors for $a/b = 1/2$ on the square matrices are a conditioning artefact of Gaussian test inputs, whose near-zero singular values are amplified by the inverse power; the milder $1/4$ exponent is unaffected, and neither is representative of the error on realised training gradients.)

G Further Optimizer Details

G.1 Kaon with SVD

`Kaon`'s performance is not specific to the chaotic recurrence used in Algorithm 2. To separate the effect of spectral randomization from the particular iterative implementation, we replace the gradient

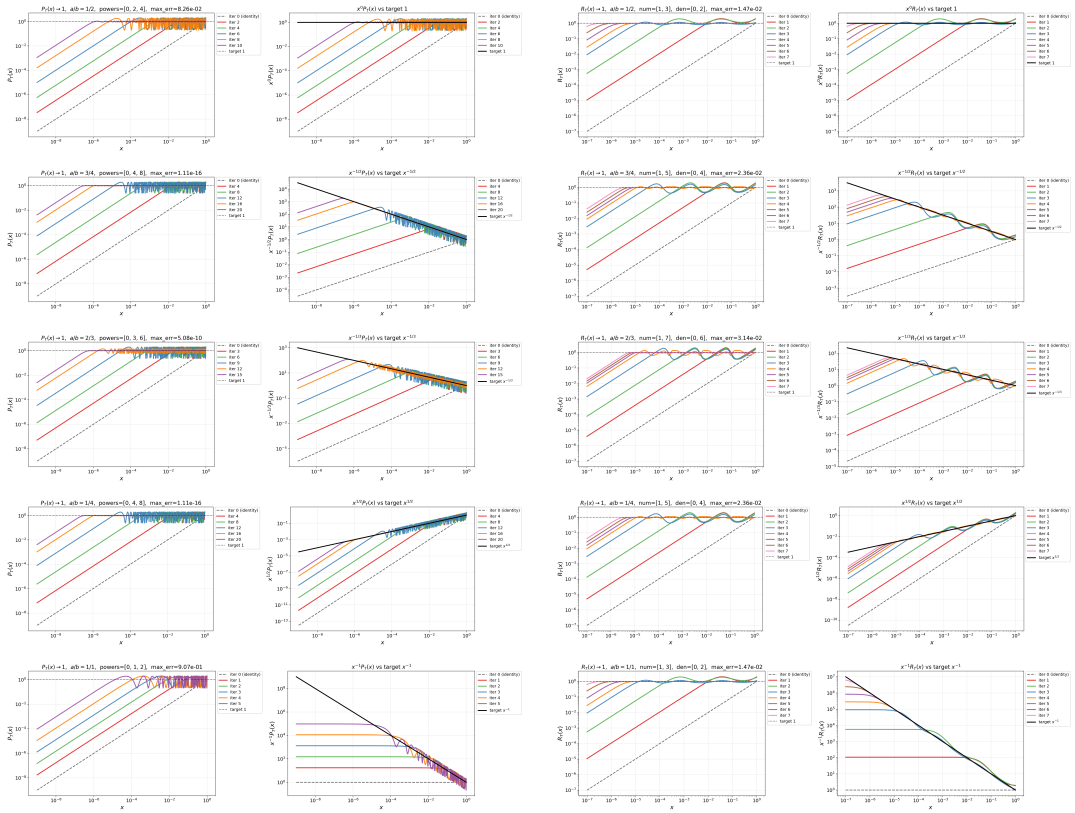


Figure F.3: Convergence of polynomial (left columns) and rational (right columns) Remez iterations for computing $(GG^\top)^{-a/b}G$, shown for $a/b \in 1/2, 3/4, 2/3, 1/4, 1/1$. Each pair of panels shares an x-axis representing a scalar $x \in [\ell, 1]$, which stands in for the singular values of the (normalised) input matrix after mapping to the unit interval. *Left panel of each pair*: the raw iterated composition $P_T(x)$ (polynomial) or $R_T(x)$ (rational), which is initialised at the identity and should converge to 1 on $[\ell, 1]$ as the number of steps T increases; curves for successive values of T are overlaid, progressing from light to dark. *Right panel of each pair*: the rescaled quantity $x^{1-2a/b}P_T(x)$ or $x^{1-2a/b}R_T(x)$ plotted against the target $x^{1-2a/b}$; since the singular values of the exact output $(GG^\top)^{-a/b}G$ are $\sigma_i^{1-2a/b}$, convergence of the composition to 1 is equivalent to convergence of the rescaled curve to the target. Polynomial coefficients are obtained obtained by the Remez algorithm on $[\ell, 1]$ with $\ell = 10^{-16}$; rational coefficients are obtained by the Remez algorithm with $\ell = 10^{-5}$. For rational approximations, exponents a/b with odd denominator (e.g. $2/3$), the rational iteration internally doubles to $4/6$ to ensure the companion-form construction remains valid.

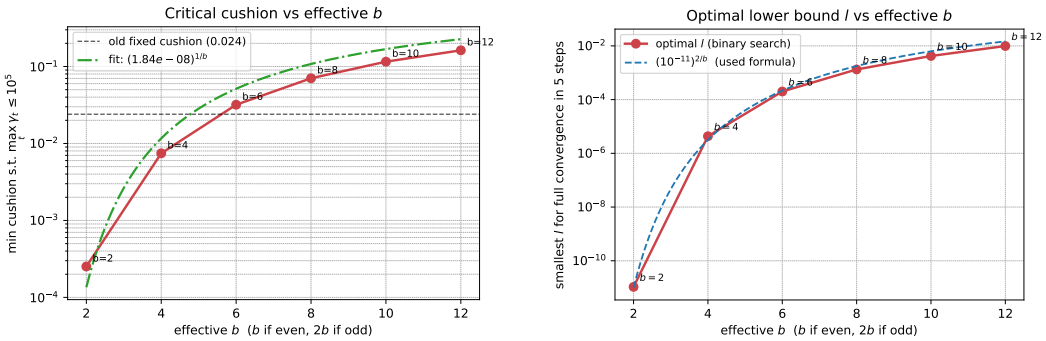


Figure F.4: Hyperparameter selection for the rational Newton–Schulz iteration. **(Left)** Minimum cushion factor required to keep the denominator coefficients bounded ($\max_t \gamma_t \leq 10^5$) as a function of effective q (where odd q is replaced by $2q$). The fitted formula $(1.84 \times 10^{-8})^{1/q}$ safely exceeds the critical threshold for all q and replaces the old fixed value of 0.024. **(Right)** Smallest lower bound l such that the iteration fully converges within five Newton–Schulz steps, i.e. the final interval width reaches its asymptotic fixed point. The formula $(10^{-11})^{2/q}$ matches the empirical optimum across all tested values of q and ensures all steps contribute to convergence.

singular values directly with i.i.d. draws from fixed distributions using an exact SVD. We evaluate Uniform$[0, 1]$ and Beta$(2, 5)$ spectra using the same GPT-2 124M setup as the main experiments (FineWeb-10B, 1907 steps, 8×H100), at `Kaon`'s best learning-rate setting from Appendix A, over three seeds. As shown in Table G.1, both direct-randomization variants closely match `Muon`, with differences comparable to run-to-run variation, and slightly outperform the iterative `Kaon` implementation. Thus, the chaotic recurrence is primarily an efficient mechanism for producing a random spectral transformation rather than the source of Kaon's behavior.

We additionally sweep the recurrence parameter λ. Performance remains competitive across a broad intermediate range, with $\lambda = 4.1$ performing best among the tested values, with small values degrading performance and large values becoming unstable. Together, these experiments indicate that Kaon's behavior is robust to both the shape of the randomized spectrum and the precise choice of recurrence, provided the resulting transformation lies in an appropriate regime.

G.2 KAON

Relation to practical Muon. The original Muon implementation already departs substantially from the exact polar factor: its finite-step Newton–Schulz iteration produces a broad output spectrum rather than unit singular values (Jordan et al., 2024). This observation partly motivated `Kaon`. However, random spectral replacement is not simply another finite-accuracy approximation to the polar factor.

First, we remove the chaotic recurrence entirely and compute an exact SVD, replacing the singular values by independent draws from either Uniform$[0, 1]$ or Beta$(2, 5)$. Both variants closely match Muon in the same GPT-2 training setup (Table G.1), showing that the effect does not depend on approximating the Newton–Schulz map.

Second, Table G.2 compares the spectral maps on a representative GPT-2 gradient. Although finite-step Muon has substantial spectral spread, its output singular values remain strongly associated with the input spectrum. By contrast, `Kaon` has comparable spectral spread while essentially destroying this ordering information. Thus the distinction is not that practical Muon is close to the exact polar factor – it need not be – but that it remains a structured transformation of the input spectrum, whereas `Kaon` tests what happens when that information is discarded.

Stationary Distribution The Kaon optimiser applies the polynomial map $f(x) = \lambda x(1 - x^2)^2$ iteratively to the singular values of the gradient matrix as a cheap alternative to the exact polar factorisation used by Muon. Unlike Newton–Schulz iterations, which converge the singular values to ± 1, the Kaon map at $\lambda = 4.1$ operates in a chaotic regime: rather than converging, the iterates explore a broad stationary distribution supported on $(0, 1.175)$. Figure G.1a displays the empirical

Algorithm 4: Coupled QDWH Cholesky Iteration for $(GG^\top)^{-a/b}G$

Input: Gradient $G \in \mathbb{R}^{k\times n}$ ($k \le n$), powers $a, b \in \mathbb{N}$, rational coefficients $\{(\alpha_t, \beta_t, \gamma_t)\}_{t=1}^{N}$, number of steps N, regularization ϵ.
Output: Approximation to $(GG^\top)^{-a/b}G$.

```
// Ensure b is even to yield an integer r for the block QR step
```
if $b \pmod 2 \neq 0$ **then**
  $a \leftarrow 2a, \quad b \leftarrow 2b$
end
$r \leftarrow b/2$
```
// Normalize gradient to improve numerical stability
```
$\nu \leftarrow \|G\|_F + \epsilon$
$G \leftarrow G/\nu$
```
// Initialize L via thin QR of augmented gradient
```
$Q_0, R_0 \leftarrow \mathrm{QR}\left(\begin{bmatrix} G^\top \\ \sqrt{\epsilon} I_k \end{bmatrix}\right)$
$L \leftarrow R_0^\top$ // Yields $LL^\top = GG^\top + \epsilon I$
$L_0 \leftarrow L$ // Saved for log-determinant;
$\log\det(GG^\top) = 2\sum_i \log[L_0]_{ii} + 2k\log\nu$
$C \leftarrow I_k$ // Accumulator
for $t = 1$ **to** N **do**
  ```
  // Compute W = α_t I + β_t A_t (I + γ_t A_t)^{-1} where A_t = LL^⊤
  ```
  $\rho_t \leftarrow \beta_t/\gamma_t$
  ```
  // Construct Block Matrix K optimally to avoid numeric
      blowup
  ```
  if $\gamma \le 1.0$ **then**
    $K \leftarrow \begin{bmatrix} \sqrt{\gamma_t} L^\top \\ I_k \end{bmatrix}$
  else
    $K \leftarrow \begin{bmatrix} L^\top \\ \frac{1}{\sqrt{\gamma_t}} I_k \end{bmatrix}$
  end
  $Q_K, R_K \leftarrow \mathrm{QR}(K)$
  $Q_2 \leftarrow$ bottom $k \times k$ block of Q_K
  $V \leftarrow Q_2 Q_2^\top$ // Identity: $V = (I + \gamma_t LL^\top)^{-1}$
  $W \leftarrow \rho_t I + (\alpha_t - \rho_t)V$
  $W \leftarrow \frac{1}{2}(W + W^\top)$ // Ensure numerical symmetry
  ```
  // Coupled state updates
  ```
  $L \leftarrow W^r L$ // Equivalent to: $L_{t+1}L_{t+1}^\top = W^b A_t$
  $C \leftarrow WC$ // Accumulate product $\prod W_t$
end
```
// Rescale and apply to gradient
```
return $\nu^{1-2a/b} C^a G$, L_0

stationary distributions for ten values of λ spanning $[3.5, 4.25]$, estimated by running $5{,}000$ particles through 500 burn-in followed by 200 collection steps of the map. The distributions shift and spread as λ increases, with the support and shape varying considerably across the range. Figure G.1b shows a cobweb diagram at $\lambda = 4.1$, confirming the chaotic character of the dynamics: trajectories starting from different initial conditions in $(0.1, 0.9)$ neither converge to a fixed point nor settle into a periodic orbit, but instead wander erratically through the attractor. This stochasticity in the effective preconditioner is an intrinsic feature of Kaon rather than a deficiency.

Cost of Kaon `Kaon` uses the same number of steps T as `Muon`'s Newton–Schulz iteration and matches its matrix-multiplication cost. Each step of both methods forms $XX^\top$, squares one $r \times r$

Table F.1: Per-call wall-clock cost of computing the a/b-power polar factor $(GG^\top)^{-a/b}G$ via exact SVD versus FREON's coupled-Cholesky rational iteration (5 Newton–Schulz steps), for two exponents. Times are on a single Quadro RTX 6000 in `float32` (median over 20–50 calls after warm-up); the SVD cost is independent of a/b. *err.* is the Frobenius distance between the two ℓ_2-normalised update directions.

			FREON, $a/b = 1/2$			FREON, $a/b = 1/4$		
Layer	Shape ($m \times n$)	SVD (ms)	ms	speedup	err.	ms	speedup	err.
`attn.c_proj` (gpt2)	768×768	44.9	30.3	1.48×	5.9e−2	32.0	1.40×	5.6e−3
`attn.c_attn` (gpt2)	768×2304	48.5	33.4	1.45×	1.2e−3	34.9	1.39×	3.5e−3
`mlp.c_fc` (gpt2)	768×3072	49.4	35.0	1.41×	1.1e−3	36.5	1.35×	3.1e−3
`attn.c_proj` (Large)	1280×1280	134.1	73.5	1.82×	6.0e−2	78.9	1.70×	7.1e−3
`mlp.c_fc` (Large)	1280×5120	145.1	84.8	1.71×	1.0e−3	91.0	1.60×	4.1e−3
square	2048×2048	423.9	174.5	2.43×	6.6e−2	196.3	2.16×	8.8e−3
square	4096×4096	3639.9	792.5	4.59×	7.8e−2	957.8	3.80×	1.2e−2

Table G.1: Robustness of `Kaon` to the spectral distribution and recurrence parameter. Results are final validation loss, reported as mean ± population standard deviation over three seeds. Direct-randomization variants use an exact SVD and contain no chaotic recurrence.

Direct spectral randomization		
Method	Final val. loss	Seeds
Muon	3.3369 ± 0.0008	3
SVD, $U[0, 1]$	3.3383 ± 0.0019	3
SVD, $\mathrm{Beta}(2, 5)$	3.3386 ± 0.0017	3
Kaon, $\lambda = 4.1$	3.3410 ± 0.0035	3
Kaon (main experiment)	3.3420 ± 0.0052	3
Recurrence parameter sweep		
λ	Final val. loss	Seeds
2.0	3.4082 ± 0.0064	3
3.0	3.3564 ± 0.0033	3
3.5	3.3707 ± 0.0024	3
4.1	3.3410 ± 0.0035	3
4.25	unstable / degraded	3

matrix, and multiplies the resulting degree-two matrix polynomial by X, costing $T(4r^2s + 2r^3)$ FLOPs to leading order.

H RANDOM FEATURE ASYMPTOTICS

In this section we show how to compute our two quantities in this asymptotic framework. Write $G = U\Sigma V^T$ for the compact SVD of the oracle gradient.

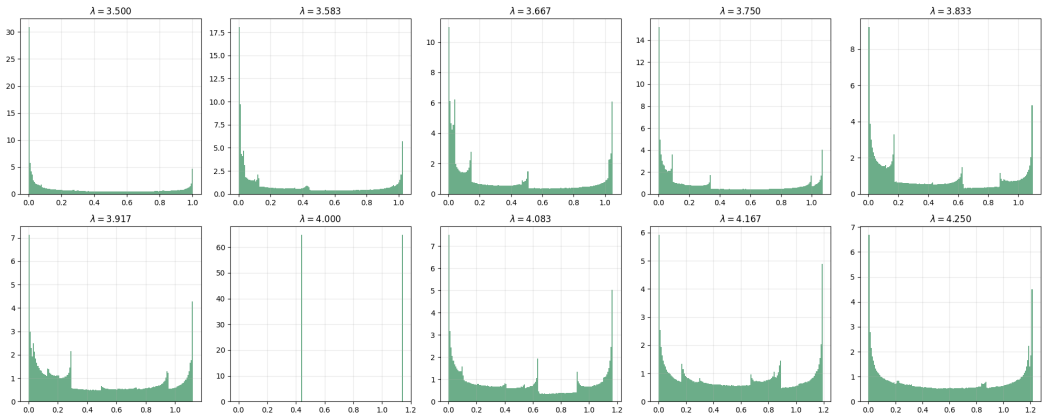
Definition H.1. The ith normalized trace is defined by

$$\tau_i = \frac{1}{n} \operatorname{tr}(C^i)$$

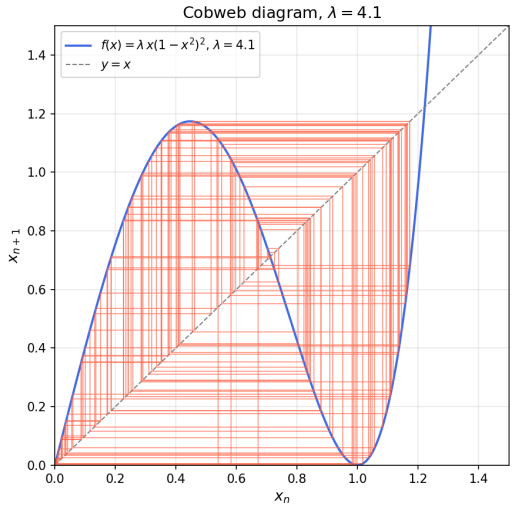
Theorem H.2. *Assume that $A = C^{1/2}Z$ with C symmetric positive semi-definite with $\limsup \|C\|_{op} < \infty$, and Z a matrix with entries which are iid Gaussians.*

Let $H = \frac{1}{b}AA^\top$. Then there exists universal polynomials p_k such that we have almost surely

$$\|V^\top H^k V - V^\top(p_k(C))V\|_{op} \to 0$$



(a) Empirical stationary distributions of the Kaon map $x_{n+1} = \lambda\, x_n(1 - x_n^2)^2$ for ten values of λ uniformly spaced in $[3.5, 4.25]$. Each histogram is obtained by evolving $5{,}000$ particles for 500 burn-in iterations followed by 200 collection steps.



(b) Cobweb diagram of the Kaon map at $\lambda = 4.1$, showing three trajectories of 40 iterations each from independent random initial conditions in $(0.1,\ 0.9)$. The map curve $y = f(x)$ (blue) and the diagonal $y = x$ (dashed) are overlaid; the chaotic wandering of the iterates is evident.

Figure G.1: Dynamics of the Kaon polynomial map $f(x) = \lambda\, x(1 - x^2)^2$ used in the Kaon optimiser as a Newton-Schulz-style spectral normalisation step.

The first three such polynomials are

$$p_1(C) = C$$
$$p_2(C) = C^2 + \delta\tau_1 C$$
$$p_3(C) = C^3 + \delta\tau_1 C^2 + (\delta\tau_2 + \delta^2\tau_1^2)C$$

Proof. Choose an orthonormal basis e_i of $\mathbb{R}^r$, as V is an $n \times r$ matrix. It is sufficient to bound this operator norm by bounding the inner product over all vectors forming the orthonormal basis. We thus fix $x = Ve_i$ and $y = Ve_j$ and compute the inner product $\langle x, Hy\rangle$. By Cauchy's Integral Formula,

Table G.2: Comparison of spectral maps on a representative GPT-2 gradient. ρ is the Spearman correlation between input and output singular values; "spread" is the standard deviation of the normalized output spectrum. Suboptimality is measured against the exact spectral-norm LMO objective. No single distance metric defines whether a map is "Muon"; the relevant distinction here is whether a broad output spectrum retains information about the input spectrum.

Map	Angle to Muon-5	Subopt.	ρ	Spread
Exact polar	33.5°	0.000	1.000	0.000
Muon, 1 step	64.9°	0.709	1.000	0.081
Muon, 2 steps	57.9°	0.473	1.000	0.174
Muon, 3 steps	44.9°	0.345	0.998	0.262
Muon, 4 steps	31.0°	0.274	0.981	0.318
Muon, 5 steps	0.0°	0.281	0.893	0.311
Polar Express, 5	15.9°	0.110	0.836	0.304
Kaon	46.8°	0.543	−0.016	0.330
SVD + Uniform	37.3°	0.308	−0.011	0.195

we have

$$\langle x, H^k y\rangle = \frac{1}{2\pi i}\oint_\Gamma z^k \langle x, (W - zI)^{-1} y\rangle dz$$

where Γ is a contour which encloses the spectrum of H. First we explicitly choose the contour for almost every H. Note that

$$H = \frac{1}{b} C^{1/2} Z Z^\top C^{1/2}$$

where Z has iid standard Gaussian columns. Therefore

$$\|H\|_{op} \le \|C\|_{op} \|\frac{1}{b} Z Z^\top\|_{op} = K \|\frac{1}{b} Z Z^\top\|_{op}$$

By standard Marchenko-Pasteur theory, the spectrum of $\frac{1}{b} Z Z^\top$ is almost surely eventually contained in the Marchenko-Pasteur bulk $[(1-\sqrt{\delta})^2, (1+\sqrt{\delta})^2]$. Therefore almost surely

$$\|H\|_{op} \le K((1+\sqrt{\delta})^2 + o(1))$$

Choose $R > \|H\|_{op}$. Take the contour Γ to be the circle of radius R.

Now we use (Couillet & Liao, 2022, Theorem 2.6). Then we have, similar to Equation 2.46 of loc. cit,

$$\langle x, H^k y\rangle = \frac{1}{2\pi i}\oint_\Gamma z^{k-1} \langle x, (I + \tilde{m}_p(z) C)^{-1} y\rangle dz + o(1)$$

where $\tilde{m}_p(z)$ solves the equation

$$z = -\frac{1}{\tilde{m}_p(z)} + \frac{\delta}{n} \operatorname{tr}(C(I + \tilde{m}_p(z) C)^{-1}) \tag{31}$$

By expanding $\tilde{m}_p(z)$ in a power series in z^{-1} we obtain a Laurent series. Combining this with the expansion for $(I + \tilde{m}_p(z) C)^{-1}$ gives a Laurent series in z^{-1} with coefficients which are polynomials in C. By standard complex analysis, the integral simply selects one of these coefficients. This gives us the result.

Now we explicitly compute the first three polynomials. Expanding the second term of Equation (31) we have

$$\frac{1}{n} \operatorname{tr}(C(I + \tilde{m}_p(z) C)^{-1}) = \tau_1 - \tilde{m}_p(z)\tau_2 + \tilde{m}_p(z)^2 \tau_3 + O(\tilde{m}_p(z)^3)$$

This allows us to solve

$$\tilde{m}_p(z) = -\frac{1}{z} - \frac{\delta\tau_1}{z^2} - \frac{\delta\tau_2 + \delta^2\tau_1^2}{z^3} + O(z^{-4})$$

For large z, $\tilde{m}_p(z)$ is small hence we may expand

$$(I + \tilde{m}_p(z) C)^{-1} = I - \tilde{m}_p(z) C + (\tilde{m}_p(z) C)^2 - (\tilde{m}_p(z) C)^3 + O(C^4)$$

Then by plugging in the expansion for $\tilde{m}_p(z)$ we obtain exactly the polynomials written above. □

Remark H.3. We can extend this same theorem, with the same polynomials to Z having entries independent mean zero, unit variances random variables, with a minor condition on the tails, but to keep with the main text we do not expand on this condition in detail.

In order to actually evaluate the limit in this theorem we need to assume that for $i = 1, 2, 3$ the following limit exists

$$V^\top C^i V \to \Gamma_i$$

Then by the above theorem we have that for $i = 1, 2, 3$, and the polynomials as defined above

$$V^\top H^i V \to p_i(\Gamma_i, \dots, \Gamma_1)$$

We define for $i = 2, 3$, $\widehat{\Gamma}_i = p_i(\Gamma_i, \dots, \Gamma_1)$.

Then almost surely

$$\gamma_{\texttt{SGD}} \to \frac{\operatorname{tr}(\Sigma^2 \Gamma_1)}{\operatorname{tr}(\Sigma^2 \widehat{\Gamma}_2)} \tag{32}$$

$$\Phi_{\texttt{SGD}} \to \frac{\operatorname{tr}(\Sigma^2 \widehat{\Gamma})^2}{\operatorname{tr}(\Sigma^2 \widehat{\Gamma}_3)} \tag{33}$$

For Muon we have, where the inverse square-root is a Moore-Penrose inverse,

$$D = \operatorname{polar}(GH) = (GH^2G^\top)^{-1/2} GH$$

Define $T^i = V^\top H^i V$, so that almost surely $T^1 \to \Gamma_1$ and for $i = 2, 3$ almost surely $T^i \to \widehat{G}_i$. Then

$$\gamma_{\texttt{Muon}} = \frac{\operatorname{tr}((GH^2G^\top)^{-1/2} GHG^\top)}{\operatorname{tr}((GH^2G^\top)^{1/2})} = \frac{\operatorname{tr}((\Sigma T_2 \Sigma)^{-1/2} \Sigma T_1 \Sigma)}{\operatorname{tr}((\Sigma T_2 \Sigma)^{1/2})} \tag{34}$$

$$\to \frac{\operatorname{tr}((\Sigma \widehat{\Gamma}_2 \Sigma)^{-1/2} \Sigma \Gamma_1 \Sigma)}{\operatorname{tr}((\Sigma \widehat{\Gamma}_2 \Sigma)^{1/2})} \tag{35}$$

$$\Phi_{\texttt{Muon}} = \frac{\operatorname{tr}((GH^2G^\top)^{1/2})^2}{\operatorname{tr}((GH^2G^\top)^{-1} GH^3G^\top)} = \frac{\operatorname{tr}((\Sigma T_2 \Sigma)^{1/2})^2}{\operatorname{tr}((\Sigma T_2 \Sigma)^{-1} \Sigma T_3 \Sigma)} \tag{36}$$

$$\to \frac{\operatorname{tr}((\Sigma \widehat{\Gamma}_2 \Sigma)^{1/2})^2}{\operatorname{tr}((\Sigma \widehat{\Gamma}_2 \Sigma)^{-1} (\Sigma \widehat{\Gamma}_3 \Sigma)} \tag{37}$$

Let $K = \Sigma \Gamma_1 \Sigma$, $M = \Sigma \widehat{\Gamma}_2 \Sigma$, $N = \Sigma \widehat{\Gamma}_3 \Sigma$. Then the SGD calculations simplify to

$$\gamma_{\texttt{SGD}} \to \frac{\operatorname{tr}(K)}{\operatorname{tr}(M)}, \quad \Phi_{\texttt{SGD}} \to \frac{\operatorname{tr}(M)^2}{\operatorname{tr}(N)} \tag{38}$$

and the Muon calculations become

$$\gamma_{\texttt{Muon}} \to \frac{\operatorname{tr}(M^{-1/2} K)}{\operatorname{tr}(M^{1/2})}, \quad \Phi_{\texttt{Muon}} \to \frac{(\operatorname{tr}(M^{1/2}))^2}{\operatorname{tr}(M^{-1} N)} \tag{39}$$

Take an orthonormal eigenbasis q_i of M. Let μ_i be the eigenvalue of q_i. Order these such that

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_r$$

Then write

$$\rho_i = \frac{q_i^T K q_i}{\mu_i}$$

Then we obtain

$$\gamma_{\texttt{SGD}} \to \frac{\sum_i \rho_i \mu_i}{\sum_i \mu_i}, \quad \gamma_{\texttt{Muon}} \to \frac{\rho_i \sqrt{\mu_i}}{\sqrt{\mu_i}}$$

Proposition H.4. *Suppose that the ρ_i are non-decreasing. Then*

$$\gamma_{\texttt{SGD}} \leq \gamma_{\texttt{Muon}}$$

Equality holds if and only if

$$(\mu_i - \mu_j)(\rho_i - \rho_j) = 0$$

for all i, j.

Proof. Direct computation gives

$$\gamma_{\texttt{Muon}} - \gamma_{\texttt{SGD}} = \frac{\sum_{i<j} \sqrt{\mu_i \mu_j}(\sqrt{\mu_i} - \sqrt{\mu_j})(\rho_j - \rho_i)}{(\sum_i \sqrt{\mu_i})(\sum_i \mu_i)}$$

The non-decreasing assumption gives that the denominator and all terms in the numerator are non-negative. This gives the first part. The second part, note that equality holds if and only if each term in the numerator is zero, which is equivalent to the statement we are trying to show. □

Now we show the aligned version of this. Assume the $CV = V\Lambda$ for some diagonal matrix Λ.

Proposition H.5. *Assume that $CV = V\Lambda$ for a diagonal matrix Λ. If the entries of Λ are in non-increasing order then $\gamma_{\mathrm{SGD}} \leq \gamma_{\mathrm{Muon}}$ with equality if and only if $\Lambda = cI$.*

Proof. Under the assumptions, we have $M = \Sigma\Lambda(\Lambda + \delta)\Sigma$ and $K = \Sigma\Lambda\Sigma$. In particular these are each diagonal. We computes

$$\mu_i = \sigma_i^2 \lambda_i (\lambda_i + \delta)$$

and

$$\rho_i = \frac{1}{\lambda_i + \delta}$$

By assumption the λ_i are non-increasing, therefore the μ_i are in non-increasing order and the ρ_i are in non-decreasing order. Now we simply apply Proposition H.4. □

Proposition H.6. *If $C = I$ then $\Phi_{\mathrm{SGD}} \geq \Phi_{\mathrm{Muon}}$ and therefore $\gamma_{\mathrm{SGD}}^2 \Phi_{\mathrm{SGD}} \geq \gamma_{\mathrm{Muon}}^2 \Phi_{\mathrm{Muon}}$*

Proof. We simply compute that $\Gamma_1 = I$, $\widehat{\Gamma}_2 = (1+\delta)I$ and $\widehat{\Gamma}_3 = (1 + 3\delta + \delta^2)I$. Then

$$\gamma_{\texttt{SGD}} = \gamma_{\texttt{Muon}} = \frac{1}{1+\delta}$$

Therefore

$$\Phi_{\texttt{SGD}} = \frac{(1+\delta)^2}{1 + 3\delta + \delta^2} \operatorname{tr}(\Sigma^2)$$

$$\Phi_{\texttt{Muon}} = \frac{(1+\delta)^2}{1 + 3\delta + \delta^2} \frac{\operatorname{tr}(\Sigma)^2}{r}$$

Then Cauchy-Schwartz gives $\Phi_{\texttt{Muon}} \leq \Phi_{\texttt{SGD}}$. □

Theorem H.7. *In the undersampled case ($\delta \to \infty$) we have $\lim_{\delta \to \infty} \frac{\gamma_{\mathrm{SGD}}^2 \Phi_{\mathrm{SGD}}}{\gamma_{\mathrm{Muon}}^2 \Phi_{\mathrm{Muon}}} \geq 1$*

Proof. As $\delta \to \infty$ we have

$$\widehat{\Gamma}_2 = \delta\tau_1\Gamma_1 + O(1), \quad \widehat{\Gamma}_3 = \delta^2\tau_1^2\Gamma_1 + O(\delta)$$

Therefore,

$$\Sigma\widehat{\Gamma}_3\Sigma = \delta^2\tau_1^2 K + O(\delta)$$

and

$$M = \Sigma\widehat{\Gamma}_2\Sigma = \delta\tau_1 K + O(1)$$

Plugging this into the Equation (38) we obtain

$$\delta^2\gamma_{\texttt{SGD}}^2\Phi_{\texttt{SGD}} = \frac{\delta^2 \operatorname{tr}(K)^2}{\delta^2\tau_1^2 \operatorname{tr}(K) + O(\delta)} \to \frac{\operatorname{tr}(K)}{\tau_1^2}$$

For Equation (39) we have

$$\delta^2 \gamma_{\texttt{Muon}}^2 \Phi_{\texttt{Muon}} = \frac{\operatorname{tr}(M^{-1/2}K)^2}{\operatorname{tr}(\widehat{\Gamma}_2 \widehat{\Gamma}_3)}$$

We compute the asymptotics of the numerator and the denominator seperately. Beginning with the numerator, note that

$$\frac{M}{\delta\tau_1} = K + O(\delta^{-1})$$

Hence $\frac{M}{\delta\tau_1} \to K$ in operator norm. Because the rank of M equals the rank of K, we have that the inverse square-root is continuous, hence

$$(\delta\tau_1)^{1/2} M^{-1/2} \to K^{-1/2}$$

Therefore,

$$\operatorname{tr}(M^{-1/2}K)^2 \to \frac{\operatorname{tr}(K^{1/2})^2}{\delta\tau_1} + o(\delta^{-1})$$

For the denominator we have

$$M^{-1}\Sigma\widehat{\Gamma}_3\Sigma = \delta\tau_1(K + O(\delta^{-1}))^{-1}(K + O(\delta^{-1})) \to \delta\tau_1 I_S$$

where S is the range of K. Here we are using the fact that we are taking Moore-Penrose inverses. Taking trace and combining these two asymptotics we obtain

$$\delta^2 \gamma_{\texttt{Muon}}^2 \Phi_{\texttt{Muon}} \to \frac{\operatorname{tr}(K^{1/2})^2}{(\dim S)\tau_1^2}$$

Finally we simply use Cauchy-Schwartz to compare the two limits,

$$\frac{\operatorname{tr}(K^{1/2})^2}{(\dim S)} \leq \operatorname{tr}(K)$$

□

I Random Feature Experiments

I.1 Detailed Setup

For the Random Feature experiments considered in this paper, we utilize the setup and code from Davis & Drusvyatskiy (2026), provided in https://github.com/damek/specgd/. The specific setup is as follows: We consider a random feature regression problem with objective $f(W) = \frac{1}{N\sqrt{d}}\|WA - Y\|_F^2$, where $W \in \mathbb{R}^{o\times d}$ is the weight matrix, $A \in \mathbb{R}^{d\times N}$ is a fixed random feature matrix, and $Y = W^\star A$ for a randomly initialised ground-truth $W^\star$. Since the loss is exactly quadratic in W, the gradient is $G = \nabla_W f = \frac{2}{N\sqrt{d}}(W - W^\star)AA^\top$ and the Hessian action along any direction D is $\langle D, \nabla^2 f[D]\rangle = \frac{2}{N\sqrt{d}}\|DA\|_F^2$. Consequently, the exact optimal step size for any update direction D is

$$\eta^\star(D) = \frac{\langle G, D\rangle}{\frac{1}{\sqrt{d}}\|DA\|_F^2}. \tag{40}$$

We run two experiments with this setup. In the **ReLU** experiment, A is obtained by applying a ReLU activation to a random linear map of $N = 400$ inputs, yielding $W \in \mathbb{R}^{120\times 100}$, $A \in \mathbb{R}^{100\times 400}$, stable rank $\operatorname{st}(A) \approx 3.06$, $\|A\|_{\text{op}} \approx 80.87$, and $\|A\|_F \approx 141.40$; training runs for 1000 iterations. In the **SwiGLU** experiment the same dimensions are used but the activation is SwiGLU, giving $\operatorname{st}(A) \approx 29.96$, $\|A\|_{\text{op}} \approx 22.15$, and $\|A\|_F \approx 121.22$; training runs for 400 iterations. Both experiments use base learning rate $\eta = 10^{-2}$, double precision (`float64`), and fixed random seed. The Optimal C methods select c and η jointly at each step by maximising the quadratic lower bound $\Delta f(c,\eta) = -\eta\, a(c) + \eta^2 b(c)$ over a grid, where $a(c) = \langle G, D_c\rangle$ and $b(c) = \frac{1}{\sqrt{d}}\|D_c A\|_F^2$ with $D_c = (GG^\top)^{-c}G$. `Freon` is implemented using SVD, while `Kaon` is implemented with SVD with random uniform noise.

I.2 Exponent Approximation for Optimal Schatten Norm

We are in the random feature model setting and assume to have $G \in \mathbb{R}^{m\times n}$ be gradient with a compact Singular Value Decomposition $G = USV^\top$, where $S = \text{diag}(\sigma_1, \ldots, \sigma_r)$. We seek a spectrally scaled matrix $D = US^aV^\top$ that maximizes optimal decrease Φ_k.

The objective is to maximize decrease at optimal step size, for $\|DA\|_F^2 = \sum k_i \sigma_i^{2a}$, where $k_i = \|A^\top v_i\|^2$ represents the projection energy of an activation matrix A:

$$\max_a g(a) = \frac{\left(\sum_{i=1}^r \sigma_i^{a+1}\right)^2}{\sum_{i=1}^r k_i \sigma_i^{2a}} \tag{41}$$

Setting the derivative of $\log g(a)$ to zero yields the stationarity condition for optimality:

$$\frac{\sum_{i=1}^r \sigma_i^{a+1} \log \sigma_i}{\sum_{i=1}^r \sigma_i^{a+1}} = \frac{\sum_{i=1}^r k_i \sigma_i^{2a} \log \sigma_i}{\sum_{i=1}^r k_i \sigma_i^{2a}} \tag{42}$$

To approximate an optimal a value, we will assume the singular values of G and the penalty weights follow a power-law decay:

$$\sigma_i \propto i^{-\alpha} \quad (\alpha > 0), \qquad k_i \propto i^{-\beta} \tag{43}$$

In this case, equality is achieved by equating exponents of LHS and RHS, i.e. $\alpha(a+1) = 2a\alpha + \beta$, meaning that

$$a = 1 - \frac{\beta}{\alpha}$$

I.3 Optimal C methods

Both adaptive-exponent methods take the update direction $D_c = (GG^\top)^{-c}G = U\,\text{diag}(\tilde{\sigma}^{1-2c})\,V^\top$, where $G = U\Sigma V^\top$ is the SVD of the gradient and $\tilde{\sigma}_i = \sigma_i/\mu_c$ are the singular values normalised by the power mean $\mu_c = \left(\frac{1}{r}\sum_i \sigma_i^{2(1-c)}\right)^{1/(2(1-c))}$. The optimal step size $\eta^\star(c) = n\,a(c)/b(c)$ is used exactly at every step, where $a(c) = \sum_i \sigma_i\,\tilde{\sigma}_i^{1-2c}$ and $b(c) = \|D_cA\|_F^2/\sqrt{d}$.

Optimal C [Greedy] determines c by exhaustive search: at each iteration a grid of 41 values $c \in [-0.5, 1.5]$ is evaluated and the value minimising the quadratic predicted loss decrease $\Delta f^\star(c) = -n\,a(c)^2/(2\,b(c))$ is selected.

Optimal C [Scaling] instead infers c analytically from the spectral structure of the gradient and the feature matrix based on Appendix I.2. Fitting $\log \sigma_i \approx -\alpha \log i$ and $\log k_i \approx -\beta \log i$ by least squares, where $k_i = \|v_i^\top A\|^2$ is the squared projection of the i-th right singular vector onto the feature matrix, the theoretically optimal exponent is

$$c^\star = \frac{\beta}{2\alpha}, \tag{44}$$

which is clamped to $[-0.5, 1.5]$ and used with step size $\eta^\star(c^\star)$.

J GPT2 Additional Experiments

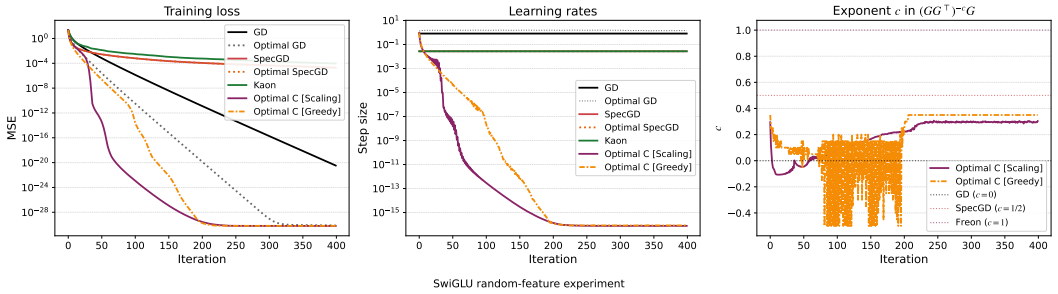


Figure I.1: **Random-feature regression with SwiGLU activations.** Same setup as Figure 5 but with a SwiGLU random-feature matrix $A \in \mathbb{R}^{100\times400}$ ($\mathrm{st}(A) \approx 29.96$, substantially higher than the ReLU case), trained for 400 iterations. *Left:* training loss. *Centre:* effective step sizes; Optimal GD exhibits strongly oscillatory step sizes throughout training. *Right:* exponent c in $(GG^\top)^{-c}G$ for the two Optimal C methods. The higher stable rank of A under SwiGLU changes the curvature landscape relative to the ReLU setting, and both Optimal C methods converge to a smaller value of c (closer to GD) than in the ReLU experiment.

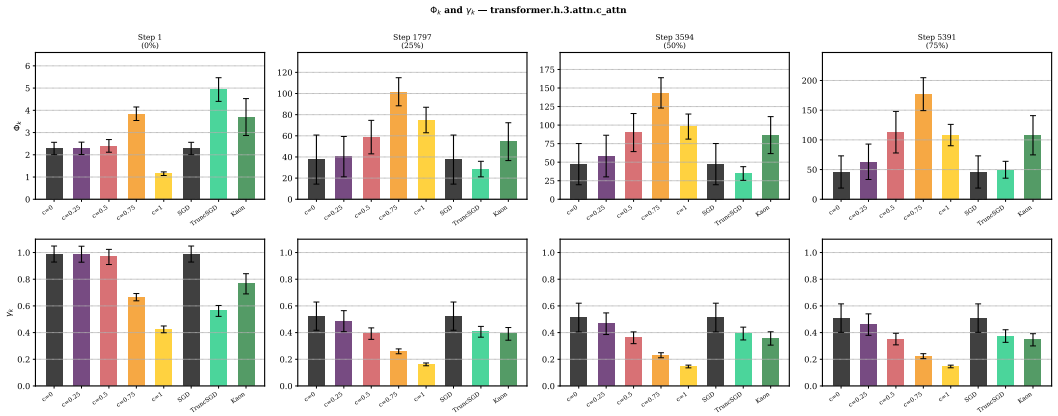


Figure J.1: Per-layer curvature alignment quantities Φ_k (top row, linear scale) and γ_k (bottom row, linear scale) for the weight matrix `h.3.attn.c_attn` at four training checkpoints (steps 1, 1797, 3594, 5391). Each bar shows the mean $\pm$ one standard deviation over validation batches, separately for each candidate update direction D_k. The directions include `Freon`'s power-mean normalised descent direction at interpolation levels $c \in 0, 0.25, 0.5, 0.75, 1$ (where $c = 0$ matches `SGD`, and $c = 1$ is the full `Freon` direction), `TruncatedSGD` (batch gradient with the top 1% of singular values zeroed), and `Kaon`. Formally, for a given direction D_k and batch gradient g_b, we calculate $\langle g_b, D_k\rangle^2$, the directional curvature $\lambda_k = \langle D_k, \mathcal{F}_{\mathrm{val}}[D_k]\rangle$ where $\mathcal{F}_{\mathrm{val}}$ is the Gauss–Newton (GGN) matrix estimated over all validation batches via Jacobian–vector products, and the batch gradient alignment $\gamma_k = \langle G_k, D_k\rangle / \langle g_b, D_k\rangle$ where G_k is the gradient averaged over all validation batches. A large Φ_k indicates that the direction is strongly aligned with the batch gradient and lies in a low-curvature region, supporting a larger step, while $\gamma_k \approx 1$ indicates that the batch gradient faithfully represents the global gradient in that direction.

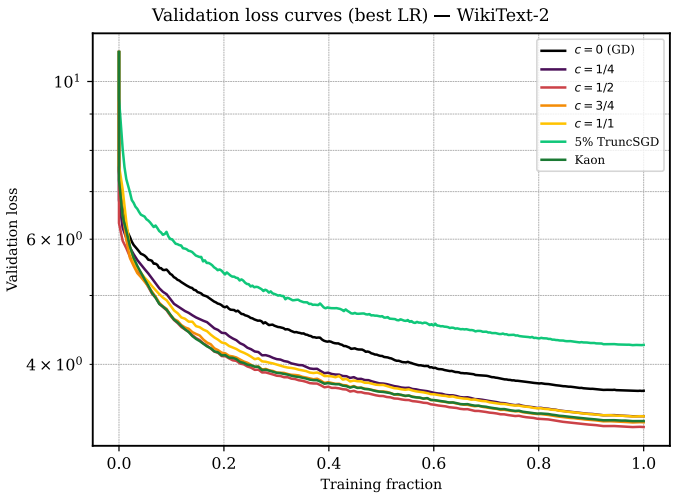


Figure J.2: Validation loss as a function of training progress for each optimiser at its best learning rate on WikiText-2. Curves shown for Freon with exponent $c \in \{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$, 5% TruncatedSGD, and Kaon.

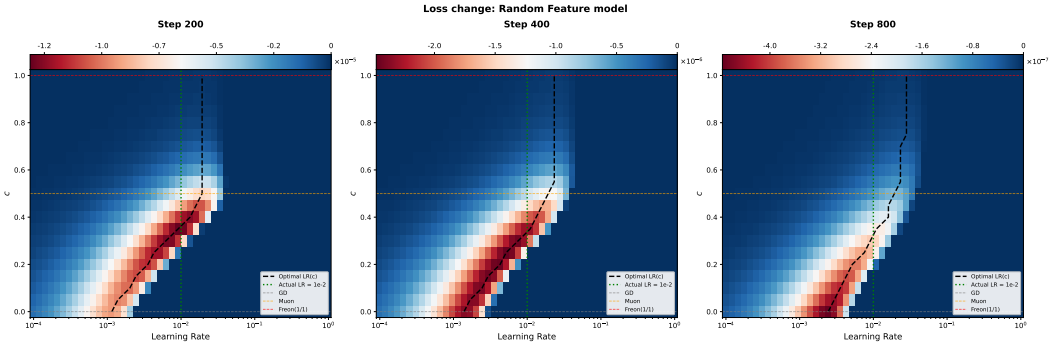


Figure J.3: Loss change Δf over the joint (c, η) grid for the quadratic model, evaluated at three training iterations of the random feature experiment. Each panel uses the gradient computed at that iteration to evaluate $\Delta f(c, \eta) = -\eta\, a(c) + \eta^2 b(c)$, where $a(c) = \langle G, D_c \rangle$ and $b(c) = \frac{1}{2n} \|D_c A\|_F^2$. The dashed black curve marks the optimal learning rate per c; the green dotted line marks the actual training learning rate $\eta = 0.01$.

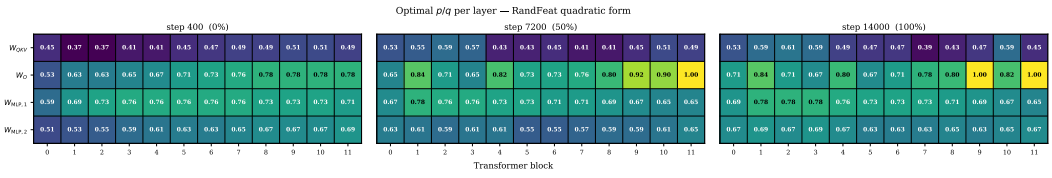


Figure J.4: **Layer-wise optimal c at three training snapshots (RandFeat quadratic model).** Each cell shows the value of $c \in [0, 1]$ that minimises the predicted one-step loss decrease for that layer under a random-feature approximation to the curvature. Panels correspond to the beginning, middle, and end of training. Rows correspond to the four weight matrices per transformer block (W_{QKV}, W_O, $W_{\mathrm{MLP},1}$, $W_{\mathrm{MLP},2}$); columns index the 12 transformer blocks. $c = 0$ recovers gradient descent, $c = \frac{1}{2}$ recovers Muon, and $c = 1$ recovers the pseudoinverse-like endpoint.

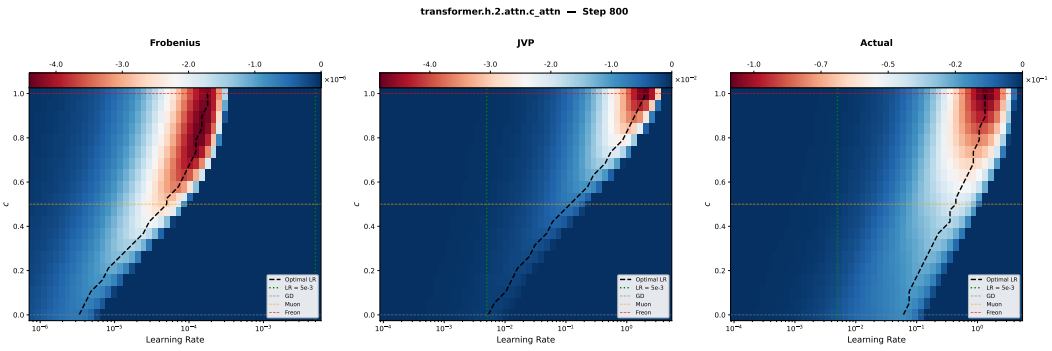


Figure J.5: **Per-step loss decrease vs. learning rate and c under three local quadratic models** (layer `h.2.attn.c_attn`, step 800). *Left (Frobenius):* The naive random feature model predicts $c \approx 2/3$ as optimal. *Centre (JVP):* Under the GGN quadratic approximation on a single batch, $c = 1$ achieves the largest per-step decrease – similar to the Newton step being optimal for a fixed-batch quadratic. *Right (Actual):* Evaluating on the actual single-batch loss agrees with the GGN picture, motivating the inter-batch SNR analysis of Section 3.2. All heatmaps are clamped at zero (only improvements shown); the dashed black curve marks the per-c optimal learning rate and the green dotted line the configured learning rate.

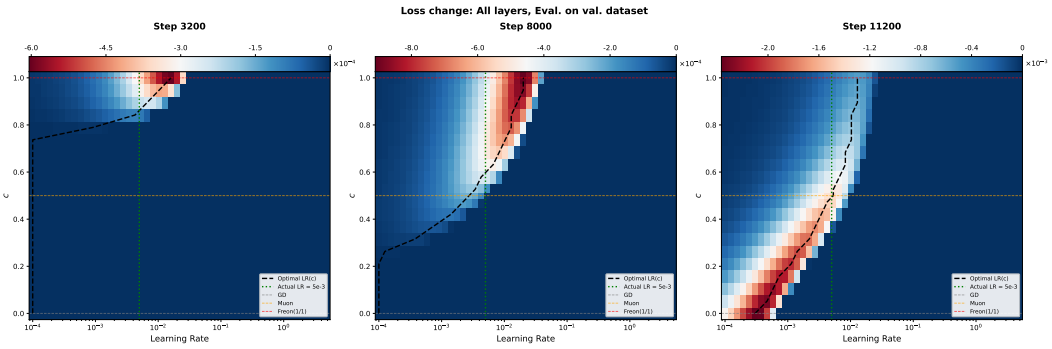


Figure J.6: Validation loss change $\Delta f(c, \eta) = f(\theta - \eta, D_c) - f(\theta)$, evaluated on the full validation set after applying a single synchronised update to all model layers, over a joint grid of direction interpolation $c \in [0, 1]$ (y-axis) and learning rate η (x-axis, logarithmic). Each panel corresponds to a different training checkpoint, i.e. the gradient G_k and momentum state used to construct the update direction are taken from that training step. For a given c, the per-layer update direction D_c is obtained by applying the Freon power-mean preconditioner with exponent c to the layer's gradient: $c = 0$ recovers the vanilla gradient-descent direction ($D = G/|G|_F$) and $c = 1$ is the fully preconditioned `Freon` direction; intermediate values interpolate smoothly between the two. The same c and η are applied to every weight matrix simultaneously, so Δf measures the *joint* effect of a global update rather than a per-layer optimum. The heatmap is clamped at zero so that only loss-decreasing regions (negative Δf) are shown; the colour axis is scaled independently per panel. The dashed black curve traces $\eta^*(c) = \arg\min_\eta \Delta f(c, \eta)$, i.e. the empirically optimal learning rate for each direction interpolation c. The green dotted vertical line marks the actual learning rate used during training, allowing direct comparison of where training operates relative to the optimal ridge.

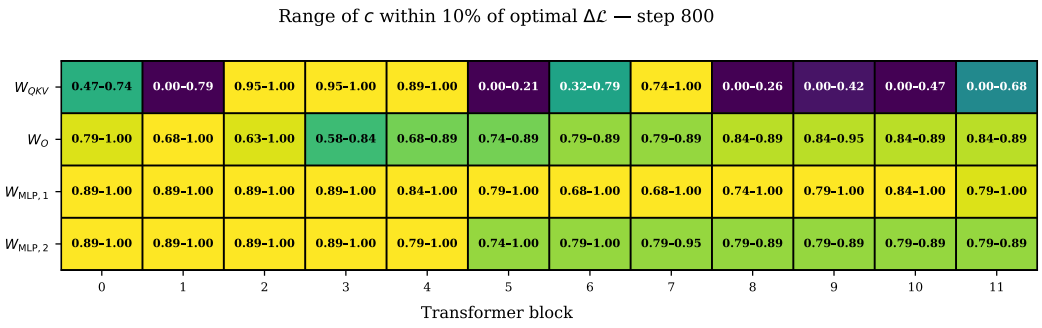


Figure J.7: **Layer-wise range of near-optimal c at step 800.** Each cell shows the range of values $c \in [0, 1]$ whose best-over-η one-step loss decrease is within 10% of the global optimum for that layer, found by a grid search over learning rates and c using real forward passes on the full validation set (actual mode). Cell colour indicates the optimal c. Rows correspond to the four weight matrices per transformer block (W_{QKV}, W_O, $W_{\text{MLP},1}$, $W_{\text{MLP},2}$); columns index the 12 transformer blocks. $c = 0$ recovers gradient descent, $c = \frac{1}{2}$ recovers Muon, and $c = 1$ recovers the pseudoinverse-like endpoint.

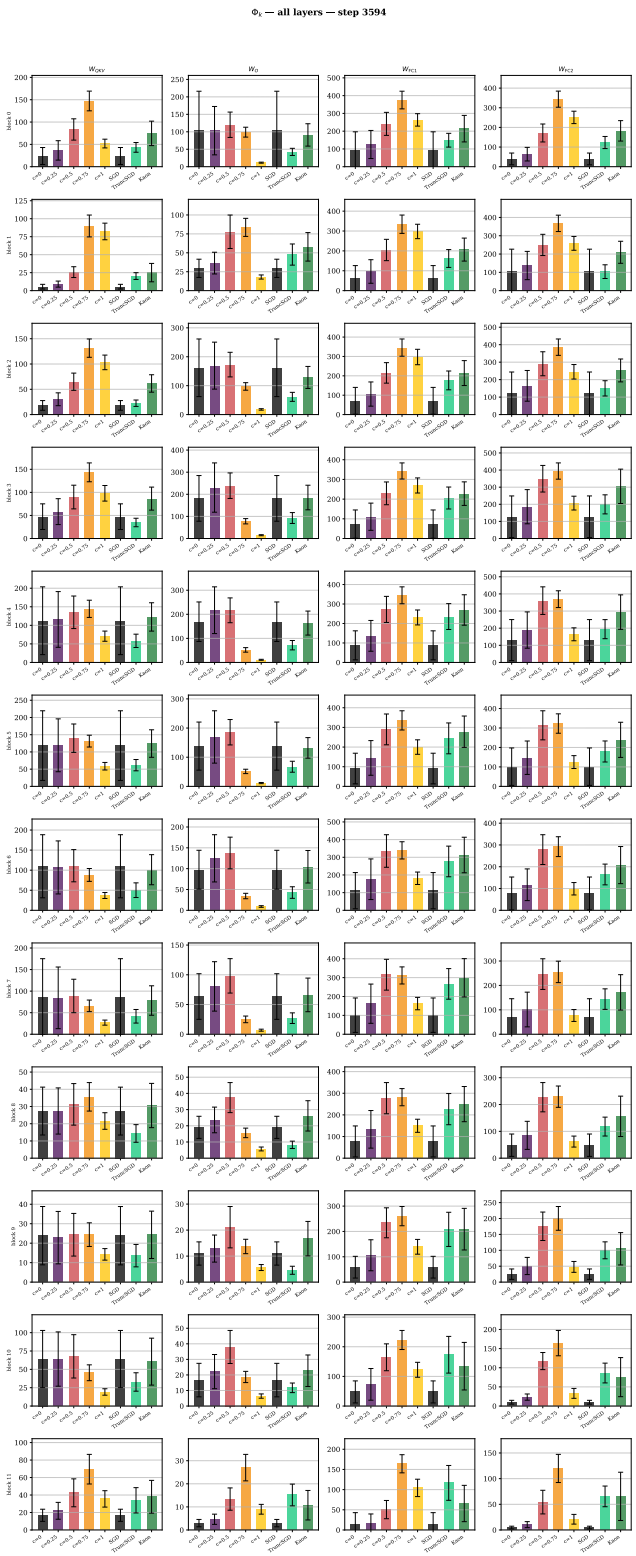


Figure J.8: Φ_k at training step 3594 ($\approx 25\%$ of training) for all 48 weight matrices, arranged as 12 transformer blocks (rows) $\times$ 4 sublayer types (columns: W_{QKV}, W_O, W_{FC1}, W_{FC2}). Each bar shows the mean $\pm$ std over validation batches for a given update direction.

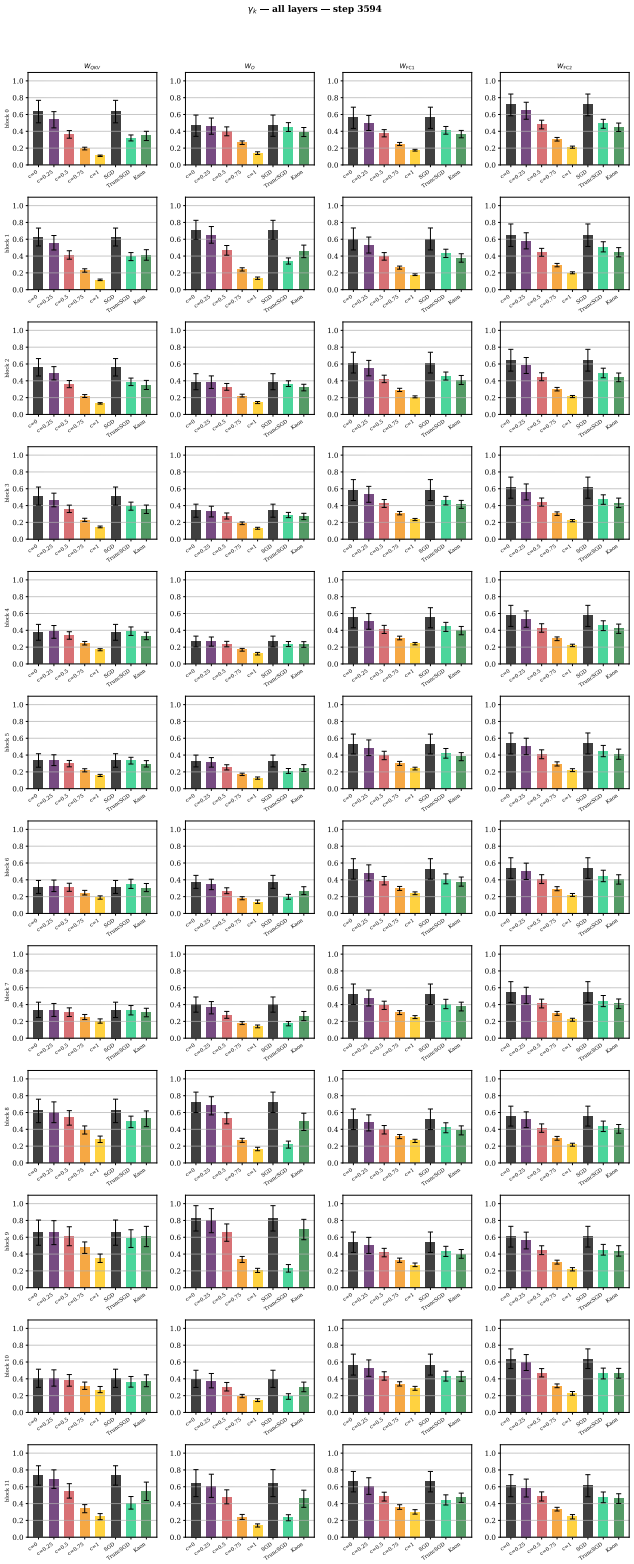


Figure J.9: Batch-gradient alignment $\gamma_k = \langle G, D\rangle / \langle g_b, D\rangle$ at training step 3594 for all 48 weight matrices (same layout as Fig. J.8). A value of $\gamma_k = 1$ indicates the direction D aligns perfectly with the full gradient; values below 1 reflect mini-batch noise or directional mismatch.

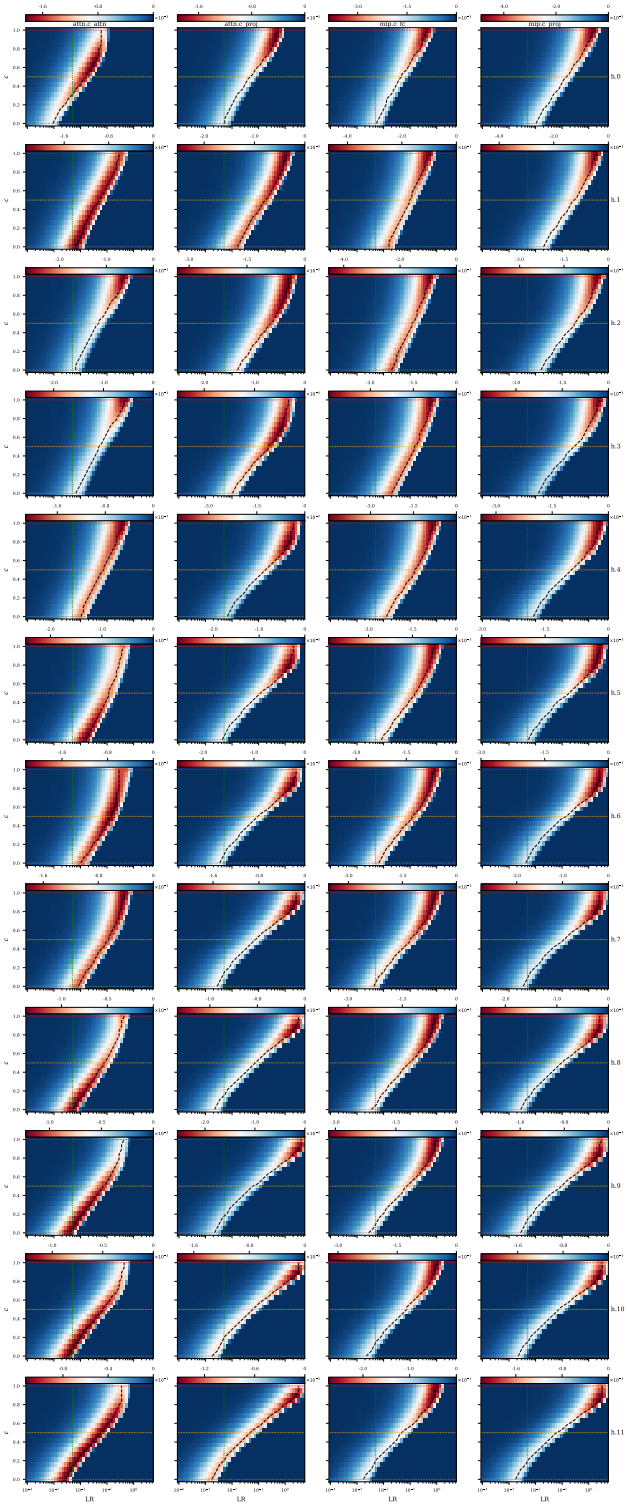


Figure J.10: Loss change Δf over the joint (c, η) grid at training step 800, evaluated per layer on the **full validation set**. Rows correspond to transformer blocks $h.0$–$h.11$; columns correspond to the four weight matrices within each block. The dashed black curve marks the optimal learning rate per c; the green dotted line marks the actual training learning rate.

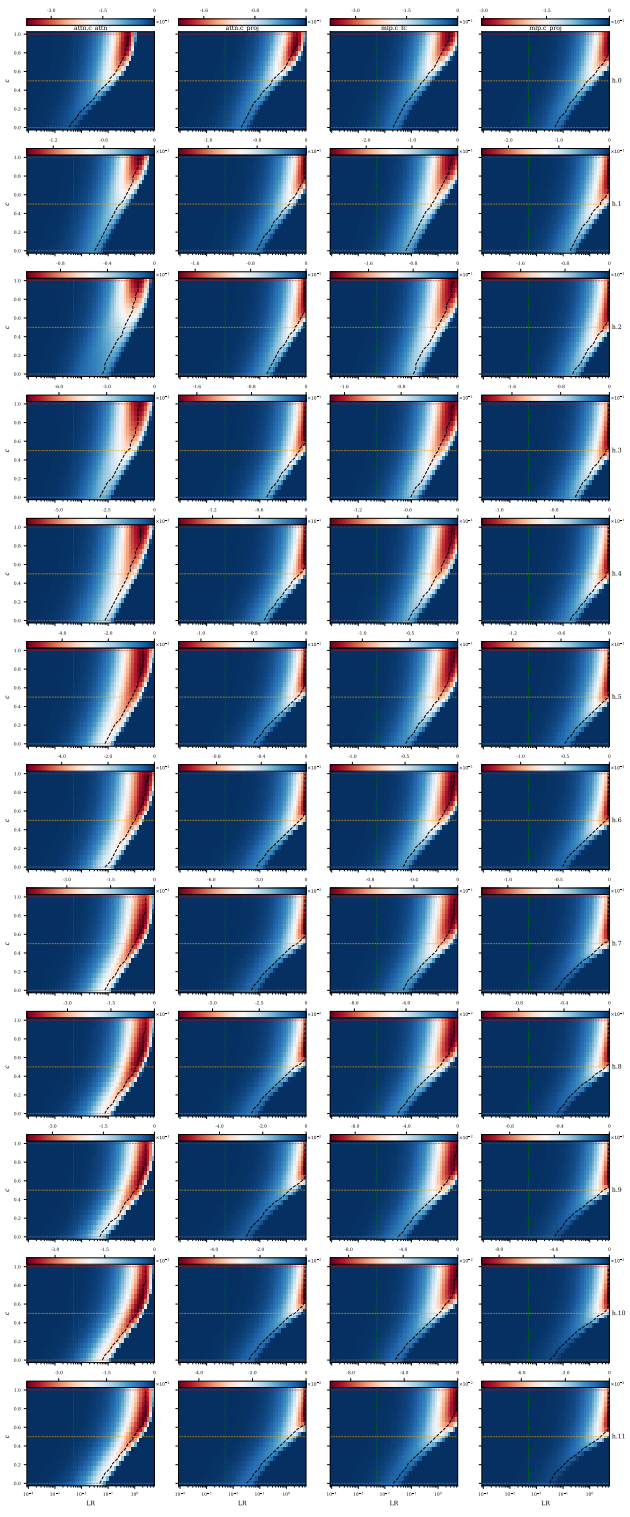


Figure J.11: Loss change Δf over the joint (c, η) grid at training step 800, evaluated per layer on the **same training batch as gradient**. Rows correspond to transformer blocks $h.0$–$h.11$; columns correspond to the four weight matrices within each block. The dashed black curve marks the optimal learning rate per c; the green dotted line marks the actual training learning rate.