MULTINORMSTEP: LEAVING THE CORNER OF STEEPEST DESCENT

Anonymous authors
Paper under double-blind review

ABSTRACT

First-order optimizers are the workhorse of large language model training, and after a decade in which AdamW was the *de facto* default, Muon has recently become the first optimizer to *consistently* beat a well-tuned AdamW across model and data scales. Yet Muon acts only on the 2D matrix weights; the remaining parameters (the normalization scales, biases, and most consequentially, the token embedding and LM head) are left as a compromise to a default AdamW. This split looks like two unrelated optimizers bolted together, but through the lens of steepest descent the pieces are more related than they appear: signSGD and Muon are the *same* ∞-norm corner read in two gauges, the ℓ_∞ box on coordinates and the Schatten-∞ ball on singular values, and Adam is in turn a *conservative* signSGD that shrinks every coordinate strictly inside that box. SignSGD sits aggressively at the corner and Adam timidly within it, yet neither leaves it. Rather than commit to a single corner, we constrain a *weighted family* of ℓ_p norms through a Young function, obtaining **MultiNormStep** (MN-step): a family with a closed-form coordinatewise update that both *includes* the single-norm methods (signSGD, PowerStep, normalized SGD) and *generates* new interior maps, a bounded *sigmoid* and an unbounded *log*. The identical construction lifts from coordinates to the spectrum, where it recovers Muon, so these maps and Muon are two faces of *one* steepest-descent principle. Under full-parameter training our bounded sigmoid step stays on par with a carefully tuned AdamW across GPT-2 124M/350M and Qwen3 0.6B/4B; its decisive role is the group Muon leaves to AdamW, where, in a Muon-hybrid that fixes Newton-Schulz Muon on the 2D weights and swaps only that branch, the sigmoid step beats the AdamW branch on validation at *both* Qwen3 0.6B and 4B and again outperforms AdamW at two scales of a proprietary mixture-of-experts model (Model-X). These consistent gains show that widening the geometry beyond a single norm genuinely helps, precisely on the parameters Muon leaves to AdamW.

1 INTRODUCTION

First-order optimizers are the workhorse of large-scale deep learning, and for most of the past decade Adam and its decoupled-decay variant AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019) have been the de facto default across architectures and scales. This has only recently changed: Muon (Jordan et al., 2024), scaled to billion-parameter language models by Moonlight (Liu et al., 2025), is the first optimizer to *consistently* outperform a well-tuned AdamW across model scales, a genuinely rare event in a field where AdamW had proven stubbornly hard to beat. Muon's success has refocused optimizer research on how the *matrix* (2D) parameters should be preconditioned, along two complementary lines. Through the lens of steepest descent, Muon is the update induced by a spectral matrix norm (Table 1), placing it in one family with SGD, signSGD and their relatives (Bernstein et al., 2018; Bernstein & Newhouse, 2024). Through the lens of Kronecker-factored preconditioning (Martens & Grosse, 2015), the Shampoo line (Gupta et al., 2018) and its successors SOAP (Vyas et al., 2025) and KL-Shampoo (Lin et al., 2026) have been put forward as strong alternatives for those same matrix parameters. Either way, the current frontier of optimizer design is centred squarely on getting the 2D weights right.

Yet remarkably little attention has gone to the parameters the standard recipe leaves to AdamW: the genuinely 1D normalization scales that govern activation magnitudes and the assorted biases,

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.

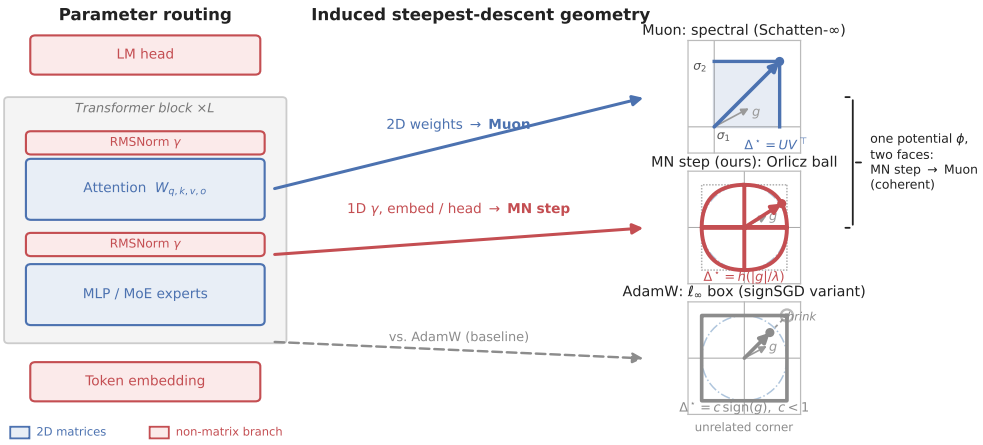


Figure 1: MN-step at a glance. *Left:* the Muon recipe keeps Muon on the 2D weights and defers the non-Muon branch (RMSNorm scales γ, token embedding, LM head) to a default AdamW; we route that branch through the MN-step. *Right:* each optimizer is a steepest-descent step under a norm ball: Muon the spectral corner ($\Delta^\star = UV^\top$), the MN-step an *interior* Orlicz ball between ℓ_2 and ℓ_∞, AdamW a shrunken ℓ_∞ corner (a signSGD variant). The Orlicz and spectral balls are two faces of *one* potential ϕ (§3.3), so the MN-step *mixes* norms between the corners, whereas AdamW only shrinks the signSGD corner and never leaves it.

Table 1: Steepest descent under different norm balls recovers familiar optimizers; the Orlicz ball generates our MN-step.

norm ball	direction	optimizer
ℓ_2 ball	$-g/\|g\|_2$	(normalized) SGD
ℓ_∞ ball	$-\operatorname{sign}(g)$	signSGD
ℓ_p ball	$-\operatorname{sign}(g) \odot \|g\|^{1/(p-1)}$	PowerStep
spectral (Schatten-∞) ball	$-UV^\top$	Muon
Orlicz ball $\mathcal{K}_\phi$	$-\operatorname{sign}(g)\, h(\|g\|/\lambda)$	**MN-step (ours)**

together with the token embedding and LM head that interface with the vocabulary (matrices Muon could act on, but which the prevailing recipe still defers). The prevailing recipe applies Muon (or a Shampoo-type method) only to the 2D hidden weights and defers every remaining parameter to a default AdamW (Jordan et al., 2024; Liu et al., 2025). This pairing looks like a patchwork of two unrelated optimizers bolted together, but through the steepest-descent lens it is more unified than it appears. By the correspondence between Schatten and vector norms, Muon is simply the *spectral* view of the very ℓ_∞ corner that signSGD occupies on coordinates: the two are one corner geometry in two gauges. AdamW, in turn, is a *shrinkage variant* of signSGD, a per-coordinate contraction that retreats just inside the same box (Orvieto & Gower, 2025). The leftover branch is therefore already governed by a member of the signSGD family, which turns the recipe into a sharper question: which member of that family is right for these parameters?

This is exactly where a single, fixed geometry becomes limiting. signSGD takes a full-magnitude step on every coordinate and is *aggressive*; AdamW contracts every coordinate and is *conservative*; and the single-ℓ_p interpolant between them (PowerStep) carries only one rigid geometric character, with no principled way to choose the best p (Lu et al., 2026). As Table 1 makes plain, every steepest-descent method to date commits to a *single* norm and hence a fixed geometry; the rich space between these isolated corners has gone largely unexplored.

We take exactly this unexplored direction. Rather than a single ℓ_p, we constrain a *weighted* family of ℓ_p norms through a Young function $\phi(s) = \sum_{p\geq 2} w_p s^p$, and define **MultiNormStep** (MN-step)

as the solution of the resulting steepest-descent problem

$$\Delta^\star = \arg\min_{\Delta} \langle g, \Delta \rangle \quad \text{s.t.} \quad \Omega_\phi(\Delta) := \sum_i \phi(|\Delta_i|) \le R, \qquad \phi(s) = \sum_{p \ge 2} w_p \, s^p. \tag{1}$$

Choosing the weight profile $\{w_p\}$ recovers the familiar corners of Table 1 as special cases; and, via the von Neumann trace inequality, the *same* construction lifts from coordinates (1D) to singular values (2D), turning one potential into a matrix-valued algorithm as well. A judicious choice of $\{w_p\}$ thus yields genuinely new optimizers carrying both a vector and a matrix face, a substantially larger design space, and a new vantage on how these updates relate. Concretely, $w_p = 1/p$ and $w_p = 1/p!$ give the *sigmoid* and *log* steps and across GPT-2 and Qwen3 pre-training they match a well-tuned AdamW while clearly improving on signSGD, showing that the multi-norm geometry does real work. Finally, in a Muon-hybrid study where Muon is fixed on the 2D weights and only the optimizer on the remaining parameters varied, our maps *surpass* AdamW on both a Qwen3 dense model and a proprietary mixture-of-experts model (*Model-X*), confirming the advantage of the multi-norm geometry and, once more, the value of taking the non-2D parameter group seriously.

Contributions.

- **A norm *family*, not a single norm.** We cast steepest descent under an Orlicz-ball constraint (Eq. 1), so that a *weighted family of norms*, rather than any one norm, fixes the update geometry. The framework both *subsumes and explains* existing single-norm methods (signSGD, PowerStep, normalized SGD, and, via a Schatten ball, Muon) and *generates* new optimizers carrying both a vector and a matrix face (§3).
- **Two concrete algorithms, with guarantees.** Instantiating the free weights $\{w_p\}$ yields **LogStep** and **SigmoidStep**, for each of which we establish an $O(1/\sqrt{T})$ non-convex convergence guarantee under a single unified analysis (§3.4, App. C.2).
- **The non-Muon group matters.** Across GPT-2 and Qwen3 at multiple scales our maps match a well-tuned AdamW while clearly improving on signSGD; in a Muon-hybrid setting on Qwen3 and the Model-X mixture-of-experts model they *stably surpass* both AdamW and signSGD on the non-Muon group. The gain reflects the richer multi-norm geometry of our maps, which a single fixed norm cannot provide (§4).

2 Related Work

Steepest descent and the choice of norm. A productive recent view interprets first-order optimizers as steepest descent under a chosen norm, so that the geometry of the trust region is what distinguishes one method from another (Bernstein et al., 2023). signSGD (Bernstein et al., 2018) and Lion (Chen et al., 2023) arise from the ℓ_∞ ball, normalized SGD from the ℓ_2 ball, the PowerStep (Lu et al., 2026) from the general ℓ_p ball, and Muon from the spectral (Schatten-∞) ball (Kovalev, 2025); the "old optimizer, new norm" program (Bernstein & Newhouse, 2024) and the modular-norm framework (Bernstein & Newhouse, 2025) make this dictionary explicit and compose norms across network layers. These accounts share a common feature: each optimizer corresponds to a fixed norm ball and thus every such method commits to a single fixed geometry. Consequently, the rich space between these isolated corners has remained largely unexplored.

Mirror descent and dual averaging. Mirror descent (Nemirovskij & Yudin, 1983; Beck & Teboulle, 2003) and dual averaging (Xiao, 2010) generalize gradient methods by taking steps in a dual space induced by a strongly convex potential and mapping back to the primal through the inverse mirror map, or *link*. This construction is tightly connected to steepest descent: choosing the potential to be half of a squared norm recovers steepest descent under that norm, so the link and the norm ball describe the same geometry from the primal and dual sides (Bernstein & Newhouse, 2024; Gunasekar et al., 2018; Amid & Warmuth, 2020). Despite this well-developed primal-dual correspondence, the mirror-descent viewpoint has seen little use for non-homogeneous potentials that mix multiple norms rather than committing to one.

Spectral and matrix-aware preconditioning. A parallel line targets the *matrix* parameters directly, moving beyond the diagonal preconditioning of AdaGrad (Duchi et al., 2011) and Adam / AdamW (Kingma & Ba, 2015; Loshchilov & Hutter, 2019). Muon (Jordan et al., 2024), scaled to billion-parameter models by Moonlight (Liu et al., 2025), semi-orthogonalizes the momentum through a Newton–Schulz iteration, while the Kronecker-factored methods precondition the 2D weights with second-order structure, including Shampoo (Gupta et al., 2018), SOAP (Vyas et al., 2025), KL-Shampoo (Lin et al., 2026), and the KFAC curvature model they build on (Martens & Grosse, 2015). These methods share a common blind spot: they act only on the hidden weights and defer the embeddings, LM head, norms and biases to a default AdamW.

3 Method: Mirror Descent under an Orlicz Geometry

3.1 Preliminaries and motivation

Steepest descent. Steepest descent takes the most g-aligned step inside a trust region of radius R,

$$\Delta^\star \;=\; \arg\min_{f(\Delta)\le R} \langle g, \Delta\rangle, \tag{2}$$

where f is the function defining the region: a norm, or more generally a convex function we constrain against. The choice of f fixes the update geometry, and different choices recover familiar optimizers (Table 1).

Schatten-∞ gives Muon; signSGD is its coordinate twin. Running equation 2 on the *spectrum* of a matrix gradient $G = U\Sigma V^\top$, with the constraint given by the Schatten-∞ (spectral) ball $\{\sigma_{\max} \le 1\}$, caps every singular value at the corner and yields $\Delta^\star = UV^\top$, which is exactly Muon's semi-orthogonalization (Jordan et al., 2024; Bernstein & Newhouse, 2024). This is the matrix analog of the coordinate ℓ_∞ corner, whose map is $\mathrm{sign}(g)$ (signSGD). Muon and signSGD are thus *one* corner geometry read in two gauges: on singular values, and on coordinates.

Adam is a shrinkage variant of signSGD. Fixing $\beta_1 = \beta_2 = \beta$, Orvieto & Gower (2025) show the Adam direction collapses to

$$d \;=\; \frac{m}{\sqrt{m\odot m + \sigma\odot\sigma}} \;=\; \frac{\mathrm{sign}(m)}{\sqrt{1 + (\sigma\odot\sigma)\oslash(m\odot m)}}, \tag{3}$$

where $\odot$ and $\oslash$ denote element-wise product and division, and σ is an online estimate of the gradient standard deviation (so $\sigma\odot\sigma$ is the coordinate-wise variance). This is signSGD's $\mathrm{sign}(m)$ scaled coordinate-wise by a factor ≤ 1 set by the (inverse) noise-to-signal ratio. Equivalently, Adam is steepest descent on an ℓ_∞ box whose radius has *shrunk* below one coordinate-wise, $|\theta_i| \le (1+\sigma_i^2/m_i^2)^{-1/2}$. Adam thus sits strictly *inside* the corner that signSGD and Muon occupy, as a conservative, per-coordinate contraction of the box. This collapse is an exact identity only at $\beta_1 = \beta_2$; for the unequal coefficients used in practice ($\beta_1 = 0.9$, $\beta_2 = 0.95$) we read it as a heuristic that still places well-tuned AdamW just inside the box corner.

Beyond a single ℓ_p. The single-norm methods above live at the extremes of the ℓ_p family. Yet an LLM gradient carries *both* bulk-typical mass (an ℓ_2 trait) and heavy-tailed extremes (an ℓ_∞ trait), and no single p, nor Adam's uniformly shrunk box, honors both at once. This motivates a constraint that mixes a *weighted family* of ℓ_p norms simultaneously: the Orlicz ball, developed next (a detailed comparison across p is deferred to App. F.1).

3.2 Steepest descent under an Orlicz ball

We constrain a *weighted family* of ℓ_p moments through a Young function $\phi(s) = \sum_{p\ge 2} w_p\, s^p$, and take the steepest-descent step inside the resulting Orlicz ball of radius R, against the heavy-ball momentum $m \leftarrow \gamma m + g$:

$$\Delta^\star = \arg\min_{\Delta} \langle m, \Delta\rangle \quad \text{s.t.} \quad \Omega_\phi(\Delta) := \sum_{i=1}^{n} \phi(|\Delta_i|) \;\le\; R. \tag{4}$$

Basic Orlicz-space facts (Musielak, 2006) (that Ω_ϕ is a modular, that $\mathcal{K}_\phi = \{x : \Omega_\phi(x) \le R\}$ is compact, convex and symmetric, and how ℓ_∞ arises as a limiting case) are collected in App. F.

General solution via KKT. Attaching a multiplier $\lambda \geq 0$ to the constraint, the Lagrangian is

$$\mathcal{L}(\Delta, \lambda) = \langle m, \Delta \rangle + \lambda\Big(\textstyle\sum_i \phi(|\Delta_i|) - R\Big). \tag{5}$$

Since ϕ is convex and separable, equation 5 decouples across coordinates; the stationarity condition $\partial\mathcal{L}/\partial\Delta_i = 0$ reads

$$m_i + \lambda\, \phi'(|\Delta_i|)\ \mathrm{sign}(\Delta_i) = 0 \iff \mathrm{sign}(\Delta_i) = -\,\mathrm{sign}(m_i), \quad \phi'(|\Delta_i|) = \frac{|m_i|}{\lambda}. \tag{6}$$

The derivative ϕ' is increasing, so it inverts to $|\Delta_i| = (\phi')^{-1}(|m_i|/\lambda)$; restoring signs and applying a learning rate η gives the update

$$\boxed{\Delta_i \;=\; -\,\eta\ \mathrm{sign}(m_i)\, h\big(|m_i|/\lambda\big), \qquad h := (\phi')^{-1} = (\phi^*)',} \tag{7}$$

with ϕ^* the convex conjugate. The link h is the mirror map of the Orlicz potential, so equation 7 is equivalently the mirror-descent / dual-averaging step under Ω_ϕ (Nemirovskij & Yudin, 1983; Beck & Teboulle, 2003; Xiao, 2010).

Setting the scale λ. Complementary slackness makes the ball size and the price two ends of one knob: at the solution $\Omega_\phi(\Delta^\star) = \sum_i \phi\big(h(|m_i|/\lambda)\big)$ is strictly decreasing in λ, so each radius R fixes a unique λ and vice versa (a larger ball means a smaller λ). We therefore drop R and parametrize the step directly by $\lambda > 0$, together with the learning rate η. What λ cannot be is arbitrary in *units*: both maps act through $|m|/\lambda$, which must be dimensionless, so λ has to scale like the gradient magnitude. Among the dimensionally admissible summaries, $\|g\|$ tunes poorly while $\mathrm{RMS}(g)$ is robust across model scales, so we set $\lambda \propto \mathrm{RMS}(g)$, estimated online exactly as Adam estimates its second moment: an exponential moving average of the squared gradients, followed by a square root and bias correction. This is also the small-R Taylor solution of the slackness identity for both maps (App. F.2). The pooling granularity used to compute RMS (per-tensor, per-row, per-column, or per-element) is a separate axis and we compare the choices in detail in §4.3.

The link h is where the geometry lives. Choosing the weight profile $\{w_p\}$ chooses h, and with it the geometry, recovering the familiar single-norm maps as special cases (Table 2). MN-step keeps the closed form equation 7 for *any* Young ϕ, opening a continuum between ℓ_2 and ℓ_∞; the two members we deploy are next.

From general theory to concrete algorithms. We instantiate equation 7 with two weight profiles (Table 2). The **log map** ($w_p = 1/p!$) is unbounded: its link grows without limit, so a large signal-to-noise coordinate can still produce a large step. The **sigmoid map** ($w_p = 1/p$) is bounded, with the Orlicz ball inscribed in the ℓ_∞ box $(-1, 1)^n$ that imposes a hard per-coordinate cap. This cap gives the sigmoid map the intuition we want: ℓ_2-adaptive in the bulk where signals are moderate, saturating like sign in the tail where a single coordinate would otherwise dominate, keeping it stable at scale. The full pseudocode (Alg. 1), per-map derivations and a probabilistic reading of the log map are deferred to App. F.

Is a non-ℓ_p constraint legitimate? The Taylor-based steepest-descent argument only needs a compact, convex, symmetric trust region, so an ℓ_p ball is not required. The set $\{\Omega_\phi \leq R\}$ is itself a norm ball, namely the unit ball of its Minkowski (Luxemburg) gauge, and the penalized form is a mirror-descent step whose Bregman potential Ω_ϕ need not be homogeneous (as with entropy on the simplex). Non-homogeneity simply means R reshapes the ball rather than rescaling it, which is the scale-adaptive geometry we want; convergence relies on strong convexity, not homogeneity. The only genuine requirement is a bounded step so the dropped second-order remainder stays controlled, which the sigmoid map enforces intrinsically and the log map obtains by clipping (App. F.3).

3.3 The matrix face: lifting to the spectrum

From coordinates to singular values. LLM parameters are predominantly matrices, where a coordinate-wise sign discards the spectral structure. We therefore lift the construction of §3.2 from coordinates to the *spectrum*, replacing each ℓ_p by its Schatten-p analog so the constraint reads

Table 2: Four maps as duals of a weight profile $\{w_p\}$. The two known single-norm methods are the edge cases (signSGD the hard-box corner, PowerStep the single moment); our two maps sit in the interior. Boundedness of h mirrors the growth of ϕ: super-linear $\phi \Rightarrow$ unbounded map; ϕ blowing up at a finite argument $\Rightarrow$ a hard per-coordinate cap.

map	w_p	$\phi(s)$	$h(x) = (\phi')^{-1}$	boundedness
signSGD	box ($p \to \infty$)	0 if $s \le 1$, else $+\infty$	$\equiv 1$ (= sign)	corner
PowerStep	single p	s^p	$x^{1/(p-1)}$	unbounded
log	$1/p!$	$e^s - 1 - s$	$\log(1+x)$	unbounded
sigmoid	$1/p$	$-s - \ln(1-s)$	$x/(1+x)$	bounded (< 1)

Table 3: One potential, two faces. The link $h = (\phi')^{-1}$ acts on coordinates (vector face) or on singular values (matrix face); both single-norm corners give Muon's $UV^\top$, and the sigmoid matrix map is a regularized Muon ($\to UV^\top$ as $\lambda \to 0$).

map	vector face	matrix face
signSGD	$\operatorname{sign}(m_i)$	$G(G^\top G)^{-1/2} = UV^\top$ (Muon)
PowerStep	$\operatorname{sign}(m_i)\,\lvert m_i\rvert^{1/(p-1)}$	$U\,\Sigma^{1/(p-1)}V^\top$ ($\to UV^\top$)
log	$\operatorname{sign}(m_i)\log\!\big(1+\frac{\lvert m_i\rvert}{\lambda}\big)$	$U\log\!\big(I+\frac{\Sigma}{\lambda}\big)V^\top$
sigmoid	$\operatorname{sign}(m_i)\dfrac{\lvert m_i\rvert}{\lambda+\lvert m_i\rvert}$	$G\big(\lambda I + (G^\top G)^{1/2}\big)^{-1}$

$\sum_j \phi(\sigma_j(\Delta)) \le R$. For a momentum SVD $G = U\Sigma V^\top$, the von Neumann trace inequality makes the inner linear-minimization oracle share G's singular vectors, collapsing the matrix problem to the same scalar problem on singular values solved coordinate-wise in §3.2, giving

$$\boxed{\Delta^\star \;=\; U\,h(\Sigma/\lambda)\,V^\top} \qquad h = (\phi')^{-1}, \tag{8}$$

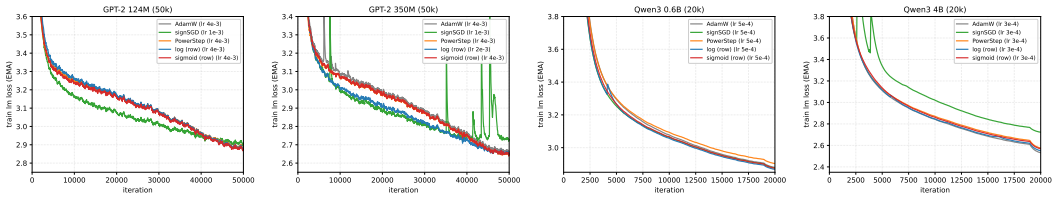
with h the same mirror map as in equation 7. Vector and matrix faces are one construction under two gauges: h acts on coordinates in the vector case and on singular values in the matrix case, mapping each row of Table 3.

Muon is the spectral corner of the family. Table 3 shows both single-norm corners collapse to $\Delta^\star = UV^\top$, exactly Muon's semi-orthogonalization (Jordan et al., 2024; Bernstein & Newhouse, 2024): Muon is not a separate heuristic but the $p \to \infty$ limit of the family. Our sigmoid matrix map is the interior λ-regularized Muon $G(\lambda I + (G^\top G)^{1/2})^{-1}$, interpolating Schatten-2 (matrix GD, $\lambda \to \infty$) and Schatten-∞ (Muon, $\lambda \to 0$). Pairing Muon on the matrices with our map on the remaining vectors thus keeps *one* geometry across the network rather than gluing two optimizers (Fig. 1): vectors carry no spectrum to orthogonalize, so the exact corner degenerates to a hard sign that drops magnitude, and we deploy instead the interior map that is ℓ_2-adaptive in the bulk and box-capped in the tail. One practical caveat: Muon has a Newton–Schulz iteration avoiding explicit SVD or inversion, whereas the log and sigmoid matrix faces admit no such shortcut and would need these expensive spectral operations, whose efficient solver we leave to future work.

3.4 Convergence

Theorem 1 ($O(1/\sqrt{T})$ to an $O(\sigma)$ neighborhood, momentum, both maps)**.** *Write the scale-normalized map $P_\lambda(u) = \lambda\,\operatorname{sign}(u)\,h(|u|/\lambda)$, so the implemented step $-\eta_0\,\operatorname{sign}(m)\,h(|m|/\lambda)$ equals $-\eta P_\lambda(m)$ with effective rate $\eta := \eta_0/\lambda$, on the heavy-ball buffer $m_t = \gamma m_{t-1} + g_t$. Let f be L-smooth with unbiased stochastic gradients of variance $\le \sigma^2$ and bounded gradient $\|\nabla f\| \le G$ (Assumptions 1–3). Running $x_{t+1} = x_t - \eta P_\lambda(m_t)$ for T steps with $\eta = \sqrt{2\Delta_f/(LMT)}$, $M = 2(G^2+\sigma^2)/(1-\gamma)^2$, $\Delta_f = f(x_0) - f^\star$, gives*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|P_\lambda(\nabla f(x_{t-1}))\|^2\big] \;\le\; \frac{1}{1-\gamma}\sqrt{\frac{2LM\Delta_f}{T}} \;+\; \frac{G\sigma}{(1-\gamma)^{3/2}} \;=\; O\Big(T^{-1/2}\Big) + O(\sigma). \tag{9}$$



(a) GPT-2 124M, 50k (own best lr). (b) GPT-2 350M, 50k (own best lr). (c) Qwen3 0.6B, 20k (shared lr $5e{-4}$). (d) Qwen3 4B, 20k (shared lr $3e{-4}$).

Figure 2: Cross-scale *training* loss for the five full-parameter optimizers (AdamW, signSGD, PowerStep, and the log / sigmoid maps at row granularity). On GPT-2 each method uses its own best lr; on Qwen3 a single shared lr per scale (App. D). At 4B the log map's update is clipped to magnitude ≤ 1. Final numbers in Table 9; validation curves in App. Fig. 6.

The bound is stated in the map's own norm, the exact quantity the oracle produces; since P_λ is an odd bijection with $P_\lambda(0) = 0$ and $P'_\lambda(0) = 1$, this norm vanishes iff ∇f does, so it converts to an ordinary $\|\nabla f\|^2$ guarantee. The full assumptions, lemmas, and proofs are given in App. C.2.

Remark 1 (Scope: fixed λ vs. the deployed schedule). Theorem 1 treats λ as a fixed scalar, the stationary idealization of the deployed online, group-wise $\lambda_{t,j} = \sqrt{\mathrm{EMA}_{\beta_2}(\overline{g^2})}$ (§3.2), which stays representative because $\lambda_{t,j}$ is a bounded, slowly varying EMA and per-group pooling only makes the preconditioner block-diagonal, so the separable argument carries over group by group (App. C.2).

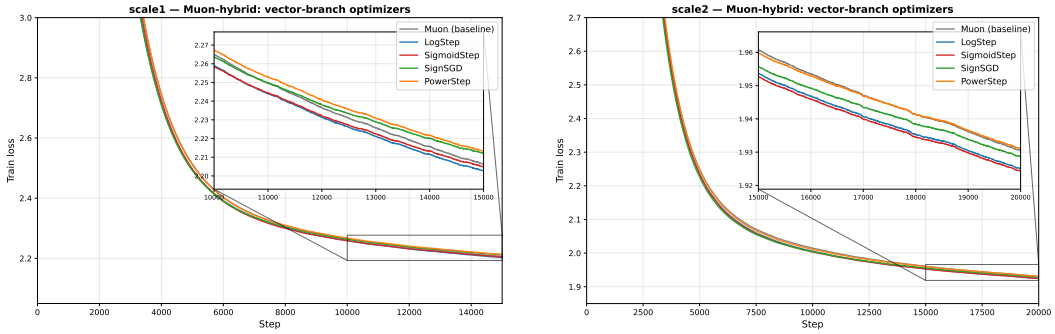
4 Experiments

We evaluate the MN-step maps in three parts across three codebases and model families: **GPT-2** (Radford et al., 2019) (nanoGPT, OpenWebText (Gokaslan et al., 2019)), **Qwen3** (Yang et al., 2025) (Megatron (Shoeybi et al., 2019), C4 (Raffel et al., 2020)), and **Model-X**, a proprietary fine-grained MoE trained on a large-scale trilingual corpus (scale1 1.4B, scale2 4.9B). §4.1 is the headline *full-parameter* comparison, where the optimizer trains *every* parameter on two GPT-2 scales (124M, 350M) and two Qwen3 scales (0.6B, 4B). §4.2 turns to the *Muon hybrid*: Newton–Schulz Muon is held fixed on the 2D matmul weights and only the optimizer on the remaining *non-Muon* parameters is swapped, on the two Qwen3 scales and two Model-X scales. §4.3 supplies the λ-granularity ablations; full model, framework, and data specifications are deferred to App. B.

Setup. GPT-2 trains for 50k steps, Qwen3 for 20k, and Model-X for 15k (scale 1) / 20k (scale 2), with a 2000-step warmup for GPT-2 and Qwen3 and 1000 for Model-X. All optimizers share momentum $\beta_1 = 0.9$, $\beta_2 = 0.95$ and decoupled weight decay 0.1; PowerStep additionally fixes its exponent $\beta_p = 0.1$ following the optimum of the original paper (Lu et al., 2026). GPT-2 and Qwen3 decay the lr on a cosine schedule to one tenth of the peak, whereas Model-X uses a linear warmup then a constant rate. Learning rates are chosen to be fair to every optimizer: full-parameter GPT-2 reports each method at its own best lr (per-method grid in App. D), whereas Qwen3 fixes a *single* shared lr per scale at AdamW's own optimum ($5e{-4}$ at 0.6B, $3e{-4}$ at 4B). The Muon hybrid likewise shares *one* base lr across the 2D and 1D parameters, the 2D (Muon) update additionally carrying Moonlight's adaptive scaling $0.2\sqrt{\max(d_{\text{in}}, d_{\text{out}})}$ (Liu et al., 2025).

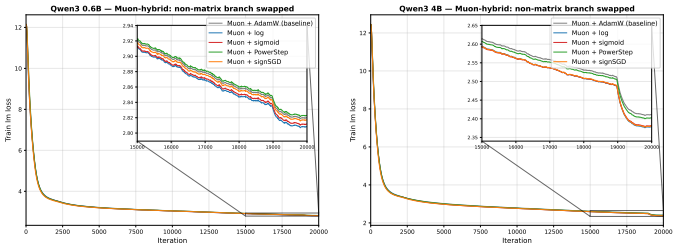
4.1 Full-parameter cross-scale comparison

Aggressive corners lead early but destabilize; the sigmoid map matches AdamW. We ask only that our maps stay on par with a well-tuned AdamW, so the Muon-hybrid gain (§4.2) is not confounded by a weak baseline; we compare AdamW, signSGD, PowerStep, and our log and sigmoid maps (row-granularity λ, Fig. 2). The aggressive maps, log and signSGD, take near full-magnitude steps and lead early but are the first to destabilize as scale grows: Fig. 2 shows signSGD oscillating sharply and the log map spiking, and at Qwen3 4B the log map needs its update clipped to stay stable. The unbounded log map is still the stronger of the two, staying with the field where



(a) Model-X scale 1 (bsz 448, lr $1e{-3}$, 15k). (b) Model-X scale 2 (bsz 896, lr $9e{-4}$, 20k).

Figure 3: Muon-hybrid on Model-X: Newton–Schulz Muon is held fixed on the matrices and only the non-Muon branch is swapped (Moonlight-default AdamW vs. our interior log / sigmoid maps and the two corner maps signSGD / PowerStep). Scale 1 is a 15k-step run, scale 2 a 20k-step run; the inset zooms the tail of each (scale 1 10–15k, scale 2 15–20k), where the interior log / sigmoid maps lead the AdamW baseline. Training loss in Table 4.



(a) Qwen3 0.6B training lm loss. (b) Qwen3 4B training lm loss.

Figure 4: Muon-hybrid on Qwen3: Newton–Schulz Muon fixed on the 2D weights, only the non-Muon branch swapped (`puremuon`=Muon+AdamW is the Moonlight default). The log / sigmoid maps sit below the AdamW baseline throughout; the tail inset shows the gap holding to 20k steps and widening at 4B on the training curve. Final training loss in Table 4; validation in Table 11.

	Qwen3		Model-X	
branch	0.6B	4B	s1	s2
AdamW	2.820	2.410	2.200	1.926
signSGD	2.817	2.379	2.207	1.924
PowerStep	2.823	2.401	2.207	1.926
log	**2.808**	**2.377**	**2.197**	1.920
sigmoid	2.812	2.380	2.199	**1.920**

Table 4: Final training loss (last-500-step average) for the Muon-hybrid, Muon fixed on matrices and the non-Muon branch swapped. Best per column in **bold**. The log / sigmoid maps win at every scale, their margin over AdamW growing with size (validation in App. Table 11).

signSGD, the exact box corner, falls behind. The sigmoid map instead reaches essentially the same final training loss as the well-tuned AdamW across all four scales, within a few thousandths of a nat: ℓ_2-adaptive in the bulk and capped like sign in the tail, it neither over-steps like the aggressive maps nor over-damps like AdamW's shrunken box or PowerStep's compressive link.

4.2 Muon-hybrid: the non-Muon branch

Holding Newton–Schulz Muon fixed on the 2D matmul weights, we vary *only* the optimizer on the branch Muon leaves to AdamW under the Moonlight recipe (embeddings, LM head, biases, normalization scales), swapping in the log or sigmoid map. This branch mixes genuine 1D tensors (norm scales, biases) with the embedding and LM-head matrices. Here λ is pooled per tensor rather than per row, as much of the branch has no natural row axis. Any difference then isolates the map's geometry where it acts, with the matrix update, the Moonlight matrix scaling, and the single shared lr held identical ($2e{-3}$/$1e{-3}$ at 0.6B/4B, AdamW-scale; setup above). We compare on Qwen3 (Fig. 4) and the Model-X MoE (Fig. 3).

Qwen3: the box corner is strong, the interior stronger. On training loss signSGD already edges AdamW at both scales (Table 4), the coordinate face of the very Schatten/box geometry Muon runs on the matrices, so the box corner is a strong step, not a weak one. The log and sigmoid maps

Table 5: λ-pooling granularity (GPT-2 124M, final val loss; best in **bold**).

map	tensor	row	column	element
log	2.935	**2.905**	2.932	2.919
sigmoid	2.918	**2.900**	2.916	2.902

Table 6: λ-EMA coefficient β_2 (GPT-2 124M, row-pooled, final val loss).

map	$\beta_2 = 0$	0.9	0.95	0.99
log	2.955	2.906	**2.905**	2.909
sigmoid	2.914	2.904	**2.900**	2.900

go further and lead throughout, the gap widening at 4B: evidence that a weighted family of norms improves on AdamW's shrunken box and PowerStep's compressive link.

Model-X: the advantage carries across architecture and scale. On Model-X the non-Muon branch additionally carries the over-embedding (Huang et al., 2025) parameters, and the AdamW baseline is so carefully tuned that at scale 1 it overtakes signSGD, exposing signSGD as too aggressive here. The log and sigmoid maps nonetheless remain the best branches at both scales (Table 4), so their advantage holds across architecture and grows with scale, mirroring Qwen3.

4.3 Ablations and discussion

Two ablations isolate the design of $\lambda = \sqrt{\mathrm{EMA}_{\beta_2}(\overline{g^2})}$ on GPT-2 124M at each map's own best lr $= 4\mathrm{e}{-3}$ (50k, cosine, wd 0.1; row baselines reuse §4.1), moving one axis of λ at a time with the link h and momentum fixed.

Where to pool λ. Pooling the RMS that sets λ per output *row* is best for both maps (Table 5): per-row λ matches each neuron's own gradient scale, whereas coarser per-column / per-tensor poolings make neurons of very different magnitude share one normalizer (costing 0.01–0.03 nat) and per-element λ gains nothing over row. Row-pooled statistics are far smaller than a per-parameter buffer, so relative to Adam our maps carry essentially one fewer full optimizer state (App. E).

How much to smooth λ. λ is a running second moment with EMA coefficient β_2 (Table 6). The EMA is necessary: dropping it ($\beta_2 = 0$, an instantaneous $\sqrt{\overline{g^2}}$) is clearly worst for both maps, but any smoothing is robust, moving by at most 0.004 nat across $\beta_2 \in \{0.9, 0.95, 0.99\}$; we adopt $\beta_2 = 0.95$, Adam's default.

Which geometry for which parameters. Our maps match AdamW full-parameter yet beat it in the Muon hybrid with a margin that *widens* with scale, and one principle reconciles both: an optimizer is a preconditioner whose value depends on whether its geometry fits the block. Hidden matrices carry cross-coordinate Kronecker curvature (Martens & Grosse, 2015; Gupta et al., 2018) where the spectral map (Muon) wins and Adam and ours tie, whereas the embedding / LM-head / norm block is Zipf (Zipf, 2016) heavy-tailed, where the concave link h gives ours $>$ Adam $\gg$ Muon; the full-parameter edge is diluted as the non-Muon mass fraction vanishes with scale (full argument and per-block tail statistics in App. A).

5 Conclusion

MN-step casts update-magnitude priors as mirror descent under an Orlicz geometry: the weight profile $\{w_p\}$ of a Young function ϕ selects the mirror map $h = (\phi^*)'$, unifying signSGD, PowerStep, normalized SGD, AdamW and, on singular values, Muon. The box-inscribed sigmoid map (ℓ_2 in the bulk, an ℓ_∞ tail cap) matches well-tuned AdamW at full parameter and beats it on the non-Muon branch (Qwen3, Model-X). We do not improve on Muon for matrices; efficient spectral solvers, learning $\{w_p\}$, and theory beyond $O(1/\sqrt{T})$ remain future work.

AI use statement

In this work, we used generative AI tools to assist in the writing of proofs, to implement methods, to assist with translation, to support qualitative and thematic data analysis, and to interpret results. We

have not used generative AI tools to help develop theoretical models or conceptual frameworks, to formulate mathematical claims, to provide critical ingredients for proving those claims, to propose or refine hypotheses, to design or provide feedback on our research methodology or experiments, or to clean and reformat datasets; the generation of synthetic data sets is not applicable to this work, as all experiments use existing corpora (OpenWebText, C4) and a proprietary training corpus rather than AI-generated data. Additionally, we used generative AI tools to edit the paper for readability and to help format references. We have reviewed all AI-assisted work: the Orlicz-ball derivations and convergence proofs were checked and modified line by line by the authors; all AI-assisted implementations were verified and tested for correctness by the authors against reference optimizer implementations; and every AI-assisted translation, data-analysis summary, and result interpretation was independently confirmed against the underlying experimental logs and figures. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Our anonymized source code, including the optimizer implementations and the training and evaluation scripts, is available at `https://anonymous.4open.science/r/multi-norm-step-B4F3/`. Full model, framework, and data specifications, together with the experimental and installation details (codebases, hyperparameters, learning-rate grids, and data-processing steps) needed to reproduce our runs, are given in App. B and App. D. The assumptions and complete proofs of our convergence guarantee are provided in App. C.2.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebrón, and Sumit Sanghai. GQA: training generalized multi-query transformer models from multi-head checkpoints. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

Ehsan Amid and Manfred K. Warmuth. Reparameterizing mirror descent as gradient descent. In *Conference on Neural Information Processing Systems*, 2020.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

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

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. In *International Conference on Machine Learning*, 2025.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SIGNSGD: compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018.

Jeremy Bernstein, Chris Mingard, Kevin Huang, Navid Azizan, and Yisong Yue. Automatic gradient descent: Deep learning without hyperparameters. *arXiv preprint arXiv:2304.05187*, 2023.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms. In *Conference on Neural Information Processing Systems*, 2023.

John C. Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus, 2019.

Suriya Gunasekar, Jason D. Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, 2018.

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

Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, and Xun Zhou. Over-tokenized transformer: Vocabulary is generally worth scaling. In *International Conference on Machine Learning*, 2025.

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

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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

Wu Lin, Scott C Lowe, Felix Dangel, Runa Eschenhagen, Zikun Xu, and Roger Grosse. Understanding and improving shampoo and soap via kullback-leibler minimization. In *International Conference on Learning Representations*, 2026.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, et al. Muon is scalable for LLM training. *arXiv preprint arXiv:2502.16982*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Yao Lu, Dengdong Fan, Shixun Zhang, and Yonghong Tian. Powerstep: Memory-efficient adaptive optimization via ℓ_p-norm steepest descent. *arXiv preprint arXiv:2605.10335*, 2026.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, 2015.

Julian Musielak. *Orlicz spaces and modular spaces*. Springer, 2006.

Arkadij Semenovič Nemirovskij and David Borisovich Yudin. Problem complexity and method efficiency in optimization. 1983.

Antonio Orvieto and Robert Gower. In search of adam's secret sauce. In *Conference on Neural Information Processing Systems*, 2025.

Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv:1911.02150*, 2019.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

R. Vershynin. *High-Dimensional Probability*. Cambridge University Press, 2018.

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 *International Conference on Learning Representations*, 2025.

Lin Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *Journal of Machine Learning Research*, 11:2543–2596, 2010.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio books, 2016.

A WHICH GEOMETRY FOR WHICH PARAMETERS

The two experiments look paradoxical at first: under *full-parameter* training the log / sigmoid / power maps only match AdamW and lose their edge as the model grows (§4.1), yet in the *Muon-hybrid* they beat the AdamW branch and the margin *widens* with scale (§4.2). Both follow from a single principle: an optimizer is a preconditioner, and its value depends on whether its geometry matches the parameter block it acts on. The main-text §4.3 states this principle in one paragraph; here we give the full argument and the gradient-tail measurements behind it.

A common preconditioner form. Write every update on momentum m as $\Delta = -\eta\, P^{-1} m$. Then

$$\text{Adam:}\quad P = \operatorname{diag}(\sqrt{v}),\ v = \mathbb{E}[g \odot g] \qquad \text{(diagonal)}, \tag{10}$$

$$\text{Muon:}\quad \Delta = -\eta\, UV^\top = -\eta\,(MM^\top)^{-1/2} M,\ M = U\Sigma V^\top \qquad \text{(spectral)}, \tag{11}$$

$$\text{ours:}\quad \Delta_i = -\eta\, \operatorname{sign}(m_i)\, h\big(|m_i|/\lambda\big),\ h = (\phi')^{-1} \qquad \text{(coordinatewise, nonlinear)}. \tag{12}$$

For a quadratic with Hessian H the rate is governed by $\kappa\big(P^{-1/2} H P^{-1/2}\big)$, minimized at $P = H$. Adam and our maps are both *diagonal* in the coordinate basis (our maps additionally reshape the magnitude through h); Muon is the only one that acts across coordinates. Which is better therefore depends on the geometry of H, and this differs sharply between the hidden matrices and the non-Muon (embedding / LM-head / norm) parameters.

Matrix block: spectral, light-tailed ⇒ Muon wins, the diagonal methods tie. A hidden-weight gradient entry $g_{ij} = \sum_t a_i^{(t)} e_j^{(t)}$ is a sum over the tokens in the batch, so by a central-limit effect it is close to Gaussian and, crucially, *homogeneous in scale* across the matrix. Its curvature carries strong cross-coordinate (Kronecker) structure $H_{\text{mat}} \approx A \otimes B$ (Martens & Grosse, 2015; Gupta et al., 2018), whose eigenvectors are not axis-aligned. A diagonal P cannot reduce κ below this off-diagonal anisotropy, whereas Muon's $(MM^\top)^{-1/2}$ removes one Kronecker factor and sharply lowers it. Among the diagonal methods there is little to gain: the block is light-tailed, so the saturation in h has no heavy tail to tame and is, if anything, a mild bias. Hence on matrices MUON $\gg$ ADAM $\gtrsim$ ours, consistent with AdamW slightly leading our maps in the full-parameter 4B run.

Vector block: Zipf-heavy-tailed, no spectral structure ⇒ Muon degenerates, our maps win. Treating $v \in \mathbb{R}^d$ as a $d \times 1$ matrix, its SVD has a single triplet $U = v/\|v\|_2,\ \Sigma = \|v\|_2,\ V = 1$, so

$$\Delta_{\text{MUON}} = -\eta\, UV^\top = -\eta\, \frac{m}{\|m\|_2}, \qquad \text{i.e. } P_{\text{MUON}} \propto \|m\|_2\, I. \tag{13}$$

Muon collapses to a single global scalar for all d coordinates, an *isotropy* assumption. But embedding / LM-head rows are the opposite of isotropic: a row receives signal in proportion to its token's frequency, which is Zipfian (Zipf, 2016), so the per-coordinate scales σ_i are heavy-tailed and $\kappa = \sigma_{\max}/\sigma_{\min}$ stays enormous under $P \propto I$. This block needs a *coordinatewise* preconditioner. Adam supplies one, but its linear $1/\sqrt{v}$ rescaling is *first-order* sensitive to a mis-estimated scale ($\hat{\lambda} = \lambda(1+\varepsilon) \Rightarrow$ update error $\propto \varepsilon$), whereas the concave h is only *logarithmically* sensitive (App. C.1) and bounds the contribution of outlier coordinates. On heavy tails this is strictly better, so on the non-Muon block ours $>$ ADAM $\gg$ MUON.

Empirical tails confirm the split. We quantify tail heaviness with two statistics. The *entrywise excess kurtosis* is the standardized fourth moment of the gradient entries minus 3: it is 0 for a Gaussian and grows large-positive when a few entries dwarf the rest, so it reads directly as "how

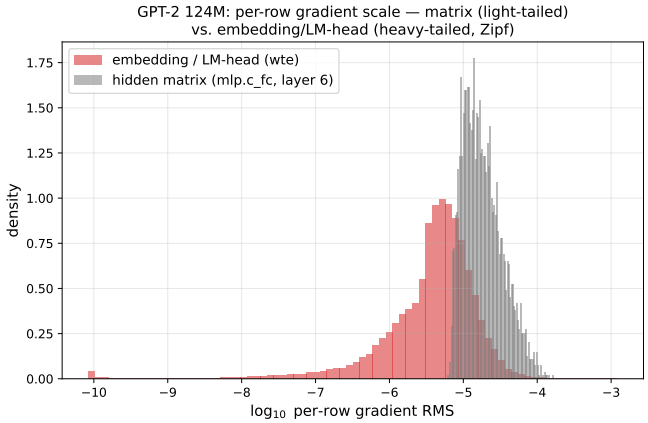


Figure 5: Per-row gradient RMS on GPT-2 124M (log scale). The hidden matrix is tightly concentrated (~ 1.5 decades); the tied embedding / LM-head spreads over ~ 6 decades with a long low-RMS tail (rare tokens), the Zipf signature. This is the geometry Muon cannot see (it applies one scalar to the whole vector) and that the coordinatewise Orlicz maps handle better than Adam.

Table 7: Gradient-tail statistics by parameter group, GPT-2 124M (median across tensors). Higher excess kurtosis / per-row spread = heavier tail. The Zipf-fed token embedding is orders of magnitude heavier than the hidden matrices; the uniform-fed position embedding is not.

parameter group	excess kurtosis	per-row RMS $p_{99.5}/p_{50}$
hidden matrix (attn/MLP)	4.4×10^2	3.5
token embedding / LM-head (Zipf)	2.4×10^5	12.4
position embedding (uniform)	3.3	2.1
norm (1D)	2.4×10^2	–

much a handful of coordinates dominate." The *per-row RMS spread* $p_{99.5}/p_{50}$ is the ratio of a near-maximal (99.5th percentile) to the median per-output-row gradient RMS, so it is ≈ 1 when all rows carry comparable signal and grows when a few rows are far louder than the rest. Both are $O(1)$ for a well-conditioned block and blow up on a heavy-tailed one. Figure 5 and Table 7 measure the accumulated gradient of a converged GPT-2 124M ($\sim 6.6 \times 10^5$ tokens). The hidden matrices are comparatively light-tailed and homogeneous (per-row RMS spread $p_{99.5}/p_{50} \approx 3.5$), while the tied embedding / LM-head is orders of magnitude heavier (entrywise excess kurtosis $\sim 2.4 \times 10^5$ vs. 4.4×10^2; per-row spread ≈ 12.4). The decisive control is the *position* embedding: same shape as the token embedding but fed by a *uniform* position frequency rather than a Zipfian one, and it is nearly Gaussian (excess kurtosis 3.3, spread 2.1). The tail is therefore a property of token-frequency statistics, not of "being an embedding", exactly the structure the bounded / sub-exponential maps are built to exploit.

Why the scale trends flip. The full-parameter metric is a parameter-count-weighted average over blocks. With hidden matrices scaling as $\sim 12Ld^2$ and the non-Muon group as $\sim 2Vd$, the non-Muon mass fraction $f_{\text{vec}} \sim V/(6Ld) \to 0$ as width and depth grow. Since our maps tie AdamW on the (dominant) matrix block,

$$\text{Edge}_{\text{full}} \approx f_{\text{mat}} \underbrace{\delta_{\text{mat}}}_{\approx 0} + f_{\text{vec}}\, \delta_{\text{vec}} \xrightarrow{\text{scale}} 0, \tag{14}$$

so the advantage is diluted away. The Muon-hybrid instead *holds the matrix branch identical*, removing f_{mat} from the comparison and exposing δ_{vec} directly; and δ_{vec} *grows* with scale because (i) larger vocabularies and token budgets make the embedding tail heavier, and (ii) once Muon has conditioned the matrices, the non-Muon block becomes the binding constraint. This is exactly the observed pattern: washed out at full-parameter, amplified in the hybrid. In short, Muon and our

maps are not competitors but the two halves of one steepest-descent picture: spectral geometry for the matrices, Orlicz coordinate geometry for the heavy-tailed remainder that Muon leaves to Adam.

B TRAINING DETAILS

This appendix gives the full specification of the three model families, training codebases, and corpora used in the experiments (§4).

GPT-2 (nanoGPT). The GPT-2 experiments use the nanoGPT codebase, a compact reimplementation of the GPT-2 decoder-only transformer (Radford et al., 2019) with learned absolute position embeddings, GELU MLPs, pre-LayerNorm blocks, and weight tying between the token embedding and the LM head. We train two standard scales: the 124M "small" model (12 layers, 12 heads, width 768) and the 350M "medium" model (24 layers, 16 heads, width 1024), both at a 1024-token context and the GPT-2 BPE vocabulary (50,304 padded entries). All full-parameter runs use 50k optimizer steps with a 2000-step linear warmup and cosine decay to one tenth of the peak rate, per-optimizer learning rates selected from the grid described in App. D.

Qwen3 (Megatron). The Qwen3 experiments use the Megatron framework (Shoeybi et al., 2019), a tensor-, pipeline-, and sequence-parallel training system for large transformers. The architecture is the Qwen3 dense decoder (Yang et al., 2025) (RMSNorm, SwiGLU MLPs, rotary position embeddings, grouped-query attention, and QK-normalization) at two scales, 0.6B and 4B parameters. Runs are 20k iterations with a 2000-step warmup; the 4B model is trained with pipeline parallelism (two stages) and the 0.6B model on a single pipeline stage. Both scales use a single shared learning rate per scale ($5e{-4}$ at 0.6B, $3e{-4}$ at 4B), anchored to AdamW's optimum as detailed in App. D.

OpenWebText. OpenWebText (Gokaslan et al., 2019) is an open reproduction of the WebText corpus used to train the original GPT-2, consisting of English web documents scraped from outbound Reddit links with a minimum-score filter, deduplicated and lightly cleaned. It contains on the order of 8B GPT-2 BPE tokens and serves as the pretraining data for both nanoGPT scales.

C4. C4 (the Colossal Clean Crawled Corpus) (Raffel et al., 2020) is a large English web corpus derived from Common Crawl with heavy heuristic cleaning: deduplication, language identification, and removal of boilerplate and offensive content. It is a standard, well-understood pretraining corpus and is used as-is for both Qwen3 scales.

Model-X. Model-X is a proprietary production-scale fine-grained mixture-of-experts transformer. We highlight the design elements that bear directly on our optimizer study. It adopts an *over-embedding* parameterization (Huang et al., 2025) that expands the input representation; because these weights are routed to the non-Muon (AdamW) group, they enter the swapped branch of our Muon-hybrid comparison, enlarging the parameter set our maps govern. A *KV-mirror* scheme (Shazeer, 2019; Ainslie et al., 2023) shares the key/value projections to reduce the key-value cache, reshaping which projection weights remain as 2D Muon-governed matrices. Dedicated *normalization* placements stabilize expert routing at scale, and the associated scale parameters are 1D and likewise fall to the swapped branch. Finally, its pretraining corpus is exceptionally diverse, a large-scale trilingual mixture, which stresses optimizer robustness across heterogeneous token distributions. We train two scales that differ in width, depth, and batch size (global batch 448 sequences at peak learning rate $1e{-3}$ for scale 1, and 896 sequences at $9e{-4}$ for scale 2, each selected by the sweep in Table 8), for 15k (scale 1) and 20k (scale 2) iterations with a 1000-step warmup. Both Model-X scales are run only in the Muon-hybrid configuration of §4.2, in which Newton–Schulz Muon governs the 2D matrices and the swapped map governs the remaining embedding, LM-head, normalization, and bias parameters. The MoE experiments are conducted on a proprietary model and dataset that cannot be released due to organizational constraints; all configurable hyperparameters are reported to facilitate future reproduction on public benchmarks.

C PROOFS

C.1 λ-INSENSITIVITY OF THE INTERIOR LINKS (LOG AND SIGMOID)

Both concave links respond only weakly to a mis-estimated scale λ, which makes λ far more forgiving to set than Adam's second moment v: where a mis-scaled v perturbs Adam's update multiplicatively (in proportion to the update itself), a mis-scaled λ shifts our maps by at most an m-independent constant. We state and prove the bound for the log map first, then its sigmoid analogue.

Lemma 1 (Logarithmic λ-insensitivity of the log map)**.** *For $u_\lambda(m) = \operatorname{sign}(m)\log(1+|m|/\lambda)$ and $\hat\lambda = \lambda(1+\varepsilon)$, $\varepsilon > -1$, one has $|u_{\hat\lambda}(m) - u_\lambda(m)| \le |\log(1+\varepsilon)|$ for all m: an additive, m-independent shift, in contrast to Adam's multiplicative sensitivity $\propto |\varepsilon|/(1+\varepsilon)\cdot|u|$.*

Proof. By oddness it suffices to take $m > 0$; write $r = |m|/\lambda \ge 0$ and $\hat\lambda = \lambda(1+\varepsilon)$ with $\varepsilon > -1$, so $|m|/\hat\lambda = r/(1+\varepsilon)$. Then

$$u_{\hat\lambda}(m) - u_\lambda(m) \;=\; \ln\!\Big(1 + \tfrac{r}{1+\varepsilon}\Big) - \ln(1+r) \;=:\; g(r). \tag{15}$$

We have $g(0) = 0$ and, as $r \to \infty$, $g(r) \to \ln \frac{r/(1+\varepsilon)}{r} = -\ln(1+\varepsilon)$. Its derivative

$$g'(r) \;=\; \frac{1}{(1+\varepsilon)+r} - \frac{1}{1+r} \tag{16}$$

has a fixed sign for all $r \ge 0$ (negative if $\varepsilon > 0$, positive if $\varepsilon \in (-1, 0)$), so g is monotone and its range is the interval between $g(0) = 0$ and $g(\infty) = -\ln(1+\varepsilon)$. Hence $\sup_m |u_{\hat\lambda}(m) - u_\lambda(m)| = \sup_{r\ge 0}|g(r)| = |\ln(1+\varepsilon)|$, attained only in the limit $|m| \to \infty$. The bound is additive and independent of m, as claimed.

For the contrast, Adam's normalized coordinate is $u = m/(\sqrt{v}+\epsilon)$; a multiplicative perturbation $\hat v = (1+\varepsilon)^2 v$ of the second-moment estimate sends $u \mapsto u/(1+\varepsilon)$ (for $\epsilon \to 0$), a change of magnitude $\frac{|\varepsilon|}{1+\varepsilon}|u|$ that scales *with* the update itself, whereas the log map's response equation 22 is capped at the constant $|\ln(1+\varepsilon)|$ regardless of $|m|$. This is the sense in which the log link is λ-insensitive and the granularity of the RMS pool (§4.3) barely moves it. □

Lemma 2 (Bounded λ-insensitivity of the sigmoid map)**.** *For $u_\lambda(m) = \operatorname{sign}(m)\,\dfrac{|m|/\lambda}{1+|m|/\lambda}$ and $\hat\lambda = \lambda(1+\varepsilon)$, $\varepsilon > -1$,*

$$|u_{\hat\lambda}(m) - u_\lambda(m)| \;=\; \frac{|\varepsilon|\,(|m|/\lambda)}{\big((1+\varepsilon)+|m|/\lambda\big)\,\big(1+|m|/\lambda\big)} \;\le\; \frac{|\varepsilon|}{\big(1+\sqrt{1+\varepsilon}\big)^2} \qquad (\forall m), \tag{17}$$

an m-independent bound that is at most $\frac{1}{4}|\varepsilon|$ for $\varepsilon \ge 0$ and never exceeds $|\varepsilon|$; moreover it vanishes *as $|m| \to 0$ and as $|m| \to \infty$.*

Proof. By oddness take $m > 0$ and set $r = |m|/\lambda \ge 0$, so $|m|/\hat\lambda = r/(1+\varepsilon)$. With the sigmoid link $h(x) = x/(1+x)$,

$$u_{\hat\lambda}(m) - u_\lambda(m) \;=\; \frac{r/(1+\varepsilon)}{1+r/(1+\varepsilon)} - \frac{r}{1+r} \;=\; \frac{r}{(1+\varepsilon)+r} - \frac{r}{1+r} \;=\; \frac{-\varepsilon\, r}{\big((1+\varepsilon)+r\big)(1+r)} \;=:\; d(r). \tag{18}$$

Thus $|d(r)| = |\varepsilon|\, f(r)$ with $f(r) = r/\big[((1+\varepsilon)+r)(1+r)\big]$, and $f(0) = 0$, $f(r) \to 0$ as $r \to \infty$. Writing $a = 1+\varepsilon$, the denominator is $D(r) = r^2 + (a+1)r + a$ and

$$f'(r) \;=\; \frac{D(r) - r\,D'(r)}{D(r)^2} \;=\; \frac{a - r^2}{D(r)^2}, \tag{19}$$

so f increases up to its unique maximizer $r^\star = \sqrt{a} = \sqrt{1+\varepsilon}$ and decreases thereafter. At the maximizer

$$f(r^\star) \;=\; \frac{\sqrt{a}}{(a+\sqrt{a})(1+\sqrt{a})} \;=\; \frac{\sqrt{a}}{\sqrt{a}\,(1+\sqrt{a})^2} \;=\; \frac{1}{(1+\sqrt{1+\varepsilon})^2}, \tag{20}$$

which gives the stated supremum $\sup_m |u_{\hat{\lambda}}(m) - u_\lambda(m)| = |\varepsilon|/(1+\sqrt{1+\varepsilon})^2$. For $\varepsilon \geq 0$ we have $\sqrt{1+\varepsilon} \geq 1$, so the denominator is ≥ 4 and the bound is $\leq \frac{1}{4}|\varepsilon|$; for $\varepsilon \in (-1, 0)$ the denominator stays > 1, so the bound never exceeds $|\varepsilon|$.

The bound is again independent of m, but the sigmoid link is even less sensitive than the log link at the extremes: whereas the log deviation *saturates* to the constant $|\ln(1+\varepsilon)|$ as $|m| \to \infty$, the sigmoid deviation decays back to zero there, because the bounded link $h < 1$ maps large coordinates of either scaling to a common value near 1. The perturbation is therefore concentrated on mid-scale coordinates ($r \approx \sqrt{1+\varepsilon}$) and is negligible on both the bulk and the tail, which is why the RMS-pooling granularity (§4.3) moves the sigmoid map least of all. □

C.2 Convergence of the Orlicz step (log and sigmoid, unified)

Scope: fixed λ vs. the deployed schedule (expanding Remark 1). The analysis below treats λ as a fixed scalar, so P_λ is a static map and the analyzed step $-\eta P_\lambda(m_t)$ is the idealization in which the scale is absorbed into η. The deployed optimizer instead uses an online, group-wise $\lambda_{t,j} = \sqrt{\mathrm{EMA}_{\beta_2}(\overline{g^2})}$ (§3.2). Two facts keep the analysis representative. First, $\lambda_{t,j}$ is a bounded, slowly varying EMA of the squared gradient rather than a free variable, so a fixed λ is the natural stationary idealization of a quantity that changes on a timescale much longer than a single step. Second, pooling λ per group only makes the preconditioner block-diagonal: the step still acts coordinate-wise through $P_{\lambda_{t,j}}$, so the separable argument below carries over group by group. A fully time-varying treatment that tracks $\lambda_{t,j}$ inside the rate is left to future work.

We analyze the *momentum* iteration actually run in §3: a heavy-ball buffer m_t accumulates the gradients and the scale-normalized map is applied *after* accumulation,

$$m_t \;=\; \gamma\, m_{t-1} + g_t, \qquad x_{t+1} \;=\; x_t - \eta\, P_\lambda(m_t), \qquad m_0 = 0, \qquad \mathbb{E}[g_t \mid x_t] = \nabla f(x_t), \tag{21}$$

where $\gamma \in [0, 1)$ is the heavy-ball momentum (Polyak, 1964) and the coordinate-wise map is

$$P_\lambda(u)_i \;=\; \lambda\, \mathrm{sign}(u_i)\, h\big(|u_i|/\lambda\big), \qquad h = (\phi')^{-1},\ i = 1, \ldots, d. \tag{22}$$

This is the gradient of the convex conjugate of the separable Orlicz modular $\psi_\lambda(x) = \lambda^2 \sum_i \Phi(|x_i|/\lambda)$, $\Phi = \phi$, i.e. $P_\lambda = \nabla \psi_\lambda^*$ (§3.2). For the two maps of interest,

$$\text{log:} \quad \Phi_{\log}(s) = e^s - s - 1,\ \ h_{\log}(x) = \ln(1+x),\ \ P_\lambda(u) = \lambda\, \mathrm{sign}(u) \ln(1 + |u|/\lambda); \tag{23}$$

$$\text{sigmoid:} \quad \Phi_{\mathrm{sig}}(s) = -s - \ln(1-s),\ \ h_{\mathrm{sig}}(x) = \tfrac{x}{1+x},\ \ P_\lambda(u) = \mathrm{sign}(u)\, \tfrac{\lambda|u|}{\lambda+|u|}. \tag{24}$$

The implemented update $\Delta = -\eta_0\, \mathrm{sign}(m)\, h(|m|/\lambda)$ of §3.3 equals $-(\eta_0/\lambda) P_\lambda(m)$; with the effective rate $\eta := \eta_0/\lambda$ the factor λ is absorbed into the effective step, so equation 21 is the implemented step up to this rescaling. Unrolling the buffer gives $m_t = \sum_{k=0}^{t} \gamma^{t-k} g_k$, whose weights sum to $(1-\gamma^{t+1})/(1-\gamma) \to (1-\gamma)^{-1}$: the unnormalized heavy-ball buffer amplifies the running gradient by a factor $\approx (1-\gamma)^{-1}$, exactly as in Polyak momentum. This constant scale is absorbed into the effective learning rate (the paper tunes η and λ jointly, §3.4), so it enters the analysis only through the second-moment constant below. All statements hold for *any* link with a 1-strongly-convex generator, i.e. $\Phi'' \geq 1$; since $\Phi''_{\log}(s) = e^s \geq 1$ and $\Phi''_{\mathrm{sig}}(s) = (1-s)^{-2} \geq 1$, both maps qualify and the proof is shared.

Assumptions.

Assumption 1 (L-smoothness)**.** $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ *for all* x, y*; equivalently* $f(y) \leq f(x) + \langle \nabla f(x), y - x\rangle + \frac{L}{2}\|y - x\|^2$.

Assumption 2 (Unbiased gradient, bounded variance)**.** *Writing* $\mathcal{F}_{t-1}$ *for the history up to (and including)* x_t*,* $\mathbb{E}[g_t \mid \mathcal{F}_{t-1}] = \nabla f(x_t)$ *and* $\mathbb{E}[\|g_t - \nabla f(x_t)\|^2 \mid \mathcal{F}_{t-1}] \leq \sigma^2$.

Assumption 3 (Bounded gradient)**.** $\|\nabla f(x)\| \leq G$ *for all* x.

Assumptions 1–2 are the standard nonconvex-SGD set; the bounded-gradient Assumption 3 is standard for signed and normalized updates and can be relaxed to $\mathbb{E}\|\nabla f\|^2 \leq G^2$ at the cost of extra $\mathbb{E}$-bookkeeping. We deliberately do *not* assume that the momentum map is, in expectation, aligned with $P_\lambda(\nabla f)$, an idealization that would make the descent term exact but hide both the buffer amplification and the Jensen gap of the nonlinear map. Instead we *prove* the alignment directly in Lemma 5 below, closing it with the fact that our P_λ is globally 1-Lipschitz and co-coercive (Lemma 3). The analysis therefore rests only on Assumptions 1–3 and the geometry of P_λ.

Key geometric property.

Lemma 3 (Firm nonexpansiveness / co-coercivity)**.** *If $\Phi'' \geq 1$ then P_λ in equation 22 satisfies, for all $a, b \in \mathbb{R}^d$,*

$$\langle a - b,\, P_\lambda(a) - P_\lambda(b)\rangle \;\geq\; \|P_\lambda(a) - P_\lambda(b)\|^2. \tag{25}$$

Consequently P_λ is 1-Lipschitz, and since $P_\lambda(0) = 0$, $\langle x,\, P_\lambda(x)\rangle \geq \|P_\lambda(x)\|^2$ for all x.

Proof. The map is separable, so it suffices to treat one coordinate through the odd scalar function $p(x) = \lambda\, \mathrm{sign}(x)\, h(|x|/\lambda)$.

Step 1: the link is a 1-Lipschitz monotone map. On $x > 0$ we have $p(x) = \lambda h(x/\lambda)$, so $p'(x) = h'(x/\lambda)$. Since $h = (\phi')^{-1}$,

$$h'(x) = \frac{1}{\phi''\big(h(x)\big)} \in (0, 1] \quad \Longleftrightarrow \quad \phi'' \geq 1,$$

which holds for both maps ($h'_{\log}(x) = (1+x)^{-1}$ and $h'_{\mathrm{sig}}(x) = (1+x)^{-2}$ both lie in $(0, 1]$). Hence p is nondecreasing and 1-Lipschitz on $\mathbb{R}$, odd symmetry extending both properties across the origin.

Step 2: the scalar co-coercivity identity. Set $D = x - y$ and $\Delta = p(x) - p(y)$. Monotonicity gives $\Delta D \geq 0$ and the 1-Lipschitz bound gives $|\Delta| \leq |D|$, so Δ and $D - \Delta$ share a sign and $\Delta(D - \Delta) \geq 0$, i.e.

$$D\,\Delta \;\geq\; \Delta^2.$$

Summing over coordinates yields equation 25.

Step 3: consequences. Cauchy–Schwarz on equation 25 gives

$$\|P_\lambda(a) - P_\lambda(b)\|^2 \;\leq\; \|a - b\|\, \|P_\lambda(a) - P_\lambda(b)\|,$$

hence P_λ is 1-Lipschitz; taking $b = 0$ with $P_\lambda(0) = 0$ gives co-coercivity. □

Equivalently, equation 25 is the Baillon–Haddad theorem: ψ_λ is 1-strongly convex ($\psi''_\lambda \geq \Phi'' \geq 1$), so its conjugate is 1-smooth and $P_\lambda = \nabla\psi^*_\lambda$ is firmly nonexpansive.

Lemma 4 (Momentum second moment)**.** *Set $M := 2(G^2 + \sigma^2)/(1-\gamma)^2$. Under Assumptions 1–3 and Lemma 3, the buffer of equation 21 obeys, for all $t \geq 1$,*

$$\mathbb{E}\|m_t\|^2 \leq M, \qquad \textit{and hence} \qquad \mathbb{E}\|P_\lambda(m_t)\|^2 \leq M. \tag{26}$$

Proof. Unroll the buffer into a signal and a noise part,

$$m_t = \sum_{k=0}^{t-1} \gamma^k g_{t-k} = a_t + b_t, \qquad a_t = \sum_{k=0}^{t-1} \gamma^k \bar{g}_{t-k}, \quad b_t = \sum_{k=0}^{t-1} \gamma^k \xi_{t-k},$$

with $\bar{g}_j := \nabla f(x_{j-1})$ and $\xi_j := g_j - \bar{g}_j$.

Signal. By the triangle inequality and Assumption 3,

$$\|a_t\| \;\leq\; \sum_{k\geq 0} \gamma^k G \;=\; \frac{G}{1-\gamma}.$$

Noise. The increments ξ_j are martingale differences ($\mathbb{E}[\xi_j \mid \mathcal{F}_{j-1}] = 0$), so cross terms vanish and

$$\mathbb{E}\|b_t\|^2 = \sum_{k=0}^{t-1} \gamma^{2k} \mathbb{E}\|\xi_{t-k}\|^2 \leq \sigma^2 \sum_{k\geq 0} \gamma^{2k} \leq \frac{\sigma^2}{1-\gamma} \qquad \Big(\|b_t\|_{L^2} \leq \tfrac{\sigma}{\sqrt{1-\gamma}}\Big).$$

Combine. Using $\|m_t\|^2 \leq 2\|a_t\|^2 + 2\|b_t\|^2$ and $\frac{1}{1-\gamma} \leq \frac{1}{(1-\gamma)^2}$,

$$\mathbb{E}\|m_t\|^2 \leq \frac{2G^2}{(1-\gamma)^2} + \frac{2\sigma^2}{1-\gamma} \leq \frac{2(G^2+\sigma^2)}{(1-\gamma)^2} = M.$$

The second inequality in equation 26 is 1-Lipschitzness of P_λ with $P_\lambda(0) = 0$, i.e. $\|P_\lambda(m_t)\| \leq \|m_t\|$ (for the sigmoid map one also has the hard cap $\|P_\lambda(m_t)\|^2 \leq d\lambda^2$). □

Gradient alignment without an unbiased-map assumption. The one nonstandard step in a momentum analysis is lower-bounding the alignment $\langle \nabla f_t, P_\lambda(m_t)\rangle$ between the true gradient and the transformed buffer, since P_λ is nonlinear and m_t is neither unbiased for ∇f_t nor independent of the past. We prove the bound directly, decomposing the buffer error into a noise part and a drift part and closing it with co-coercivity and 1-Lipschitzness of P_λ. Unroll $m_t = \sum_{k=0}^{t-1} \gamma^k g_{t-k} = a_t + b_t$ with signal $a_t = \sum_{k=0}^{t-1} \gamma^k \nabla f_{t-k}$ ($\nabla f_j := \nabla f(x_{j-1})$) and noise $b_t = \sum_{k=0}^{t-1} \gamma^k \xi_{t-k}$, $\xi_j := g_j - \nabla f_j$. Write $S_t := \sum_{k=0}^{t-1} \gamma^k = (1-\gamma^t)/(1-\gamma) \in [1, (1-\gamma)^{-1})$ and split the signal into a scaled gradient plus a drift, $a_t = S_t \nabla f_t + r_t$, $r_t := \sum_{k=0}^{t-1} \gamma^k (\nabla f_{t-k} - \nabla f_t)$.

Lemma 5 (Gradient alignment)**.** *Under Assumptions 1–3 and Lemma 3, for all $t \geq 1$,*

$$\mathbb{E}\big[\langle \nabla f_t, P_\lambda(m_t)\rangle\big] \;\geq\; (1-\gamma)\,\mathbb{E}\|P_\lambda(\nabla f_t)\|^2 \;-\; C_0 \;-\; C_1\,\eta, \tag{27}$$

with the noise floor $C_0 = G\sigma/\sqrt{1-\gamma}$ *and the drift constant* $C_1 = GL\gamma\sqrt{M}/(1-\gamma)^2$ *(M from Lemma 4).*

Proof. Decompose the alignment at the scaled gradient $S_t \nabla f_t$:

$$\langle \nabla f_t, P_\lambda(m_t)\rangle = \underbrace{\langle \nabla f_t, P_\lambda(S_t \nabla f_t)\rangle}_{\text{(I)}} + \underbrace{\langle \nabla f_t, P_\lambda(m_t) - P_\lambda(S_t \nabla f_t)\rangle}_{\text{(II)}}.$$

Term (I): co-coercivity. Applying Lemma 3 at $S_t \nabla f_t$,

$$\langle S_t \nabla f_t,\, P_\lambda(S_t \nabla f_t)\rangle \;\geq\; \|P_\lambda(S_t \nabla f_t)\|^2.$$

Dividing by $S_t > 0$, then using $1/S_t \geq 1-\gamma$ together with $\|P_\lambda(S_t \nabla f_t)\| \geq \|P_\lambda(\nabla f_t)\|$ (each coordinate of the odd, nondecreasing link grows with $S_t \geq 1$),

$$\text{(I)} \;\geq\; (1-\gamma)\,\|P_\lambda(\nabla f_t)\|^2.$$

Term (II): 1*-Lipschitz control of noise and drift.* Since $m_t - S_t \nabla f_t = r_t + b_t$, the 1-Lipschitzness of P_λ (Lemma 3) and $\|\nabla f_t\| \leq G$ give

$$\text{(II)} \;\geq\; -G\,\|P_\lambda(m_t) - P_\lambda(S_t \nabla f_t)\| \;\geq\; -G\big(\|r_t\| + \|b_t\|\big).$$

The noise b_t is a sum of martingale differences, so

$$\mathbb{E}\|b_t\| \leq (\mathbb{E}\|b_t\|^2)^{1/2} = \Big(\textstyle\sum_k \gamma^{2k} \mathbb{E}\|\xi_{t-k}\|^2\Big)^{1/2} \leq \frac{\sigma}{\sqrt{1-\gamma}}.$$

For the drift, L-smoothness and the update give $\|\nabla f_{t-k} - \nabla f_t\| \leq L\|x_{t-1-k} - x_{t-1}\| \leq L\eta \sum_i \|P_\lambda(m_i)\|$ over the k intervening steps, so, with $\mathbb{E}\|P_\lambda(m_i)\| \leq \sqrt{M}$ (Lemma 4),

$$\mathbb{E}\|r_t\| \leq \sum_k \gamma^k L\eta k \sqrt{M} \leq L\eta\sqrt{M} \sum_{k\geq 0} k\gamma^k = \frac{L\eta\sqrt{M}\,\gamma}{(1-\gamma)^2}.$$

Collecting the two bounds,

$$\mathbb{E}[\text{(II)}] \;\geq\; -\frac{G\sigma}{\sqrt{1-\gamma}} - \frac{GL\gamma\sqrt{M}\,\eta}{(1-\gamma)^2} \;=\; -C_0 - C_1\eta,$$

which combined with (I) gives equation 27. □

Main result.

Theorem 2 ($O(1/\sqrt{T})$ to an $O(\sigma)$ neighborhood, momentum)**.** *Under Assumptions 1–3, run equation 21 with momentum $\gamma \in [0,1)$ and constant step $\eta = \sqrt{2\Delta_f/(LMT)}$, where $\Delta_f := f(x_0) - f^\star$ and $M = 2(G^2+\sigma^2)/(1-\gamma)^2$ (Lemma 4). Then*

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\big[\|P_\lambda(\nabla f_t)\|^2\big] \;\leq\; \frac{1}{1-\gamma}\sqrt{\frac{2LM\Delta_f}{T}} \;+\; \frac{G\sigma}{(1-\gamma)^{3/2}} \;+\; O\Big(T^{-1/2}\Big). \tag{28}$$

That is, the running-average squared map-gradient converges at rate $O(1/\sqrt{T})$ to a neighborhood of radius $C_0/(1-\gamma) = G\sigma/(1-\gamma)^{3/2}$; when $\sigma = 0$ (full-batch) the neighborhood collapses and the rate is the clean $O(1/\sqrt{T})$.

Proof. Step 1: one-step descent. Write $\nabla f_t := \nabla f(x_{t-1})$. By L-smoothness and the update $x_t = x_{t-1} - \eta P_\lambda(m_t)$,

$$f(x_t) \le f(x_{t-1}) - \eta \langle \nabla f_t, P_\lambda(m_t) \rangle + \tfrac{L\eta^2}{2} \|P_\lambda(m_t)\|^2.$$

Take total expectation, apply the alignment Lemma 5 to the linear term and Lemma 4 ($\mathbb{E}\|P_\lambda(m_t)\|^2 \le M$) to the quadratic term:

$$\mathbb{E} f(x_t) \le \mathbb{E} f(x_{t-1}) - \eta\big[(1-\gamma)\mathbb{E}\|P_\lambda(\nabla f_t)\|^2 - C_0 - C_1\eta\big] + \tfrac{L\eta^2}{2} M.$$

Because the noise sits in the *constant* floor C_0 and the flat second-moment term rather than multiplying $\|P_\lambda(\nabla f_t)\|^2$, no step-size cap is needed.

Step 2: telescope. Summing $t = 1, \dots, T$ and using $\mathbb{E} f(x_T) \ge f^\star$,

$$\eta(1-\gamma) \sum_{t=1}^{T} \mathbb{E}\|P_\lambda(\nabla f_t)\|^2 \le \Delta_f + \eta T C_0 + \eta^2 T C_1 + \tfrac{L\eta^2}{2} MT.$$

Dividing by $\eta(1-\gamma)T$,

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}\|P_\lambda(\nabla f_t)\|^2 \le \frac{1}{1-\gamma}\Big(\frac{\Delta_f}{\eta T} + \frac{L\eta M}{2}\Big) + \frac{C_0}{1-\gamma} + \frac{C_1\eta}{1-\gamma}.$$

Step 3: balance the step. The choice $\eta = \sqrt{2\Delta_f/(LMT)}$ balances the first bracket, giving

$$\frac{1}{1-\gamma}\Big(\frac{\Delta_f}{\eta T} + \frac{L\eta M}{2}\Big) = \frac{1}{1-\gamma}\sqrt{\frac{2LM\Delta_f}{T}}, \qquad \frac{C_1\eta}{1-\gamma} = O(T^{-1/2}),$$

while the residual $C_0/(1-\gamma) = G\sigma/(1-\gamma)^{3/2}$ is the stated neighborhood. □

Because the proof uses only co-coercivity and 1-Lipschitzness (Lemma 3) and the flat second-moment bound (Lemma 4), all holding for any $\Phi'' \ge 1$, it applies verbatim to the log and sigmoid maps, and needs *no* unbiased-map assumption. Momentum enters only through the constant $M \propto (1-\gamma)^{-2}$ and the noise floor $C_0 = G\sigma/\sqrt{1-\gamma}$.

From the map norm back to ℓ_2. Theorem 2 is stated in the map's own norm $\|P_\lambda(\nabla f)\|$, which is the exact quantity the descent argument produces and the natural currency of a steepest-descent oracle (as with signSGD and Muon, whose guarantees live in their dual norms). It is a bona fide stationarity measure: P_λ is an odd, strictly increasing bijection with $P_\lambda(0) = 0$ and $P_\lambda'(0) = 1$, so $\|P_\lambda(\nabla f)\| = 0 \iff \nabla f = 0$ and the two norms agree to first order near a stationary point. Under the bounded-gradient Assumption 3 the guarantee also converts to ordinary ℓ_2 stationarity with an explicit constant.

Corollary 1 (ℓ_2 stationarity)**.** *Let $\psi(a) := |P_\lambda(a)|$ for $a \ge 0$. Both links give $\psi(a) \le a$, and since ψ is concave with decreasing derivative,*

$$\psi(a) \;\ge\; \psi'(G)\, a \qquad \text{on } [0, G], \qquad \psi'(a) = \begin{cases} \lambda/(\lambda + a) & (\textit{log}), \\ \lambda^2/(\lambda + a)^2 & (\textit{sigmoid}). \end{cases}$$

Hence under Assumption 3,

$$\|\nabla f\|^2 \;\le\; \big(1 + \tfrac{G}{\lambda}\big)^2 \|P_\lambda(\nabla f)\|^2 \quad (\textit{log}), \qquad \|\nabla f\|^2 \;\le\; \big(1 + \tfrac{G}{\lambda}\big)^4 \|P_\lambda(\nabla f)\|^2 \quad (\textit{sigmoid}). \tag{29}$$

Combining with Theorem 2 turns the $O(1/\sqrt{T})$-to-$O(\sigma)$ bound into the same statement for $\frac{1}{T}\sum_t \mathbb{E}\|\nabla f_t\|^2$, scaled by these constants.

The constant is finite but conservative: G is a full-vector bound, so G/λ carries the usual dimension dependence, the same $\sqrt{n}$-type ℓ_1/ℓ_2 gap that signSGD-style analyses pay, and it collapses toward 1 as gradients shrink, consistent with the first-order equivalence above.

Table 8: Model-X shared-base-lr search: training lm loss at a fixed token budget (both scales at sequence length 8192). "NaN" marks a diverged run; the selected rate is in **bold**. Scale 1 (bsz 448) is compared at $\approx 5.0 \times 10^{10}$ tokens ($\sim$ 13.6k steps), scale 2 (bsz 896) at $\approx 1.26 \times 10^{11}$ tokens ($\sim$17.2k steps).

scale 1 lr	5e−4	6e−4	7e−4	8e−4	9e−4	1e−3	1.1e−3	1.2e−3	1.4e−3
train loss	NaN	NaN	NaN	2.2200	2.2194	**2.2186**	2.2193	NaN	NaN

scale 2 lr	3e−4	4e−4	5e−4	6e−4	7e−4	8e−4	9e−4	1e−3
train loss	NaN	NaN	NaN	1.9513	1.9483	1.9470	**1.9464**	NaN

Why the neighborhood is intrinsic. The $O(\sigma)$ floor C_0 is not an artifact of our proof: it is the alignment/Jensen gap of *any* nonlinear map applied to a noisy momentum buffer, arising from the same noise/drift decomposition used above with $\|b_t\| \le \sigma/\sqrt{1-\gamma}$. We therefore state the neighborhood explicitly rather than absorbing it into a decreasing step schedule (which would leave a non-vanishing floor of the same order), and two structural properties of our maps sharpen it. First, global 1-Lipschitzness (Lemma 3) makes the *drift* enter as $C_1\eta = O(\eta)$, controlled by the step size. Second, the systematic (conditional-mean) part of the gap is $O(1/\lambda)$: a Taylor expansion of equation 22 gives

$$P_\lambda(u) = u - \tfrac{1}{2\lambda} u|u| + O(\lambda^{-2}) \quad \text{(log; sigmoid analogous)},$$

so the bias $\mathbb{E}[P_\lambda(m_t) \mid \mathcal{F}_{t-1}] - P_\lambda(\nabla f_t)$ shrinks with λ. In the idealized limit where the buffer is unbiased for ∇f_t (equivalently $\sigma = 0$, or the map taken exact), $C_0 = 0$ and the clean $O(1/\sqrt{T})$ is recovered. This matches the main-text statement that the residual is a stochastic floor with an $O(1/\lambda)$ systematic part.

D Additional experimental details

Batch size, sequence length, and hardware. GPT-2 uses a sequence length of 1024 and a global batch of 480 sequences ($\approx$ 0.49M tokens) at both the 124M and 350M scales, following the nanoGPT OpenWebText recipe on a single 8×GPU node. Qwen3 uses a sequence length of 2048 and a global batch of 256 sequences ($\approx$ 0.52M tokens) at both the 0.6B and 4B scales. The 0.6B runs use data-parallel size 8 on a single node (tensor- and pipeline-parallel 1); the 4B runs span 2 nodes, 16 NVIDIA H800 GPUs in total, with tensor-parallel 1, pipeline-parallel 2, and data-parallel 8. Model-X (scale1 1.4B, scale2 4.9B) uses the two global batch sizes reported in App. B (448 and 896 sequences).

Learning-rate grid anchored to AdamW. The Part-1 lr grid is chosen so that AdamW sits at its own optimum, so no map is credited a learning rate that AdamW was denied. We sweep the 50k evidence grid $\{1, 2, 4, 8\}$e−3, on which AdamW improves monotonically toward an interior optimum (2.940 → 2.922 → 2.908 across 1 → 2 → 4e−3 at 124M) and turns back over by 8e−3 (2.909), so the grid brackets AdamW's optimum on both sides. On that grid AdamW's own 50k optimum is 4e−3 at both GPT-2 scales (Table 10), which is where AdamW is reported in Table 9 and every other optimizer is reported at its own grid optimum. On Qwen we instead fix a single shared lr per scale (5e−4 at 0.6B, 3e−4 at 4B), so every optimizer runs at the identical AdamW-tuned rate.

Model-X learning-rate search. Model-X uses a sequence length of 8192 and, lacking a pre-tuned AdamW anchor, we searched the single shared base lr directly for each scale, comparing training lm loss at a fixed token budget (Table 8). The stable band is narrow and shifts with batch size. At scale 1 (bsz 448, compared at $\approx 5.0 \times 10^{10}$ tokens, $\sim$13.6k steps) every rate at or below 7e−4 and at or above 1.2e−3 diverges to NaN, leaving a shallow interior optimum at 1e−3. At scale 2 (bsz 896, compared at $\approx 1.26 \times 10^{11}$ tokens, $\sim$17.2k steps) the band is 6–9e−4, monotonically improving to its edge optimum at 9e−4 before diverging at 1e−3. These selected rates (1e−3 and 9e−4) are the ones reported in Fig. 3 and Table 4.

Table 9: Final validation loss across four scales (full-parameter training), companion to the training curves of Fig. 2. GPT-2 reports each map at its own best lr from the 50k grid $\{1, 2, 4, 8\}$e-3 (Table 10): 124M AdamW / PowerStep / sigmoid / log 4e-3, signSGD 1e-3; 350M AdamW / PowerStep / sigmoid 4e-3, log 2e-3, signSGD 1e-3. Qwen uses a shared lr per scale (5e-4 at 0.6B, 3e-4 at 4B). †at 4B the unclipped log-row diverges; we report the *clipped* log-row, which is stable (App. D.1). Best per column in **bold**. GPT-2 runs are 50k steps, Qwen 20k. Full-parameter differences are small: every map stays within a few thousandths of a well-tuned AdamW, the sanity check §4.1 asks for.

method	GPT-2 124M	GPT-2 350M	Qwen3 0.6B	Qwen3 4B
AdamW	2.908	2.702	2.963	2.804
signSGD	2.931	2.765	2.957	2.864
PowerStep	2.907	**2.687**	2.983	2.823
log (row)	2.905	2.696	**2.949**	**2.792**†
sigmoid (row)	**2.900**	2.690	2.957	2.803

Table 10: Per-method learning-rate sweep (final validation lm loss, 50k) on the grid $\{1, 2, 4, 8\} \times 10^{-3}$, the own-best selection behind Fig. 2 and Table 9. "div" = diverged (signSGD past its narrow ℓ_∞ band, with the 124M 2e-3 blow-up observed and higher-lr cells inferred; the log-row also diverges at 350M $4/8$e-3); "c" = crashed run (AdamW 350M 1e-3, anomalous early spike). Best per scale in **bold**.

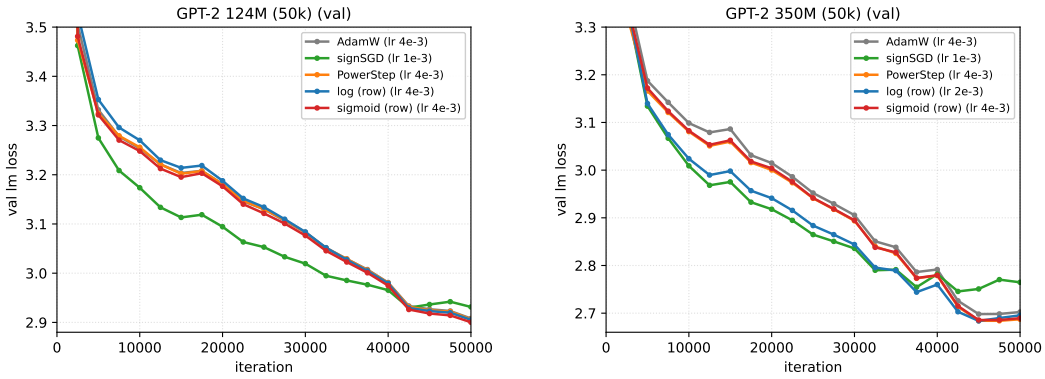
	GPT-2 124M				GPT-2 350M			
method	1	2	4	8	1	2	4	8
AdamW	2.940	2.922	2.908	2.909	c	2.731	2.702	2.709
signSGD	2.931	div	div	div	2.765	div	div	div
PowerStep	2.940	2.913	2.907	2.907	2.717	2.702	**2.687**	2.702
log (row)	2.917	2.941	2.905	2.928	2.705	2.696	div	div
sigmoid (row)	2.932	2.909	**2.900**	2.907	2.711	2.693	2.690	div

Optima and stability bands. Two features separate the interior maps from the box corner in Table 10. The interior maps and PowerStep have a smooth optimum inside the grid (at 350M PowerStep improves $2.717 \to 2.702 \to 2.687$ across $1 \to 2 \to 4$e-3, and both row maps reach their best by 4e-3), so no map is credited a rate at which it happens to be unstable. signSGD, the exact ℓ_∞ corner, has by contrast the narrowest stable band: competitive at 1e-3 but diverging once the step doubles (124M 2e-3 blows up to 6.97), the corner-versus-interior reading of §3.3 made empirical. The lr-tolerant exception is the per-row family, whose self-normalized update decouples the effective step from scale. The divergence ceiling tightens with model scale: at 4B and the shared 3e-4 the unclipped log-row map diverges while the bounded sigmoid-row map, PowerStep, and AdamW stay stable, and clipping the log update restores its stability and makes it the best 4B branch (App. D.1).

Muon-hybrid validation. Table 11 reports the final validation loss and perplexity for the Muon-hybrid Qwen3 runs whose training loss is summarized in Table 4. The full-parameter validation gaps are small, as expected from a shared, AdamW-tuned learning rate: signSGD is train-competitive yet val-worst at 4B, whereas the interior log and sigmoid maps both beat AdamW on validation at *both* scales, consistent with the interior-geometry argument of §3.3.

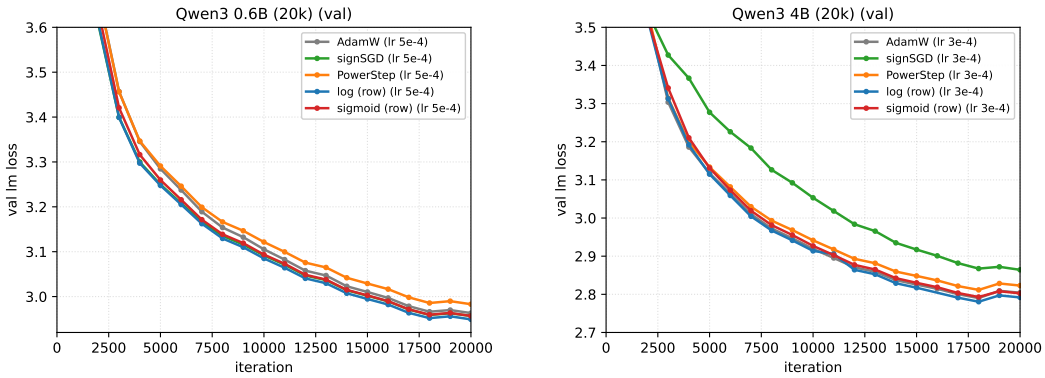
D.1 Boundedness and learning-rate sensitivity of the log map at 4B

The log map carries the richest Orlicz geometry of our family: its Young function $\Phi_{\log}(s) = e^s - s - 1$ is the critical sub-exponential form, giving a Gaussian bulk with only mild (ℓ_2-to-log) tail growth, and on the smaller scales it is the strongest non-Muon branch (Table 9). But that geometry is *unbounded*: $h_{\log}(x) = \ln(1 + x)$ grows without limit, so a single large-magnitude coordinate can still drive an arbitrarily large step. At 4B this becomes fatal. Figure 7 shows the per-row log map at a shared 3e-4 diverging near iter $\sim$2600, and the divergence is not an artifact of one learning rate: the *same* unclipped map diverges at every grid point we tried, 2.5e-4 (blows up near iter 17k),



(a) GPT-2 124M, 50k.

(b) GPT-2 350M, 50k.



(c) Qwen3 0.6B, 20k (shared lr $5e{-}4$).

(d) Qwen3 4B, 20k (shared lr $3e{-}4$).

Figure 6: Part-1 *validation* loss (companion to the training curves of Fig. 2 and the final numbers of Table 9). At 4B the unclipped log-row diverges at a shared $3e{-}4$ (analyzed in App. D.1); the clipped log-row and the intrinsically bounded sigmoid-row map both stay stable and beat AdamW.

	0.6B		4B	
branch	val	PPL	val	PPL
AdamW	2.899	18.15	2.761	15.82
log	**2.889**	**17.97**	2.760	15.81
sigmoid	2.892	18.03	**2.757**	**15.75**
PowerStep	2.902	18.22	2.760	15.79
signSGD	2.898	18.14	2.764	15.87

Table 11: Final validation lm loss / PPL at 20k for the Qwen3 Muon-hybrid, Muon fixed on matrices and the non-Muon branch swapped (shared lr). Best per scale in **bold**.

$3e{-}4$, and $5e{-}4$ (early, near iter 8k), so there is no stable lr band for it at this scale. Imposing an explicit clip on the log update (an *external* bound that mimics the box-cap the sigmoid map carries *intrinsically*) removes the divergence entirely and makes the clipped log-row the best 4B branch (val 2.792, Table 9). The contrast is exactly the boundedness thesis of §3.3: it is not the value of the loss but the boundedness of h that governs stability at scale. The intrinsically bounded sigmoid map and the magnitude-compressing PowerStep both survive the shared $3e{-}4$ without any clip, whereas the unbounded log map needs one; the log map thus has the better interior geometry but the worse learning-rate tolerance, and clipping is what lets us keep the former without paying the latter.

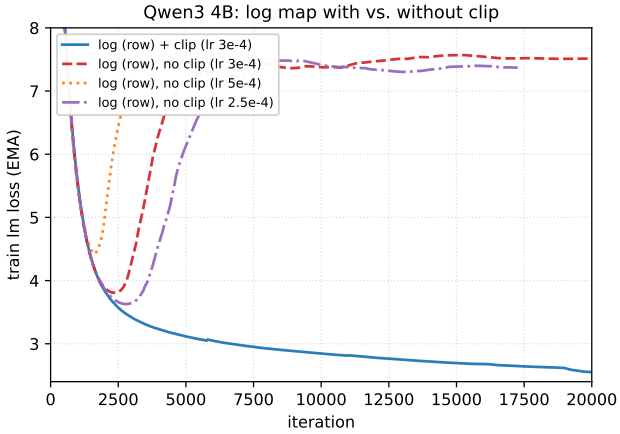


Figure 7: Qwen3 4B, per-row log map, with vs. without an explicit clip on the update. The unbounded (no-clip) log map diverges at *every* learning rate on the grid ($2.5\mathrm{e}{-4}$, $3\mathrm{e}{-4}$, and $5\mathrm{e}{-4}$ all blow up), so it has no stable band at 4B. Clipping the update (an imposed bound, echoing the sigmoid map's intrinsic box-cap) restores stability at the shared $3\mathrm{e}{-4}$ and yields the best 4B branch (val 2.792). The log map's superior interior geometry buys nothing at scale unless its tail is bounded.

Table 12: Per-coordinate update RMS on the Muon-hybrid vector branch (GPT-2 124M, OpenWebText), measured on a shared AdamW-driven gradient stream. All maps land within one order of magnitude, so a single learning rate sets a comparable effective step for every branch. The Orlicz maps (log, sigmoid) are self-normalizing, so their $\mathrm{RMS}(u)$ is pinned to $O(1)$ by construction independent of gradient scale; signSGD is exactly 1 and PowerStep compresses magnitude toward 1.

vector-branch map	mean $\mathrm{RMS}(u)$	std
AdamW	0.147	0.037
log	0.833	0.219
sigmoid	0.500	0.088
signSGD	1.000	0.000
PowerStep	0.326	0.040

D.2 Update-magnitude calibration for the shared vector-branch LR

The Muon-hybrid runs use a single learning rate for the whole optimizer, including the vector branch (Table 11). To check that this is not an accident of tuning, we measure the per-coordinate update magnitude each map actually produces on the parameters it governs. We drive one realistic AdamW trajectory (GPT-2 124M on OpenWebText, $\beta = (0.9, 0.95)$, lr $6\mathrm{e}{-4}$, 400 steps) and, on the *same* gradient stream, compute as side buffers the update u of every candidate map, namely AdamW, log, sigmoid, signSGD, and PowerStep ($\beta = 0.1$), with the four sign-family maps sharing the identical heavy-ball momentum ($\gamma = 0.9$) and the Orlicz maps using $\lambda = \mathrm{EMA}(\mathrm{RMS}(g))$. Table 12 reports $\mathrm{RMS}(u)$ averaged over the Block-B vector branch (embeddings, LM head, and all $\mathrm{ndim} \leq 1$ norm/bias parameters) and over measurement steps.

E Optimizer-state memory

The row-pooled maps we run in practice are also lighter on memory than AdamW. AdamW keeps two parameter-sized `fp32` buffers per weight: the first moment m and the full second moment v, so an $m \times n$ matrix carries $2mn$ optimizer scalars. Our maps keep the same heavy-ball momentum but replace v by the pooled statistic $\lambda = \sqrt{\mathrm{EMA}_{\beta_2}(\overline{g^2})}$. At row granularity λ has one entry per output row, so an $m \times n$ matrix stores only $mn + m$ optimizer scalars: essentially the momentum buffer alone, with a per-row overhead of $m/(mn) = 1/n$ that is negligible at transformer widths.

Table 13: Per-rank GPU memory on Qwen3 0.6B (DP$=8$, TP=PP$=1$, no distributed optimizer), read from the Megatron memory report: persistent `allocated` (master weights, gradients, optimizer state) and peak `max allocated` (including activations). Full-parameter trains every parameter; the Muon-hybrid holds Newton–Schulz Muon on the 2D weights and swaps only the non-Muon branch (`puremuon`=Muon+AdamW is the Moonlight default). The row-pooled log/sigmoid maps drop AdamW's second-moment buffer: 2271.80 MB ($\approx 21.5\%$ of persistent state) full-parameter and 593.69 MB on the hybrid, matching the 2273.6 MB `fp32` buffer that $596{,}042{,}752$ parameters would occupy almost exactly. signSGD/PowerStep, momentum-only, sit a further 1.9 MB below the row maps (the entire row-λ overhead).

	full-parameter		Muon-hybrid	
optimizer	allocated	max alloc.	allocated	max alloc.
AdamW / puremuon	10552.75	41746.91	8872.75	40066.91
log (row)	8280.95	39475.10	8279.06	39473.21
sigmoid (row)	8280.95	39475.10	8279.06	39473.21
signSGD	8279.03	39473.19	8279.06	39473.21
PowerStep	8279.03	39473.19	8279.06	39473.21

signSGD and PowerStep, which carry momentum but no λ, are the mn lower bound. Thus row-pooled log/sigmoid save almost exactly one of AdamW's two buffers, whereas a *per-element* λ coincides with Adam's $\sqrt{\hat{v}}$ and recovers Adam's footprint (the ablation of §4.3 shows per-element buys nothing over per-row, so the saving comes for free).

Measured footprint. Table 13 reports the per-rank figures from the Qwen3 0.6B runs, taken from the clean setting in which no optimizer uses Megatron's distributed (ZeRO) optimizer, so every method holds its full optimizer state on each rank and the comparison is purely algorithmic. Full-parameter, the row maps save 2271.80 MB of persistent state against AdamW, which is within 2 MB of the 2273.6 MB an `fp32` buffer of all $596{,}042{,}752$ parameters would take: the maps drop AdamW's entire second moment and add back only the tiny per-row λ. On the Muon-hybrid the same saving is 593.69 MB, the fraction that lives on the non-Muon branch AdamW governs (Muon itself is momentum-only on the matrices), and the row-λ overhead there is under 0.03 MB.

A caveat on distributed sharding. The clean comparison above fixes the distributed optimizer off. Our current integration shards only the *per-element* optimizer state across the data-parallel group, since element-wise λ matches Adam's per-parameter layout, whereas the row- and column-pooled statistics reduce along a tensor axis and are not yet ZeRO-partitioned. At 4B with a wide data-parallel group this inverts the per-rank picture: Adam's state is sharded across ranks while the row maps replicate theirs, so the per-rank `allocated` for the row maps can exceed the sharded Adam figure. This is an artifact of the sharding integration, not of the algorithm: Adam, precisely because it must store the second-moment buffer v, has to turn the distributed optimizer on at 4B or it runs out of memory.

The buffer is large enough to matter. The saving is not academic at scale. Running AdamW on Qwen3 4B with the distributed optimizer *off* exhausts the 80 GB H800 memory and crashes with a CUDA out-of-memory error; the 4B AdamW runs complete only once the optimizer state is ZeRO-sharded across the data-parallel group. The second-moment buffer AdamW carries on the non-Muon branch is therefore consequential even though that branch holds a minority of the parameters, because the token embedding and LM head are individually among the largest tensors in the model. Dropping one full optimizer buffer on exactly those parameters, as the row-pooled maps do for free, directly relaxes the memory ceiling that forces the sharding workaround.

F Orlicz-space background

This section collects the classical Orlicz-space objects the method rests on and fixes notation. A *Young function* $\phi : [0, \infty) \to [0, \infty]$ is convex, nondecreasing, with $\phi(0) = 0$ and $\phi(s)/s \to \infty$; extended evenly to $\mathbb{R}$ it plays the role of a per-coordinate cost. It induces the *Orlicz modular*

Algorithm 1 MN-step (log / sigmoid): a single procedure differing only in the link h. The vector face applies h coordinate-wise; the matrix face applies it to the singular values of the momentum. The scale $\lambda = \mathrm{EMA}(\mathrm{RMS}(g))$ is the online second-moment estimate, pooled per group.

Require: parameters θ, learning rate η, momentum γ, EMA rate β_2
Require: link $h(x) = \ln(1+x)$ (log map) **or** $h(x) = x/(1+x)$ (sigmoid map)
$m \leftarrow 0, \quad v \leftarrow 0$ ▷ momentum and second-moment buffers
for $t = 1, 2, \ldots, T$ **do**
  $g \leftarrow \nabla f(\theta)$ ▷ stochastic gradient
  $m \leftarrow \gamma\, m + g$ ▷ heavy-ball momentum
  $v \leftarrow \beta_2\, v + (1-\beta_2)\,\mathrm{RMS}(g)^2; \quad \lambda \leftarrow \sqrt{\hat{v}}$ ▷ $\lambda = \mathrm{EMA}(\mathrm{RMS}(g))$, bias-corrected
  $\theta \leftarrow \theta - \eta\ \mathrm{sign}(m)\, h(|m|/\lambda)$ ▷ vector face (coordinate-wise)
    or $\theta \leftarrow \theta - \eta\, U\, h(\Sigma/\lambda)\, V^\top$ for a matrix $m = U\Sigma V^\top$ ▷ matrix face
end for
return θ

$\Omega_\phi(x) = \sum_i \phi(|x_i|)$ and, through it, the *Luxemburg gauge* $\|x\|_\phi = \inf\{t > 0 : \Omega_\phi(x/t) \le 1\}$, which is a genuine norm whose unit ball is the sublevel set $\{\Omega_\phi \le 1\}$ (the well-posedness of this construction is the subject of §F.3). Our two maps are the two Young functions $\phi_{\log}(s) = e^s - 1 - s$ and $\phi_{\mathrm{sig}}(s) = -s - \ln(1-s)$: the first is the critical *sub-exponential* growth rate that appears in Bernstein-type concentration (Vershynin, 2018), giving an unbounded but tail-suppressing ball; the second is finite only on $[0, 1)$ and diverges at $s = 1$, so its ball is inscribed in and capped by the ℓ_∞ box. The remainder of the section motivates leaving the single-ℓ_p corner (§F.1), derives the dimensional calibration of λ (§F.2), and verifies that the non-homogeneous constraint set is a legitimate steepest-descent trust region (§F.3).

Algorithm 1 states the resulting optimizer in full. Both maps share a single loop and differ only in the link h: the vector face applies h coordinate-wise, while the matrix face applies it to the singular values of the momentum. The scale λ is the online $\mathrm{EMA}(\mathrm{RMS}(g))$ pooled per group (per tensor or per row, §4.3), estimated exactly as Adam estimates its second moment.

F.1 The ℓ_p family and why we leave the corner

At a fixed radius the ℓ_p balls are nested and *grow* with p, $B_{\ell_2} \subset B_{\ell_p} \subset B_{\ell_\infty}$ for $2 \le p \le \infty$, sweeping from the round ℓ_2 ball out to the sharp ℓ_∞ box. The single-norm methods live at the extremes: normalized SGD at ℓ_2, signSGD/Muon at the ℓ_∞ corner. Adam merely retreats a shrunken step inside that box, whose per-coordinate radius is set by the (inverse) noise-to-signal ratio. Committing to a single p (or to a uniformly contracted corner) is needlessly conservative when the gradient carries both a bulk-typical ℓ_2 component and heavy-tailed ℓ_∞-like extremes, and motivates the weighted family developed in §3.2.

Probabilistic reading of the two maps. Both Orlicz constraints have a clean prior interpretation through the Gibbs correspondence: a modular penalty Ω_ϕ on the update induces the density $p(\Delta) \propto \exp\big(-\lambda \sum_i \phi(|\Delta_i|)\big)$, so a steepest step inside the ϕ-ball is the maximum a posteriori (MAP) move under that prior. The two Young functions therefore encode two different beliefs about the update coordinates. For the log map, $\phi_{\log}(s) = e^s - 1 - s$ is exactly the critical sub-exponential moment-generating form that appears in Bernstein-type concentration (Vershynin, 2018): near the origin $\phi_{\log}(s) = s^2/2 + O(s^3)$, so the prior is Gaussian in the bulk and the map acts like ℓ_2, while the exponential growth of $\phi_{\log}$ in the tail suppresses large coordinates faster than any ℓ_p, giving a super-Gaussian but still full-support prior: large steps are penalized, never forbidden, which is the unbounded link $h(x) = \ln(1+x)$. For the sigmoid map, $\phi_{\mathrm{sig}}(s) = -s - \ln(1-s)$ shares the same $s^2/2$ bulk but *diverges* as $s \to 1$, so its prior has *compact support* on the box $(-1, 1)^d$: coordinates are Gaussian-distributed in the interior and assigned zero probability beyond the box edge. This is the probabilistic form of the ℓ_∞ cap: the bounded link $h(x) = x/(1+x) < 1$ can never push a coordinate past the wall, and is why the sigmoid ball of Fig. 8 is inscribed in the ℓ_∞ box while the log ball escapes it.

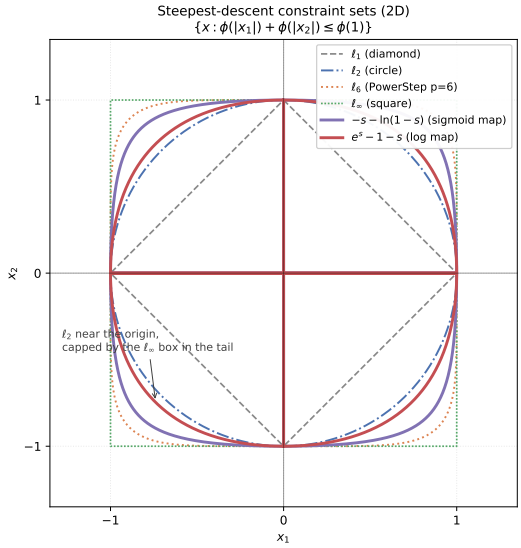


Figure 8: 2D steepest-descent constraint sets: the $\ell_1/\ell_2/\ell_6/\ell_\infty$ balls against the two interior Orlicz maps, the log ($\phi(s) = e^s - 1 - s$) and sigmoid ($\phi(s) = -s - \ln(1-s)$) maps, all normalized to pass through $(\pm 1, 0)$ and $(0, \pm 1)$. Both interior maps are round like ℓ_2 near the origin and bulge toward the ℓ_∞ box in the tail; the sigmoid ball is the boxier of the two and is inscribed in (capped by) the ℓ_∞ box, whereas the log ball is unbounded and keeps a milder curvature.

F.2 Setting λ: complementary slackness and the RMS calibration

We give here the derivation supporting the calibration $\lambda \propto \mathrm{RMS}(g)$ used in §3.2. There is no canonical value of λ: complementary slackness ties it to the radius R, which is itself a free scale, so any consistent choice of λ picks a corresponding ball. What is fixed is its *dimension*: in $\mathrm{sign}(m) \cdot h(|m|/\lambda)$ the ratio $|m|/\lambda$ must be dimensionless, so λ must scale like a gradient magnitude. Both $\|g\|$ and $\mathrm{RMS}(g)$ are dimensionally valid; empirically $\|g\|$ tunes poorly (its scale drifts with parameter count d), whereas $\mathrm{RMS}(g) = \sqrt{\|g\|^2/d}$ is stable across model scales and is what we use.

We also record a slackness-based derivation that lands on the same expression. Write the constraint as $\Omega_\phi(\Delta^\star) = R$ and let $\eta = \gamma d$ with $\gamma \in (0, 1)$ (a fixed fraction of the ambient dimension).

Sigmoid map ($\phi_{\mathrm{sig}}(s) = -s - \ln(1-s)$, $h(x) = x/(1+x)$). Substituting $|\Delta_i^\star| = |g_i|/(\lambda + |g_i|)$ into $\Omega_\phi(\Delta^\star) = \eta$ gives

$$\sum_{i=1}^{d} \left[\log\!\left(1 + \tfrac{|g_i|}{\lambda}\right) - \tfrac{|g_i|}{\lambda + |g_i|} \right] = \eta. \tag{30}$$

Under $\lambda \gg \max_i |g_i|$ the bracket is $\frac{1}{2}(|g_i|/\lambda)^2 + O((|g_i|/\lambda)^3)$, so the equation reduces to $\|g\|_2^2/(2\lambda^2) = \gamma d$, i.e. $\lambda \approx \mathrm{RMS}(g)/(\sqrt{2}\,\gamma)$.

Log map ($\phi_{\log}(s) = e^s - 1 - s$, $h(x) = \log(1+x)$). The slackness identity is

$$\sum_{i=1}^{d} \left[\tfrac{|g_i|}{\lambda} - \log\!\left(1 + \tfrac{|g_i|}{\lambda}\right) \right] = \eta, \tag{31}$$

whose second-order expansion under the same assumption is again $\|g\|_2^2/(2\lambda^2) = \gamma d$, giving the identical $\lambda \approx \mathrm{RMS}(g)/(\sqrt{2}\,\gamma)$.

The assumption $\lambda > \max_i |g_i|$ is strong (in fact strictly incompatible with $\lambda = \mathrm{RMS}(g)/(\sqrt{2}\gamma)$ once any coordinate deviates far from the bulk), and slackness by itself pins λ only relative to a chosen R. We therefore treat the expression above as a dimensional calibration rather than as an

exact prescription: it fixes units and gives a natural constant $\sqrt{2}\gamma$ which we absorb into the learning-rate sweep. The pooling granularity of $\mathrm{RMS}(g)$ (per-tensor, per-row, per-column, per-element) is a separate design axis, discussed in §3.2 and ablated in §4.3.

F.3 Well-posedness: the Orlicz constraint is a legitimate steepest-descent trust region

Steepest descent from a first-order Taylor model only requires a compact, convex, symmetric trust region with 0 in its interior; a norm ball is one choice among many. We record why replacing that ball with the non-homogeneous Orlicz sublevel set $C_R = \{x : \Omega_\phi(x) \le R\}$ raises no difficulty.

It is still a norm ball. C_R is convex (as ϕ is convex), symmetric (ϕ even), bounded (ϕ super-linear, or $+\infty$ past a finite argument), and contains 0 in its interior ($\phi(0) = 0$ and ϕ finite near 0). Any such set is the unit ball of its Minkowski gauge

$$\gamma_{C_R}(x) = \inf\{t > 0 : \Omega_\phi(x/t) \le R\}, \tag{32}$$

which is a genuine norm: positive homogeneity and the triangle inequality follow from the gauge construction and convexity, symmetry from that of C_R, and definiteness from boundedness. For R equal to the normalizing constant, γ_{C_R} is exactly the Luxemburg norm $\|\cdot\|_\phi$ and C_R its unit ball. Hence equation 4 *is* steepest descent under a norm, just not under any single ℓ_p.

What non-homogeneity changes (and why it is harmless). Ω_ϕ is a modular, not a norm: it is not 1-homogeneous, so increasing R *reshapes* C_R rather than rescaling it (near-ℓ_2 at the origin, drifting toward the box in the tail, Fig. 8), and the optimal direction depends on the scale λ, unlike an ℓ_p ball, where direction and step size decouple. This is the source of the scale-adaptive geometry we want, and it is why λ must be set meaningfully (§3.4). Homogeneity is never needed for correctness: the penalized form $\min_\Delta \langle m, \Delta\rangle + \lambda\Omega_\phi(\Delta)$ is a mirror-descent / proximal step with Bregman potential Ω_ϕ, whose solution is $\Delta^\star = \nabla\Omega_\phi^*(-m/\lambda)$ (identical to equation 7). Mirror-descent potentials are routinely non-homogeneous (e.g. the negative entropy on the simplex), and our convergence analysis (App. C.2) rests only on the strong convexity $\Phi'' \ge 1$ / co-coercivity of P_λ, not on any homogeneity.

The one requirement: a bounded trust region. For the linearization to stay valid the trust region must bound the step, so that the dropped remainder $\frac{L}{2}\|\Delta\|_2^2$ is controlled. The sigmoid map enforces this intrinsically ($\|\Delta\|_\infty < 1$); the log map does not (its link is unbounded), so at large signal-to-noise the step can leave the region where the first-order model is trusted, precisely the divergence observed at 4B, which re-imposing a bounded region by clipping removes (App. D.1).