LIONMUON: ALTERNATING SPECTRAL AND SIGN DESCENT FOR EFFICIENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Pretraining a language model takes enormous compute, and the right optimizer can save a good part of it. `Muon`'s spectral step gives a stronger direction than a sign step, but it is expensive. Every step runs Newton–Schulz iterations on the full matrix and, in distributed training, an extra all-reduce. Sign steps, as in `Lion` and `Signum`, are cheap and stay local to each device. We propose `LionMuon`, which takes one `Muon` step every P iterations and `Lion` steps in between, with a single dual-EMA momentum buffer shared by both. `Muon`'s compute and communication are paid once per P steps, and the optimizer state is half of `AdamW`'s. A single-EMA variant, `SignMuon`, already improves on `Muon`. We prove complexity bounds under heavy-tailed noise in which the period sets an interpolation between `Muon`'s and `Lion`'s smoothness and noise constants, and which say when `LionMuon` is faster than both. On 124M and 355M models trained on FineWeb, `LionMuon` with P=2 and P=5 reaches a lower loss than `Muon`, `AdamW`, `Lion` and `Signum` at the same number of tokens. Under 4-GPU data-parallel training it reaches `Muon`'s final loss with a third less wall-clock on PCIe, and it beats the communication-efficient `Muon` variants `Dion` and `MuonBP` on loss at no more exposed communication, while keeping the exact gradient.
Code: `https://anonymous.4open.science/r/lionmuon/`.

1 INTRODUCTION

Training Large Language Models (LLMs) is a billion-parameter, million-step optimization problem in which per-step cost determines the final compute bill (Hoffmann et al., 2022; Team et al., 2025). Finding update rules that are both FLOP-cheap per step and fast to converge is therefore a central question for modern deep learning (Dahl et al., 2023; Kasimbeg et al., 2025).

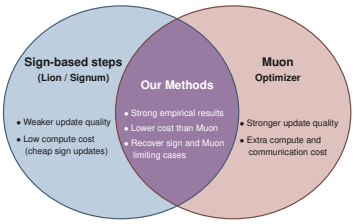

Figure 1: Sign steps are cheap, `Muon` steps are strong but expensive, and our methods alternate between the two.

A useful way to organize this design space is the Linear Minimization Oracle (LMO) viewpoint, which originates from Frank-Wolfe optimization (Jaggi, 2013). Recent work reinterprets a wide family of first-order optimizers as norm-constrained linear oracles (Chen et al., 2024; Veprikov et al., 2025). In particular, the parameter update at step t takes the form:

$$W_{t+1} \;=\; W_t + \eta_t \,\mathrm{LMO}_{\|\cdot\|}(\hat{G}_t), \qquad \mathrm{LMO}_{\|\cdot\|}(G) := \arg\min_{\|S\|\le 1} \langle G, S\rangle, \tag{1}$$

where $\hat{G}_t$ is a momentum-smoothed gradient, $\|\cdot\|$ is a chosen norm and $\eta_t > 0$ is the learning rate, usually with warm-up and decay (Goyal et al., 2017; Loshchilov & Hutter, 2017; Riabinin et al., 2026). The choice of norm picks the optimizer: Frobenius norm $\|\cdot\|_F$ gives `normalized SGD` (Hazan et al., 2015); $\|\cdot\|_\infty$ gives `signSGD` with its momentum variants, `Signum` (Bernstein et al., 2018) and `Lion` (Chen et al., 2023); and the spectral norm $\|\cdot\|_2$ gives `Muon` (Jordan et al., 2024). These methods now drive production LLM training, with `Muon` and its variants powering Moonlight (Liu et al., 2025), Kimi K2 (Team et al., 2025), and DeepSeek V4 (DeepSeek-AI, 2026).

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.

Within this family, sign-based methods sit at the cheap end: `Signum` updates with the sign of a single momentum buffer, while `Lion` uses two EMA timescales but keeps the same coordinate-wise sign step. `Muon` sits at the opposite end, computing the matrix sign $\mathrm{msign}(X) = UV^\top$ via Newton–Schulz iterations (Bernstein & Newhouse, 2024). The resulting spectral direction is often much stronger than a coordinate-wise sign step (Chen et al., 2026a), but it is also much more expensive: each `Muon` step runs several Newton–Schulz iterations of matrix multiplications and, in distributed training, needs extra communication (Essential AI, 2025; Chen et al., 2026b).

Distributed training infrastructure is built around element-wise optimizers. Every device holds a copy or a shard of each matrix and updates it with a local rule, so the optimizer step never needs the whole matrix and never communicates. `Muon` breaks this. Its update couples the entire matrix, so a sharded matrix has to be gathered before Newton–Schulz (Khaled et al., 2026; Essential AI, 2025), and under data parallelism the Newton–Schulz work is either repeated on every device or dealt out across them and exchanged afterwards (Liu et al., 2025; Chen et al., 2026b). Either way the optimizer adds a synchronization that cannot hide behind the backward pass, on top of the Newton–Schulz iterations themselves. The reported price ranges from a 5 to 10% throughput loss under tensor parallelism (Khaled et al., 2026) to more than twice the cost of forward and backward in a naive implementation (Chen et al., 2026b). Careful implementations shrink this price but pay it on every step. A sign step needs none of it: it is element-wise, exactly what the infrastructure assumes (Figure 1). This raises a natural question:

> *Can we keep the update quality of* `Muon` *while paying its compute and communication cost only once every P steps?*

We answer this question positively. Starting from `Signum`, inserting one `Muon` step every P iterations gives `SignMuon`. Replacing its momentum with `Lion`'s dual-EMA rule then yields `LionMuon`, the main method we study. The contributions below separate these two steps.

Contributions.

- `SignMuon` and `LionMuon` (Section 3): one `Muon` step every P iterations and sign steps in between, with `Signum`'s single EMA or `Lion`'s dual EMA. Both add one integer P to `Lion`, and `Muon`, `Lion` and `Signum` are special cases. The Newton–Schulz compute and the communication of `Muon` are paid once per P steps, the sign steps are local, and the state is one buffer (Appendix D.1).
- Complexity bounds under heavy-tailed noise, with and without weight decay (Section 4.2, Appendix B). The period sets an interpolation between `Muon`'s and `Lion`'s smoothness and noise constants, and the bound says when `LionMuon` is faster than both (Section 4.3).
- Experiments at 124M on FineWeb (Penedo et al., 2024) and WikiText-103 (Merity et al., 2017), tuned per method with three seeds, and at 355M on FineWeb with transferred hyperparameters (Section 5). `LionMuon` with $P \in \{2, 5\}$ beats `Muon`, `AdamW`, `Lion` and `Signum` at both sizes. Under 4-GPU data-parallel training we count the bytes each step sends, time the step on PCIe and NVLink, and compare with `Dion` (Ahn et al., 2025) and `MuonBP` (Khaled et al., 2026).

2 Related work

Sign-based methods. Sign-based methods first appeared as a communication-efficient solution for distributed optimization (Bernstein et al., 2019). The element-wise sign update $W_{t+1} = W_t - \eta_t \,\mathrm{sign}(\hat{G}_t)$ is cheap to compute, to parallelize and to transmit. Sign methods are also valued in LLM training for their memory efficiency, for zeroth-order fine-tuning (Petrov et al., 2025), and for their robustness to heavy noise (Kornilov et al., 2025; Yu et al., 2026) and to complex models (Crawshaw et al., 2022). `Signum` (Bernstein et al., 2018) is the simplest first-moment-only sign optimizer, and `Lion` (Chen et al., 2023) extends it with a separate interpolation EMA before the sign step. Our path from `SignMuon` to `LionMuon` mirrors this progression.

Spectral methods. `Muon` (Jordan et al., 2024) moves from element-wise sign to its matrix analogue computed by Newton–Schulz (NS) iterations: $W_{t+1} = W_t - \eta_t \,\mathrm{NS}_K(\hat{G}_t)$. This update yields a

much stronger spectral direction, but each step is also significantly more expensive. The MuonClip variant powers Kimi K2 (Team et al., 2025). Other works refine Muon itself, e.g., Gluon (Riabinin et al., 2025) and HTMuon (Pang et al., 2026). A second line makes it cheaper to run at scale. Dion (Ahn et al., 2025) replaces Newton–Schulz by a power iteration that keeps a low-rank factorization of the momentum with error feedback, and synchronizes the two factors instead of the gradient, so what is sent grows with the rank rather than with the matrix. MuonBP (Khaled et al., 2026) orthogonalizes each tensor-parallel shard on its own device and gathers the full matrix only every P-th step, which removes the all-gather from most steps at the price of a block-wise update in between. Layer sharding (Essential AI, 2025) keeps whole matrices on one device so that Newton–Schulz needs no gather. DMuon (Chen et al., 2026b) keeps the update exactly and removes the redundant work of naive distributed implementations: each matrix is assigned to one owner rank and Newton–Schulz runs in its Gram form, which brings Muon's step time to within a few percent of AdamW's on 8 to 256 GPUs.

Two concurrent methods target Muon's per-step cost directly. LiMuon (Huang et al., 2025) replaces Newton–Schulz with a low-rank randomized SVD of the momentum, and OLion (Wang et al., 2026) composes orthogonalization with an element-wise sign inside every step. We work along a different axis, the iteration axis. The spectral oracle stays a black box and is called only every P iterations, so either of them, or a faster polynomial iteration for the matrix sign (Bernstein & Newhouse, 2024; Amsel et al., 2025; Grishina et al., 2025), can be dropped in and the savings compound. The same holds for implementations such as DMuon: they make each Muon step cheaper, we make them rarer.

Lion-$\mathcal{K}$ framework. Chen et al. (2024) view Lion and signSGD as solving a constrained problem with $\mathcal{K} = \|\cdot\|_1$ through a Lyapunov analysis, and the same machinery covers Muon with $\mathcal{K} = \|\cdot\|_{\text{nuc}}$ (Chen et al., 2026a). The stochastic Frank-Wolfe view (Sfyraki & Wang, 2025) recovers the rates of both families in one language. Our analysis builds on this common basis.

Optimizer switching. Combining different optimizers within a single run is an established idea: SWATS (Keskar & Socher, 2017) switches from Adam to SGD, AdaBound (Luo et al., 2019) interpolates between them through learning-rate clipping, and AGD (Yue et al., 2023) gates between the two adaptively. To our knowledge, no prior work studies the periodic switching between Muon and Lion-style steps that we propose here.

3 Algorithm

3.1 Notation

We work in the matrix parameter space $\mathbb{R}^{m\times n}$ and denote the parameter matrix at iteration t by $W_t \in \mathbb{R}^{m\times n}$ and its stochastic gradient by $G_t \in \mathbb{R}^{m\times n}$. This space is equipped with the Frobenius inner product $\langle X, Y\rangle := \operatorname{tr}(X^\top Y)$, $\|X\|_F^2 = \langle X, X\rangle$ and with the following matrix norms:

$$\|X\|_2 := \sigma_1, \quad \|X\|_\infty := \max_{ij} |X_{ij}|, \quad \|X\|_{\text{nuc}} := \sum_k \sigma_k, \quad \|X\|_1 := \sum_{ij} |X_{ij}|,$$

where $\sigma_1 \geq \cdots \geq \sigma_{\min(m,n)}$ are the sorted singular values of matrix $X \in \mathbb{R}^{m\times n}$. The dual norm $\|X\|_\star := \sup_{\|S\|\leq 1}\langle X, S\rangle$ gives dual pairs $\|\cdot\|_{2,\star} = \|\cdot\|_{\text{nuc}}$ and $\|\cdot\|_{\infty,\star} = \|\cdot\|_1$. For all matrices $X \in \mathbb{R}^{m\times n}$, the considered norms satisfy the following inequalities:

$$\|X\|_\infty \leq \|X\|_2 \leq \|X\|_F \leq \sqrt{mn}\,\|X\|_\infty \quad \text{and} \quad \tfrac{1}{\sqrt{mn}}\|X\|_1 \leq \|X\|_F \leq \|X\|_{\text{nuc}} \leq \|X\|_1. \quad (2)$$

We use the spectral norm LMO to calculate the matrix-sign operation $\text{LMO}_{\|\cdot\|_2}(G) = -\text{msign}(G)$ and the infinity norm LMO to calculate the element-wise sign $\text{LMO}_{\|\cdot\|_\infty}(G) = -\text{sign}(G)$.

3.2 LionMuon

Algorithm 1 keeps a single momentum buffer M_t, updated at every step. Each iteration forms the direction $\hat{G}_t$ as Lion does, by interpolating between M_{t-1} and the current gradient G_t. Every P-th iteration takes a Muon step, which orthogonalizes $\hat{G}_t$ with Newton–Schulz, and every other iteration takes an element-wise sign step. Both use decoupled weight decay λ.

Algorithm 1 `LionMuon` and `SignMuon` for a single 2D parameter $W \in \mathbb{R}^{m\times n}$

Require: Horizon T, period $P \in \{1, 2, \ldots\} \cup \{\infty\}$ ($P{=}\infty$ means the `Muon` branch is never taken), learning rates η_M (`Muon`) and η_L (`Lion`), betas $\beta_1, \beta_2 \in [0, 1)$, weight decay $\lambda \geq 0$, NS steps K_{NS}, initial parameters W_0 and momentum $M_{-1} = 0$, and $c(A) := 0.2\sqrt{\max(\mathrm{rows}(A), \mathrm{cols}(A))}$.
for $t = 0, 1, \ldots, T-1$ **do**
  $G_t = \nabla_W \mathcal{L}_t$ ▷ Stochastic gradient
  $\hat{G}_t = \beta_1 M_{t-1} + (1-\beta_1) G_t$ ▷ `Lion` interpolation (direction)
  if $t \bmod P = 0$ **then**
    $W_{t+1} = W_t - \eta_M \left(c(\hat{G}_t)\, \mathrm{NS}_{K_{\mathrm{NS}}}(\hat{G}_t) + \lambda W_t \right)$ ▷ `Muon` step
  else
    $W_{t+1} = W_t - \eta_L \left(\mathrm{sign}(\hat{G}_t) + \lambda W_t \right)$ ▷ `Lion` step
  end if
  $M_t = \beta_2 M_{t-1} + (1-\beta_2) G_t$ ▷ Momentum update (every step)
end for

Implementation notes. `LionMuon` keeps one buffer M_t per matrix, since $\hat{G}_t$ is computed in place, so its state matches `Lion` and `Muon` and is half of `AdamW`'s. All 2D matrices of a transformer, embeddings included, take the `LionMuon` update. The 1D parameters (biases and norm gains) use `AdamW` at a fixed 10^{-3}, the usual `Muon` convention (Jordan et al., 2024). Giving them the `Lion` step instead is clearly worse (Appendix E.2). Table 3 lists the special cases.

4 Convergence analysis

This section analyzes `LionMuon` (Algorithm 1): the assumptions (Section 4.1), the convergence bound (Section 4.2), and what it says about the ratio of the two learning rates and the period P (Section 4.3). The main text treats the case without weight decay ($\lambda = 0$). Appendix B covers weight decay, with more technical work and the same conclusions.

4.1 Assumptions

We use standard assumptions on the objective and on the noise.

Assumption 1 (Smoothness and lower boundness)**.** *The objective function $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ is lower bounded by $f_\star$ and L-smooth with respect to a primal norm $\|\cdot\|$:*

$$\|\nabla f(W) - \nabla f(W')\|_\star \leq L \|W - W'\|, \quad \text{for all } W, W' \in \mathbb{R}^{m\times n}.$$

We use smoothness constants L_2 and L_∞ for norms $\|\cdot\|_2$ and $\|\cdot\|_\infty$, respectively.

From the norm inequalities (2), we can bound the smoothness ratio $1 \leq L_\infty / L_2 \leq mn$. Following Sadiev et al. (2023), Hübler et al. (2025) and Chezhegov et al. (2026), we allow heavy-tailed noise, which is what LLM training shows (Gürbüzbalaban et al., 2021).

Assumption 2 (Bounded κ-th moment)**.** *Stochastic gradients G_t are unbiased estimates of the true gradient $\nabla f(W_t)$, and have bounded κ-th moment for some $\kappa \in (1, 2]$ and $\sigma \geq 0$:*

$$\mathbb{E}[G_t] = \nabla f(W_t), \qquad \mathbb{E}\big[\|G_t - \nabla f(W_t)\|_F^\kappa\big] \leq \sigma^\kappa.$$

We measure this exponent rather than assume it. A Hill estimator on the norms of the gradient noise, over checkpoints of a 124M `LionMuon` $P{=}2$ run on WikiText-103, gives tail indices with medians of 46 at batch 32, 22 at batch 8 and 12 at batch 2, and the smallest value we saw was 10.6. All of them are above 2, so Assumption 2 holds with $\kappa = 2$, and the tails thicken as the batch shrinks, which is the direction the bound predicts. The noise level also depends on the dual norm. This norm equivalence in expectation follows Kornilov et al. (2023) and Hübler et al. (2025).

Assumption 3 (Noise norm equivalence)**.** *For any linear combination $\sum_\tau a_\tau \epsilon_\tau$ of independent gradient noise terms $\epsilon_\tau := G_\tau - \nabla f(W_\tau)$, we have:*

$$\mathbb{E}\big[\|\textstyle\sum_\tau a_\tau \epsilon_\tau\|_\star\big] \leq \rho_\star \cdot \mathbb{E}\big[\|\textstyle\sum_\tau a_\tau \epsilon_\tau\|_F\big] \quad \text{for some level } \rho_\star > 0.$$

We use noise levels ρ_{nuc} and ρ_1 for dual norms $\|\cdot\|_{\mathrm{nuc}}$ and $\|\cdot\|_1$, respectively.

The norm inequalities (2) give $\rho_{\text{nuc}} \leq \sqrt{\min\{m,n\}}$ and $\rho_1 \leq \sqrt{mn}$, but the actual levels depend on the noise distribution and can be far smaller (Table 2). Both bounds are per matrix. The measured ρ_{nuc} is about half of its bound and the measured ρ_1 sits well inside its range, so neither is near the worst case the proof has to allow for.

4.2 Convergence bound

With these assumptions in place, we present our main convergence Theorem 1 and optimal parameters Corollary 1 for our `LionMuon` (Algorithm 1). We provide all proofs in Appendix A.

Theorem 1 (Convergence bound of `LionMuon`)**.** *Let the objective function f satisfy Assumption 1 with respect to $\|\cdot\|_2$ with constant L_2, and with respect to $\|\cdot\|_\infty$ with constant L_∞. Let noise Assumptions 2 and 3 hold with noise constants σ, ρ_{nuc} and ρ_1. Fix a horizon T, period $P \in [1,\infty]$, momentum parameters $\beta_1, \beta_2 \in [0,1)$ and learning rates η_M and η_L.*

Define the period-averaged learning rate, noise level and smoothness:

$$\bar{\eta} := \tfrac{\eta_M}{P} + \tfrac{(P-1)\eta_L}{P}, \quad \bar{\rho} := \tfrac{\eta_M}{P\bar{\eta}}\rho_{nuc} + \tfrac{(P-1)\eta_L}{P\bar{\eta}}\rho_1, \quad \bar{L} := \tfrac{\eta_M\tilde{\eta}_{\max}}{P\bar{\eta}^2}L_2 + \tfrac{(P-1)\eta_L\eta_{\max}}{P\bar{\eta}^2}L_\infty, \tag{3}$$

where $\tilde{\eta}_{\max} = \max\{\eta_M, \sqrt{mn}\,\eta_L\}$ and $\eta_{\max} = \max\{\eta_M, \eta_L\}$ for intermediate $P \in (1,\infty)$, with the boundary cases $\tilde{\eta}_{\max} = \eta_{\max} = \eta_M$ at $P = 1$ and $\tilde{\eta}_{\max} = \eta_{\max} = \eta_L$ at $P = \infty$.

Then, our `LionMuon` algorithm starting with $\Delta_0 := f(W_0) - f_\star$, $E_0 = \nabla f(W_0) - M_0$ guarantees the following bound on the period-averaged gradient dual norm:

$$\min_{i<\frac{T}{P}}\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} \leq \frac{\Delta_0}{\bar{\eta}T} + \frac{4\bar{L}\bar{\eta}}{(1-\beta_2)} + \frac{2\beta_1}{\beta_2}\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + 2\left|1-\frac{\beta_1}{\beta_2}\right|\bar{\rho}\sigma + \frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar{\eta}T(1-\beta_2)},$$

$$\text{where } \mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|] := \frac{\left(\eta_M \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P})\|_{nuc}] + \sum_{j=1}^{P-1}[\eta_L \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P+j})\|_1]]\right)}{\eta_M + (P-1)\eta_L}. \tag{4}$$

Prior analyses treat `Muon` or `Lion` on their own (Li & Hong, 2025; Shen et al., 2025; An et al., 2025; Riabinin et al., 2025). Our bound covers both steps, with their different norms and constants, under heavy-tailed noise, and it depends on the schedule only through the period-averaged learning rate, noise and smoothness, which interpolate between the pure-`Muon` and pure-`Lion` regimes. With these constants the bound has the optimal form for momentum-based norm-constrained methods under heavy-tailed noise (Liu & Zhou, 2025; Kornilov et al., 2025), and its boundary cases recover the known bounds for `Muon` and `Lion` (Yu et al., 2026; Nagashima & Iiduka, 2026; Iiduka, 2026).

Corollary 1 (Optimal Parameters for `LionMuon`)**.** *Let the objective function f and the noise satisfy Assumptions 1, 2 and 3 with the period-averaged constants $\bar{L}$, σ and $\bar{\rho}$ defined in* (3)*.*

- *Fix a period $P \in (1,\infty)$ and the learning-rate ratio $\alpha = \eta_M/\eta_L$.*

  To achieve accuracy $\min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} \leq \varepsilon$, our `LionMuon` requires T iterations:

  $$T = O\left(\bar{L}\Delta_0 \cdot \max\left\{\frac{(\bar{\rho}\sigma)^{\frac{\kappa}{\kappa-1}}}{\varepsilon^{\frac{3\kappa-2}{\kappa-1}}}, \frac{1}{\varepsilon^2}\right\}\right), \tag{5}$$

  with the optimal parameters:

  $$1-\beta_2 = \min\left\{\left(\frac{\varepsilon}{16\bar{\rho}\sigma}\right)^{\frac{\kappa}{\kappa-1}}, 1\right\}, \beta_1 \in \beta_2\left[\max\left\{1-\frac{\varepsilon}{16\bar{\rho}\sigma}, 0\right\}, 1\right], \eta_L = \frac{\varepsilon(1-\beta_2)}{32\left(\frac{\alpha}{P} + \frac{P-1}{P}\right)\cdot\bar{L}}.$$

- *Pure `Muon` ($P = 1$) and `Lion` ($P = \infty$) keep the same momentums β_1, β_2, number of iterations T and only single learning rate $\eta_M = \frac{\varepsilon(1-\beta_2)}{32\cdot L_2}$ or $\eta_L = \frac{\varepsilon(1-\beta_2)}{32\cdot L_\infty}$.*
- *We can set single-EMA $\beta_1 = \beta_2$ to get optimal parameters for our `SignMuon`.*

Our complexity (5) has the optimal dependence on the accuracy ε and on the constants $\bar{L}, \bar{\rho}$: with the Frobenius smoothness L_F and noise ρ_F in their place it is the known optimal rate (Zhang et al., 2020).

Tightening up the constants. In the analysis of our LionMuon, we mix Lion and Muon steps and handle different norms within them, applying the worst-case norm inequalities (2) which cover all possible matrices. For this reason, the interpolated smoothness $\bar{L}$ from (3) has extra conservative factors such as $\eta_{\max}$ or $\sqrt{mn}$ which disappear in pure regimes.

In practice, gradients and updates in deep networks tend to be dense (Bernstein et al., 2018), and ours are too (Table 2). For these dense matrices, the norm inequalities usually yield the approximate equalities:

$$\|\nabla f(W_t)\|_{\text{nuc}} \approx \alpha \cdot \|\nabla f(W_t)\|_1 \quad \text{for some large constant } \alpha \lesssim \sqrt{mn}, \tag{6}$$

and we can obtain a more natural interpolation $\bar{L} = \frac{\eta_M^2}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L^2}{P\bar{\eta}^2} L_\infty$ (see Appendix A.4).

4.3 Discussion

Choice of the learning-rate ratio. The scale $\frac{\eta_M}{\eta_L}$ determines the interpolation between the smoothness constants, noise levels and gradient norms of Muon and Lion. By the norm relations (2), the gradient norm $\|\cdot\|_1$ of Lion is always larger than $\|\cdot\|_{\text{nuc}}$ of Muon, especially for dense matrices, so without scaling the Lion updates dominate the metric (4) simply because they are large and frequent. We therefore choose a large scale $\eta_M/\eta_L = \alpha$ that brings the gradient norms (6) to the same order of magnitude: $\min_{i<\frac{T}{P}}\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} \approx \frac{\alpha P \cdot \min_t\{\mathbb{E}[\|\nabla f(W_t)\|_{\text{nuc}}]\}}{\alpha+(P-1)}$. The grid search over (η_M, η_L) in Appendix E confirms that a large ratio is best, and that the tuned ratio grows with P (Section 5). The theory cannot fix its value, so we tune the ratio and η_M.

Choice of period P. The reason to take an intermediate P is cost. A Muon step costs about K_{NS} times a Lion step in FLOPs, and it is the step that communicates. The bound also says that an intermediate P can reach a given accuracy in fewer operations than pure Muon. Our complexity (5) interpolates between the pure-Muon (P=1, $\bar{L} = L_2$, $\bar{\rho} = \rho_{\text{nuc}}$) and pure-Lion ($P$=$\infty$, $\bar{L} = L_\infty$, $\bar{\rho} = \rho_1$) regimes via the period-averaged smoothness $\bar{L}$ and noise $\bar{\rho}$. For typical dense gradients (6), we set the ratio $\eta_M/\eta_L = \alpha$ to equalize the $\|\cdot\|_1$ and $\|\cdot\|_{\text{nuc}}$ gradient norms in the minimal metric (4). Then, the averaged learning rate $\bar{\eta}$ can be estimated by $\bar{\eta} \approx \eta_M/P$, and the refined averaged smoothness and noise become $\bar{L} \approx P^2\frac{L_2}{P} + P^2\frac{(P-1)}{P}\frac{L_\infty}{\alpha^2}$ and $\bar{\rho} \approx P\frac{\rho_{\text{nuc}}}{P} + P\frac{(P-1)}{P}\frac{\rho_1}{\alpha}$. We can now compare the number of operations N that pure Muon and LionMuon need for the same accuracy $\min_t\{\mathbb{E}[\|\nabla f(W_t)\|_{\text{nuc}}]\} \le \varepsilon$. LionMuon does $\frac{K_{\text{NS}}+(P-1)}{P}$ operations per iteration and only needs the weaker accuracy $P \cdot \varepsilon$ in (5):

$$N = O\Bigg[\underbrace{\left(\frac{1}{P}\right)^{\frac{3\kappa-2}{\kappa-1}}\left(\frac{1}{P}+\frac{1}{K_{\text{NS}}}\right)\left(P + P(P-1)\frac{L_\infty}{\alpha^2 L_2}\right)\cdot\left(1+(P-1)\frac{\rho_1}{\alpha\rho_{\text{nuc}}}\right)^{\frac{\kappa}{\kappa-1}}}_{=:\phi(P,\frac{L_\infty}{\alpha^2L_2},\frac{\rho_1}{\alpha\rho_{\text{nuc}}})} \cdot \underbrace{K_{\text{NS}}\cdot L_2\Delta_0\cdot\frac{(\rho_{\text{nuc}}\sigma)^{\frac{\kappa}{\kappa-1}}}{\varepsilon^{\frac{3\kappa-2}{\kappa-1}}}}_{=N_{\text{Muon}}}\Bigg].$$

The trade-off factor $\phi(P, \frac{L_\infty}{\alpha^2 L_2}, \frac{\rho_1}{\alpha\rho_{\text{nuc}}}) \approx \left(\frac{1}{P} + (1-\frac{1}{P})\frac{L_\infty}{\alpha^2 L_2}\right)\cdot\left(\frac{1}{P} + (1-\frac{1}{P})\frac{\rho_1}{\alpha\rho_{\text{nuc}}}\right)^{\frac{\kappa}{\kappa-1}}$ is a polynomial in $1/P$ defined by the scaled smoothness and noise ratios $\frac{L_\infty}{\alpha^2 L_2}$ and $\frac{\rho_1}{\alpha\rho_{\text{nuc}}}$. For $P \in (1, +\infty)$ satisfying $\phi(P, \frac{L_\infty}{\alpha^2 L_2}, \frac{\rho_1}{\alpha\rho_{\text{nuc}}}) < 1$, our LionMuon outruns Muon. The optimal regime $P^* \in [1, \infty]$ minimizes the trade-off factor and can be approximately determined from the trade-off ratios:

1. If $\frac{L_\infty}{\alpha^2 L_2}, \frac{\rho_1}{\alpha\rho_{\text{nuc}}} \gtrsim 1$, the costly Muon is more preferable ($\phi \uparrow$ when $P \uparrow$);
2. If $\frac{L_\infty}{\alpha^2 L_2}, \frac{\rho_1}{\alpha\rho_{\text{nuc}}} < 1$, *intermediate* values P (possibly up to Lion) are the fastest ($\phi \downarrow$ when $P \uparrow$);
3. If $\frac{L_\infty}{\alpha^2 L_2}, \frac{\alpha\rho_{\text{nuc}}}{\rho_1} < 1$ (or > 1), some *intermediate* P^* (can be Muon) is the best ($\phi \downarrow$ then $\phi \uparrow$).

Section 5 finds the third case: the loss against FLOPs is lowest at small intermediate periods, $P^* \in \{2, 5\}$.

Match of theory and practice. We measure the constants of the bound during 124M training (Appendix C, Table 2 and Figure 5). The gradient ratio α comes out close to the tuned ratio η_M/η_L on both datasets. The trade-off ratios on FineWeb put us in the regime with an interior optimal period, and that is what Section 5 finds.

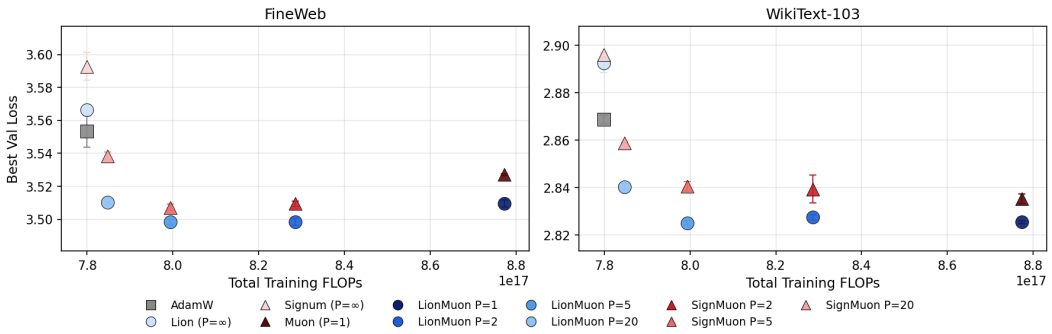

Figure 2: Loss against training FLOPs at 124M on FineWeb (left) and WikiText-103 (right). Every optimizer is tuned, and the error bars are over three seeds. `LionMuon` with $P \in \{2, 5\}$ is lowest on both datasets and uses fewer FLOPs than `Muon`.

5 Experiments

5.1 Setup

All runs use GPT-2-style decoders from the `llm-baselines` benchmark of Semenov et al. (2025), a cosine schedule, weight decay 0.1 and `AdamW` for 1D parameters (Section 3).

Single GPU. At 124M (12 layers, width 768, batch 32×512 tokens, 64,000 steps, 1.05B tokens) we train on FineWeb (Penedo et al., 2024) and WikiText-103 (Merity et al., 2017) and compare `AdamW` (Loshchilov & Hutter, 2019), `Signum` (Bernstein et al., 2018), `Lion` (Chen et al., 2023), `Muon` (Jordan et al., 2024), and `SignMuon` and `LionMuon` with $P \in \{1, 2, 5, 20\}$. Learning rates are tuned on full-length runs: η_M over 10^{-4} to 10^{-1} and, for the alternating methods, the ratio η_M/η_L over 1 to 3000 (Appendix E). Momentum was tuned as well (Appendix E.1), and the best values are $(0.9, 0.99)$ for `Lion` and `LionMuon`, 0.9 for `Muon` and `SignMuon`, and $(0.8, 0.999)$ for `AdamW`. Each tuned setting is rerun with three seeds. At 355M (24 layers, width 1024, sequence 1024, batch 512, 15,650 steps, 8.2B tokens, above the Chinchilla budget (Hoffmann et al., 2022)) we train on FineWeb with every hyperparameter copied from the 124M optimum, so this setting also tests how the tuned values transfer.

Four GPUs. The 124M model is trained end to end with PyTorch DDP on four H200 GPUs, 8 sequences per GPU, for 150,000 steps (2.46B tokens, the Chinchilla budget). Every hyperparameter comes from the 124M tuning, and the rates of `Muon`, `Lion` and `Dion` were rechecked at this horizon (Table 4). Every GPU holds a full copy of the model, and the framework averages the gradients bucket by bucket while the backward pass is still running. `Muon` then orthogonalizes each matrix on one GPU and all-reduces the updates, as in Moonlight (Liu et al., 2025). `Dion` (Ahn et al., 2025) and `MuonBP` (Khaled et al., 2026) use the settings of their papers and the learning rate of `Muon`. Both were designed for sharded models. `MuonBP` is a tensor-parallel and FSDP method, not a data-parallel one, so we emulate its layout with four column blocks per matrix, as a four-way tensor-parallel split would give. Its block step is then local and its full step is `Muon`'s, so it differs from `LionMuon` only in what it does between full steps. `Dion`'s low-rank gradient sync is the data-parallel mode its paper proposes. Appendix D gives the details and pseudocode. Step times are measured on this node and on four A100 GPUs over PCIe, where communication is more expensive.

5.2 Single-GPU training: 124M tuned, 355M transferred

Figure 2 shows the tuned 124M runs. `LionMuon` P=2 and P=5 reach the lowest loss on both datasets, below `Muon`, `AdamW`, `Lion` and `Signum`, and the gaps to `Muon` are more than ten times the seed spread. Appendix E lists every value.

Figure 3 repeats the FineWeb comparison at 355M with every hyperparameter copied from 124M, so it also tests how the tuned values transfer. Every alternating setting except `SignMuon` P=5 beats

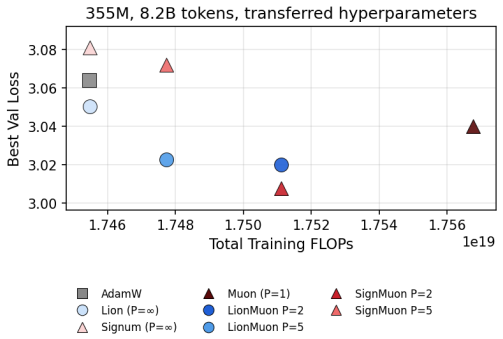

Figure 3: Loss against training FLOPs at 355M on FineWeb, with every hyperparameter copied from 124M. One seed per method, values in Table 6.

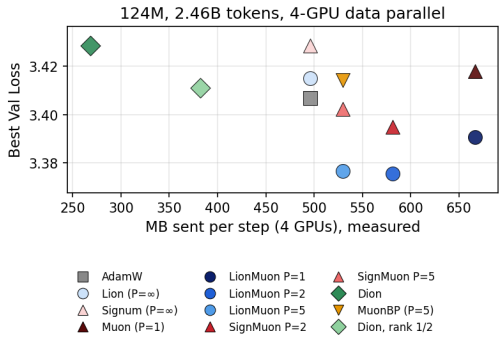

Figure 4: Loss against the bytes one step sends, for 124M trained on four GPUs with data parallelism. The bytes are counted from every collective the run issues. Baselines that had a rate sweep are shown at their best rate. Values in Table 4.

`Muon`, which beats `AdamW`, at fewer FLOPs. `Lion` lands between `Muon` and `AdamW` and `Signum` ends last, so at this size too alternating beats both of its endpoints (Table 6).

Discussion. Alternation does most of the work and the dual EMA adds a little. Going from `Muon` to `SignMuon` $P{=}2$ closes about two thirds of the gap to the best run, and the dual EMA of `LionMuon` closes the rest, at $P{=}5$ and on WikiText-103 as well. The loss is flat between $P{=}2$ and $P{=}5$ on both datasets and rises slightly at $P{=}20$, which still beats `Muon` on FineWeb, so $P{=}5$ gives the cheapest step at no cost in loss. The tuned ratio η_M/η_L grows with P (Appendix E), so a practitioner who changes P should retune this ratio and can leave η_M near its `Muon` value. The period does not have to be fixed either: a `Muon` step drawn with probability $1/P$ at every iteration gives the same loss (Appendix E.2). When hyperparameters are transferred rather than retuned, the dual-EMA variant is the safer one. `LionMuon` $P{=}5$ transfers to 355M unchanged, while `SignMuon` $P{=}5$, with its larger spectral rate, does not. Finally, the constants measured in Section 4.3 predict an interior optimal period on FineWeb, which is what we see.

5.3 Data-parallel training on four GPUs

Figure 4 compares all methods under data parallelism. `LionMuon` $P{=}2$ and $P{=}5$ reach the lowest loss. Both endpoints of the family, pure `Muon` and the pure sign methods, end higher, and so do `AdamW` and the two `Muon` variants. The order is the same as in the single-GPU runs. `Lion` needs a rate three times below its 64k optimum to stay stable over 150,000 steps, and at that rate it ends just below `Muon`. Alternating beats both by a clear margin (Table 4).

Table 1 gives what each step costs. `LionMuon` $P{=}5$ sends a fifth less than `Muon`. `MuonBP` $P{=}5$ sends exactly the same amount and ends level with `Muon`, because it still runs Newton–Schulz on every step. `Dion` at rank $\min(m,n)/4$ sends the least and ends highest. At rank $\min(m,n)/2$ it sends more, still less than `AdamW`, and ends level with `Muon` and `MuonBP`, so its rank trades bytes for loss, and at neither rank does it come near `LionMuon`. `AdamW` sends slightly less than `LionMuon` $P{=}5$ and ends higher, so against `AdamW` our claim is the loss, not the cost.

Step times depend on the interconnect (Table 1). On PCIe, where the all-reduce is expensive, `Dion` and `MuonBP` are also cheaper per step than `Muon`. On NVLink they become slower than `Muon`, since both trade bytes for extra arithmetic, while `LionMuon` still saves. Replicating the orthogonalization on every device would remove `Muon`'s all-reduce at the price of about a third more arithmetic (Appendix D.1). `LionMuon` divides both costs by P, so it is cheaper either way.

Table 5 in Appendix D puts the two axes together and gives what each method needs to reach `Muon`'s own final loss, relative to `Muon`. `LionMuon` $P{=}5$ gets there with the fewest FLOPs and the least

Table 1: What one step costs on top of forward and backward, at 124M on four GPUs. Exposed communication is the optimizer's own all-reduce, which cannot start before the backward pass ends. Step times are relative to `Muon`, measured by running all methods back to back in a random order within each round and taking the ratio inside the round, so that any background load on the node falls on every method alike (30 rounds on PCIe, 40 on NVLink, 95% intervals within ± 0.01 and ± 0.08). `Muon`'s median step took 1.56 s on PCIe and 0.24 s on NVLink. `Dion`'s bytes replace the gradient all-reduce. State is per 2D parameter W.

		Exposed		Step time / `Muon`	
Optimizer	Newton–Schulz	MB/step	State	PCIe	NVLink
`AdamW`	no	0	$2\|W\|$	0.75	0.85
`Lion` / `Signum`	no	0	$\|W\|$	–	–
`Muon`	every step	170	$\|W\|$	1.00	1.00
`LionMuon` $P{=}2$	every 2nd step	85	$\|W\|$	0.88	0.92
`LionMuon` $P{=}5$	every 5th step	34	$\|W\|$	**0.81**	**0.89**
`MuonBP` $P{=}5$	blocks each, full every 5th	34	$\|W\|$	0.94	1.25
`Dion`, rank $\min(m,n)/4$	no, QR on factors	269	$\|W\| + nr$	0.76	1.50

time on either interconnect, and $P{=}2$ is close behind. On bytes alone `Dion` at rank $1/2$ is cheapest, but it stops level with `Muon`, and `AdamW` is about as cheap as `LionMuon` $P{=}5$ but stops higher.

Each distributed run has one seed, and the seed spread at 124M is an order of magnitude below the gaps here. `Muon` was also run at rates three times lower and three times higher than its transferred one, and neither improves on it (Table 4).

6 Conclusion

We present `LionMuon`, an optimizer that takes one `Muon` step per P iterations and `Lion` steps in between, sharing a single dual-EMA momentum buffer and adding only the integer P. The step cost follows directly: Newton–Schulz compute and the optimizer all-reduce of `Muon` are divided by P, the sign steps are local, and the state is half of `AdamW`'s. A complexity bound (5) shows that the compute-optimal period is set by the ratios $\frac{L_\infty}{L_2}$ and $\frac{\rho_1}{\rho_{\text{nuc}}}$ and says when `LionMuon` beats both `Muon` and `Lion`. Empirically, `LionMuon` with $P \in \{2, 5\}$ reaches a lower loss than `Muon`, `AdamW`, `Lion` and `Signum` at 124M and 355M, and under 4-GPU data-parallel training at $1\times$ Chinchilla it reaches `Muon`'s final loss with a third less wall-clock on PCIe and ends below every baseline. Larger models, multi-node training, an adaptive period and an analysis that accounts for the Newton–Schulz error are the natural next steps.

Ethics statement

This work studies optimizers for language-model pretraining. It uses public datasets, introduces no new data and involves no human subjects. Its aim, the same loss for less compute, raises no ethical concerns beyond those of language-model training in general.

Reproducibility statement

Algorithm 1 is the complete update rule; Appendix D lists every architecture, schedule and hyperparameter, Appendix E the tuning grids and selected values. The anonymized code with the training scripts, the round-robin distributed protocol and the plotting scripts is at `https://anonymous.4open.science/r/lionmuon/`. All proofs are in Appendices A and B.

AI use statement

Large language models were used to help write the experiment and plotting scripts, to draft and edit the text, and to support the literature review. All experiments were designed, run and verified by the authors, who take responsibility for the final content.

REFERENCES

Kwangjun Ahn, Byron Xu, Natalie Abreu, Ying Fan, Gagik Magakyan, Pratyusha Sharma, Zheng Zhan, and John Langford. Dion: Distributed orthonormalized updates, 2025. URL https://arxiv.org/abs/2504.05295.

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The Polar Express: Optimal matrix sign methods and their application to the Muon algorithm, 2025. URL https://arxiv.org/abs/2505.16932.

Kang An, Yuxing Liu, Rui Pan, Yi Ren, Shiqian Ma, Donald Goldfarb, and Tong Zhang. ASGO: Adaptive structured gradient optimization. In *Advances in Neural Information Processing Systems 38*, pp. 140553–140592. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2025. doi: 10.52202/085713-4229. URL https://doi.org/10.52202/085713-4229.

Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology, 2024. URL https://arxiv.org/abs/2409.20325.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. SIGNSGD: compressed optimisation for non-convex problems. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 559–568. PMLR, 2018. URL http://proceedings.mlr.press/v80/bernstein18a.html.

Jeremy Bernstein, Jiawei Zhao, Kamyar Azizzadenesheli, and Anima Anandkumar. signSGD with majority vote is communication efficient and fault tolerant. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=BJxhijAcY7.

Lizhang Chen, Bo Liu, Kaizhao Liang, and Qiang Liu. Lion secretly solves a constrained optimization: As Lyapunov predicts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*, 2024. URL https://openreview.net/forum?id=e4xS9ZarDr.

Lizhang Chen, Jonathan Li, and Qiang Liu. Muon optimizes under spectral norm constraints. *Trans. Mach. Learn. Res.*, 2026, 2026a. URL https://openreview.net/forum?id=Blz4hjxLwU.

Vincent Chen, Starrick Liu, Regis Cheng, Dance Yang, Shalfun Li, Ryan Yu, Lucy Liang, Hang Su, Roy Gan, Hao Wang, and Qian Wang. DMuon: Efficient distributed Muon training with Near-Adam overhead, 2026b. URL https://arxiv.org/abs/2606.27153.

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 *Advances in Neural Information Processing Systems 36*, pp. 49205–49233. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-2140. URL https://doi.org/10.52202/075280-2140.

Savelii Chezhegov, Daniela Angela Parletta, Andrea Paudice, and Eduard Gorbunov. High-probability bounds for the last iterate of clipped SGD. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4sGEvpwyxN.

Michael Crawshaw, Mingrui Liu, Francesco Orabona, Wei Zhang, and Zhenxun Zhuang. Robustness to unbounded smoothness of generalized SignSGD. In *Advances in Neural Information Processing Systems 35*, pp. 9955–9968. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2022. doi: 10.52202/068431-0723. URL https://doi.org/10.52202/068431-0723.

George E. Dahl, Frank Schneider, Zachary Nado, Naman Agarwal, Chandramouli Shama Sastry, Philipp Hennig, Sourabh Medapati, Runa Eschenhagen, Priya Kasimbeg, Daniel Suo, Juhan Bae, Justin Gilmer, Abel L. Peirson, Bilal Khan, Rohan Anil, Mike Rabbat, Shankar Krishnan,

Daniel Snider, Ehsan Amid, Kongtao Chen, Chris J. Maddison, Rakshith Vasudev, Michal Badura, Ankush Garg, and Peter Mattson. Benchmarking neural network training algorithms, 2023. URL https://arxiv.org/abs/2306.07179.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Essential AI. Layer sharding for large-scale training with Muon. Essential AI Blog, may 2025. URL https://www.essential.ai/blog/infra.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: Training ImageNet in 1 hour, 2017. URL https://arxiv.org/abs/1706.02677.

Ekaterina Grishina, Matvey Smirnov, and Maxim Rakhuba. Accelerating Newton-Schulz iteration for orthogonalization via Chebyshev-type polynomials, 2025. URL https://arxiv.org/abs/2506.10935.

Mert Gürbüzbalaban, Umut Simsekli, and Lingjiong Zhu. The heavy-tail phenomenon in SGD. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3964–3975. PMLR, 2021. URL http://proceedings.mlr.press/v139/gurbuzbalaban21a.html.

Elad Hazan, Kfir Y. Levy, and Shai Shalev-Shwartz. Beyond convexity: Stochastic quasi-convex optimization. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1594–1602, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/934815ad542a4a7c5e8a2dfa04fea9f5-Abstract.html.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL https://arxiv.org/abs/2203.15556.

Feihu Huang, Yuning Luo, and Songcan Chen. LiMuon: Light and fast Muon optimizer for large models, 2025. URL https://arxiv.org/abs/2509.14562.

Florian Hübler, Ilyas Fatkhullin, and Niao He. From gradient clipping to normalization for heavy tailed SGD. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Mohammad Emtiyaz Khan (eds.), *International Conference on Artificial Intelligence and Statistics, AISTATS 2025, Mai Khao, Thailand, 3-5 May 2025*, volume 258 of *Proceedings of Machine Learning Research*, pp. 2413–2421. PMLR, 2025. URL https://proceedings.mlr.press/v258/hubler25a.html.

Hideaki Iiduka. Muon converges under heavy-tailed noise: Nonconvex Hölder-smooth empirical risk minimization, 2026. URL https://arxiv.org/abs/2603.15059.

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 427–435. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/jaggi13.html.

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

Priya Kasimbeg, Frank Schneider, Runa Eschenhagen, Juhan Bae, Chandramouli Shama Sastry, Mark Saroufim, Boyuan Feng, Less Wright, Edward Z. Yang, Zachary Nado, Sourabh

Medapati, Philipp Hennig, Michael Rabbat, and George E. Dahl. Accelerating neural network training: An analysis of the AlgoPerf competition. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL https://openreview.net/forum?id=CtM5xjRSfm.

Nitish Shirish Keskar and Richard Socher. Improving generalization performance by switching from Adam to SGD, 2017. URL https://arxiv.org/abs/1712.07628.

Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. MuonBP: Faster Muon via block-periodic orthogonalization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mHouLSUQP5.

Nikita Kornilov, Ohad Shamir, Aleksandr Lobanov, Darina Dvinskikh, Alexander Gasnikov, Innokentiy Shibaev, Eduard Gorbunov, and Samuel Horváth. Accelerated zeroth-order method for non-smooth stochastic convex optimization problem with infinite variance, 2023. URL https://arxiv.org/abs/2310.18763.

Nikita Kornilov, Philip Zmushko, Andrei Semenov, Mark Ikonnikov, Alexander Gasnikov, and Alexander Beznosikov. Sign operator for coping with heavy-tailed noise in non-convex optimization: High probability bounds under (l_0, l_1)-smoothness, 2025. URL https://arxiv.org/abs/2502.07923.

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

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

Zijian Liu and Zhengyuan Zhou. Nonconvex stochastic optimization under heavy-tailed noises: Optimal convergence without gradient clipping. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*, 2025. URL https://openreview.net/forum?id=NKotdPUc3L.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL https://openreview.net/forum?id=Bkg3g2R9FX.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Shuntaro Nagashima and Hideaki Iiduka. Improved convergence rates of Muon optimizer for nonconvex optimization, 2026. URL https://arxiv.org/abs/2601.19400.

Tianyu Pang, Yujie Fang, Zihang Liu, Shenyang Deng, Lei Hsiung, Shuhua Yu, and Yaoqing Yang. HTMuon: Improving Muon via heavy-tailed spectral correction. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36504–36535. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1819. URL https://doi.org/10.18653/v1/2026.findings-acl.1819.

Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin A. Raffel, Leandro von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. doi: 10.52202/079017-0970. URL http://papers.nips.cc/paper_files/paper/2024/hash/370df50ccfdf8bde18f8f9c2d9151bda-Abstract-Datasets_and_Benchmarks_Track.html.

Egor Petrov, Grigoriy Evseev, Aleksey Antonov, Andrey Veprikov, Nikolay Bushkov, Stanislav Moiseev, and Aleksandr Beznosikov. Leveraging coordinate momentum in SignSGD and Muon: Memory-optimized zero-order, 2025. URL https://arxiv.org/abs/2506.04430.

Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making Muon & Scion great again! (bridging theory and practice of LMO-based optimizers for LLMs), 2025. URL https://arxiv.org/abs/2505.13416.

Artem Riabinin, Andrey Veprikov, Arman Bolatov, Martin Takáč, and Aleksandr Beznosikov. Where does warm-up come from? adaptive scheduling for norm-constrained optimizers, 2026. URL https://arxiv.org/abs/2602.05813.

Abdurakhmon Sadiev, Marina Danilova, Eduard Gorbunov, Samuel Horváth, Gauthier Gidel, Pavel E. Dvurechensky, Alexander V. Gasnikov, and Peter Richtárik. High-probability bounds for stochastic optimization and variational inequalities: the case of unbounded variance. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29563–29648. PMLR, 2023. URL https://proceedings.mlr.press/v202/sadiev23a.html.

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

Maria-Eleni Sfyraki and Jun-Kun Wang. Lions and Muons: Optimization via stochastic Frank-Wolfe under heavy-tailed noise, 2025. URL https://arxiv.org/abs/2506.04192.

Wei Shen, Ruichuan Huang, Minhui Huang, Cong Shen, and Jiawei Zhang. On the convergence analysis of Muon, 2025. URL https://arxiv.org/abs/2505.23737.

Kimi Team, Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Chenxiao Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Yuyao Ge, Shangyi Geng, Qizheng Gu, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Yunjia He, Chao Hong, Hao Hu, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yang Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Haoyu Lu, Lijun Lu, Yashuo Luo, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Zeyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Lin Sui, Xinjie Sun, Flood Sung, Yunpeng Tai, Heyi Tang, Jiawen Tao, Qifeng Teng, Chaoran Tian, Chensi Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Jianzhou Wang, Jiaxing Wang, Jinhong Wang, Shengjie Wang, Shuyi Wang, Si Wang, Xinyuan Wang, Yao Wang, Yejie Wang, Yiqin Wang, Yuxin Wang, Yuzhi Wang, Zhaoji Wang, Zhengtao Wang, Zhengtao Wang, Zhexu Wang, Chu Wei, Qianqian Wei, Haoning Wu, Wenhao Wu, Xingzhe Wu, Yuxin Wu, Chenjun Xiao, Jin Xie, Xiaotong Xie, Weimin Xiong, Boyu Xu, Jinjing Xu, L. H. Xu, Lin Xu, Suting Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ziyao Xu, Jing Xu, Jing Xu, Junjie Yan,

Yuzi Yan, Hao Yang, Xiaofei Yang, Yi Yang, Ying Yang, Zhen Yang, Zhilin Yang, Zonghan Yang, Haotian Yao, Xingcheng Yao, Wenjie Ye, Zhuorui Ye, Bohong Yin, Longhui Yu, Enming Yuan, Hongbang Yuan, Mengjie Yuan, Siyu Yuan, Haobing Zhan, Dehao Zhang, Hao Zhang, Wanlu Zhang, Xiaobin Zhang, Yadong Zhang, Yangkun Zhang, Yichi Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Haotian Zhao, Yikai Zhao, Zijia Zhao, Huabin Zheng, Shaojie Zheng, Longguang Zhong, Jianren Zhou, Xinyu Zhou, Zaida Zhou, Jinguo Zhu, Zhen Zhu, Weiyu Zhuang, and Xinxing Zu. Kimi K2: Open agentic intelligence, 2025. URL https://arxiv.org/abs/2507.20534.

Andrey Veprikov, Arman Bolatov, Aleksandr Bogdanov, Samuel Horváth, Aleksandr Beznosikov, Martin Takáč, and Slavomir Hanzely. Preconditioned norms: A unified framework for steepest descent, Quasi-Newton and adaptive methods, 2025. URL https://arxiv.org/abs/2510.10777.

Zixiao Wang, Yifei Shen, and Huishuai Zhang. OLion: Approaching the Hadamard ideal by intersecting spectral and ℓ_∞ implicit biases, 2026. URL https://arxiv.org/abs/2602.01105.

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

Yun Yue, Zhiling Ye, Jiadi Jiang, Yongchao Liu, and Ke Zhang. AGD: an auto-switchable optimizer using stepwise gradient difference for preconditioning matrix. In *Advances in Neural Information Processing Systems 36*, pp. 45812–45832. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023. doi: 10.52202/075280-1985. URL https://doi.org/10.52202/075280-1985.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J. Reddi, Sanjiv Kumar, and Suvrit Sra. Why are adaptive methods good for attention models? In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b05b57f6add810d3b7490866d74c0053-Abstract.html.

A MISSING PROOFS

This appendix collects the missing proofs of Theorem 1 and Corollary 1. We first state and prove two technical lemmas (the descent Lemma 1 and the momentum error bound Lemma 2) that are the building blocks of our analysis. Then, we assemble these lemmas into the main proof.

A.1 BUILDING-BLOCK LEMMAS

Lemma 1 (`LionMuon` Descent Lemma)**.** *Let the objective function f satisfy Assumption 1 with respect to a norm $\|\cdot\|$, and let $\|\cdot\|_\star$ be its dual norm. Then, for update $W_{t+1} = W_t + \eta_t U_t$ with $U_t = \textit{LMO}_{\|\cdot\|}(\hat{G}_t)$, momentums $M_t = \beta_2 M_{t-1} + (1-\beta_2)G_t$ and $\hat{G}_t = \beta_1 M_{t-1} + (1-\beta_1)G_t$, the following bound holds:*

$$f(W_{t+1}) \leq f(W_t) - \eta_t \cdot \|\nabla f(W_t)\|_\star + \frac{2\eta_t\beta_1}{\beta_2}\|\nabla f(W_t) - M_t\|_\star + 2\eta_t \left|1 - \frac{\beta_1}{\beta_2}\right| \|\nabla f(W_t) - G_t\|_\star + \frac{L\eta_t^2}{2}.$$

Proof. We begin our proof with bounding the value $f(W_{t+1})$ after the update using the smoothness Assumption 1:

$$\begin{aligned}
f(W_{t+1}) &= f(W_t + \eta_t U_t) \\
&\leq f(W_t) + \eta_t \langle \nabla f(W_t), U_t \rangle + \frac{L\eta_t^2}{2}\|U_t\|^2 \\
&\leq f(W_t) + \eta_t \langle \nabla f(W_t), U_t \rangle + \frac{L\eta_t^2}{2} \\
&= f(W_t) + \eta_t \langle \hat{G}_t, U_t \rangle + \eta_t \langle \nabla f(W_t) - \hat{G}_t, U_t \rangle + \frac{L\eta_t^2}{2}.
\end{aligned}$$

Then, we define the optimal matrix $\hat{V}_t := \arg\max_{\|V\| \leq 1} \langle V, -\nabla f(W_t) \rangle$ and continue bounding:

$$\begin{aligned}
f(W_{t+1}) &= f(W_t) + \eta_t \langle \hat{G}_t, U_t \rangle + \eta_t \langle \nabla f(W_t) - \hat{G}_t, U_t \rangle + \frac{L\eta_t^2}{2} \\
&\leq f(W_t) + \eta_t \langle \hat{G}_t, \hat{V}_t \rangle + \eta_t \langle \nabla f(W_t) - \hat{G}_t, U_t \rangle + \frac{L\eta_t^2}{2} \\
&= f(W_t) + \eta_t \langle \hat{G}_t, \hat{V}_t - U_t \rangle + \eta_t \langle \nabla f(W_t), U_t \rangle + \frac{L\eta_t^2}{2} \\
&= f(W_t) + \eta_t \langle \nabla f(W_t), \hat{V}_t \rangle + \eta_t \langle \nabla f(W_t) - \hat{G}_t, U_t - \hat{V}_t \rangle + \frac{L\eta_t^2}{2} \\
&\leq f(W_t) - \eta_t \|\nabla f(W_t)\|_\star + \eta_t \|\nabla f(W_t) - \hat{G}_t\|_\star \|U_t - \hat{V}_t\| + \frac{L\eta_t^2}{2} \\
&\leq f(W_t) - \eta_t \|\nabla f(W_t)\|_\star + \|\nabla f(W_t) - \hat{G}_t\|_\star 2\eta_t + \frac{L\eta_t^2}{2}.
\end{aligned}$$

Furthermore, we can switch to the bound with the main momentum M_t:

$$\begin{aligned}
\|\nabla f(W_t) - \hat{G}_t\|_\star &= \|\nabla f(W_t) - M_t + M_t - \hat{G}_t\|_\star \\
&= \left\| \nabla f(W_t) - M_t + \left(1 - \frac{\beta_1}{\beta_2}\right)(M_t - G_t) \right\|_\star \\
&= \left\| \frac{\beta_1}{\beta_2}(\nabla f(W_t) - M_t) + \left(1 - \frac{\beta_1}{\beta_2}\right)(\nabla f(W_t) - G_t) \right\|_\star \\
&\leq \frac{\beta_1}{\beta_2}\|\nabla f(W_t) - M_t\|_\star + \left|1 - \frac{\beta_1}{\beta_2}\right| \|\nabla f(W_t) - G_t\|_\star.
\end{aligned}$$

Thus, the final bound is

$$f(W_{t+1}) \leq f(W_t) - \eta_t \cdot \|\nabla f(W_t)\|_\star + \frac{2\eta_t\beta_1}{\beta_2}\|\nabla f(W_t) - M_t\|_\star + 2\eta_t \left|1 - \frac{\beta_1}{\beta_2}\right| \|\nabla f(W_t) - G_t\|_\star + \frac{L\eta_t^2}{2}.$$

□

Lemma 2 (`LionMuon` Momentum Error Bound)**.** *Let the objective function f and corrupting noise satisfy Assumptions 1, 2, 3 with a norm $\|\cdot\|$ and let momentum M_τ be defined as: $M_\tau = \beta_2 M_{\tau-1} + (1-\beta_2)G_\tau$. Then, for updates $W_{\tau+1} = W_\tau + \eta_\tau U_\tau$, the following bound holds*

$$\mathbb{E}[\|E_t\|_\star] \leq \beta_2^t \|E_0\|_\star + \frac{LA\beta_2}{1-\beta_2} + \rho_\star \sigma (1-\beta_2)^{\frac{\kappa-1}{\kappa}}.$$

where $E_t := \nabla f(W_t) - M_t$ and $\max_{\tau \leq t}\{\eta_\tau \cdot \|U_\tau\|\} \leq A$.

Proof. Using the momentum definition, we write down the recursive step:

$$\begin{aligned}
E_t &= \nabla f(W_t) - M_t = \nabla f(W_t) - \big(\beta_2 M_{t-1} + (1-\beta_2)G_t\big) \\
&= \beta_2 \nabla f(W_t) + (1-\beta_2)\nabla f(W_t) - \beta_2 M_{t-1} - (1-\beta_2)G_t \\
&= \beta_2\big(\nabla f(W_t) - M_{t-1}\big) + (1-\beta_2)\big(\nabla f(W_t) - G_t\big) \\
&= \beta_2\big(\nabla f(W_t) - \nabla f(W_{t-1}) + \nabla f(W_{t-1}) - M_{t-1}\big) + (1-\beta_2)\big(\nabla f(W_t) - G_t\big) \\
&= \beta_2\big(\nabla f(W_t) - \nabla f(W_{t-1})\big) + \beta_2\big(\nabla f(W_{t-1}) - M_{t-1}\big) + (1-\beta_2)\big(\nabla f(W_t) - G_t\big).
\end{aligned}$$

Further, we use the notations $S_t = \nabla f(W_t) - G_t$ and $R_t = \nabla f(W_t) - \nabla f(W_{t-1})$ to unroll the recursion:

$$E_t = \beta_2 E_{t-1} + (1-\beta_2)S_t + \beta_2 R_t = \beta_2^t E_0 + \sum_{j=0}^{t-1} \beta_2^j \big[(1-\beta_2)S_{t-j} + \beta_2 R_{t-j}\big].$$

Now, we observe that

$$\|R_{t-j}\|_\star = \|\nabla f(W_{t-j}) - \nabla f(W_{t-j-1})\|_\star \leq L\|W_{t-j} - W_{t-j-1}\| = L\eta_{t-1}\|U_{t-j-1}\| \leq LA.$$

Therefore, we estimate using the norm equivalence Assumption 3:

$$\begin{aligned}
\mathbb{E}[\|E_t\|_\star] &\leq \beta_2^t \cdot \|E_0\|_\star + \mathbb{E}\left[\left\|\sum_{j=0}^{t-1} \beta_2^j \big[(1-\beta_2)S_{t-j} + \beta_2 R_{t-j}\big]\right\|_\star\right] \\
&\leq \beta_2^t \cdot \|E_0\|_\star + \sum_{j=0}^{t-1} \beta_2^{j+1}\mathbb{E}[\|R_{t-j}\|_\star] + \mathbb{E}\left[\left\|\sum_{j=0}^{t-1} \beta_2^j (1-\beta_2) S_{t-j}\right\|_\star\right] \\
&\leq \beta_2^t \cdot \|E_0\|_\star + \frac{LA\beta_2}{1-\beta_2} + \rho\left(\mathbb{E}\left[\left\|\sum_{j=0}^{t-1} \beta_2^j (1-\beta_2) S_{t-j}\right\|_F^\kappa\right]\right)^{\frac{1}{\kappa}}.
\end{aligned}$$

For the linear combination of corrupting noises, we apply a batching lemma on the reduction of the κ-th moment, proposed and developed in works (Kornilov et al., 2023; Hübler et al., 2025):

Lemma 3. *Let $X_1, \ldots, X_B$ be a matrix martingale difference sequence (i.e. $\mathbb{E}[X_j | X_{j-1}, \ldots, X_1] = 0$ for $1 < j \leq B$) such that $\mathbb{E}[\|X_j\|_F^\kappa | X_{j-1}, \ldots, X_1] \leq \sigma_j^\kappa$ for $1 < \kappa \leq 2$. Then, we have*

$$\mathbb{E}\left[\left\|\sum_{j=1}^{B} X_i\right\|_F^\kappa\right] \leq \sum_{j=1}^{B} \sigma_i^\kappa.$$

Namely, we treat the sequence $\{\beta_2^j(1-\beta_2)\cdot S_{t-j}\}_{j=0}^{t-1}$ as the required martingale difference sequence with $\sigma_j = \beta_2^j(1-\beta_2)\sigma$ and apply Lemma 3:

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{j=0}^{t-1} \beta_2^j (1-\beta_2) S_{t-j}\right\|_F^\kappa\right] &\leq \sum_{j=0}^{t-1} \beta_2^{\kappa j}(1-\beta_2)^\kappa \sigma^\kappa \\
&\leq \sigma^\kappa (1-\beta_2)^\kappa \sum_{j=0}^{t-1} \beta_2^{\kappa j} \\
&\leq \frac{\sigma^\kappa (1-\beta_2)^\kappa}{1-\beta_2^\kappa}.
\end{aligned}$$

Hence, we get

$$\mathbb{E}[\|E_t\|_\star] \leq \beta_2^t \cdot \|E_0\|_\star + \frac{LA\beta_2}{1-\beta_2} + \rho\sigma\frac{1-\beta_2}{(1-\beta_2^\kappa)^{\frac{1}{\kappa}}}.$$

Since $0 < 1-\beta_2 \leq 1-\beta_2^\kappa$, we further simplify the bound to:

$$\mathbb{E}[\|E_t\|_\star] \leq \beta_2^t \cdot \|E_0\|_\star + \frac{LA\beta_2}{1-\beta_2} + \rho\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}}.$$

□

A.2 Proof of LionMuon Convergence Theorem 1

Proof. We divide the iteration indices $t \in \{0, \ldots, T-1\}$ into two disjoint sets: the set of Muon steps $S_{\text{muon}} = \{t \mid t \equiv 0 \pmod P\}$ and the set of block Lion steps $S_{\text{lion}} = \{t \mid t \not\equiv 0 \pmod P\}$.

Step 1: Analysis of the Muon Steps ($t \in S_{\text{muon}}$). For $t \in S_{\text{muon}}$, the update utilizes the spectral norm $\|\cdot\|_2$. To use Lemmas 1 and 2, we find the uniform upper bound constant A_2 such that $\max_{\tau \leq t}\{\eta_\tau \|U_\tau\|_2\} \leq A_2$ for all previous steps $\tau \leq t$:

- If $\tau \in S_{\text{muon}}$, then all updates $U_\tau = \text{LMO}_{\|\cdot\|_2}(\hat{G}_\tau)$ are bounded by $\|U_\tau\|_2 = 1$, and the stepsize is $\eta_\tau = \eta_M$.
- If $\tau \in S_{\text{lion}}$, then the updates $U_\tau = \text{LMO}_{\|\cdot\|_\infty}(\hat{G}_\tau)$ utilize the infinity norm LMO, yielding $\|U_\tau\|_\infty = 1$. Using the norm equivalence ($\|W\|_2 \leq \sqrt{mn}\|W\|_\infty$), we have $\|U_\tau\|_2 \leq \sqrt{mn}$ and stepsize $\eta_\tau = \eta_L$.

Taking the maximum over these two cases for $P \in (1, \infty)$, we get $A_2 = \max\{\eta_M, \sqrt{mn} \cdot \eta_L\}$. When $P = 1$, all τ steps belong only to S_{muon} and $A_2 = \eta_M$.

Thus, combining Lemmas 1 and 2 with A_2 and dual variance factor ρ_{nuc}, we bound the gradient dual norm:

$$\begin{aligned}
\eta_M \cdot \mathbb{E}[\|\nabla f(W_t)\|_{\text{nuc}}] &\leq \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \frac{2\eta_M\beta_1}{\beta_2}\mathbb{E}[\|\nabla f(W_t) - M_t\|_{\text{nuc}}] \\
&\quad + 2\eta_M\left|1 - \frac{\beta_1}{\beta_2}\right|\mathbb{E}[\|\nabla f(W_t) - G_t\|_{\text{nuc}}] + \frac{L_2\eta_M^2}{2} \\
&\leq \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] \\
&\quad + \frac{2\eta_M\beta_1}{\beta_2}\left(\beta_2^t\|E_0\|_{\text{nuc}} + \frac{L_2A_2\beta_2}{1-\beta_2} + \rho_{\text{nuc}}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}}\right) \\
&\quad + 2\eta_M\left|1 - \frac{\beta_1}{\beta_2}\right|\rho_{\text{nuc}}\sigma + \frac{L_2\eta_M^2}{2} =: \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \text{Error}_t^{\text{muon}}.
\end{aligned} \tag{7}$$

Step 2: Analysis of the Lion Steps ($t \in S_{\text{lion}}$). For $t \in S_{\text{lion}}$, the update utilizes the infinite norm $\|\cdot\|_\infty$. To use Lemmas 1 and 2, we find the uniform upper bound constant A_∞ such that $\max_{\tau \leq t}\{\eta_\tau \|U_\tau\|_\infty\} \leq A_\infty$ for all past steps $\tau \leq t$:

- If $\tau \in S_{\text{muon}}$, then all updates $U_\tau = \text{LMO}_{\|\cdot\|_2}(\hat{G}_\tau)$ are bounded by $\|U_\tau\|_\infty \leq \|U_\tau\|_2 = 1$ and the stepsize is $\eta_\tau = \eta_M$.
- If $\tau \in S_{\text{lion}}$, then the updates $U_\tau = \text{LMO}_{\|\cdot\|_\infty}(\hat{G}_\tau)$ utilize the infinity norm LMO, yielding $\|U_\tau\|_\infty = 1$ and stepsize $\eta_\tau = \eta_L$.

Taking the maximum over these two cases for $P \in (1, \infty)$, we get $A_\infty = \max(\eta_M, \eta_L)$. When $P = \infty$, all τ steps belong only to S_{lion} and $A_\infty = \eta_L$.

Similarly combining Lemmas 1 and 2 with A_∞ and dual variance factor ρ_1, we bound the gradient dual norm:

$$\eta_L \cdot \mathbb{E}[\|\nabla f(W_t)\|_1] \le \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \frac{\eta_L \beta_1}{\beta_2}\left(\beta_2^t \|E_0\|_1 + \frac{L_\infty A_\infty \beta_2}{1-\beta_2} + \rho_1 \sigma (1-\beta_2)^{\frac{\kappa-1}{\kappa}}\right)$$
$$+ 2\eta_L \left|1 - \frac{\beta_1}{\beta_2}\right| \rho_1 \sigma + \frac{L_\infty \eta_L^2}{2} =: \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \text{Error}_t^{\text{lion}}. \quad (8)$$

Step 3: Telescoping Sum. We sum the bounds for `Muon` (7) and `Lion` (8) steps over $t = 0$ to $T-1$. Note that we group the terms over $\frac{T}{P}$ periods of length P, and the total numbers of each step type are $|S_{\text{muon}}| = \frac{T}{P}$ and $|S_{\text{lion}}| = \frac{T(P-1)}{P}$:

$$\sum_{i=0}^{\frac{T}{P}-1}\left(\eta_M \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P})\|_{\text{nuc}}] + \sum_{j=1}^{P-1}[\eta_L \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P+j})\|_1]]\right) \le f(W_0) - f_\star + \sum_{t\in S_{\text{muon}}} \text{Error}_t^{\text{muon}}$$
$$+ \sum_{t\in S_{\text{lion}}} \text{Error}_t^{\text{lion}}.$$

For the left-hand side, we consider the minimal period-averaged gradient dual norm:

$$\sum_{i=0}^{\frac{T}{P}-1}\left(\eta_M \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P})\|_{\text{nuc}}] + \sum_{j=1}^{P-1}[\eta_L \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P+j})\|_1]]\right)$$
$$= \sum_{i=0}^{\frac{T}{P}-1}(\eta_M + \eta_L(P-1)) \cdot \underbrace{\frac{\left(\eta_M \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P})\|_{\text{nuc}}] + \sum_{j=1}^{P-1}[\eta_L \cdot \mathbb{E}[\|\nabla f(W_{i\cdot P+j})\|_1]]\right)}{\eta_M + \eta_L(P-1)}}_{:=\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]}$$
$$\ge \sum_{i=0}^{\frac{T}{P}-1}(\eta_M + \eta_L(P-1)) \cdot \min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\}$$
$$= \frac{T}{P}(\eta_M + \eta_L(P-1)) \cdot \min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} = T \cdot \bar{\eta} \cdot \min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\},$$

where the period-averaged stepsize is $\bar{\eta} := \frac{\eta_M}{P} + \frac{\eta_L(P-1)}{P}$.

Note that when $P = 1$ or $P = \infty$ the minimal averaged norm becomes the minimal nuclear dual norm over all intermediate points $\min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} = \min_i\{\mathbb{E}[\|\nabla f(W_i)\|_{\text{nuc}}]\}$ or $\min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} = \min_i\{\mathbb{E}[\|\nabla f(W_i)\|_1]\}$.

For the right-hand side, we apply the geometric series upper bound $\sum_{t=0}^{T-1}\beta_2^t \le \frac{1}{1-\beta_2}$ for the intermediate momentum errors. Grouping the constant terms matching the lengths of sets S_{muon} and S_{lion} and dividing the entire inequality by $T\bar{\eta}$, we obtain the overall bound:

$$\min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} \le \frac{\Delta_0}{\bar{\eta}T} + \frac{2\eta_M\beta_1}{\beta_2}\frac{\|E_0\|_{\text{nuc}}}{T\bar{\eta}(1-\beta_2)} + \frac{1}{P}\frac{2\eta_M\beta_1}{\beta_2}\frac{L_2A_2\beta_2}{(1-\beta_2)\bar{\eta}}$$
$$+ \frac{1}{P}\frac{2\eta_M\beta_1}{\beta_2\bar{\eta}}\rho_{\text{nuc}}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + \frac{1}{P}\frac{2\eta_M}{\bar{\eta}}\left|1-\frac{\beta_1}{\beta_2}\right|\rho_{\text{nuc}}\sigma + \frac{1}{P}\frac{L_2\eta_M^2}{2\bar{\eta}}$$
$$+ \frac{2\eta_L\beta_1}{\beta_2}\frac{\|E_0\|_1}{T\bar{\eta}(1-\beta_2)} + \frac{2\eta_L\beta_1}{\beta_2}\frac{P-1}{P}\frac{L_\infty A_\infty\beta_2}{(1-\beta_2)\bar{\eta}}$$
$$+ \frac{P-1}{P}\frac{2\eta_L\beta_1}{\beta_2\bar{\eta}}\rho_1\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + 2\frac{P-1}{P}\frac{\eta_L}{\bar{\eta}}\left|1-\frac{\beta_1}{\beta_2}\right|\rho_1\sigma + \frac{P-1}{P}\frac{L_\infty\eta_L^2}{2\bar{\eta}}.$$

We can combine the momentum β_2 terms:

$$\frac{1}{P}\frac{L_2\eta_M^2}{2\bar{\eta}} \le \frac{1}{P}\frac{L_2A_2\eta_M}{2(1-\beta_2)\bar{\eta}} \le \frac{1}{P}\frac{L_2A_2\eta_M}{2(1-\beta_2)\bar{\eta}^2}\bar{\eta}$$

and

$$\frac{P-1}{P}\frac{L_\infty \eta_L^2}{2\bar{\eta}} \le \frac{P-1}{P}\frac{L_\infty \eta_L A_\infty}{2(1-\beta_2)\bar{\eta}} \le \frac{P-1}{P}\frac{L_\infty \eta_L A_\infty}{2(1-\beta_2)\bar{\eta}^2}\bar{\eta}.$$

Then, we can bound the initial norm term:

$$\frac{2\eta_M\beta_1}{\beta_2}\frac{\|E_0\|_{\text{nuc}}}{T\bar{\eta}(1-\beta_2)} + \frac{2\eta_L\beta_1}{\beta_2}\frac{\|E_0\|_1}{T\bar{\eta}(1-\beta_2)} \le \frac{2\beta_1}{\beta_2}\frac{\max\{\eta_L,\eta_M\}\|E_0\|_1}{T\bar{\eta}(1-\beta_2)}.$$

When $P = 1$ or $P = \infty$, only one of the terms appears, and the bound still holds true.

Next, we define the period-averaged noise and smoothness constants:

$$\begin{aligned}\bar{\rho} &:= \frac{\eta_M}{P\bar{\eta}}\rho_{\text{nuc}} + \frac{(P-1)\eta_L}{P\bar{\eta}}\rho_1,\\ \bar{L} &:= \frac{\eta_M A_2}{P\bar{\eta}^2}L_2 + \frac{(P-1)\eta_L A_\infty}{P\bar{\eta}^2}L_\infty.\end{aligned} \tag{9}$$

Employing the averaged constants, we further simplify the bound:

$$\min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} \le \frac{\Delta_0}{\bar{\eta}T} + \frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar{\eta}T(1-\beta_2)} + \frac{4\bar{L}\bar{\eta}}{(1-\beta_2)} + \frac{2\beta_1}{\beta_2}\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + 2\left|1-\frac{\beta_1}{\beta_2}\right|\bar{\rho}\sigma.$$

□

A.3 Proof of Optimal Parameters Corollary 1

Proof. In Theorem 1, we obtained the convergence bound of `LionMuon` algorithm under arbitrary parameters:

$$\min_i\{\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|]\} \le \frac{\Delta_0}{\bar{\eta}T} + \frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar{\eta}T(1-\beta_2)} + \frac{4\bar{L}\bar{\eta}}{(1-\beta_2)} + \frac{2\beta_1}{\beta_2}\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + 2\left|1-\frac{\beta_1}{\beta_2}\right|\bar{\rho}\sigma, \tag{10}$$

$$\begin{aligned}\mathbb{E}[\|\overline{\nabla} f(W_{i\cdot P})\|] &:= \frac{\left(\eta_M\cdot\mathbb{E}[\|\nabla f(W_{i\cdot P})\|_{\text{nuc}}] + \sum_{j=1}^{P-1}[\eta_L\cdot\mathbb{E}[\|\nabla f(W_{i\cdot P+j})\|_1]]\right)}{\eta_M + (P-1)\eta_L}\\ &\ge \min_j \mathbb{E}[\|\nabla f(W_{i\cdot P+j})\|_{\text{nuc}}].\end{aligned}$$

Fixed period $P \in (1,\infty)$**.** To achieve accuracy ε, we choose the optimal horizon T, momentums β_1, β_2, stepsizes η_L and $\eta_M = \alpha\eta_L$, whereas period P and stepsizes scale α are treated as hyperparameters.

First, we pick the smaller momentum $\beta_1 \le \beta_2$ close to β_2 to limit the last term in (10):

$$2\left|1-\frac{\beta_1}{\beta_2}\right|\bar{\rho}\sigma \le \frac{\varepsilon}{8} \quad\Longrightarrow\quad \beta_1 = \beta_2\cdot[\max\{1-\frac{\varepsilon}{16\bar{\rho}\sigma},0\},1].$$

Now, all ratios $\frac{\beta_1}{\beta_2}$ can be upper-bounded by 1. We continue with the noise term:

$$\frac{2\beta_1}{\beta_2}\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} \le 2\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} \le \frac{\varepsilon}{8} \quad\Longrightarrow 1-\beta_2 = \left(\frac{\varepsilon}{16\bar{\rho}\sigma}\right)^{\frac{\kappa}{\kappa-1}}.$$

To simplify the following smoothness term, we also satisfy the condition $1-\beta_2 \le 1$, i.e., $1-\beta_2 = \min\{\left(\frac{\varepsilon}{16\bar{\rho}\sigma}\right)^{\frac{\kappa}{\kappa-1}}, 1\}$. Next, we upper-bound the third term:

$$\frac{4\bar{L}\bar{\eta}}{(1-\beta_2)} = \frac{4\bar{L}(\frac{\alpha}{P}+\frac{P-1}{P})\eta_L}{(1-\beta_2)} \le \frac{\varepsilon}{8} \quad\Longrightarrow\quad \eta_L = \frac{\varepsilon(1-\beta_2)}{32(\frac{\alpha}{P}+\frac{P-1}{P})\bar{L}}, \eta_M = \alpha\eta_L.$$

To proceed to the second term, we note that the period-averaged stepsize $\bar{\eta} := \frac{\eta_M}{P} + \frac{\eta_L(P-1)}{P}$ can be lower-bounded by $\bar{\eta} \geq \frac{1}{P}\max\{\eta_M, \eta_L\} = \frac{\eta_L}{P}\max\{1, \alpha\}$ as a convex combination:

$$\frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar{\eta}T(1-\beta_2)} \leq \frac{2P\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\eta_{\max}T(1-\beta_2)} \leq \frac{2P\|E_0\|_1}{\bar{\eta}T(1-\beta_2)} \leq \frac{\varepsilon}{8} \quad \Longrightarrow$$

$$T \geq \frac{16P\|E_0\|_1}{(1-\beta_2)\varepsilon} = P \cdot \max\left\{\frac{(32\bar{\rho}\sigma)^{\frac{\kappa}{\kappa-1}}\|E_0\|_1}{\varepsilon^{\frac{2\kappa-1}{\kappa-1}}}, \frac{16\|E_0\|_1}{\varepsilon}\right\}.$$

Finally, we bound the first term:

$$\frac{\Delta_0}{\bar{\eta}T} \leq \frac{\varepsilon}{8} \quad \Longrightarrow \quad T \geq \frac{8\Delta_0}{\varepsilon\bar{\eta}} = 2^9 \cdot \bar{L}\Delta_0 \cdot \max\left\{\frac{(16\bar{\rho}\sigma)^{\frac{\kappa}{\kappa-1}}}{\varepsilon^{\frac{3\kappa-2}{\kappa-1}}}, \frac{1}{\varepsilon^2}\right\}.$$

The bound obtained in the previous term is an order of magnitude smaller than this bound due to the larger power of ε factor. Hence, we keep only the last bound:

$$T = O\left(\bar{L}\Delta_0 \cdot \max\left\{\frac{(\bar{\rho}\sigma)^{\frac{\kappa}{\kappa-1}}}{\varepsilon^{\frac{3\kappa-2}{\kappa-1}}}, \frac{1}{\varepsilon^2}\right\}\right).$$

Cases $P = 1$ and $P = \infty$. In these cases, the proof is identical with constants $\bar{L} = L_2, \bar{\rho} = \rho_{\text{nuc}}, A_{\max} = \eta_M$ or $\bar{L} = L_\infty, \bar{\rho} = \rho_1, A_{\max} = \eta_L$ until the stepsize pick. The considered stepsizes become $\bar{\eta} = \eta_M$ or $\bar{\eta} = \eta_L$.

□

A.4 Remark about the constants for dense matrices

In our proofs, we use the worst-case norm inequalities (2) which cause extra conservative factors in the obtained bound from Theorem 1. Fortunately, the gradients and update matrices during LLM training tend to have a dense structure, as we also observe in our experiments (Table 2). Thus, this case is worth a separate analysis.

We call an update matrix $U_t = \text{LMO}_{\|\cdot\|}(\hat{G}_\tau)$ *dense*, if we have an approximate equivalence:

$$\|U_t\|_2 \approx \alpha\|U_t\|_\infty \quad \text{for some constant } \alpha \lesssim \sqrt{mn}. \tag{11}$$

Now, we can estimate the refined constants A_2 and A_∞ in the proof of Theorem 1 at **Steps** 1 **and** 2.

Step 1: Refined analysis of the `Muon` Steps ($t \in S_{\text{muon}}$). For $t \in S_{\text{muon}}$, the update utilizes the spectral norm $\|\cdot\|_2$. To use Lemmas 1 and 2, we estimate the uniform upper bound constant A_2 such that $\max_{\tau \leq t}\{\eta_\tau\|U_\tau\|_2\} \leq A_2$ for all previous steps $\tau \leq t$:

- If $\tau \in S_{\text{muon}}$, then all updates $U_\tau = \text{LMO}_{\|\cdot\|_2}(\hat{G}_\tau)$ are bounded by $\|U_\tau\|_2 = 1$, and the stepsize is $\eta_\tau = \eta_M$.
- If $\tau \in S_{\text{lion}}$, then the updates $U_\tau = \text{LMO}_{\|\cdot\|_\infty}(\hat{G}_\tau)$ utilize the infinity norm LMO, yielding $\|U_\tau\|_\infty = 1$. Using the norm equality (11), we have $\|U_\tau\|_2 \approx \alpha$ and stepsize $\eta_\tau = \eta_L$.

Taking the maximum over these two cases for $P \in (1, \infty)$, we get $A_2 = \max\{\eta_M, \alpha \cdot \eta_L\}$. When $P = 1$, all τ steps belong only to S_{muon} and $A_2 = \eta_M$.

Step 2: Refined analysis of the `Lion` Steps ($t \in S_{\text{lion}}$). For $t \in S_{\text{lion}}$, the update utilizes the infinite norm $\|\cdot\|_\infty$. To use Lemmas 1 and 2, we estimate the uniform upper bound constant A_∞ such that $\max_{\tau \leq t}\{\eta_\tau\|U_\tau\|_\infty\} \leq A_\infty$ for all past steps $\tau \leq t$:

- If $\tau \in S_{\text{muon}}$, then all updates $U_\tau = \text{LMO}_{\|\cdot\|_2}(\hat{G}_\tau)$ are bounded by $\|U_\tau\|_\infty \approx \frac{1}{\alpha}\|U_\tau\|_2 = \frac{1}{\alpha}$ and the stepsize is $\eta_\tau = \eta_M$.
- If $\tau \in S_{\text{lion}}$, then the updates $U_\tau = \text{LMO}_{\|\cdot\|_\infty}(\hat{G}_\tau)$ utilize the infinity norm LMO, yielding $\|U_\tau\|_\infty = 1$ and stepsize $\eta_\tau = \eta_L$.

Taking the maximum over these two cases for $P \in (1, \infty)$, we get $A_\infty = \max(\frac{1}{\alpha}\eta_M, \eta_L)$. When $P = \infty$, all τ steps belong only to S_{lion} and $A_\infty = \eta_L$.

Refined interpolated smoothness. With new refined uniform constants A_2 and A_∞, we can similarly derive new interpolated smoothness from (9):

$$\bar{L} = \frac{\eta_M A_2}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L A_\infty}{P\bar{\eta}^2} L_\infty = \frac{\eta_M \max\{\eta_M, \alpha \cdot \eta_L\}}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L \max(\frac{1}{\alpha}\eta_M, \eta_L)}{P\bar{\eta}^2} L_\infty.$$

Next, we apply the scale $\eta_M/\eta_L = \alpha$ to equalize the different norms and get new smoothness:

$$\begin{aligned} \bar{L} &= \frac{\eta_M \max\{\eta_M, \alpha \cdot \eta_L\}}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L \max(\frac{1}{a}\eta_M, \eta_L)}{P\bar{\eta}^2} L_\infty \\ &= \frac{\eta_M \max\{\eta_M, \frac{\alpha}{\alpha}\eta_M\}}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L \max(\frac{\alpha}{\alpha}\eta_L, \eta_L)}{P\bar{\eta}^2} L_\infty \\ &\approx \frac{\eta_M^2}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L^2}{P\bar{\eta}^2} L_\infty. \end{aligned} \tag{12}$$

The refined smoothness (12) more naturally and smoothly interpolates between pure `Muon` L_2 and pure `Lion` L_∞ when going from $P = 1$ to $P = \infty$.

B Weight Decay Analysis

Notations. We denote the closed $\|\cdot\|$-norm ball of radius r by $B_{\|\cdot\|}(r) := \{S \in \mathbb{R}^{m\times n} : \|S\| \le r\}$ and rewrite LMO as $\text{LMO}_{B_{\|\cdot\|}(r)}(G) = \arg\min_{S \in B_{\|\cdot\|}(r)} \langle G, S\rangle$. We use the spectral norm LMO to calculate the matrix-sign operation $\text{LMO}_{B_{\|\cdot\|_2}(r)}(G) = -r \cdot \text{msign}(G)$ and the infinity norm LMO to calculate the element-wise sign $\text{LMO}_{B_{\|\cdot\|_\infty}(r)}(G) = -r \cdot \text{sign}(G)$.

Constrained optimization view. With these new notations, LMO update (1) with a weight decay $\lambda > 0$ can be restated as a Frank-Wolfe step:

$$W_{t+1} = W_t + \eta_t \,\text{LMO}_{B_{\|\cdot\|}(1)}(\hat{G}_t) - \eta_t \lambda W_t \quad \Leftrightarrow \quad W_{t+1} = (1-\eta_t\lambda)W_t + \eta_t\lambda \,\text{LMO}_{B_{\|\cdot\|}(1/\lambda)}(\hat{G}_t).$$

This Frank-Wolfe algorithm solves the *constrained* optimization problem (Chen et al., 2024; 2026a; Sfyraki & Wang, 2025):

$$\min_{W \in B_{\|\cdot\|}(\frac{1}{\lambda})} f(W).$$

As a convergence criterion, we use the Frank-Wolfe gap for a set $\mathcal{C} \subseteq \mathbb{R}^{m\times n}$:

$$\mathcal{G}_{\mathcal{C}}(W) := \max_{V \in \mathcal{C}} \langle V - W, -\nabla f(W)\rangle$$

which equals exactly zero at the KKT points of $\mathcal{C}$.

Our `LionMuon` iterates between working within the $B_{\|\cdot\|_2}(1/\lambda)$ ball at `Muon` iterations and within the larger $B_{\|\cdot\|_\infty}(1/\lambda)$ ball at `Lion` ones. Hence, our method preserves fast convergence to `Muon` inner KKT points, while also being able to reach `Lion` KKT points away from the `Muon` ball.

We generalize Theorem 1 to bound the minimal smaller Frank-Wolfe gap $\mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W)$ on a smaller set and obtain almost identical convergence bound (the differences are highlighted):

$$\begin{aligned} \min_t\{\lambda \cdot \mathbb{E}[\mathcal{G}_{B_{\|\cdot\|_2}(\frac{1}{\lambda})}(W_t)]\} \le \frac{\Delta_0}{\bar{\eta}T} + \frac{8\bar{L}\bar{\eta}}{\min\{(1-\beta_2), 1/\sqrt{mn}\}} + \frac{2\beta_1}{\beta_2}\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} \\ + 2\left|1 - \frac{\beta_1}{\beta_2}\right|\bar{\rho}\sigma + \frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar{\eta}T(1-\beta_2)}, \end{aligned}$$

where the *new smoothness* $\bar{L} := \frac{\eta_M\sqrt{mn}\cdot\eta_{\max}}{P\bar{\eta}^2} L_2 + \frac{(P-1)\eta_L\eta_{\max}}{P\bar{\eta}^2} L_\infty$ is applied. The full Theorem 2 with Corollary 2 about the optimal parameters are located below.

We extend the prior work (Sfyraki & Wang, 2025) which provides analysis of simple momentums-equipped LMO updates with weight decay and heavy-tailed noise. We consider a wider class of switching LMO updates and obtain better noise dependence for pure `Muon` and `Lion` in Corollary 2.

Key differences from the non-weight-decay case:

- **New gap metric.** First, we cannot apply all norm inequalities to the gaps in different sets. We can only guarantee that gap on a smaller $\|\cdot\|_2$-ball is lower than gap on a $\|\cdot\|_\infty$-ball. Thus, we do not bound the weighted gap in the bounds, but we still use learning rates scale to equalize the smoothness and noise constants.

  Second, due to switching between the balls, some matrices W_t can go out of the `Muon` ball, and the gap can become negative. Nevertheless, it is still an informative metric as both zero and negative gaps indicate that no direction towards the `Muon` ball will yield improvement.

- **New interpolation.** When matrix W_t goes out of the `Muon` ball during `Lion` steps, it may slow the convergence for next `Muon` steps. For this reason, a bit worse factors $\sqrt{mn}$ appear in the new weight decay bound and smoothness.

All other discussions about optimal parameters, learning rates scale and period remain the same.

B.1 Building-block lemmas

First, we prove the modified version of building-blocks lemmas.

Lemma 4 (`LionMuon` Descent Lemma with Weight Decay)**.** *Let the objective function f satisfy Assumption 1 with respect to a norm $\|\cdot\|$, and let $\|\cdot\|_\star$ be its dual norm. Then, for the update $W_{t+1} = (1-\lambda\eta_t)W_t + \lambda\eta_t U_t$ with $U_t = \mathrm{LMO}_{B_{\|\cdot\|}(1/\lambda)}(\hat{G}_t)$, momentums $M_t = \beta_2 M_{t-1} + (1-\beta_2)G_t$ and $\hat{G}_t = \beta_1 M_{t-1} + (1-\beta_1)G_t$, the following bound holds:*

$$f(W_{t+1}) \leq f(W_t) - \lambda\eta_t \cdot \mathcal{G}_{B_{\|\cdot\|}(1/\lambda)}(W_t) + \frac{2\eta_t\beta_1}{\beta_2}\|\nabla f(W_t) - M_t\|_\star + 2\eta_t\left|1 - \frac{\beta_1}{\beta_2}\right|\|\nabla f(W_t) - G_t\|_\star + 2LC^2\eta_t^2,$$

where the Frank-Wolfe gap is $\mathcal{G}_{B_{\|\cdot\|}(1/\lambda)}(W_t) := \max_{V \in B_{\|\cdot\|}(1/\lambda)}\langle V - W_t, -\nabla f(W_t)\rangle$, and update and intermediate matrices are confined to $\max\{\|W_t\|, \|U_t\|\} \leq C/\lambda$.

Proof. We begin by bounding $f(W_{t+1})$ using the smoothness Assumption 1:

$$\begin{aligned}
f(W_{t+1}) &= f(W_t + \lambda\eta_t(U_t - W_t)) \\
&\leq f(W_t) + \lambda\eta_t\langle\nabla f(W_t), U_t - W_t\rangle + \frac{L\lambda^2\eta_t^2}{2}\|U_t - W_t\|^2 \\
&\leq f(W_t) + \lambda\eta_t\langle\nabla f(W_t), U_t - W_t\rangle + \frac{L}{2}\lambda^2C^2\eta_t^2 \cdot \frac{4}{\lambda^2} \\
&= f(W_t) + \lambda\eta_t\langle\hat{G}_t, U_t - W_t\rangle + \lambda\eta_t\langle\nabla f(W_t) - \hat{G}_t, U_t - W_t\rangle + 2LC^2\eta_t^2.
\end{aligned}$$

We define $\hat{V}_t := \arg\max_{V \in B_{\|\cdot\|}(1/\lambda)}\langle V - W_t, -\nabla f(W_t)\rangle$ and continue:

$$\begin{aligned}
f(W_{t+1}) &= f(W_t) + \lambda\eta_t\langle\hat{G}_t, U_t - W_t\rangle + \lambda\eta_t\langle\nabla f(W_t) - \hat{G}_t, U_t - W_t\rangle + 2LC^2\eta_t^2 \\
&\leq f(W_t) + \lambda\eta_t\langle\hat{G}_t, \hat{V}_t - W_t\rangle + \lambda\eta_t\langle\nabla f(W_t) - \hat{G}_t, U_t - W_t\rangle + 2LC^2\eta_t^2 \\
&= f(W_t) + \lambda\eta_t\langle\hat{G}_t, \hat{V}_t - U_t\rangle + \lambda\eta_t\langle\nabla f(W_t), U_t - W_t\rangle + 2LC^2\eta_t^2 \\
&= f(W_t) + \lambda\eta_t\langle\nabla f(W_t), \hat{V}_t - W_t\rangle + \lambda\eta_t\langle\nabla f(W_t) - \hat{G}_t, U_t - \hat{V}_t\rangle + 2LC^2\eta_t^2 \\
&\leq f(W_t) - \lambda\eta_t\mathcal{G}_{B_{\|\cdot\|}(1/\lambda)}(W_t) + \lambda\eta_t\|\nabla f(W_t) - \hat{G}_t\|_\star\|U_t - \hat{V}_t\| + 2LC^2\eta_t^2 \\
&\leq f(W_t) - \lambda\eta_t\mathcal{G}_{B_{\|\cdot\|}(1/\lambda)}(W_t) + \lambda\|\nabla f(W_t) - \hat{G}_t\|_\star \cdot \frac{2\eta_t}{\lambda} + 2LC^2\eta_t^2.
\end{aligned}$$

We can switch to a bound using the main momentum M_t:

$$\begin{aligned}\|\nabla f(W_t) - \hat{G}_t\|_\star &= \|\nabla f(W_t) - M_t + M_t - \hat{G}_t\|_\star \\ &= \left\|\nabla f(W_t) - M_t + \left(1 - \frac{\beta_1}{\beta_2}\right)(M_t - G_t)\right\|_\star \\ &= \left\|\frac{\beta_1}{\beta_2}(\nabla f(W_t) - M_t) + \left(1 - \frac{\beta_1}{\beta_2}\right)(\nabla f(W_t) - G_t)\right\|_\star \\ &\leq \frac{\beta_1}{\beta_2}\|\nabla f(W_t) - M_t\|_\star + \left|1 - \frac{\beta_1}{\beta_2}\right| \|\nabla f(W_t) - G_t\|_\star.\end{aligned}$$

Finally, we yield the required bound:

$$\begin{aligned}f(W_{t+1}) \leq f(W_t) - \lambda\eta_t \cdot \mathcal{G}_{B_{\|\cdot\|}(1/\lambda)}(W_t) + \frac{2\eta_t\beta_1}{\beta_2}\|\nabla f(W_t) - M_t\|_\star \\ + 2\eta_t \left|1 - \frac{\beta_1}{\beta_2}\right| \|\nabla f(W_t) - G_t\|_\star + 2LC^2\eta_t^2.\end{aligned}$$

□

Lemma 5 (`LionMuon` Momentum Error Bound with Weight Decay)**.** *Let the objective function f and corrupting noise satisfy Assumptions 1, 2, 3 with norm $\|\cdot\|$ and let momentum M_τ be defined as $M_\tau = \beta_2 M_{\tau-1} + (1-\beta_2)G_\tau$. Then, for updates $W_{\tau+1} = (1-\lambda\eta_\tau)W_\tau + \lambda\eta_\tau U_\tau$ with $U_\tau = \mathrm{LMO}_{B_{\|\cdot\|}(1/\lambda)}(\hat{G}_\tau)$, the following bound holds:*

$$\mathbb{E}[\|E_t\|_\star] \leq \beta_2^t\|E_0\|_\star + \frac{2LA\beta_2}{1-\beta_2} + \rho_\star\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}},$$

where $E_t := \nabla f(W_t) - M_t$ and $\max_{\tau\leq t}\{\eta_\tau\|W_\tau\|, \eta_\tau\|U_\tau\|\} \leq A/\lambda$.

Proof. Using the momentum definition, we write down the recursive step:

$$\begin{aligned}E_t = \nabla f(W_t) - M_t &= \nabla f(W_t) - \big(\beta_2 M_{t-1} + (1-\beta_2)G_t\big) \\ &= \beta_2\nabla f(W_t) + (1-\beta_2)\nabla f(W_t) - \beta_2 M_{t-1} - (1-\beta_2)G_t \\ &= \beta_2\big(\nabla f(W_t) - M_{t-1}\big) + (1-\beta_2)\big(\nabla f(W_t) - G_t\big) \\ &= \beta_2\big(\nabla f(W_t) - \nabla f(W_{t-1}) + \nabla f(W_{t-1}) - M_{t-1}\big) + (1-\beta_2)\big(\nabla f(W_t) - G_t\big) \\ &= \beta_2\big(\nabla f(W_t) - \nabla f(W_{t-1})\big) + \beta_2\big(\nabla f(W_{t-1}) - M_{t-1}\big) + (1-\beta_2)\big(\nabla f(W_t) - G_t\big).\end{aligned}$$

Using the notations $S_t = \nabla f(W_t) - G_t$ and $R_t = \nabla f(W_t) - \nabla f(W_{t-1})$, we unroll the recursion:

$$E_t = \beta_2 E_{t-1} + (1-\beta_2)S_t + \beta_2 R_t = \beta_2^t E_0 + \sum_{j=0}^{t-1}\beta_2^j\big[(1-\beta_2)S_{t-j} + \beta_2 R_{t-j}\big].$$

By smoothness, we have:

$$\begin{aligned}\|R_{t-j}\|_\star &= \|\nabla f(W_{t-j}) - \nabla f(W_{t-j-1})\|_\star \\ &\leq L\|W_{t-j} - W_{t-j-1}\| = L\lambda\eta_{t-j-1}\|U_{t-j-1} - W_{t-j-1}\| \leq 2LA.\end{aligned}$$

We continue with the norm-equivalence Assumption 3 and Jensen's inequality for math expectation:

$$\begin{aligned}\mathbb{E}[\|E_t\|_\star] &\leq \beta_2^t\cdot\|E_0\|_\star + \mathbb{E}\left[\left\|\sum_{j=0}^{t-1}\beta_2^j\big[(1-\beta_2)S_{t-j} + \beta_2 R_{t-j}\big]\right\|_\star\right] \\ &\leq \beta_2^t\cdot\|E_0\|_\star + \sum_{j=0}^{t-1}\beta_2^{j+1}\mathbb{E}[\|R_{t-j}\|_\star] + \mathbb{E}\left[\left\|\sum_{j=0}^{t-1}\beta_2^j(1-\beta_2)S_{t-j}\right\|_\star\right] \\ &\leq \beta_2^t\cdot\|E_0\|_\star + \frac{2LA\beta_2}{1-\beta_2} + \rho_\star\left(\mathbb{E}\left[\left\|\sum_{j=0}^{t-1}\beta_2^j(1-\beta_2)S_{t-j}\right\|_F^\kappa\right]\right)^{1/\kappa}.\end{aligned}$$

We treat the sequence $\{\beta_2^j(1-\beta_2)\cdot S_{t-j}\}_{j=0}^{t-1}$ as a martingale difference sequence with $\sigma_j = \beta_2^j(1-\beta_2)\sigma$ and apply the batching lemma 3 :

$$\begin{aligned}\mathbb{E}\left[\left\|\sum_{j=0}^{t-1}\beta_2^j(1-\beta_2)S_{t-j}\right\|_F^{\kappa}\right] &= \sum_{j=0}^{t-1}\beta_2^{\kappa j}(1-\beta_2)^{\kappa}\sigma^{\kappa} \\ &\le \sigma^{\kappa}(1-\beta_2)^{\kappa}\sum_{j=0}^{t-1}\beta_2^{\kappa j} \\ &\le \frac{\sigma^{\kappa}(1-\beta_2)^{\kappa}}{1-\beta_2^{\kappa}}.\end{aligned}$$

Hence, we have the required bound:

$$\mathbb{E}[\|E_t\|_\star] \le \beta_2^t\cdot\|E_0\|_\star + \frac{2LA\beta_2}{1-\beta_2} + \rho_\star\sigma\cdot\frac{1-\beta_2}{(1-\beta_2^{\kappa})^{1/\kappa}}.$$

Since $0 < 1-\beta_2 \le 1-\beta_2^{\kappa}$, we further simplify:

$$\mathbb{E}[\|E_t\|_\star] \le \beta_2^t\cdot\|E_0\|_\star + \frac{2LA\beta_2}{1-\beta_2} + \rho_\star\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}}.$$

□

B.2 LionMuon Convergence Theorem with weight decay

Theorem 2 (Convergence of LionMuon, $\lambda > 0$)**.** *Let the objective function f satisfy Assumption 1 with respect to $\|\cdot\|_2$ with constant L_2 and with respect to $\|\cdot\|_\infty$ with constant L_∞. Let noise Assumptions 2 and 3 hold with noise constants σ, ρ_{nuc} and ρ_1. Fix a horizon T, period $P \in [1,\infty]$, weight decay λ, momentum parameters $\beta_1,\beta_2 \in [0,1)$ and learning rates η_M and η_L.*

Define the period-averaged learning rate, noise level and smoothness:

$$\bar\eta := \tfrac{\eta_M}{P} + \tfrac{(P-1)\eta_L}{P},\quad \bar\rho := \tfrac{\eta_M}{P\bar\eta}\rho_{nuc} + \tfrac{(P-1)\eta_L}{P\bar\eta}\rho_1,\quad \bar L := \tfrac{\eta_M\eta_{\max}C_2}{P\bar\eta^2}L_2 + \tfrac{(P-1)\eta_L\eta_{\max}}{P\bar\eta^2}L_\infty, \tag{13}$$

where $\eta_{\max} = \max\{\eta_M,\eta_L\}$ and $C_2 = \sqrt{mn}$ for intermediate $P \in (1,\infty)$ with the boundary cases $\eta_{\max} = \eta_M, C_2 = 1$ at $P = 1$ and $\eta_{\max} = \eta_L, C_2 = 1$ at $P = \infty$.

Then, our LionMuon algorithm starting with $\Delta_0 := f(W_0) - f_\star$, $E_0 = \nabla f(W_0) - M_0$ guarantees the following bound on the period-averaged Frank-Wolfe gap norm:

$$\begin{aligned}\min_t \mathbb{E}\big[\lambda\cdot\mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)\big] \le \frac{\Delta_0}{\bar\eta T} + \frac{8\bar L\bar\eta}{\min\{1-\beta_2,1/C_2\}} + \frac{2\beta_1}{\beta_2}\bar\rho\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} \\ + 2\left|1-\frac{\beta_1}{\beta_2}\right|\bar\rho\sigma + \frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar\eta T(1-\beta_2)},\end{aligned}$$

For $P = \infty$ (pure Lion), the same bound holds with the larger Frank-Wolfe gap $\mathcal{G}_{B_{\|\cdot\|_\infty}(1/\lambda)}(W_t)$.

Proof. We divide the iteration indices $t \in \{0,\dots,T-1\}$ into two disjoint sets: the set of Muon steps $S_{\text{muon}} = \{t \mid t \equiv 0 \pmod P\}$ and the set of Lion steps $S_{\text{lion}} = \{t \mid t \not\equiv 0 \pmod P\}$.

Step 1: Analysis of the Muon steps ($t \in S_{\text{muon}}$). For $t \in S_{\text{muon}}$, the update uses the spectral norm $\|\cdot\|_2$. To apply Lemmas 4 and 5, we need the bound C_2 such that $\max\{\|W_t\|_2,\|U_t\|_2\} \le C_2/\lambda$, and the uniform bound A_2 such that $\max_{\tau\le t}\{\eta_\tau\|W_\tau\|_2,\eta_\tau\|U_\tau\|_2\} \le A_2/\lambda$ for all past steps $\tau \le t$:

- If $\tau \in S_{\text{muon}}$, all updates $U_\tau = \mathrm{LMO}_{B_{\|\cdot\|_2}(1/\lambda)}(\hat G_\tau)$ are bounded by $\|U_\tau\|_2 = 1/\lambda$ with stepsize $\eta_\tau = \eta_M$.
- If $\tau \in S_{\text{lion}}$, the updates $U_\tau = \mathrm{LMO}_{B_{\|\cdot\|_\infty}(1/\lambda)}(\hat G_\tau)$ use the infinity-norm LMO, yielding $\|U_\tau\|_\infty = 1/\lambda$. By norm equivalence, we have $\|U_\tau\|_2 \le \sqrt{mn}/\lambda$ with stepsize $\eta_\tau = \eta_L$.

- When $P \in (1, \infty)$, all W_τ lie in the `Lion` ball $B_{\|\cdot\|_\infty}(1/\lambda)$, so we have $\|W_\tau\|_2 \leq \sqrt{mn}\,\|W_\tau\|_\infty \leq \sqrt{mn}/\lambda$ with alternating stepsizes $\eta_\tau \leq \max\{\eta_M, \eta_L\}$.
- When $P = 1$, all W_τ lie in the `Muon` ball $B_{\|\cdot\|_2}(1/\lambda)$, so we have $\|W_\tau\|_2 \leq 1/\lambda$ with single stepsize $\eta_\tau = \eta_M$.

Taking the maximum over these cases, we have $\eta_{\max} = \max\{\eta_M, \eta_L\}, C_2 = \sqrt{mn}, A_2 = C_2 \cdot \eta_{\max}$ if $P \in (1, \infty)$ and $\eta_{\max} = \eta_M, C_2 = A_2 = 1$ if $P = 1$ (no `Lion` steps).

Combining Lemmas 4 and 5 with C_2, A_2 and dual variance factor ρ_{nuc}, we bound the gap:

$$\begin{aligned}
\lambda\eta_M \cdot \mathbb{E}[\mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)] &\leq \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \frac{2\eta_M\beta_1}{\beta_2}\mathbb{E}[\|\nabla f(W_t) - M_t\|_{\text{nuc}}] \\
&\quad + 2\eta_M\left|1 - \frac{\beta_1}{\beta_2}\right|\mathbb{E}[\|\nabla f(W_t) - G_t\|_{\text{nuc}}] + 2L_2C_2^2\eta_M^2 \\
&\leq \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] \\
&\quad + \frac{2\eta_M\beta_1}{\beta_2}\left(\beta_2^t\|E_0\|_{\text{nuc}} + \frac{2L_2C_2\eta_{\max}\beta_2}{1-\beta_2} + \rho_{\text{nuc}}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}}\right) \\
&\quad + 2\eta_M\left|1 - \frac{\beta_1}{\beta_2}\right|\rho_{\text{nuc}}\sigma + 2L_2C_2^2\eta_M^2 \\
&=: \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \text{Error}_t^{\text{muon}}. \qquad (14)
\end{aligned}$$

Step 2: Analysis of the `Lion` steps ($t \in S_{\text{lion}}$). For $t \in S_{\text{lion}}$, the update uses the infinity norm $\|\cdot\|_\infty$. To apply Lemmas 4 and 5, we need the bound C_∞ such that $\max\{\|W_t\|_\infty, \|U_t\|_\infty\} \leq C_\infty/\lambda$, and the uniform bound A_∞ such that $\max_{\tau \leq t}\{\eta_\tau\|W_\tau\|_\infty, \eta_\tau\|U_\tau\|_\infty\} \leq A_\infty/\lambda$ for all past steps $\tau \leq t$:

- If $\tau \in S_{\text{muon}}$, all updates $U_\tau = \text{LMO}_{B_{\|\cdot\|_2}(1/\lambda)}(\hat{G}_\tau)$ satisfy $\|U_\tau\|_\infty \leq \|U_\tau\|_2 = 1/\lambda$ with stepsize $\eta_\tau = \eta_M$.
- If $\tau \in S_{\text{lion}}$, the updates $U_\tau = \text{LMO}_{B_{\|\cdot\|_\infty}(1/\lambda)}(\hat{G}_\tau)$ use the infinity-norm LMO, yielding $\|U_\tau\|_\infty = 1/\lambda$ with stepsize $\eta_\tau = \eta_L$.
- All steps W_τ lie in the ball $B_{\|\cdot\|_\infty}(1/\lambda)$, so we have $\|W_\tau\|_\infty \leq 1/\lambda$ with alternating stepsizes $\eta_\tau \leq \max\{\eta_M, \eta_L\}$ if $P \in (1, \infty)$ or single stepsize $\eta_\tau = \eta_L$ if $P = \infty$.

Taking the maximum, we get $\eta_{\max} = \max(\eta_M, \eta_L), C_\infty = 1, A_\infty = \eta_{\max}$ if $P \in (1, \infty)$ and $\eta_{\max} = \eta_L, C_\infty = A_\infty = 1$ if $P = \infty$ (no `Muon` steps).

Similarly combining Lemmas 4 and 5 with C_∞, A_∞ and dual variance factor ρ_1, we bound the Frank-Wolfe gap:

$$\begin{aligned}
\lambda\eta_L \cdot \mathbb{E}[\mathcal{G}_{B_{\|\cdot\|_\infty}(1/\lambda)}(W_t)] &\leq \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] \\
&\quad + \frac{2\eta_L\beta_1}{\beta_2}\left(\beta_2^t\|E_0\|_1 + \frac{2L_\infty\eta_{\max}\beta_2}{1-\beta_2} + \rho_1\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}}\right) \\
&\quad + 2\eta_L\left|1 - \frac{\beta_1}{\beta_2}\right|\rho_1\sigma + 2L_\infty\eta_L^2.
\end{aligned}$$

Since $\|X\|_\infty \leq \|X\|_2$ (2), we have an inclusion $B_{\|\cdot\|_2}(1/\lambda) \subseteq B_{\|\cdot\|_\infty}(1/\lambda)$. Hence, the maximum defining the Frank-Wolfe gap $\mathcal{G}_{B_{\|\cdot\|_\infty}(1/\lambda)}(W_t)$ over the $\|\cdot\|_\infty$-ball is taken over a larger set and we can safely lower-bound $\mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t) \leq \mathcal{G}_{B_{\|\cdot\|_\infty}(1/\lambda)}(W_t)$. The final bound is

$$\begin{aligned}
\lambda\eta_L \cdot \mathbb{E}[\mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)] &\leq \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] \\
&\quad + \frac{2\eta_L\beta_1}{\beta_2}\left(\beta_2^t\|E_0\|_1 + \frac{2L_\infty\eta_{\max}\beta_2}{1-\beta_2} + \rho_1\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}}\right) \\
&\quad + 2\eta_L\left|1 - \frac{\beta_1}{\beta_2}\right|\rho_1\sigma + 2L_\infty\eta_L^2 \\
&=: \mathbb{E}[f(W_t)] - \mathbb{E}[f(W_{t+1})] + \text{Error}_t^{\text{lion}}. \qquad (15)
\end{aligned}$$

Step 3: Telescoping sum. We sum the bounds for Muon (14) and Lion (15) steps over $t = 0, \dots, T-1$. The number of steps of each type are $|S_{\text{muon}}| = T/P$ and $|S_{\text{lion}}| = T(P-1)/P$:

$$\sum_{t=0}^{T-1} \left(\mathbf{1}_{t\in S_{\text{muon}}}\eta_M + \mathbf{1}_{t\in S_{\text{lion}}}\eta_L\right) \cdot \mathbb{E}[\lambda \cdot \mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)] \le f(W_0) - f_\star + \sum_{t\in S_{\text{muon}}} \text{Error}_t^{\text{muon}} + \sum_{t\in S_{\text{lion}}} \text{Error}_t^{\text{lion}}.$$

The coefficient on the left-hand side sums exactly to $T(\eta_M/P + \eta_L(P-1)/P) = T\bar{\eta}$, so we lower-bound it by $T\bar{\eta} \cdot \min_t \mathbb{E}[\lambda \cdot \mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)]$. On the right-hand side, we apply the geometric series bound $\sum_{t=0}^{T-1} \beta_2^t \le 1/(1-\beta_2)$ for the intermediate momentum errors. Grouping constants matched to $|S_{\text{muon}}|$ and $|S_{\text{lion}}|$ and dividing by $T\bar{\eta}$, we get:

$$\begin{aligned}
\min_t \mathbb{E}[\lambda \cdot \mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)] \le{}& \frac{\Delta_0}{\bar{\eta}T} + \frac{2\eta_M\beta_1}{\beta_2}\frac{\|E_0\|_{\text{nuc}}}{T\bar{\eta}(1-\beta_2)} \\
&+ \frac{1}{P}\frac{2\eta_M\beta_1}{\beta_2}\frac{2L_2C_2\eta_{\max}\beta_2}{(1-\beta_2)\bar{\eta}} \\
&+ \frac{1}{P}\frac{2\eta_M\beta_1}{\beta_2\bar{\eta}}\rho_{\text{nuc}}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + \frac{1}{P}\frac{2\eta_M}{\bar{\eta}}\left|1-\frac{\beta_1}{\beta_2}\right|\rho_{\text{nuc}}\sigma + \frac{1}{P}\frac{2L_2C_2^2\eta_M^2}{\bar{\eta}} \\
&+ \frac{2\eta_L\beta_1}{\beta_2}\frac{\|E_0\|_1}{T\bar{\eta}(1-\beta_2)} + \frac{2\eta_L\beta_1}{\beta_2}\frac{P-1}{P}\frac{2L_\infty\eta_{\max}\beta_2}{(1-\beta_2)\bar{\eta}} \\
&+ \frac{P-1}{P}\frac{2\eta_L\beta_1}{\beta_2\bar{\eta}}\rho_1\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + 2\frac{P-1}{P}\frac{\eta_L}{\bar{\eta}}\left|1-\frac{\beta_1}{\beta_2}\right|\rho_1\sigma \\
&+ \frac{P-1}{P}\frac{2L_\infty\eta_L^2}{\bar{\eta}}.
\end{aligned}$$

Next, we unite the momentum β_2 terms:

$$\frac{1}{P}\frac{2L_2C_2^2\eta_M^2}{\bar{\eta}} \le \frac{1}{P}\frac{2L_2C_2\eta_{\max}\eta_M}{\min\{1-\beta_2, 1/C_2\}\bar{\eta}} = \frac{1}{P}\frac{2\bar{\eta}\cdot L_2C_2\eta_{\max}\eta_M}{\min\{1-\beta_2, 1/C_2\}\bar{\eta}^2}$$

and

$$\frac{P-1}{P}\frac{2L_\infty\eta_L^2}{\bar{\eta}} \le \frac{P-1}{P}\frac{2L_\infty\eta_L\eta_{\max}}{\min\{1-\beta_2, 1/C_2\}\bar{\eta}} = \frac{P-1}{P}\frac{2\bar{\eta}\cdot L_\infty\eta_L\eta_{\max}}{\min\{1-\beta_2, 1/C_2\}\bar{\eta}^2}.$$

Then, we bound the initial-norm term:

$$\frac{2\eta_M\beta_1}{\beta_2}\frac{\|E_0\|_{\text{nuc}}}{T\bar{\eta}(1-\beta_2)} + \frac{2\eta_L\beta_1}{\beta_2}\frac{\|E_0\|_1}{T\bar{\eta}(1-\beta_2)} \le \frac{2\eta_{\max}\beta_1}{\beta_2}\frac{\|E_0\|_1}{T\bar{\eta}(1-\beta_2)}.$$

When $P = 1$ or $P = \infty$, only one of the two terms appears, and the bound still holds.

Next, we define the period-averaged noise level and smoothness:

$$\begin{aligned}
\bar{\rho} &:= \frac{\eta_M}{P\bar{\eta}}\rho_{\text{nuc}} + \frac{(P-1)\eta_L}{P\bar{\eta}}\rho_1, \\
\bar{L} &:= \frac{\eta_M\eta_{\max}C_2}{P\bar{\eta}^2}L_2 + \frac{(P-1)\eta_L\eta_{\max}}{P\bar{\eta}^2}L_\infty.
\end{aligned}$$

Using these constants, we further simplify:

$$\begin{aligned}
\min_t \mathbb{E}[\lambda \cdot \mathcal{G}_{B_{\|\cdot\|_2}(1/\lambda)}(W_t)] \le{}& \frac{\Delta_0}{\bar{\eta}T} + \frac{2\beta_1}{\beta_2}\frac{\eta_{\max}\|E_0\|_1}{\bar{\eta}T(1-\beta_2)} + \frac{8\bar{L}\bar{\eta}}{\min\{1-\beta_2, 1/C_2\}} \\
&+ \frac{2\beta_1}{\beta_2}\bar{\rho}\sigma(1-\beta_2)^{\frac{\kappa-1}{\kappa}} + 2\left|1-\frac{\beta_1}{\beta_2}\right|\bar{\rho}\sigma.
\end{aligned}$$

For the pure-Lion case $P = \infty$, the same bound holds for the larger Frank-Wolfe gap $\min_t \mathbb{E}[\lambda \cdot \mathcal{G}_{B_{\|\cdot\|_\infty}(1/\lambda)}(W_t)]$. □

B.3 LionMuon Optimal Parameters Corollary with weight decay

Corollary 2 (Optimal Parameters for LionMuon, $\lambda > 0$)**.** *Let the objective function f and the noise satisfy Assumptions 1, 2 and 3 with the period-averaged constants $\bar{L}$, σ and $\bar{\rho}$ defined in* (13)*.*

- *Fix a weight decay $\lambda > 0$, period $P \in (1, \infty)$ and learning rates scale $\alpha = \eta_M/\eta_L$.*

  To achieve accuracy $\min_t \mathbb{E}[\lambda \cdot \mathcal{G}_{B_{\|\cdot\|_2}(\frac{1}{\lambda})}(W_t)] \le \varepsilon$*, our* LionMuon *requires T iterations*

$$T = O\left(\bar{L}\Delta_0 \cdot \max\left\{\frac{(\bar{\rho}\sigma)^{\frac{\kappa}{\kappa-1}}}{\varepsilon^{\frac{3\kappa-2}{\kappa-1}}}, \frac{C_2}{\varepsilon^2}\right\}\right), \tag{16}$$

  with the optimal parameters:

$$1-\beta_2 = \min\{\left(\frac{\varepsilon}{16\bar{\rho}\sigma}\right)^{\frac{\kappa}{\kappa-1}}, \frac{1}{C_2}\}, \quad \beta_1 \in \beta_2 \cdot [\max\{1-\frac{\varepsilon}{16\bar{\rho}\sigma}, 0\}, 1] \quad \eta_L = \frac{\varepsilon(1-\beta_2)}{64(\frac{\alpha}{P} + \frac{P-1}{P}) \cdot \bar{L}}.$$

- *Pure* Muon *($P = 1$) and* Lion *($P = \infty$) keep the same momentums β_1, β_2, number of iterations T and only single learning rate $\eta_M = \frac{\varepsilon(1-\beta_2)}{64 \cdot L_2}$ or $\eta_L = \frac{\varepsilon(1-\beta_2)}{64 \cdot L_\infty}$.*
- *We can set single-EMA $\beta_1 = \beta_2$ to get optimal parameters for our* SignMuon *with weight decay.*

Proof. The proof exactly copies the proof of non-weight-decay Corollary 1 from Appendix A.3. The two main difference are new interpolated smoothness (13) from instead of (3) and extra condition on momentum $1 - \beta_2 \le \frac{1}{C_2}$.

□

C Estimated constants during training

We record the quantities of Assumptions 3 and 1 at every optimizer step of a 124M run with P=2 on each dataset, as the median over the 2D parameters, and report the mean over training in Table 2: the gradient-norm ratio $\alpha = \|G_t\|_1/\|G_t\|_{\text{nuc}}$ (6), the noise levels $\rho_{\text{nuc}} \approx \|G_t - M_t\|_{\text{nuc}}/\|G_t - M_t\|_{\text{F}}$ and $\rho_1 \approx \|G_t - M_t\|_1/\|G_t - M_t\|_{\text{F}}$ (Assumption 3, with momentum as a less noisy gradient estimate), and the smoothness constants $L_2 \approx \|G_{t+1} - G_t\|_{\text{nuc}}/\|W_{t+1} - W_t\|_2$ and $L_\infty \approx \|G_{t+1} - G_t\|_1/\|W_{t+1} - W_t\|_\infty$ (Assumption 1).

Table 2: Constants of the bound, measured during 124M training with P=2. At each step we take the median over the 2D parameters, and the table gives the mean over the run (100 records per run). L_2 and L_∞ are single-batch estimates and therefore noisier. The last two columns are the ratios the bound depends on.

	α	ρ_{nuc}	ρ_1	L_2	L_∞	$L_\infty/(\alpha^2 L_2)$	$\rho_1/(\alpha\rho_{\text{nuc}})$
FineWeb	67	13.6	894	0.79	857	0.24	0.98
WikiText-103	75	10.9	835	1.46	1312	0.16	1.02

Two things follow. The gradient ratio α falls quickly and then holds near 67 on FineWeb and 75 on WikiText-103, the same order as the ratio η_M/η_L that the sweep prefers (Appendix E). The scale the theory asks for is close to the scale that trains best. Both trade-off ratios are of order one. That is the regime in which the bound has an interior optimal period P^* rather than preferring P=1 or P=∞, which matches the flat optimum between P=2 and P=5 in Figure 2.

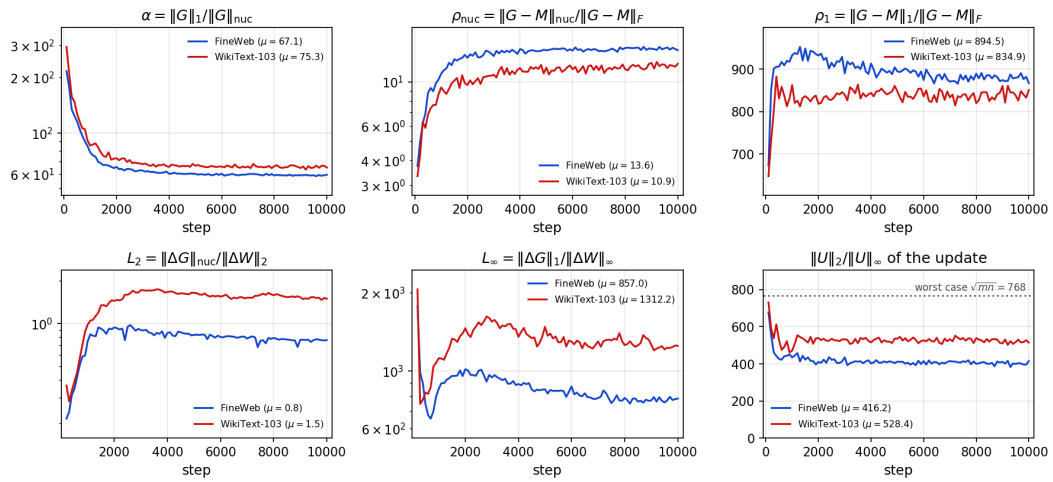

Figure 5: The same constants over the course of training, on 124M runs with P=2, with the mean over the run in each legend. The gradient ratio α falls over the first 2000 steps and then holds. The last panel is the update ratio $\|U\|_2/\|U\|_\infty$, which stays far below the worst case $\sqrt{mn}$ that the dense approximation would otherwise have to assume.

D Full Experimental Setup

Table 3: Special cases of `LionMuon`. Both conditions, on β_1, β_2 and on P, must hold.

Optimizer	Momentum	Period
`Signum` (Bernstein et al., 2018)	$\beta_1 = \beta_2$	$P = \infty$
`Lion` (Chen et al., 2023)	$\beta_1 \neq \beta_2$ (dual-EMA)	$P = \infty$
`Muon` (Jordan et al., 2024)	$\beta_1 = \beta_2$	$P = 1$
`SignMuon` **(this work)**	$\beta_1 = \beta_2$	any P
`LionMuon` **(this work)**	$\beta_1 \neq \beta_2$ (dual-EMA)	any P

D.1 Cost of a step

Table 1 lists what a step costs on top of the forward and backward pass.

Compute. Five Newton–Schulz iterations add about a tenth to `Muon`'s step at 124M. `LionMuon` pays them once in P steps.

Memory. `LionMuon` keeps one momentum buffer per matrix, like `Lion`, `Signum` and `Muon`, and half of `AdamW`.

Communication. Under data parallelism every method all-reduces the gradients, and most of that traffic hides behind the backward pass, because each bucket is sent as soon as it is ready. `Muon` then needs a second all-reduce that cannot hide: each matrix is orthogonalized on one device and the update is sent to the others (Liu et al., 2025). At 124M that is 170 MB per step in bf16, on top of 496 MB of gradients. A sign step is elementwise, so every device applies it to its own copy of the momentum and sends nothing. `LionMuon` therefore pays the second all-reduce only on `Muon` steps, $170/P$ MB per step on average.

`Dion` and `MuonBP` cut the same cost in other ways. `Dion` (Ahn et al., 2025) replaces the gradient all-reduce by an all-reduce of two low-rank factors with error feedback. At rank $\min(m, n)/4$ that is 113 MB, plus 155 MB for the parameters it does not factorize, 269 MB in total, or 382 MB at rank $\min(m, n)/2$. It is the least of any method here, but all of it waits for the backward pass to finish. `MuonBP` (Khaled et al., 2026) orthogonalizes column blocks locally and takes a full `Muon`

step every P-th iteration (P=5 in their paper), so it sends what LionMuon sends at the same P but runs Newton–Schulz on every step.

Replicated Muon. Muon can also avoid its all-reduce by orthogonalizing every matrix on every device. That trades the 170 MB for redundant Newton–Schulz work. At 124M on four GPUs a device then does 1.53×10^{12} FLOPs of Newton–Schulz per step instead of 0.38×10^{12}, on top of 3.31×10^{12} for forward and backward, about 31% more arithmetic. LionMuon divides both the transfer and the Newton–Schulz work by P, so it is cheaper than Muon either way, and the loss is the same in both versions.

Table 4: 124M on four GPUs with data parallelism, 150,000 steps, one seed per run. Loss is the best validation loss of the run, bytes are what one step sends (Table 1), and the last column is the first evaluation at or below Muon's own final loss (evaluations every 1,000 steps).

Method	η_M / η_L	Loss	MB per step	Steps to Muon's loss
AdamW	10^{-3}	3.407	496	137k
Muon	10^{-3}	3.418	667	147k
Muon	3×10^{-4}	3.467	667	never
Muon	3×10^{-3}	3.756 at 96k, stopped	667	never
LionMuon P=1	10^{-3}	3.391	667	133k
SignMuon P=2	3×10^{-3} / 10^{-4}	3.395	581	136k
LionMuon P=2	10^{-3} / 10^{-4}	3.376	581	126k
SignMuon P=5	10^{-2} / 3.3×10^{-5}	3.402	530	139k
LionMuon P=5	3×10^{-3} / 10^{-4}	3.377	530	127k
Signum	3×10^{-4}	3.429	496	never
Lion	10^{-4}	3.415	496	140k
Lion	3×10^{-4}	3.880, loss spikes	496	never
MuonBP P=5	10^{-3}	3.414	530	142k
Dion, rank 1/4	10^{-3}	3.429	269	never
Dion, rank 1/4	3×10^{-3}	3.451	269	never
Dion, rank 1/2	10^{-3}	3.411	382	141k

Table 5: What it costs to reach Muon's final loss, relative to Muon, at 124M on four GPUs. Steps are the first evaluation at or below that loss, and the other columns multiply them by the per-step costs of Table 1. Step times are given for the methods of Table 1. Dion at rank 1/4 and Signum never reach it.

	Steps	FLOPs	Bytes	Time, PCIe	Time, NVLink
Muon	1.00	1.00	1.00	1.00	1.00
LionMuon P=2	0.86	0.81	0.75	0.75	0.79
LionMuon P=5	0.86	**0.79**	0.69	**0.70**	**0.77**
SignMuon P=2	0.93	0.88	0.81	0.81	0.85
AdamW	0.93	0.84	0.69	0.70	0.79
Lion	0.95	0.85	0.71	–	–
MuonBP P=5	0.97	0.93	0.77	0.91	1.21
Dion, rank 1/2	0.96	0.88	**0.55**	–	–

Distributed setup. All distributed runs use PyTorch DDP on one node with four H200 GPUs (NVLink), one process per GPU and 8 sequences of 512 tokens per GPU, the global batch of the 124M grid. Each process holds a full replica of the model and the optimizer state, and DDP averages the gradients bucket by bucket during the backward pass. What happens after it differs by method. Muon: the 2D parameters are dealt round-robin to the four processes, each runs Newton–Schulz on its share, and one all-reduce of the updates gives every process all of them, as in Moonlight (Liu et al., 2025). LionMuon and SignMuon: the same on Muon steps, nothing on sign steps. MuonBP: four column blocks stand in for a 4-way tensor-parallel split, the block step is local, and the full step every P=5 iterations is Muon's. Dion: DDP's gradient all-reduce is off, each process keeps a local momentum and all-reduces two low-rank factors per matrix, plus one all-reduce for the parameters it does not factorize, which is the data-parallel mode of their paper (their Section 3.3).

Table 6: 355M on FineWeb, 8.2B tokens, every hyperparameter copied from 124M without retuning. Best validation loss, one seed per method.

Method	η_M / η_L	Val loss
`SignMuon` P=2	3×10^{-3} / 10^{-4}	**3.008**
`LionMuon` P=2	10^{-3} / 10^{-4}	3.020
`LionMuon` P=5	3×10^{-3} / 10^{-4}	3.023
`Muon`	10^{-3}	3.040
`Lion`	3×10^{-4}	3.050
`AdamW`	10^{-3}	3.064
`SignMuon` P=5	10^{-2} / 3.3×10^{-5}	3.072
`Signum`	3×10^{-4}	3.081

Algorithms 2 and 3 give both exactly as run. Both use `Muon`'s update scaling, so `Muon`'s learning rate transfers, and `AdamW` for the embedding and 1D parameters.

Both baselines were built for sharded models: `MuonBP` for tensor parallelism and FSDP, where `Muon` has to all-gather the shards of a matrix first (their Section 2.2), and `Dion` for FSDP and tensor parallelism, with the data-parallel sync as an option. Neither paper measures the saving end to end under data parallelism. `MuonBP` reports throughput under tensor parallelism, and `Dion` simulates step times on one GPU without communication (their Figure 1). One node under data parallelism is where we can count every byte and time every method under the same load, so that is what we use.

Algorithm 2 `MuonBP` for a single 2D parameter $W \in \mathbb{R}^{m \times n}$ under data parallelism

Require: Period P, number of column blocks B, learning rate η, momentum β, weight decay λ, Newton–Schulz steps K_{NS}, and $c(A) := 0.2\sqrt{\max(\mathrm{rows}(A), \mathrm{cols}(A))}$
for $t = 0, 1, \ldots, T-1$ **do**
  $G_t = \nabla_W \mathcal{L}_t$ ▷ Gradient, averaged over devices by the framework
  $M_t = \beta M_{t-1} + (1-\beta)\, G_t$ ▷ Momentum, replicated on every device
  if $t \bmod P = 0$ **then**
    $U_t = c(M_t)\, \mathrm{NS}_{K_{\mathrm{NS}}}(M_t)$ ▷ Full step, each device orthogonalizes its share
    all-reduce U_t ▷ The only communication of the optimizer
  else
    split M_t into column blocks $M_t^{(1)}, \ldots, M_t^{(B)}$
    $U_t = \big[\, c(M_t^{(1)})\, \mathrm{NS}_{K_{\mathrm{NS}}}(M_t^{(1)}) \;\cdots\; c(M_t^{(B)})\, \mathrm{NS}_{K_{\mathrm{NS}}}(M_t^{(B)}) \,\big]$ ▷ Block step, local
  end if
  $W_{t+1} = W_t - \eta \left(U_t + \lambda W_t\right)$
end for

E Hyperparameter Tuning

Every number in the main text comes from a learning rate chosen on the *full* training horizon: each cell of the sweep is a complete 64,000-step run of the same model, data and schedule as the headline runs (12 layers, width 768, batch 32×512 tokens, cosine schedule, evaluation every 500 steps), and a cell's score is the best validation loss it reaches. We tune two quantities: the spectral-step learning rate η_M and the ratio $\alpha = \eta_M/\eta_L$ to the sign-step learning rate, which is the pair the theory in Section 4.3 identifies. For pure `Muon` (P=1) and `Signum` (P=∞) there is only one rate to sweep.

Grid and pruning. η_M is swept over 3×10^{-4} to 10^{-1} and α over 1 to 3000, both in half-decade steps, with the swept band following the optimum as it moves with P. Running every pair at full length would be wasteful, so a cell is stopped early once it is clearly out of the running: from 34 finished curves, no eventual winner was ever more than 0.10 behind the best cell of its own (P, β) group at 6,000 steps or 0.20 behind at 16,000, and we stop cells that exceed those margins. Tables 8 and 9 report how many cells ran to completion and how many were stopped this way; a stopped neighbour is written "cut".

Algorithm 3 `Dion` for a single 2D parameter $W \in \mathbb{R}^{m \times n}$ on D devices

Require: Rank r, error-feedback rate γ, learning rate η, weight decay λ, $M_{-1}^{(d)} = 0$ on every device d, $V_{-1} \in \mathbb{R}^{n \times r}$ random with orthonormal columns and the same on every device, and $c := 0.2\sqrt{\max(m, n)}$

for $t = 0, 1, \ldots, T-1$ **do**

$G_t^{(d)} = \nabla_W \mathcal{L}_t^{(d)}$ ▷ Local gradient, the framework's gradient all-reduce is off

$M_t^{(d)} = M_{t-1}^{(d)} + G_t^{(d)}$ ▷ Local momentum, accumulated without decay

$P_t = \frac{1}{D} \sum_d M_t^{(d)} V_{t-1}$ ▷ all-reduce of an $m \times r$ factor

$U_t = \mathrm{QR}(P_t)$ ▷ Orthonormal columns, one power iteration

$R_t^{(d)} = M_t^{(d)\top} U_t, \quad R_t = \frac{1}{D} \sum_d R_t^{(d)}$ ▷ all-reduce of an $n \times r$ factor

$M_t^{(d)} \leftarrow M_t^{(d)} - \gamma\, U_t R_t^{(d)\top}$ ▷ Error feedback, local

$V_t = R_t \,\mathrm{diag}\big(\|R_t e_1\|_2, \ldots, \|R_t e_r\|_2\big)^{-1}$ ▷ Columns normalized

$W_{t+1} = W_t - \eta \left(c\, U_t V_t^\top + \lambda W_t\right)$

end for

Table 7: Full experimental configuration.

Model architecture	
Number of layers	12
Number of heads	12
Embedding dim	768
Sequence length	512
Vocabulary size	50,304 (GPT-2 BPE)
Architectures	GPT-2 base
Training schedule	
Iterations	64,000
Warmup steps	3,000
Batch size	32
Gradient accumulation	1
LR scheduler	cosine
Weight decay	0.1
Gradient clipping	0.5
Eval interval	every 500 steps
Optimizer-specific	
Newton–Schulz steps	$K_{\mathrm{NS}} = 5$
NS scaling	$0.2\sqrt{\max(m, n)}$
`AdamW` (β_1, β_2)	$(0.8, 0.999)$
`Lion` (β_1, β_2)	$(0.9, 0.99)$
`Signum` momentum	$\beta = 0.9$ (EMA)
`Muon` momentum	$\beta = 0.9$ (EMA), no Nesterov
`SignMuon` momentum	$\beta = 0.9$ (EMA), no Nesterov
`LionMuon` (β_1, β_2)	$(0.9, 0.99)$
1D-param backup (hybrids)	`AdamW` with $\eta_{\mathrm{1D}} = 10^{-3}$
Distributed run (Section 5.3)	
Framework	PyTorch DDP, NCCL, one process per GPU
Hardware	one node, 4× H200 (NVLink)
Per-GPU batch	8 sequences × 512 tokens (global batch 32)
Iterations / warmup	150,000 / 7,500
Eval	every 1,000 steps, 32 batches
`MuonBP`	4 column blocks, full step every 5, momentum 0.9
`Dion`	rank $\min(m, n)/4$, error feedback 0.05, gradient all-reduce off

Selected cells. For each method we take the best cell and use it verbatim: for the three-seed runs at 124M, and, without any further tuning, for 355M. The tables give the chosen pair together with the

losses of the neighbouring η_M values, which is what makes the optimum interior rather than an edge of the grid. Two observations transfer to practice. The ratio α grows with the period (10, 30, 100 at P=2, 5, 20 for `LionMuon` on both datasets), so a practitioner who changes P should retune α and can leave η_M near its `Muon` value. And the sign-step rate itself stays in a narrow band ($\eta_L \approx 10^{-4}$ on FineWeb), so the ratio, not the second rate, is the knob that matters.

Why not a cheap pilot. An earlier version of this work chose learning rates on 3,000-step pilot runs and transferred them. We found that this mis-ranks the candidates: the cell a pilot prefers is not the cell that wins at 64,000 steps, and the gap between the pilot's choice and the full-horizon choice is larger than the differences we report between methods. Every tuned number in this paper therefore comes from the full-horizon sweep; an earlier pilot-tuned grid over three datasets and two architectures gave the same ranking of methods, but we do not report it here.

Table 8: The tuning sweep at 124M on FineWeb, every cell a full 64,000-step run. "Cells" counts the runs that went to the end plus the ones the pruning rule stopped early. The last column is the loss of the chosen cell, with the losses at the next lower and next higher η_M (same α) in brackets.

method	η_M swept	α swept	cells	chosen (η_M, α)	loss (next η_M down / up)
`Muon` (P=1)	0.0003–0.003	–	3+0	0.001	**3.528** (3.596 / 3.534)
`SignMuon` P=2	0.001–0.01	3–300	8+3	(0.003, 30)	**3.511** (3.529 / cut)
`SignMuon` P=5	0.001–0.03	3–1000	13+3	(0.01, 300)	**3.508** (3.534 / cut)
`SignMuon` P=20	0.003–0.1	10–1000	8+4	(0.01, 100)	**3.540** (3.581 / 3.551)
`LionMuon` P=1	0.0003–0.003	–	2+1	0.001	**3.511** (3.540 / cut)
`LionMuon` P=2	0.0003–0.01	1–300	12+2	(0.001, 10)	**3.501** (3.538 / 3.507)
`LionMuon` P=5	0.001–0.01	3–300	9+2	(0.003, 30)	**3.500** (3.523 / cut)
`LionMuon` P=20	0.001–0.03	10–300	8+3	(0.01, 100)	**3.511** (3.538 / cut)

Table 9: The same sweep on WikiText-103.

method	η_M swept	α swept	cells	chosen (η_M, α)	loss (next η_M down / up)
`Muon` (P=1)	0.001–0.01	–	2+1	0.003	**2.838** (2.884 / cut)
`SignMuon` P=2	0.0003–0.03	1–3000	12+5	(0.003, 3)	**2.835** (2.885 / cut)
`SignMuon` P=5	0.0001–0.03	1–100	10+5	(0.01, 30)	**2.842** (2.890 / cut)
`SignMuon` P=20	0.001–0.1	3–3000	9+8	(0.03, 100)	**2.859** (2.902 / cut)
`LionMuon` P=1	0.001–0.01	–	2+1	0.003	**2.826** (2.861 / cut)
`LionMuon` P=2	0.0003–0.01	1–3000	8+8	(0.003, 10)	**2.827** (2.881 / cut)
`LionMuon` P=5	0.0003–0.03	1–100	8+5	(0.01, 30)	**2.825** (2.872 / cut)
`LionMuon` P=20	0.001–0.1	3–300	9+5	(0.03, 100)	**2.839** (2.877 / cut)

E.1 Momentum

The momentum pair was tuned on FineWeb at 124M with P=2. For each (β_1, β_2) the rate η_M and the ratio α were swept as above, and Table 10 reports the best cell. The diagonal $\beta_1 = \beta_2$ is `SignMuon`, the rest is `LionMuon`. Every off-diagonal pair beats every diagonal one, and $(0.9, 0.99)$ is best, so `LionMuon` and `Lion` use it. The diagonal is flat within 0.002, so `SignMuon` and `Muon` keep the usual 0.9. `AdamW` uses $(0.8, 0.999)$, the pair tuned by the benchmark, which also beat $(0.9, 0.95)$ in our check.

Table 10: Loss at 124M on FineWeb with P=2 for each momentum pair, after tuning the rate and the ratio for that pair. The diagonal is `SignMuon`, the rest is `LionMuon`.

$\beta_1 \backslash \beta_2$	0.9	0.95	0.99
0.9	3.511	3.508	**3.501**
0.95		3.509	3.502
0.99			3.510

E.2 Two ablations: a random period, and Lion on the 1D parameters

Two questions came up in earlier reviews. Does the period have to be fixed, and do the 1D parameters need AdamW? For the first we replace the fixed period by a Muon step drawn with probability $1/P$ at every iteration, so the mean spacing is the same but the steps are irregular. For the second we give the biases and norm gains the same Lion step as the matrices take on sign steps, at the sign-step rate and betas, instead of AdamW. Both run at 124M on FineWeb for $64{,}000$ steps at the tuned rates of Appendix E, with three seeds (Table 11). The random period changes nothing. Lion on the 1D parameters is clearly worse, so the AdamW fallback is worth its second buffer for those few parameters.

Table 11: Loss at 124M on FineWeb, $64{,}000$ steps, mean over three seeds with the spread in brackets. The first column is the tuned setting of Section 5, the other two are the ablations.

	Fixed period	Random period	Lion on 1D parameters
LionMuon P=2	3.498 (0.002)	3.499 (0.003)	3.535 (0.001)
LionMuon P=5	3.498 (0.002)	3.499 (0.003)	3.538 (0.002)

F Heavy-ball versus EMA momentum in SignMuon

Our initial SignMuon implementation followed the heavy-ball convention $M_t = \mu M_{t-1} + G_t$ used in the llm-baselines codebase of (Semenov et al., 2025), but we found that it requires careful joint tuning of μ and learning rate. Switching to the Lion-style EMA $M_t = \beta M_{t-1} + (1-\beta) G_t$ proved to be much more robust across the grid, even though the two updates are equivalent up to a constant rescaling of η.

Proof. We consider two momentum recursions starting from $M_{-1} = M'_{-1} = 0$:

$$\text{(HB)} \quad M_t = \mu\, M_{t-1} + G_t, \qquad \text{(EMA)} \quad M'_t = \beta\, M'_{t-1} + (1-\beta)\, G_t.$$

With $\mu = \beta$, induction gives $M'_t = (1-\beta)\, M_t$ for all t.

Base case: $M'_{-1} = (1-\beta) M_{-1} = 0$.

Inductive step: assuming $M'_{t-1} = (1-\beta) M_{t-1}$, we have

$$M'_t = \beta M'_{t-1} + (1-\beta) G_t = \beta(1-\beta) M_{t-1} + (1-\beta) G_t = (1-\beta)\big(\beta M_{t-1} + G_t\big) = (1-\beta) M_t.$$

Since both msign and sign are positively homogeneous of degree zero (i.e., $\mathrm{msign}(\alpha X) = \mathrm{msign}(X)$ for any $\alpha > 0$), the update directions $\mathrm{msign}(M_t)$ and $\mathrm{msign}(M'_t)$ are identical, and similarly for sign. The two parametrizations therefore generate the same iterate sequence under

$$\eta_{\text{HB}} \;=\; (1-\beta)\, \eta_{\text{EMA}}.$$

□

The practical consequence is that the LR ranges that "feel right" under the two parametrizations differ by a factor of $1/(1-\beta) \approx 100$ at $\beta = 0.99$. This difference explains why the EMA form is more forgiving on a fixed grid: a typical LR around 10^{-3}–10^{-4} already sits in its useful range, whereas the corresponding heavy-ball LR is around 10^{-5}–10^{-6} and easy to miss when sweeping.