PRECISION-AWARE OPTIMAL NEWTON–SCHULZ ITERATIONS FOR THE MATRIX SIGN

Anonymous authors
Paper under double-blind review

ABSTRACT

The MUON optimizer orthogonalizes each gradient matrix by running a short quintic Newton–Schulz (NS) iteration that drives its singular values to one—an approximate matrix sign function, $\mathrm{msign}(M) = UV^\top$. Recent work (POLAR EXPRESS) shows that the per-step *coefficients* of this iteration should not be hand-tuned but *solved for*: a per-step minimax (Remez) fit yields provably fastest convergence in exact arithmetic. But MUON actually runs the iteration in *bfloat16*, and no prior analysis characterizes these learned iterations under low precision. We give a precision-aware characterization of the coefficient *design space*, deliberately scoped to the *idealized scalar reduction* of the iteration—the singular-value band $[1/\kappa, 1]$ with per-scalar rounding—and bounded honestly against real gradients and real training. We study the sign iteration as an *optimal–structured–hybrid* family—P1, a free per-step minimax fit (POLAR EXPRESS); P2, the same minimax under a boundedness constraint; and P3, a hybrid that runs free steps early and bounded steps late—and measure error-versus-step curves in fp32 and bf16 across $\kappa \in \{10, 30, 100\}$. Three findings. **(i)** On the idealized band, the fp32 per-step-optimal P1 crushes fixed MUON at a matched budget (10^{-8} vs. 0.318 at $\kappa = 10$; 7.8×10^{-6} vs. 0.318 at $\kappa = 100$). **(ii)** On that *same idealized band*, bf16 is fragile at high conditioning: at $\kappa = 100$ the aggressive P1 coefficients make the bf16 scalar iteration *diverge* (terminal 4×10^{18}; P2 overflows to $+\infty$), while the hybrid P3 stays finite (bf16 floor 0.141, beating fixed's 0.328); we give a first-order model of this floor. **(iii) This bf16-*specificity* is a feature of the idealized band, not of real gradients.** Running the *true matrix* iteration on real 124M gradient matrices (fp32-accumulate tensor cores), the optimal P1 coefficients are fragile in *both* precisions—blowing up on $8/8$ matrices in bf16 and $7/8$ in fp32, so the clean bf16-only divergence appears for just $1/8$—while fixed MUON stays stable (residual ≈ 0.32). **And the deployed iteration is bf16-safe end-to-end:** a nanoGPT-124M premise test trains equally well in bf16 and fp32 (a null, -0.038 nats). Our contribution is thus a precision-aware map of the coefficient design space *in the idealized setting*, honestly bounded: the optimal-coefficient instability does not clearly transfer to real gradients bf16-specifically, and today's MUON is already bf16-safe—not a fix for a real deployment problem. Extending precision-aware optimal iterations to general spectral functions is left to future work.

1 INTRODUCTION

The MUON optimizer (Jordan et al., 2024) takes a step by replacing the raw gradient (momentum) matrix $M = U\Sigma V^\top$ with its orthogonal polar factor $\mathrm{msign}(M) = UV^\top$—the matrix in which every singular value has been mapped to one—and it computes this factor not by an SVD but by a handful of matrix multiplications: a quintic Newton–Schulz (NS) polynomial iteration run, crucially, in bfloat16. The same orthogonalization primitive underlies preconditioners such as Shampoo (Gupta et al., 2018) and the modular-duality view of optimization (Bernstein & Newhouse, 2024a), and the matrix sign / polar factor is a classical object in numerical linear algebra (Higham, 2008).

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.

The coefficients should be solved for, not guessed. The NS iteration at MUON's heart has free per-step coefficients (a_t, b_t, c_t)—acting on each singular value as $\sigma \mapsto a_t\sigma + b_t\sigma^3 + c_t\sigma^5$—and the original optimizer fixes them to a single hand-tuned triple used at every step. POLAR EXPRESS (Amsel et al., 2025) made the sharp observation that these coefficients are the solution of an optimization problem: at each step one should choose the odd polynomial that *minimaxes* the distance between the current singular-value interval and the target value 1, exactly the classical Remez problem (Trefethen, 2013). Solving step-by-step yields a schedule that converges faster than any fixed choice and comes with clean approximation-theoretic guarantees. A subsequent hardware-aware Gram-matrix reformulation reaches similar conclusions from a different route (Zhang et al., 2026). This is a genuinely appealing idea: replace tuning with a solver.

The gap: the analysis is full-precision, the deployment is bf16. Both lines of work analyze the iteration in exact (or fp32) arithmetic. Yet MUON runs it in bf16, and the interaction between polynomial conditioning and rounding is uncharted for these solved schedules. The per-step-optimal coefficients are *aggressive*: to collapse a wide interval onto 1 as fast as possible, an intermediate step may transiently *expand* singular values before contracting them. In fp32 this is harmless; in bf16, whose dynamic range is exercised far more than its precision suggests, an expanding step can, *in the idealized scalar model of the iteration*, send an iterate into a self-amplifying blow-up (numerical *divergence*) and, in the extreme, into *overflow*—so within that model the fp32-optimal and bf16-stable iterations need not coincide. Whether this idealized hazard *transfers* to real gradients and real training is a separate, load-bearing question we answer explicitly (and negatively) below.

This paper. We give the first precision-aware account of the MUON sign iteration. We keep the quintic NS template but treat the choice of per-step coefficient *schedule* as the design variable, organized into an *optimal–structured–hybrid* family (Section 4): **P1**, a free per-step minimax fit (the POLAR EXPRESS idea); **P2**, the same minimax under a boundedness constraint that keeps each step's image within a fixed range so iterates stay bf16-representable; and **P3**, a hybrid that runs free (P1) steps early—to make fast progress while the interval is wide—and bounded (P2) steps late—to stay safe as it converges. Against the fixed MUON quintic as a transparent baseline, we measure full error-versus-step curves ($T = 1 \dots 8$) in both fp32 and bf16 across condition numbers $\kappa \in \{10, 30, 100\}$ *on the idealized scalar band* $[1/\kappa, 1]$—MUON's spectral-norm-normalized working picture—and we make four contributions.

- **C1 (fp32: solving wins).** The per-step minimax P1 reaches terminal error 10^{-8} ($\kappa = 10$) to 7.8×10^{-6} ($\kappa = 100$) at $T = 8$, versus 0.318 for the fixed MUON quintic at the *same* budget; we report full curves, so it is a fair error-vs-step trajectory, not a cherry-picked ratio (Section 5, E1).
- **C2 (bf16 on the idealized band: the fp32-optimal iteration breaks).** On the idealized scalar band, at $\kappa = 100$ the free P1 coefficients *diverge* in bf16 (terminal 4.07×10^{18}—finite but $\sim$20 orders too large, and headed for genuine overflow; best-over-T floor stuck at 0.387), and a naive boundedness constraint P2 diverges even faster and *overflows* to $+\infty$ (floor 0.688). Within this idealized design space the message is sharp: the schedule you would solve for in fp32 is not the one that stays stable in bf16 (E2)—a phenomenon of the idealized band, scoped honestly in C4, not a statement that real MUON training destabilizes in bf16.
- **C3 (the hybrid is the precision-robust choice on the idealized band).** P3 is the only method finite in bf16 at *all* κ; at $\kappa = 100$ its bf16 floor (0.141) beats fixed MUON (0.328), and at $\kappa \leq 30$ all learned methods are bf16-stable and beat fixed by 5–20$\times$ (E2). We back this with a first-order bf16 error-floor model (Proposition 1) and are explicit about the price: P3 trades some fp32 accuracy for stability (fp32 0.235 at $\kappa = 100$, still better than fixed's 0.318). No free lunch.
- **C4 (honest bound: the bf16-specificity does not transfer, and deployed MUON is bf16-safe).** Two controls scope the C2 finding. First, running the *true matrix* iteration on real 124M gradient matrices (torch, fp32-accumulate tensor cores), the aggressive P1 coefficients are fragile in *both* precisions—they blow up on $8/8$ matrices in bf16 and $7/8$ in fp32, so the clean bf16-*only* divergence appears for just $1/8$—while fixed MUON stays stable (residual ≈ 0.32). The bf16-*specificity* of E2 is therefore an artifact of the idealized band, not a property of real near-rank-one gradients (E4). Second, an end-to-end premise test (nanoGPT-124M, E5) shows the *deployed fixed* iteration trains equally well in bf16 and

fp32 (a null, -0.038 nats, faster in bf16). We chart a design-space hazard of the aggressive *optimal* coefficients in an idealized model—not a defect in today's MUON.

The core study is cheap to reproduce (CPU-bound minimax solving on singular-value grids, minutes). Our thesis is a *mapping*, not a deployment recommendation: the coefficient design space for MUON's sign iteration has fp32-optimal and bf16-stable answers that *diverge in the idealized model*, and characterizing that space—the optimal/structured/hybrid family plus a bf16 error-floor model, honestly delimited by C4—is the contribution. We do *not* claim to fix a real bf16 problem in MUON.

2 RELATED WORK

Newton–Schulz and matrix-sign iterations. Computing the matrix sign function and the closely related polar factor by polynomial or rational iteration is classical. The Newton–Schulz iteration $X_{t+1} = \frac{1}{2}X_t(3I - X_t^\top X_t)$ converges globally from a suitably scaled start and cubically near the fixed point (Higham, 2008; Kenney & Laub, 1991). Its roots go back to Schulz's iterative matrix inverse (Schulz, 1933), the iterative pseudoinverse literature (Ben-Israel, 1966; Söderström & Stewart, 1974; Pan & Schreiber, 1991), orthogonality-improving iterations (Kovarik, 1970), and fast polar-decomposition schemes (Higham & Schreiber, 1990), from which the MUON-style orthogonalization inherits its matmul-only structure. Higher-order and rational variants—Padé/Halley iterations, the QDWH scheme, and Zolotarev's optimal rational approximants—trade multiplications for conditioning (Higham, 1986; Nakatsukasa et al., 2010; Nakatsukasa & Freund, 2016), and the sign function has a long history in control and model reduction, from the Denman–Beavers iteration (Denman & Beavers, 1976) through the algebraic Riccati equation (Roberts, 1980; Byers, 1987) and the scaled Newton variants that accelerate it (Kenney & Laub, 1992); modern surveys and the spectral divide-and-conquer viewpoint are in Higham (2008); Nakatsukasa & Higham (2013); Golub & Van Loan (2013); Demmel (1997). Related spectral iterations—fractional matrix powers and pth roots for Shampoo-style preconditioning—have their own literature (Iannazzo, 2006; Bini et al., 2005; Higham, 1997; Gawlik, 2019); we do not study them here (future work). All of this classical work fixes a target and studies *how fast* a given iteration reaches it in exact arithmetic; we instead ask which *per-step coefficient schedule* is best for the sign target under the bf16 precision MUON actually runs.

MUON and spectral optimizers. MUON (Jordan et al., 2024) orthogonalizes the momentum matrix each step via a fixed quintic Newton–Schulz iteration run in bf16, and is competitive with adaptive optimizers (Kingma & Ba, 2015; Duchi et al., 2011; Loshchilov & Hutter, 2019; Reddi et al., 2018)—themselves descendants of classical momentum and acceleration (Polyak, 1964; Nesterov, 1983)—on language-model pretraining at lower memory. It sits in a family of spectral and preconditioning optimizers: Shampoo (Gupta et al., 2018) and its scalable / Adam-stabilized variants (Anil et al., 2020; Vyas et al., 2024), Kronecker-factored curvature (Martens & Grosse, 2015), the operator-norm / trust-region view of learning (Bernstein et al., 2020; Pethick et al., 2025), maximal-update parameterization (Yang & Hu, 2021; Yang et al., 2021), the scalable MUON variant of Liu et al. (2025), and the modular-duality and normed-descent framework of Bernstein & Newhouse (2024a;b); Large et al. (2024) that recasts such updates as steepest descent under an operator norm. What unites them is a spectral matrix function computed by matmul-only iteration; our focus is the numerical behavior of that inner iteration for the sign target under low precision.

POLAR EXPRESS: optimal coefficients, in exact arithmetic. Closest to us, POLAR EXPRESS (Amsel et al., 2025) shows that the per-step coefficients of the MUON iteration should be chosen by a minimax (Remez) fit against the target value 1 over the current singular-value interval, yielding the fastest-converging schedule with clean L^∞ guarantees; a subsequent hardware-aware Gram-matrix reformulation reaches a related optimum with better numerical behavior (Zhang et al., 2026). Our P1 schedule *is* this per-step minimax, and we reproduce its fp32 dominance (E1). Our delta is precision-awareness: *on the idealized scalar band* we show that this fp32-optimal schedule *diverges* in bf16 at high conditioning (E2), that a single boundedness constraint does not fix it, and that a free-early/bounded-late hybrid restores bf16 stability across all conditioning (E3)—a low-precision failure mode and remedy that the exact-arithmetic analysis does not surface. We bound the claim: a true-matrix control on real gradients shows this bf16-*specificity* does not transfer (the optimal

coefficients are fragile in both precisions there, E4), so ours is a characterization of an *idealized design space*, not a fix for real MUON.

Approximation theory and minimax fitting. Our coefficient solver is standard best-approximation machinery: the Remez exchange algorithm and Chebyshev/minimax theory (Remez, 1934; Powell, 1981; Cheney, 1966; Rivlin, 1969; Trefethen, 2013; Pachón & Trefethen, 2009), cast per-step as a small optimization over the quintic coefficients (Boyd & Vandenberghe, 2004; Diamond & Boyd, 2016). The novelty is not the solver but that it is pointed at a *precision-constrained* schedule: the bounded minimax (5) adds a representability constraint absent from the classical problem.

Low-precision numerical linear algebra. Mixed-precision and low-bit training are now standard (Micikevicius et al., 2018; Kalamkar et al., 2019; Gupta et al., 2015; Wang et al., 2018; Dettmers et al., 2022), and rounding-error analysis of iterative linear algebra is classical (Higham, 2002), with a recent survey of mixed-precision numerical linear algebra (Higham & Mary, 2022). What is missing—and what this paper supplies—is a precision-aware account of the *learned* MUON sign iteration: how, *in the idealized scalar model*, bf16's narrow range turns the fp32-optimal schedule into a numerical-divergence hazard at high condition number, and how—honestly bounded (E4/E5)—this bf16-specific hazard is a property of the idealized band, not of real gradients or of deployed MUON.

3 BACKGROUND AND PROBLEM SETUP

The matrix sign / polar factor. Let $M \in \mathbb{R}^{m\times n}$ with (thin) singular value decomposition $M = U\Sigma V^\top$, $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_r)$, $\sigma_i \geq 0$. The orthogonal polar factor (matrix sign, for the symmetric-part convention) is

$$\mathrm{msign}(M) \;=\; U\,V^\top \;=\; U\,\mathbf{1}(\Sigma)\,V^\top, \tag{1}$$

i.e. the spectral matrix function (Higham, 2008; Horn & Johnson, 2012; Bhatia, 1997) that maps every singular value to 1 while fixing the singular vectors. This is the object MUON needs at every step. A direct SVD costs $O(mn\min(m,n))$ (Golub & Van Loan, 2013; Golub & Kahan, 1965) and runs poorly in the low-precision, matmul-only regime that motivates MUON (it relies on orthogonal reductions and divisions ill-served by bf16 tensor cores). The alternative is an *iteration built from matrix multiplications*, the route MUON and its relatives take (Kenney & Laub, 1992; 1991).

The quintic Newton–Schulz step acts coordinate-wise. MUON uses a fixed quintic NS step, and every method we study keeps the same template

$$X_{t+1} \;=\; a_t X_t + b_t\,(X_t X_t^\top)X_t + c_t\,(X_t X_t^\top)^2 X_t \;=\; X_t\,p_t\big(X_t^\top X_t\big), \tag{2}$$

an odd matrix polynomial with per-step coefficients (a_t, b_t, c_t) (the fixed MUON choice is $(3.4445, -4.7750, 2.0315)$, used at every step). The reason such iterations are tractable is a structural fact we use throughout: if X_0 shares the singular vectors of M—in particular if $X_0 = M/\alpha$ for a scalar α—then every iterate keeps those vectors and acts only on the singular values.

Lemma 1 (Coordinate reduction)**.** *If $X_t = UD_tV^\top$ with $D_t = \mathrm{diag}(x_1, \ldots, x_r)$, then* (2) *gives $X_{t+1} = UD_{t+1}V^\top$ with each coordinate updated independently by the scalar map $x \mapsto x\,p_t(x^2) = a_t x + b_t x^3 + c_t x^5$.*

(Proof: $X_t^\top X_t = VD_t^2V^\top$ is diagonal in the V basis, so $p_t(X_t^\top X_t) = Vp_t(D_t^2)V^\top$ and the factors telescope; full statement in Appendix A.) Running T steps composes T scalar maps $h_t(x) = a_t x + b_t x^3 + c_t x^5$, and the entire matrix problem reduces to a *scalar approximation problem*: choose the coefficients so that the composite $\varphi_T = h_T \circ \cdots \circ h_1$ satisfies $\varphi_T(\sigma) \approx 1$ for all singular values σ in the range of interest.

Normalization and the working interval. Because the reduction is coordinate-wise, the only spectral information that matters is the interval containing the (scaled) singular values. We prescale $X_0 = M/\alpha$ by a cheap upper bound $\alpha \geq \sigma_{\max}$ so the spectrum lands in $[1/\kappa, 1]$, where $\kappa = \sigma_{\max}/\sigma_{\min}$—exactly MUON's spectral-norm-normalized picture, and the single knob controlling how hard the scalar problem is: a wider band forces the composite to crush a longer interval onto 1. We sweep $\kappa \in \{10, 30, 100\}$ (E1–E2), standard for sign/polar iterations (Kenney & Laub, 1992; Higham, 2008).

Minimax (Remez) fitting. Given the target value 1 on the current interval I and the odd-quintic family, the best-approximation problem

$$\min_{a,b,c} \max_{x \in I} \left| ax + bx^3 + cx^5 - 1 \right| \tag{3}$$

is the classical L^∞ (Chebyshev) problem, solved by Remez exchange or, on a fine grid, a small program (Trefethen, 2013; Diamond & Boyd, 2016). (3) is the atom from which every schedule is built; the design choices are (i) what interval to fit at each step and (ii) whether the fit is free or subject to a boundedness constraint keeping the step's image bf16-representable—which we treat next.

Why precision is a first-order concern. bf16 has an 8-bit mantissa (unit roundoff $u \approx 2^{-8} \approx 3.9 \times 10^{-3}$) but the *same* 8-bit exponent as fp32, so its failure mode on these iterations is less about mantissa rounding than about *dynamic range*. A per-step map h_t that transiently pushes an iterate to a large value—which the fp32-optimal coefficients do on a wide interval—can, *in the idealized scalar model of the iteration over a uniform band*, send the iterate into a self-amplifying blow-up (numerical *divergence*); in the extreme it exceeds the largest bf16 number and overflows to ∞, after which no later step can recover. This numerical instability is the phenomenon our experiments expose at high κ *on the idealized band* (Section 5); we then bound it honestly (E4/E5), so the takeaway is a design-space caution, not a claim that current MUON is bf16-unstable.

4 AN OPTIMAL–STRUCTURED–HYBRID FAMILY FOR THE SIGN ITERATION

By Lemma 1 every method we consider runs the same quintic template (2); what differs is the per-step coefficient *schedule* $\{(a_t, b_t, c_t)\}_{t=1}^T$. We describe four schedules—the fixed MUON baseline and three solved variants forming an optimal–structured–hybrid family—then analyze how they behave under bf16 rounding. Throughout, the scalar target is $\varphi_T(\sigma) \to 1$ on the working band $[1/\kappa, 1]$ (Section 3).

4.1 FOUR COEFFICIENT SCHEDULES

Baseline: fixed MUON. The stock optimizer uses one hand-tuned quintic $(3.4445, -4.7750, 2.0315)$ at every step. It is conservative—its step map keeps iterates bounded, so it never overflows bf16—but far from per-step optimal, and its terminal error plateaus (at ≈ 0.318) rather than driving to zero. We show its real error-vs-step curve alongside every learned schedule.

(P1) Free per-step minimax (POLAR EXPRESS). The direct application of the "solve, don't tune" idea (route ② of the source discussion (Su, 2025)) chooses each step's quintic *greedily* to minimize worst-case distance to 1 over the current image interval. Writing φ_{t-1} for the composite so far and $I_{t-1} = \varphi_{t-1}([1/\kappa, 1])$ for its image,

$$(a_t, b_t, c_t) \;=\; \arg\min_{a,b,c} \max_{x \in I_{t-1}} \left| ax + bx^3 + cx^5 - 1 \right|. \tag{4}$$

This is exactly the POLAR EXPRESS per-step minimax (Amsel et al., 2025), and $I_t \to \{1\}$. P1 is the "optimal" vertex: fastest fp32 convergence (solved numerically by Nelder–Mead, so near-optimal rather than certified Remez; Appendix B), but its coefficients are unconstrained and can be large, so a step's image may transiently exceed 1 by a wide margin—the seed of the bf16 problem below.

(P2) Bounded / structured minimax. To make the schedule precision-safe we solve the *same* minimax but constrain each step's map to stay within a fixed magnitude B over the representable sub-interval (route ①, the "structured" idea, of Su, 2025):

$$(a_t, b_t, c_t) \;=\; \arg\min_{a,b,c} \max_{x \in I_{t-1}} \left| ax + bx^3 + cx^5 - 1 \right| \quad \text{s.t.} \quad \max_{y \in [0,B]} \left| ay + by^3 + cy^5 \right| \le B. \tag{5}$$

The constraint ($B = 1.3$) forces every step to map $[0, B]$ into $[-B, B]$, so no intermediate iterate inflates past B and the trajectory stays inside the bf16 range. P2 is the "structured" vertex: it sacrifices some per-step optimality for a bounded image. As we will see, boundedness alone is *not* enough at the hardest conditioning—a fixed B safe on a narrow band can be overrun once the interval is very wide.

	Fixed Muon	**P1 Free**	**P2 Bounded**	**P3 Hybrid**
Coefficients per step	one fixed triple	minimax (4)	bounded minimax (5)	free early / bounded late
Per-step optimal (fp32)	no	**yes**	partial	**yes** (early)
Image bounded (bf16-safe)	yes	no	by design	late steps
fp32 terminal error	plateau	**best**	good	good
bf16 stable at high κ	yes	**no**	**no**	**yes**

Table 1: The optimal–structured–hybrid family for the sign iteration. POLAR EXPRESS occupies the P1 vertex; the fixed MUON quintic is the conservative baseline. On the idealized band, only P3 is both fast in fp32 and stable in bf16 across all conditioning; the "bf16 stable at high κ" row refers to that band, not to real gradients, where the optimal coefficients are fragile in any precision (E4).

(P3) Hybrid (free early, bounded late). The two ideas are complementary in time: early the interval is wide and the priority is fast progress; late it is near 1 and the priority is not to overflow while polishing. P3 runs the free schedule (4) for the first $\lceil T/2 \rceil$ steps and the bounded schedule (5) for the rest:

$$(a_t, b_t, c_t) \text{ from } \begin{cases} (4) & t \le \lceil T/2 \rceil \quad \text{(free)}, \\ (5) & t > \lceil T/2 \rceil \quad \text{(bounded)}. \end{cases} \tag{6}$$

P3 is the "optimal $\cap$ structured" vertex: it inherits P1's aggressive early contraction while capping the late steps so the composite never leaves the bf16 range. Table 1 summarizes the family.

Budget knobs. We fix the per-step degree to the quintic $d = 5$ (matching MUON) and sweep $T = 1 \dots 8$, so differences are attributable to the schedule, not the budget. The same construction targets any odd spectral function g by replacing the fit target 1 in (4)–(5) with g; here we study $g = \mathrm{msign}$ only.

4.2 LOW-PRECISION (BF16) BEHAVIOR

bf16 has unit roundoff $u \approx 2^{-8} \approx 3.9 \times 10^{-3}$ but only an 8-bit exponent, so on these iterations the binding constraint is dynamic range as much as precision. We model each executed step *in the idealized scalar reduction* (Lemma 1, with per-scalar rounding) as the exact step plus a rounding perturbation ε_t with $\|\varepsilon_t\| \lesssim c\,u$; the schedule decides both how these perturbations accumulate and whether an iterate can leave the representable range entirely. This model is deliberately idealized in two ways we revisit in Section 5: a uniform band $[1/\kappa, 1]$ rather than a real spectrum, and per-scalar rounding, *harsher* than fp32-accumulate tensor cores.

Proposition 1 (Error floor and overflow, informal)**.** *Consider running the schedule in bf16.* (i) *If every step map keeps its image bounded, $|h_t(x)| \le B$ for all reachable x, then no iterate overflows and, by the first-order model, the terminal error settles at a floor of order $c\,u$ (mantissa-limited) that does not grow with T.* (ii) *If a step map can* expand*—$\max_x |h_t(x)|$ exceeds the current iterate magnitude, as the free minimax* (4) *does on a wide interval $[1/\kappa, 1]$—then on a sufficiently wide band (large κ) some coordinate of the idealized scalar iterate blows up (numerical* divergence*, growing by orders of magnitude per step while still finite) and, pushed far enough, is driven past the largest representable bf16 value and overflows to ∞, after which no later step recovers it. The best attainable error is then the* floor before the blow-up*, $\min_t$ of the error-vs-step curve, not the terminal value.*

Proof sketch. For a bounded schedule the linearized recursion $e_{t+1} = \rho_t e_t + \varepsilon_t$ has $|\rho_t| < 1$ near the target, giving a geometric steady state $c\,u/(1 - \bar{\rho})$, and boundedness guarantees every x_t is finite and representable. For an expanding step, let $\mu_t = \max_{x \in I_{t-1}} |h_t(x)|$; whenever the running product of expansions carries a coordinate above bf16's maximum ($\approx 3.4 \times 10^{38}$), that coordinate rounds to ∞ and propagates. Because (4) optimizes fp32 progress with no range constraint, μ_t grows with the interval width $1 - 1/\kappa$, so the overflow is triggered precisely at large κ. Full model in Appendix D. □

Proposition 1 is a design-level prediction *for the idealized scalar model*, not a worst-case rounding proof; its role is to make the bf16 story falsifiable. It predicts (a) that the free P1 schedule diverges in bf16 at large κ *on the idealized band* (and, pushed further, overflows) while the fixed MUON baseline does not, and (b) that keeping late steps bounded (P3) prevents the divergence. Section 5 checks

both predictions on the idealized band ($\kappa = 100$ confirms them), then tests whether they transfer to real training: they do *not* transfer bf16-specifically—a true-matrix control (E4) finds the aggressive coefficients fragile in *both* precisions on real gradients, so the model's bf16-specificity is a feature of the idealization.

4.3 COST

Every schedule is matmul-only. For $M \in \mathbb{R}^{m\times n}$ ($m \geq n$) one step of (2) forms $X^\top X$ once ($O(mn^2)$) and evaluates the quintic by Horner's rule, so a T-step schedule costs $O(Tmn^2)$ flops with no SVD, root, or division. P1/P2/P3 run the *same* template at the same per-step cost as fixed MUON; they change *which* coefficients are used, not the operation count, so any drop-in replacement is cost-neutral at matched T. The coefficient solve (4)–(5) is an offline CPU-bound minimax over three scalars per step, producing a small table indexed by (κ, T) loaded without re-solving.

5 EXPERIMENTS

We test the claims of Section 4 in two tiers. On the *idealized scalar band* (E1–E3): C1 that the free per-step minimax (P1) beats the fixed MUON quintic in fp32; C2 that in bf16 the fp32-optimal schedule breaks at high conditioning; C3 that the hybrid P3 is the precision-robust choice there. Then two *honesty controls* that bound how far the idealized story reaches: E4, a true-matrix control on real gradients showing the bf16-*specificity* does not transfer, and E5, an end-to-end null showing the deployed iteration is already bf16-safe. All numbers below are *measured*; the divergence/overflow cells report the actual numerical failure (it is the finding, not an omission).

Setup and metric. By the coordinate reduction (Lemma 1) the matrix iteration is exactly a scalar iteration on the singular values *in exact arithmetic*, so for E1–E3 we evaluate each schedule on a dense grid of 600 points over the *idealized* band $[1/\kappa, 1]$ (uniform spectrum, per-scalar bf16 rounding) for $\kappa \in \{10, 30, 100\}$. (E4 drops both idealizations and runs the true matrix iteration on real gradients.) For a step budget T we report the terminal worst-case error $\max_{x\in[1/\kappa,1]} |\varphi_T(x) - 1|$, i.e. the largest deviation of any singular value from the target 1 after T quintic steps. We compute the full curve for $T = 1 \ldots 8$ so the comparison is a transparent trajectory, and we additionally report, for bf16, the *floor*: the best (minimum) error attained over $T \leq 8$, which is the right summary when a schedule overflows at large T. Coefficient schedules are solved once in fp32 by the greedy/bounded minimax (4)/(5) (Nelder–Mead per step, warm-started from the MUON triple); bf16 is simulated by round-to-nearest-even truncation of every intermediate product (Appendix B).

E1 — fp32 accuracy: solving wins (C1). The fp32 columns of Table 2 confirm the POLAR EXPRESS thesis cleanly. At a matched budget $T = 8$ the free per-step minimax P1 reaches 1.0×10^{-8} ($\kappa = 10$) to 7.77×10^{-6} ($\kappa = 100$), versus 3.18×10^{-1} for the fixed MUON quintic at *every* κ and T—a 7–10 order-of-magnitude improvement. Because we show the full curves (Appendix C)—P1 improving monotonically while the fixed baseline saturates by $T \approx 3$—it is a fair error-vs-step trajectory, not a cherry-picked ratio.

E2 — bf16 stability on the idealized band: the fp32-optimal schedule breaks (C2). On the idealized band the bf16 columns tell the opposite story at high conditioning. At $\kappa \leq 30$ all learned schedules stay bf16-stable and beat fixed by 5–20$\times$ on the floor. But at $\kappa = 100$ the free P1 coefficients drive the idealized scalar iterate into a self-amplifying blow-up: the terminal bf16 error *diverges* to 4.07×10^{18} (floor stuck at 3.87×10^{-1}, worse than fixed), and the naive boundedness constraint P2 *overflows* to $+\infty$ (floor 6.88×10^{-1}, the worst of all), because a fixed bound B calibrated on a narrow band is overrun once the band stretches to $[1/100, 1]$. Scoped to this idealized design space, the schedule you would solve for in fp32 is not the one that stays stable in bf16, and a single boundedness knob is insufficient—exactly the mode Proposition 1(ii) predicts. This is a bf16-*specific* failure *of the idealized band*; E4 shows it does not carry over to real gradients bf16-specifically.

E3 — the hybrid is the robust choice on the idealized band, at an honest price (C3). On the idealized band, P3 is the only schedule finite in bf16 at *all* κ (Table 2, green cells): at $\kappa = 100$ its

κ	Method	fp32 err (T=8)	bf16 err (T=8)	bf16 floor (min over T)
10	fixed MUON	3.18×10^{-1}	3.28×10^{-1}	3.28×10^{-1}
	P1 free	$\mathbf{1.0 \times 10^{-8}}$	1.56×10^{-2}	1.56×10^{-2}
	P2 bounded	7.05×10^{-3}	5.47×10^{-2}	$\mathbf{1.17 \times 10^{-2}}$
	P3 hybrid	6.83×10^{-3}	7.81×10^{-2}	3.12×10^{-2}
30	fixed MUON	3.18×10^{-1}	3.28×10^{-1}	3.28×10^{-1}
	P1 free	$\mathbf{3.45 \times 10^{-7}}$	1.56×10^{-2}	$\mathbf{1.56 \times 10^{-2}}$
	P2 bounded	6.95×10^{-2}	7.03×10^{-2}	7.81×10^{-3}
	P3 hybrid	1.65×10^{-2}	5.47×10^{-2}	5.47×10^{-2}
100	fixed MUON	3.18×10^{-1}	3.28×10^{-1}	3.28×10^{-1}
	P1 free	$\mathbf{7.77 \times 10^{-6}}$	4.07×10^{18} (diverges)	3.87×10^{-1}
	P2 bounded	1.59×10^{-2}	$+\infty$ (overflow)	6.88×10^{-1}
	P3 hybrid	2.35×10^{-1}	$\mathbf{2.81 \times 10^{-1}}$ (stable)	$\mathbf{1.41 \times 10^{-1}}$

Table 2: *Idealized scalar band* (uniform band, per-scalar bf16 rounding). Terminal worst-case error $\max_{x\in[1/\kappa,1]} |\varphi_T(x) - 1|$ at T=8 in fp32 and bf16, and the bf16 floor (best over $T \leq 8$), for the fixed MUON quintic and the three solved schedules. In fp32 the free minimax P1 beats fixed by 7–10 orders; at κ=100 in bf16 the aggressive P1 *diverges* (4.07×10^{18}, headed for overflow) and P2 *overflows* to $+\infty$, while only the hybrid P3 stays finite (floor 1.41×10^{-1}, beating fixed's 3.28×10^{-1}). This bf16 divergence is band-specific: on real gradients the optimal coefficients are fragile in *both* precisions (E4). Bold marks the best value per column-block; full T=1 . . . 8 curves in Appendix C.

bf16 floor 1.41×10^{-1} *beats* fixed MUON's 3.28×10^{-1} while the fp32-optimal schedules have blown up—running free steps early captures most of P1's fast contraction while the interval is wide, then bounded steps late cap the images before any can overflow. We are explicit about the cost: P3 trades fp32 accuracy for stability (at $\kappa = 100$ its fp32 error 2.35×10^{-1} is far worse than P1's 7.77×10^{-6}, still better than fixed's 3.18×10^{-1}), and at $\kappa \leq 30$ it offers no advantage over P1. No single schedule is best everywhere; we do not claim a free lunch. (All of E1–E3 is on the idealized band; E4 tests transfer.)

E4 — true-matrix control: the bf16-specificity does *not* transfer to real gradients. The load-bearing question is whether E2's bf16-specificity survives on real gradients under real arithmetic. We logged the 2D gradient/momentum matrices MUON orthogonalizes during a real 124M GPT run and ran the *true matrix* quintic on them in torch with fp32-accumulate tensor cores—dropping *both* idealizations of E1–E3 (uniform band and per-scalar rounding). Real gradients do reach the high-κ regime: they are near-rank-one (stable rank $\approx$ 1.07), and the effective condition number $\kappa_{\text{eff}} = \sigma_{\max}/\sigma_{@99\%}$ has median 9.0, p90 108 (Appendix E). But the divergence there is *not* bf16-specific: solving the P1 coefficients per matrix and running the real iteration on 8 ill-conditioned matrices ($\kappa_{\text{eff}} \approx$ 65–109), the aggressive P1 schedule blows up on $8/8$ **matrices in bf16 *and*** $7/8$ **in fp32**—the clean "bf16-only" divergence of E2 appears for only $1/8$—while fixed MUON stays stable in both precisions (residual ≈ 0.32; Table 5, Appendix F). On real near-rank-one gradients the per-step-optimal coefficients are thus fragile in *any* precision; the bf16-*specificity* of E2 is a property of the idealized band and its per-scalar rounding, *not* of real gradient matrices. This is the paper's central honesty bound: nothing here implies that adopting the optimal coefficients would destabilize real MUON training in bf16 specifically.

E5 — end-to-end premise test: is the deployed iteration bf16-safe? (yes; an honest bound). E2/E4 warn about the *aggressive optimal* coefficients (P1); they do not say the iteration MUON ships is broken in bf16. We test that premise directly: train nanoGPT-124M (Karpathy, 2022) for 3000 steps (2 seeds) with the deployed *fixed* quintic, changing *only* the sign-iteration precision. Validation loss is statistically indistinguishable—fp32 5.086 ± 0.015 vs. bf16 5.048 ± 0.024 nats, a gap of -0.038 nats (bf16 marginally *lower*, within $\sim$two std), while bf16 is $\sim$23% faster (Appendix G). *The verdict is a null.* Together with E4, this bounds the paper honestly: the bf16 instability is an *idealized-band* property of the per-step-*optimal* schedule (not bf16-specific on real matrices, absent from the deployed fixed iteration), so P3 is a design-space remedy for those who adopt the aggressive coefficients, not a fix current MUON needs. Every idealized-band number comes from a single released CPU script; details and curves are in Appendices B–G.

6 DISCUSSION

Why precision-awareness is the point (in the design space). The usual account is full-precision: pick the coefficients that converge fastest, run a few steps, done. POLAR EXPRESS is the sharp version, and it is correct *in fp32*. Our contribution is to show that, *in the idealized scalar model of the iteration*, the limiting quantity is not convergence speed but whether the iteration stays stable at all: our $\kappa = 100$ result shows the fp32-optimal schedule *diverging* (bf16 error $\to 4 \times 10^{18}$; P2 to genuine overflow) while the conservative fixed baseline sits calmly at 0.328. On this idealized band "optimal" is precision-relative. We are careful not to overclaim: on real gradient matrices the same aggressive coefficients are fragile in *both* precisions (E4), so the value here is a *map of the design space*, not a bf16 fix for deployed MUON.

Boundedness is necessary but not sufficient; the hybrid is the fix. Constraining each step's image (P2) is the right instinct but not enough: a fixed bound calibrated for a narrow interval is overrun once the band stretches to $[1/100, 1]$, and P2 overflows at $\kappa = 100$ with the *worst* floor of all (0.688). What works is to separate the schedule in time—aggressive (free) early when the interval is wide, safe (bounded) late when expansion is pure overflow risk. That is P3, the only method finite in bf16 at every idealized-band κ; this decoupling of early speed and late safety is the practical content of the design-space analysis.

The honest tradeoff. We do not claim P3 dominates everywhere. It buys bf16 stability by giving up fp32 accuracy (at $\kappa = 100$ its fp32 error 0.235 is far worse than P1's), and at $\kappa \leq 30$ it offers no benefit over P1. On the idealized band no single schedule is simultaneously fp32-optimal and bf16-safe across all conditioning, which suggests a conditioning-indexed table $(\kappa, T, \text{precision})$—but this recommendation is *scoped to the idealized band*. On *real* gradients it collapses: E4 shows the optimal coefficients fragile in *any* precision, so there the fixed baseline (bf16-safe end-to-end, E5) is already the safe choice and no bf16-specific hazard is opted into. The idealized-band hazard is a cautionary map of the coefficient space, not a live deployment risk in today's MUON.

Scope and limitations. (1) The bf16 model (Proposition 1) is a first-order prediction of the overflow/floor behavior, not a rigorous componentwise rounding bound; its value is that it is falsifiable and the $\kappa = 100$ measurement confirms it *on the idealized band*. (2) The E1–E3 curves live in the *idealized* scalar reduction (Lemma 1) over a uniform band with per-scalar bf16 rounding—exact in exact arithmetic, but *not* a faithful stand-in for the bf16 behavior of real gradients, and we say so plainly: the per-scalar rounding is harsher than fp32-accumulate tensor cores, and, the load-bearing point, our true-matrix control on real 124M gradients (E4, Appendix F) shows the aggressive P1 coefficients blow up in *both* bf16 ($8/8$) and fp32 ($7/8$), so the bf16-*specificity* does *not* transfer to real near-rank-one gradients. What is robust is the qualitative existence of an optimal-coefficient instability and the idealized-band stability ordering; what is *not* claimed is any bf16-specific real-gradient effect. Our one end-to-end measurement (E5) returns a null (-0.038 nats), so we show no end-to-end gain from the learned schedules and no bf16 defect in deployed MUON; P3 is an idealized-design-space remedy and an end-to-end P1/P3-swap study is future work. (3) We study a single target ($g = \mathrm{msign}$) and the single quintic degree MUON uses; the general-spectral-function extension is future work (Section 7).

7 CONCLUSION

We gave a precision-aware account of the Newton–Schulz sign iteration at the heart of MUON, treating the per-step coefficient schedule as the design variable (optimal–structured–hybrid family P1/P2/P3) *on the idealized scalar band*. There the per-step-optimal P1 beats the fixed quintic in fp32 by 7–10 orders, but *diverges* in bf16 at high conditioning ($\kappa = 100$, to 4×10^{18}; P2 to overflow), and the hybrid P3 is the resolution. We bound this honestly on two fronts: a true-matrix control on real 124M gradients shows the aggressive coefficients fragile in *both* precisions ($8/8$ bf16, $7/8$ fp32), so the bf16-*specificity* is a feature of the idealized band, not of real gradients; and a nanoGPT-124M premise test returns a null (-0.038 nats, bf16 vs. fp32) for the *deployed fixed* iteration. The result is a characterization of an *idealized coefficient design space*, not a claim that current MUON is broken in bf16. Extending it to general spectral functions is the broader program this paper opens.

Reproducibility Statement

The results of this paper are cheap to reproduce because the coefficient schedules are solved offline on CPU and, by the coordinate reduction (Lemma 1, proved in Appendix A), the matrix iteration is exactly a deterministic scalar iteration on the singular-value band. The schedule construction is fully specified in Section 4—the free minimax objective (4), the bounded minimax (5), and the hybrid rule (6)—and expanded in Appendix B, including the exact bf16 rounding simulation. The low-precision error/overflow model (Proposition 1) is derived in Appendix D, and the iteration is written out in Appendix H. Every number in Table 2 and the full curves in Table 3 (Appendix C) is produced by a single released CPU script over seed-fixed grids for $\kappa \in \{10, 30, 100\}$ and $T = 1 \dots 8$, so all fp32 and bf16 curves regenerate without a GPU. The real-gradient conditioning measurement (Appendix E) and the true-matrix control (E4; Appendix F), which runs the real quintic on logged 124M gradient matrices with fp32-accumulate matmuls in both precisions and finds the optimal coefficients fragile in *both* (bf16 $8/8$, fp32 $7/8$), are logged from a real MUON run; the matrices, per-matrix coefficients, and control harness are released. The end-to-end premise test (Section 5, E5; Appendix G) is a nanoGPT-124M run of 3000 steps over 2 seeds per condition (fixed MUON, fp32 vs. bf16 sign iteration); its training script, seeds, and per-seed losses are released. We report a null (bf16 vs. fp32 $= -0.038$ nats); we make no end-to-end claim for the P1/P3 schedules, which we scope as future work.

Ethics Statement

This work is a numerical-methods contribution to optimizer internals: it studies the coefficient schedule of the MUON sign iteration and its stability in low precision. It introduces no new data, no human subjects, and no generative capability, and we foresee no direct ethical concerns beyond the general efficiency implications of faster large-model training.

Use of Large Language Models

Large language models were used as writing and engineering assistance: drafting and copy-editing prose, refactoring LaTeX and analysis scaffolding, and literature triage — surfacing candidate related work, which the authors then verified against primary sources. Entries that could not be confirmed against a paper page, project page or proceedings entry were removed rather than retained on the strength of a model's summary. The research contribution — the precision-aware characterization of the coefficient design space, the optimal–structured–hybrid (P1/P2/P3) family, the first-order bf16 floor model, and the real-matrix and end-to-end controls — is the authors' own, and the authors take full responsibility for all content, including any errors. No results, data, or citations were generated by a language model.

REFERENCES

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the muon algorithm. *arXiv preprint arXiv:2505.16932*, 2025.

Rohan Anil, Vineet Gupta, Tomer Koren, Kevin Regan, and Yoram Singer. Scalable second order optimization for deep learning. *arXiv preprint arXiv:2002.09018*, 2020.

Adi Ben-Israel. A note on an iterative method for generalized inversion of matrices. *Mathematics of Computation*, 20(95):439–440, 1966.

Jeremy Bernstein and Laker Newhouse. Modular duality in deep learning. *arXiv preprint arXiv:2410.21265*, 2024a.

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

Jeremy Bernstein, Arash Vahdat, Yisong Yue, and Ming-Yu Liu. On the distance between two neural networks and the stability of learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Rajendra Bhatia. *Matrix Analysis*, volume 169 of *Graduate Texts in Mathematics*. Springer, 1997.

Dario A. Bini, Nicholas J. Higham, and Beatrice Meini. Algorithms for the matrix pth root. *Numerical Algorithms*, 39(4):349–378, 2005.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Ralph Byers. Solving the algebraic riccati equation with the matrix sign function. *Linear Algebra and its Applications*, 85:267–279, 1987.

Elliott W. Cheney. *Introduction to Approximation Theory*. McGraw-Hill, 1966.

James W. Demmel. *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics (SIAM), 1997.

Eugene D. Denman and Alex N. Beavers. The matrix sign function and computations in systems. *Applied Mathematics and Computation*, 2(1):63–94, 1976.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations (ICLR)*, 2022.

Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

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

Evan S. Gawlik. Zolotarev iterations for the matrix square root. *SIAM Journal on Matrix Analysis and Applications*, 40(2):696–719, 2019.

Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the SIAM: Series B, Numerical Analysis*, 2(2):205–224, 1965.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 4th edition, 2013.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning (ICML)*, 2015.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Nicholas J. Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986.

Nicholas J. Higham. Stable iterations for the matrix square root. *Numerical Algorithms*, 15(2): 227–242, 1997.

Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Society for Industrial and Applied Mathematics (SIAM), 2nd edition, 2002.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (SIAM), 2008.

Nicholas J. Higham and Theo Mary. Mixed precision algorithms in numerical linear algebra. *Acta Numerica*, 31:347–414, 2022.

Nicholas J. Higham and Robert S. Schreiber. Fast polar decomposition of an arbitrary matrix. *SIAM Journal on Scientific and Statistical Computing*, 11(4):648–655, 1990.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2nd edition, 2012.

Bruno Iannazzo. On the Newton method for the matrix pth root. *SIAM Journal on Matrix Analysis and Applications*, 28(2):503–523, 2006.

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

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, et al. A study of BFLOAT16 for deep learning training. *arXiv preprint arXiv:1905.12322*, 2019.

Andrej Karpathy. nanogpt. `https://github.com/karpathy/nanoGPT`, 2022.

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

Charles S. Kenney and Alan J. Laub. On scaling Newton's method for polar decomposition and the matrix sign function. *SIAM Journal on Matrix Analysis and Applications*, 13(3):688–706, 1992.

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

Zdislav V. Kovarik. Some iterative methods for improving orthonormality. *SIAM Journal on Numerical Analysis*, 7(3):386–389, 1970.

Tim Large, Yang Liu, Minyoung Huh, Hyojin Bahng, Phillip Isola, and Jeremy Bernstein. Scalable optimization in the modular norm. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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 (ICLR)*, 2019.

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

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. In *International Conference on Learning Representations (ICLR)*, 2018.

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

Yuji Nakatsukasa and Nicholas J. Higham. Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD. *SIAM Journal on Scientific Computing*, 35(3):A1325–A1349, 2013.

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

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Ricardo Pachón and Lloyd N. Trefethen. Barycentric-Remez algorithms for best polynomial approximation in the chebfun system. *BIT Numerical Mathematics*, 49(4):721–741, 2009.

Victor Pan and Robert Schreiber. An improved Newton iteration for the generalized inverse of a matrix, with applications. *SIAM Journal on Scientific and Statistical Computing*, 12(5):1109–1130, 1991.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *International Conference on Machine Learning (ICML)*, 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.

M. J. D. Powell. *Approximation Theory and Methods*. Cambridge University Press, 1981.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

Evgeny Ya. Remez. Sur la détermination des polynômes d'approximation de degré donnée. *Communications de la Société mathématique de Kharkov*, 10:41–63, 1934.

Theodore J. Rivlin. *An Introduction to the Approximation of Functions*. Blaisdell Publishing Company, 1969.

J. D. Roberts. Linear model reduction and solution of the algebraic riccati equation by use of the sign function. *International Journal of Control*, 32(4):677–687, 1980.

Günther Schulz. Iterative Berechnung der reziproken Matrix. *Zeitschrift für Angewandte Mathematik und Mechanik*, 13(1):57–59, 1933.

Torsten Söderström and G. W. Stewart. On the numerical properties of an iterative method for computing the Moore–Penrose generalized inverse. *SIAM Journal on Numerical Analysis*, 11(1): 61–74, 1974.

Jianlin Su. The brute-force aesthetics of matrix function approximation. Blog post (in Chinese). `https://spaces.ac.cn/archives/11787`, 2025. Blog post and comment thread; source of the residual and greedy-fitting routes.

Lloyd N. Trefethen. *Approximation Theory and Approximation Practice*. Society for Industrial and Applied Mathematics (SIAM), 2013.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. SOAP: Improving and stabilizing Shampoo using Adam. *arXiv preprint arXiv:2409.11321*, 2024.

Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Greg Yang and Edward J. Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning (ICML)*, 2021.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs V: Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Jack Zhang, Noah Amsel, Berlin Chen, and Tri Dao. Gram Newton-Schulz: A fast, hardware-aware Newton–Schulz algorithm for Muon. Dao AI Lab blog. `https://tridao.me/blog/2026/gram-newton-schulz/`, 2026.

A Coordinate Reduction (Proof of Lemma 1)

Let $X_t = UD_tV^\top$ with $D_t = \mathrm{diag}(x_1, \ldots, x_r)$ and $U^\top U = V^\top V = I_r$. Then $X_t^\top X_t = VD_t^2V^\top$, and for any polynomial p, $p(X_t^\top X_t) = V\,p(D_t^2)\,V^\top$ because V is orthonormal and powers telescope: $(VD_t^2V^\top)^k = VD_t^{2k}V^\top$. Hence

$$X_{t+1} = X_t\,p_t(X_t^\top X_t) = UD_tV^\top\,Vp_t(D_t^2)V^\top = U\left(D_t\,p_t(D_t^2)\right)V^\top,$$

so $D_{t+1} = D_t\,p_t(D_t^2)$ acts coordinate-wise, $x_i \mapsto x_i\,p_t(x_i^2) = a_tx_i + b_tx_i^3 + c_tx_i^5$. In particular, if $X_0 = M/\alpha$ then U, V are the singular vectors of M and remain fixed for all t, so the entire matrix iteration is the scalar iteration on the singular values that we evaluate in the experiments. □

B Coefficient Solving Details

Per-step minimax. Each step's coefficients solve the worst-case (Chebyshev) fit of Section 3. For the free schedule (P1) this is (4); for the bounded schedule (P2) it is (5), the same objective plus the representability constraint $\max_{y\in[0,B]} |ay + by^3 + cy^5| \le B$ with $B = 1.3$; P3 uses the free rule for the first $\lceil T/2 \rceil$ steps and the bounded rule thereafter. We solve each three-parameter problem by Nelder–Mead, warm-started from the fixed Muon triple $(3.4445, -4.7750, 2.0315)$, evaluating the inner maximum on a grid of 400 points over the current image interval (and, for the bounded case, 300 points over $[0, B]$ with the constraint added as a large penalty). The image interval $I_t = \varphi_t([1/\kappa, 1])$ is tracked numerically (min/max over a 600-point grid) after each step and supplies the domain for the next fit. Coefficients are precision-agnostic: the schedule is solved once in fp32 and then evaluated in both fp32 and bf16, exactly as a practitioner would ship a single table and run it in either precision. Because Nelder–Mead is a non-convex local solve on a finite grid, the resulting coefficients are *near-optimal numerical solutions*—close to but not certified equioscillating Remez minima; we therefore describe P1 as a numerically-solved per-step minimax rather than a provably optimal one.

bf16 simulation and its fidelity caveat. bf16 is simulated by rounding every intermediate quantity to nearest-even at 8-bit mantissa: each product x^2, x^3, x^5 and each scaled term ax, bx^3, cx^5 and their sum is truncated to bf16 before the next operation. A coordinate that grows without bound eventually exceeds the largest representable bf16 value and becomes ∞; we report this as $+\infty$ (genuine overflow) and never clip or clamp it, since the instability is the phenomenon under study. We are explicit that this per-scalar rounding is a *harsher* arithmetic than real hardware: production tensor cores accumulate the matrix products $X^\top X$ and the quintic in fp32 and round only on write-back, which is more forgiving than rounding every scalar operation to bf16. Our scalar simulation may therefore *overstate* the absolute instability, and—more importantly—its clean bf16-*specificity* is itself an artifact of the idealization: the true-matrix control of Appendix F, run with fp32-accumulate matmuls on real gradients, shows the aggressive coefficients blowing up in *both* precisions ($8/8$ bf16, $7/8$ fp32). What *is* robust is only the qualitative existence of an optimal-coefficient instability at high conditioning and the design-space stability ordering on the idealized band (fixed and P3 stable, P1/P2 not); the bf16-specific reading holds *only* on the idealized band and does not transfer to real gradient matrices.

C Full Error-versus-Step Curves

Table 3 gives the complete $T = 1 \ldots 8$ curves summarized at $T = 8$ in Table 2, for every $(\kappa, \text{method}, \text{precision})$. These are the transparent trajectories behind the headline numbers: the fixed Muon baseline saturates by $T \approx 3$ (at ≈ 0.318 fp32 / ≈ 0.328 bf16 for all κ); P1 improves monotonically in fp32 but, at $\kappa = 100$, its bf16 curve turns upward after $T = 4$ and diverges ($3.75 \to 4543 \to 4.07 \times 10^{18}$); P2's bf16 curve at $\kappa = 100$ diverges even faster ($1.47 \to 75 \to 3.5 \times 10^9 \to \infty$); and P3's bf16 curve is the only one at $\kappa = 100$ that stays finite throughout, bottoming out at 1.41×10^{-1} (its floor) before a mild rise to 2.81×10^{-1} at $T = 8$.

D Low-Precision Error Model (Expanded)

Model the executed step as $x_{t+1} = h_t(x_t) + \varepsilon_t$ with $|\varepsilon_t| \le c\,u$, $u \approx 2^{-8}$ for bf16, and $h_t(x) = a_tx + b_tx^3 + c_tx^5$. Near the target $x_\star = 1$ the linearization $e_{t+1} = \rho_te_t + \varepsilon_t$ with $\rho_t = h_t'(1)$

κ	method / prec	Terminal error $\max_{x\in[1/\kappa,1]} \lvert\varphi_T(x)-1\rvert$ at step T							
		T=1	2	3	4	5	6	7	8
10	fixed / fp32	6.60×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}
	fixed / bf16	6.58×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}
	P1 / fp32	4.55×10^{-1}	7.65×10^{-2}	2.34×10^{-2}	2.44×10^{-3}	1.08×10^{-4}	4.80×10^{-6}	2.10×10^{-7}	1.00×10^{-8}
	P1 / bf16	5.00×10^{-1}	1.25×10^{-1}	4.69×10^{-2}	3.12×10^{-2}	1.56×10^{-2}	1.56×10^{-2}	1.56×10^{-2}	1.56×10^{-2}
	P2 / fp32	6.07×10^{-1}	1.07×10^{-1}	3.14×10^{-3}	1.61×10^{-1}	7.45×10^{-3}	6.07×10^{-3}	6.54×10^{-3}	7.05×10^{-3}
	P2 / bf16	6.09×10^{-1}	1.41×10^{-1}	1.17×10^{-2}	1.64×10^{-1}	2.73×10^{-2}	3.91×10^{-2}	3.12×10^{-2}	5.47×10^{-2}
	P3 / fp32	4.55×10^{-1}	7.65×10^{-2}	2.34×10^{-2}	2.44×10^{-3}	1.60×10^{-1}	5.88×10^{-3}	6.34×10^{-3}	6.83×10^{-3}
	P3 / bf16	5.00×10^{-1}	1.25×10^{-1}	4.69×10^{-2}	3.12×10^{-2}	1.95×10^{-1}	7.42×10^{-2}	9.38×10^{-2}	7.81×10^{-2}
30	fixed / fp32	8.85×10^{-1}	6.12×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}
	fixed / bf16	8.85×10^{-1}	6.11×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}
	P1 / fp32	8.32×10^{-1}	5.15×10^{-1}	9.40×10^{-2}	3.06×10^{-2}	3.95×10^{-3}	1.77×10^{-4}	7.83×10^{-6}	3.45×10^{-7}
	P1 / bf16	8.44×10^{-1}	5.62×10^{-1}	1.56×10^{-1}	9.38×10^{-2}	7.81×10^{-2}	4.69×10^{-2}	1.56×10^{-2}	1.56×10^{-2}
	P2 / fp32	8.66×10^{-1}	5.00×10^{-1}	5.87×10^{-2}	1.56×10^{-1}	1.96×10^{-4}	6.92×10^{-2}	2.46×10^{-4}	6.95×10^{-2}
	P2 / bf16	8.65×10^{-1}	5.31×10^{-1}	2.19×10^{-1}	1.72×10^{-1}	7.81×10^{-3}	7.03×10^{-2}	2.34×10^{-2}	7.03×10^{-2}
	P3 / fp32	8.32×10^{-1}	5.15×10^{-1}	9.40×10^{-2}	3.06×10^{-2}	2.52×10^{-2}	3.69×10^{-2}	2.04×10^{-2}	1.65×10^{-2}
	P3 / bf16	8.44×10^{-1}	5.62×10^{-1}	1.56×10^{-1}	9.38×10^{-2}	7.81×10^{-2}	7.81×10^{-2}	6.25×10^{-2}	5.47×10^{-2}
100	fixed / fp32	9.66×10^{-1}	8.82×10^{-1}	6.00×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}	3.18×10^{-1}
	fixed / bf16	9.66×10^{-1}	8.81×10^{-1}	6.00×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}	3.28×10^{-1}
	P1 / fp32	9.44×10^{-1}	8.41×10^{-1}	5.02×10^{-1}	8.66×10^{-2}	3.03×10^{-2}	3.89×10^{-3}	1.73×10^{-4}	7.77×10^{-6}
	P1 / bf16	9.53×10^{-1}	9.06×10^{-1}	7.03×10^{-1}	3.87×10^{-1}	4.69×10^{-1}	3.75×10^{0}	4.54×10^{3}	4.07×10^{18}
	P2 / fp32	9.58×10^{-1}	8.27×10^{-1}	3.82×10^{-1}	2.94×10^{-2}	2.43×10^{-2}	3.63×10^{-2}	1.97×10^{-2}	1.59×10^{-2}
	P2 / bf16	9.69×10^{-1}	8.75×10^{-1}	6.88×10^{-1}	1.47×10^{0}	7.50×10^{1}	3.52×10^{9}	$+\infty$	$+\infty$
	P3 / fp32	9.44×10^{-1}	8.41×10^{-1}	5.02×10^{-1}	8.66×10^{-2}	6.56×10^{-2}	5.61×10^{-2}	1.09×10^{-1}	2.35×10^{-1}
	P3 / bf16	9.53×10^{-1}	9.06×10^{-1}	7.03×10^{-1}	3.87×10^{-1}	2.58×10^{-1}	1.56×10^{-1}	1.41×10^{-1}	2.81×10^{-1}

Table 3: Full error-versus-step curves ($T = 1 \dots 8$) for every (κ, method, precision). $+\infty$ denotes bf16 overflow. The $T = 8$ columns and the per-row bf16 minima reproduce Table 2.

governs the mantissa-limited regime: when the schedule is contractive ($|\rho_t| \le \bar\rho < 1$) and bounded, the steady state obeys $|e_\infty| \le c\,u/(1-\bar\rho)$—a floor of order u independent of T, re-attracted after any transient, matching the flat bf16 tails of the bounded schedules at $\kappa \le 30$ in Table 3.

The distinct low-precision failure is *range*, not rounding. Let $\mu_t = \max_{x\in I_{t-1}} |h_t(x)|$ be the per-step image magnitude. The free minimax (4) places no bound on μ_t; to collapse a wide interval $[1/\kappa, 1]$ onto 1 as fast as possible it uses large coefficients whose step map *overshoots* before settling, so μ_t grows with the interval width $1 - 1/\kappa$. On a wide-enough band the running product of overshoots makes the iterate blow up: it first *diverges*—growing by orders of magnitude each step while still finite (P1 at $\kappa = 100$ reaches 4.07×10^{18} by $T = 8$, some 20 orders below bf16's maximum $\approx 3.4 \times 10^{38}$; it would overflow given more steps)—and, in the more aggressive case, actually reaches that maximum and rounds to ∞, from which no later step recovers (P2 at $\kappa = 100$ hits $+\infty$ by $T = 7$). This is why the P1 bf16 curve at $\kappa = 100$ (Table 3) turns upward after $T = 4$ and diverges, why the fixed MUON quintic—whose conservative single step keeps $\mu_t \le 1$—never destabilizes, and why capping the late steps (P3) removes the divergence while keeping the aggressive early steps. The model is first-order; a rigorous componentwise analysis of the divergence/overflow onset is future work.

E REAL-GRADIENT CONDITIONING (SUPPORTING E4)

To locate where the idealized-band hazard *could* matter, we first ask whether the high-κ regime that destabilizes the free schedule on the idealized band is a synthetic worst case or something real MUON encounters. We logged the singular spectra of the 2D hidden gradient/momentum matrices that MUON actually orthogonalizes during a real training run of a 124M-parameter GPT, over 40 training steps (192 matrices of shapes 2304×768, 768×768, 768×3072, 3072×768). Table 4 reports the distribution. Two facts stand out. First, the *full* condition number $\kappa = \sigma_{\max}/\sigma_{\min}$ is astronomically large (median $\approx 5.6 \times 10^7$) simply because real gradients are near-rank-deficient: the median stable rank $\|M\|_F^2/\|M\|_2^2$ is only ≈ 1.07, i.e. the energy is concentrated in essentially one direction, with a long near-null tail of tiny singular values. The full κ is therefore *not* the quantity that governs the iteration. Second, the meaningful conditioning is the *effective* condition number $\kappa_{\text{eff}} = \sigma_{\max}/\sigma_{@99\%}$ on the energy-bearing subspace (the singular values up to 99% of cumulative spectral energy): here we measure median $\kappa_{\text{eff}} = 9.0$ and 90th percentile $\kappa_{\text{eff}} = 108.4$. So real gradients *do* reach the $\kappa \sim 100$ regime, for an ill-conditioned $\sim 10\%$ tail.

Quantity (over 192 real matrices)	median	p90	max
Full condition number $\kappa = \sigma_{\max}/\sigma_{\min}$	5.64×10^7	1.56×10^{10}	1.22×10^{11}
Effective cond. $\kappa_{\text{eff}} = \sigma_{\max}/\sigma_{@99\%}$	9.02	108.4	—
Stable rank $\|M\|_F^2/\|M\|_2^2$	1.07	—	—

Table 4: Conditioning of real MUON gradient/momentum matrices (124M GPT, 40 steps). The full κ is huge only because of a near-null tail (stable rank ≈ 1); the energy-bearing κ_{eff} has median 9 and p90 108, so real gradients *do* reach the $\kappa \sim 100$ regime for an ill-conditioned tail. **Whether the idealized bf16-specific divergence actually occurs there is answered—negatively—by the true-matrix control of Appendix F, not by a scalar replay.**

Why we do *not* report a scalar replay here. It is tempting to plug these measured κ_{eff} values back into the idealized scalar model and report that P1 "diverges in bf16 at $\kappa_{\text{eff}} \approx 108$": the scalar model does predict exactly that. But such a replay would inherit both idealizations (uniform band, per-scalar rounding) and would therefore *overstate* bf16-specificity—as Appendix F shows, when the *true matrix* iteration is run on the actual gradient matrices, the aggressive coefficients blow up in *both* precisions ($8/8$ bf16, $7/8$ fp32), so the divergence is not bf16-specific on real gradients. We therefore report the conditioning here and the honest, non-idealized outcome in Appendix F, and we do *not* claim the ill-conditioned tail is a bf16-specific real-gradient hazard.

One further honest note: the fixed-MUON bf16 orthogonalization residual measured on the *full* spectrum is ≈ 1.0, but this is dominated by the near-null tail singular values that no finite msign iteration can lift to 1; it should not be over-interpreted as a failure of MUON, which orthogonalizes the dominant energy-bearing subspace, not the near-null tail. All spectra and the logging script are released with the code.

F TRUE-MATRIX CONTROL ON REAL GRADIENTS (E4)

The E1–E3 curves live in the idealized scalar reduction (uniform band, per-scalar bf16 rounding). E4 removes *both* idealizations: we take real gradient matrices logged from the 124M GPT run (Appendix E), select the ill-conditioned tail (8 matrices of shapes 3072×768 and 768×3072 with $\kappa_{\text{eff}} \approx 65$–$109$), solve the free P1 coefficients (4) *per matrix* in fp32, and then run the *true* quintic matrix iteration (2) on each matrix in torch, using fp32-accumulate tensor-core matmuls, in both fp32 and bf16. We compare against the fixed MUON quintic. The metric is the orthogonalization residual $\|X_T^\top X_T - I_r\|/\sqrt{r}$ on the energy-bearing subspace at $T = 8$ ("blow-up" = residual $\gg 1$ or non-finite).

Iteration (true matrix, fp32-accumulate)	fp32 residual	bf16 residual
P1 free (optimal, solved per matrix)	**blows up on** $7/8$	**blows up on** $8/8$
median residual over the 8 matrices	$\approx 9.6 \times 10^1$	$\approx 4.6 \times 10^{19}$
# matrices clean (≈ 0.3)	$1/8$	$0/8$
fixed MUON (conservative)	≈ 0.32 (stable)	≈ 0.32 (stable)

Table 5: **True-matrix control (E4).** Running the *real* matrix quintic on 8 ill-conditioned real 124M gradient matrices, with fp32-accumulate tensor cores, in both precisions. The per-step-optimal P1 coefficients blow up on $8/8$ matrices in bf16 *and* $7/8$ in fp32—the clean "bf16-only" divergence of E2 occurs for just $1/8$—so on real near-rank-one gradients the aggressive coefficients are fragile in *any* precision, and the bf16-*specificity* seen on the idealized band (Table 2) does *not* transfer. Fixed MUON is stable in both precisions (residual ≈ 0.32). This is the paper's central honesty bound.

The reading is unambiguous. On the idealized band (E2) the failure was cleanly bf16-specific: fp32-optimal P1 was accurate in fp32 and diverged only in bf16. On *real* gradient matrices that clean separation disappears: P1 blows up in fp32 too ($7/8$), and the single matrix where fp32 survives while bf16 diverges is the lone $1/8$ case. The cause is structural—real gradients are near rank one (stable rank ≈ 1.07), so the "interval" the free minimax is fitting is really a spike plus a near-null tail, and the aggressive coefficients that the per-step minimax chooses to collapse a *uniform* band overshoot catastrophically on this spectrum regardless of precision. Two consequences we state plainly. (i)

The bf16-specific divergence is a property of the idealized band, not of real gradients; nothing in this paper should be read as claiming that adopting the optimal coefficients would break real MUON training in bf16 specifically. (ii) The conservative fixed iteration MUON deploys is stable on these real matrices in both precisions, consistent with the E5 end-to-end null (Appendix G). The per-matrix solved coefficients, the real gradient matrices, and the control harness are released with the code.

G END-TO-END PREMISE TEST (E5): FIXED-MUON PRECISION SWAP

E5 tests the load-bearing premise of the paper: does the precision of the sign iteration matter for the iteration MUON *actually deploys*? We train nanoGPT-124M (Karpathy, 2022) (the standard 12-layer, 768-width GPT-2 configuration) on OpenWebText for 3000 steps with the MUON optimizer, holding *everything* fixed—architecture, data order, learning-rate schedule, step budget, and the deployed fixed quintic coefficients $(3.4445, -4.7750, 2.0315)$—and changing *only* the arithmetic precision of the Newton–Schulz sign iteration inside the optimizer: condition (A) runs the msign iteration in fp32, condition (B) runs the identical iteration in bf16. Each condition is repeated over 2 seeds. Table 6 reports the terminal validation loss and training throughput.

	Sign-iteration precision	Val. loss (nats, mean $\pm$ std)	Throughput (tokens/s)
A	fixed MUON, fp32 iter	5.086 ± 0.015	55,210
B	fixed MUON, bf16 iter	5.048 ± 0.024	67,995
	Difference $B - A$	-0.038 (bf16 marginally lower)	$+23\%$

Table 6: End-to-end premise test (E5): nanoGPT-124M, 3000 steps, 2 seeds per condition, swapping *only* the precision of the *deployed fixed* MUON sign iteration. Per-seed validation losses: A $= \{5.101, 5.070\}$, B $= \{5.072, 5.024\}$. The bf16$-$fp32 gap is -0.038 nats—bf16 is if anything marginally *better*—and is well within $\sim$two standard deviations of either condition, i.e. statistically indistinguishable, while bf16 trains $\sim 23\%$ faster in wall-clock throughput. **Verdict: a null.** With the fixed coefficients MUON actually ships, running the sign iteration in bf16 rather than fp32 does not degrade end-to-end training loss (and is faster). This is one of the two honest bounds on the paper (the other being the true-matrix control, E4): the bf16 instability we characterize (Section 5, E2, *on the idealized band*) is a property of the *aggressive per-step-optimal* schedule (P1) at high conditioning, not of the status-quo iteration, which is already bf16-safe end-to-end.

The result is deliberately a bound rather than a promotion of our method. It says the precision-aware analysis is a warning about a *design choice* (adopting the fp32-optimal aggressive coefficients) and a characterization of when that choice is safe, not a claim that current MUON is mis-configured. Because the deployed iteration is already bf16-safe, we do *not* run the P3-swap end-to-end and do not claim an end-to-end training improvement from it: P3's value is as the precision-robust member of the design space one would reach for *if* chasing the fp32-optimal accuracy in low precision, and quantifying any end-to-end gain from that route is future work. The training script, seeds, and logged losses are released with the code.

H ALGORITHM

Algorithm 1 Precision-aware MUON sign step

Input: matrix M, step budget T, precision π, schedule $\in \{\text{fixed, P1, P2, P3}\}$, coefficient table $\{(a_t, b_t, c_t)\}_{t=1}^{T}$ (solved offline per (κ, T))
scale $X_0 \leftarrow M/\alpha$ with $\alpha \geq \sigma_{\max}$ so its spectrum lies in $[1/\kappa, 1]$
for $t = 1, \ldots, T$ **do**
    $G \leftarrow X_{t-1}^{\top} X_{t-1}$ ▷ Gram matrix
    $X_t \leftarrow a_t X_{t-1} + b_t X_{t-1} G + c_t X_{t-1} G^2$
    all products in precision π (e.g. bf16)
end for
return $X_T \approx \mathrm{msign}(M)$

The only difference between the schedules is which coefficient table is loaded; the per-step matmul cost is identical (Section 4.3). For bf16-deployed, potentially ill-conditioned workloads the recommended table is P3; for fp32 or well-conditioned workloads it is P1; fixed MUON is the overflow-proof fallback. Coefficient schedules, the scaling rule for α, seeds, and the evaluation harness are released with the code.