MUONMIX: A MIXED-LMO BLOCK-PERIODIC ALGORITHM FOR EFFICIENT DISTRIBUTED MUON TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

The Muon optimizer accelerates language model pretraining by applying Newton–Schulz orthogonalization to hidden weight matrices, but scaling it to distributed training requires gathering gradient shards across devices at every step. MuonBP reduces this cost through block-periodic orthogonalization, but applying the spectral-norm linear minimization oracle (LMO) to local shards can introduce a partition-dependent approximation error. Since max-norm LMO updates do not suffer from this error, we propose MuonMix, a mixed-LMO block-periodic algorithm for distributed Muon training that assigns each hidden weight matrix to either the spectral-norm LMO or the max-norm LMO under a shared synchronization period p, based on a per-layer selection heuristic motivated by a layerwise convergence upper bound. Layers assigned to the max-norm LMO require no gather or scatter operations and introduce no block approximation error. We evaluate the proposed algorithm on language model pretraining at 160M, 960M, and 1.2B parameters on FineWeb, and show that it improves throughput by 3.0%, 4.8%, and 6.5% over MuonBP and achieves faster early convergence across all model sizes, and reaches target validation losses up to 7.1% faster in wall-clock time.

1 INTRODUCTION

Training large language models is compute-intensive, making optimizer choice critical for reducing time and cost. Large-scale training partitions model states and computation across devices using the zero redundancy optimizer (ZeRO) (Rajbhandari et al., 2020), fully sharded data parallelism (FSDP) (Zhao et al., 2023), and tensor parallelism (TP) (Shoeybi et al., 2019). The optimizer must make meaningful progress per step while keeping per-step communication and computation costs low. Geometry-aware optimizers have recently emerged as a promising approach. Unlike Euclidean-based stochastic gradient descent (SGD) (Bottou, 2010) and AdamW (Loshchilov & Hutter, 2017), Muon (Jordan et al., 2024b) applies Newton–Schulz (NS) orthogonalization to each hidden-layer momentum matrix to match the spectral geometry of model parameters. Recent work (Pethick et al., 2025; Shulgin et al., 2025; Frans et al., 2025; Li et al., 2025; Jiang et al., 2026; Gruntkowska et al., 2025; Tang et al., 2025) reports improved training speed and stability over Euclidean-based updates. However, NS is not elementwise: under TP or FSDP, each weight matrix is partitioned across devices, so computing its orthogonalized update requires gathering all shards before applying NS. This adds 5%–10% communication overhead over AdamW (Khaled et al., 2025). The overhead grows with model parallelism and becomes a dominant bottleneck in large-scale distributed training.

While Muon-type spectral updates can provide stronger single-machine optimization behavior, they may incur larger approximation errors with partitioning and local updates. Elementwise updates such as AdamW (Loshchilov & Hutter, 2017) or Signum (Bernstein et al., 2018) may have weaker single-machine behavior under the same linear minimization oracle (LMO)-based viewpoint, but are more compatible with partitioning: local shard updates concatenate to the full-matrix direction. Thus, distributed training trades update-geometry behavior against error from partitioning and local computation. Existing distributed Muon variants, including Moonlight (Liu et al., 2025) and MuonBP (Khaled et al., 2025), do not explicitly account for this tradeoff and typically apply the same spectral-norm LMO to all partitioned matrix layers. This uniform choice may not be optimal: some parameter groups may benefit more from spectral updates, while others may favor elementwise updates that avoid sharding-induced directional error.

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.

Our approach. We propose MuonMix, a mixed-LMO block-periodic algorithm for distributed Muon training that assigns LMO geometries to parameter groups to balance single-machine optimization behavior, communication cost, and local-update approximation error. Under a shared synchronization period p, each hidden weight matrix uses one of two LMOs.[1] This lets each layer use the norm best matching its geometry while reducing MuonBP's communication cost and block approximation error. Layers in the set $\mathcal{N}$ use the spectral-norm LMO as in MuonBP, while layers in $\mathcal{S}$ use the max-norm LMO, implemented as Signum in both convergence analysis and main experiments.[2] Layers in $\mathcal{S}$ require no all-gather or scatter for orthogonalization, removing NS computation and communication on those layers. A profiling heuristic motivated by the convergence upper bound selects layers. For each layer s, a *descent advantage value* measures the difference in the *per-step descent lower bounds* evaluated as noiseless surrogates on a fixed reference trajectory. A positive value assigns s to the max-norm LMO. Our contribution can be summarized as follows.

- We identify a fundamental limitation of MuonBP. Applying the spectral-norm LMO to each local shard can introduce an approximation error with a partition-dependent worst-case bound. This error is absent for max-norm LMO.
- We propose MuonMix (Algorithm 1) that assigns each hidden weight matrix to either the spectral-norm LMO or the max-norm LMO. Layers assigned to the max-norm LMO incur no block approximation error, and require no all-gather or scatter at the orthogonalization step on either block or full iterations.
- We establish a convergence upper bound for MuonMix under layerwise smoothness (Theorem 1). Unlike the spectral-norm LMO terms in the upper bound, the contribution from max-norm LMO does not depend on the number of shards.
- We derive a per-layer LMO selection rule from the convergence bound. For each layer, we define a scalar descent advantage that compares the per-step decrease attained by the two LMOs on that layer. The sign of this value determines the LMO for the layer, and it is estimated directly from gradient differences and parameter updates observed during a short warm-up phase.
- Empirically, we show that MuonMix outperforms MuonBP in transformer language model pretraining on FineWeb across three model sizes, 160M, 960M, and 1.2B parameters. Profiling experiment assigns the query and key projection matrices to the max-norm LMO. MuonMix improves throughput by 3.0%, 4.8%, and 6.5% over MuonBP respectively, and reaches target validation losses up to 7.1% faster in wall-clock time.

2 Related Work and Preliminaries

2.1 Related Work

Geometry-aware optimization. Standard optimizers such as SGD (Bottou, 2010), Adagrad (Duchi et al., 2011), Adam (Kingma & Ba, 2014), and AdamW (Loshchilov & Hutter, 2017) are defined in the standard Euclidean parameter space. In particular, adaptive methods such as Adagrad, Adam, and AdamW use elementwise scaling based on diagonal second-moment estimates, and therefore do not explicitly exploit the matrix geometry in the network. Shampoo (Gupta et al., 2018) and K-FAC (Martens & Grosse, 2015) introduce structured preconditioning methods that better capture matrix geometry, but require higher memory and computation. SOAP (Vyas et al., 2024) improves the practical efficiency of Shampoo by using Adam-style updates in the preconditioner's eigenbasis, but its per-step cost remains higher than standard first-order methods. The Muon optimizer (Jordan et al., 2024b) is a lightweight geometry-aware optimizer for 2D hidden-layer parameters. It applies Newton-Schulz orthogonalization to the SGD-momentum update, producing an approximately semi-orthogonal direction motivated by spectral-norm geometry.

LMO-based optimization. The LMO originates from the Frank-Wolfe framework for constrained optimization (Jaggi, 2013). Recent work has extended norm-constrained LMOs to unconstrained deep learning, showing that the choice of norm induces different update geometries within a common framework (Pethick et al., 2025). For a matrix $\mathbf{B} \in \mathbb{R}^{a\times b}$, let $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ be its SVD with

[1] MuonMix can be extended to any number of layer groups with different LMOs.

[2] We focus on Signum in this paper, but it can be extended to other norm geometries and corresponding LMOs.

$r = \min(a, b)$, including singular vectors for zero singular values. $\mathbf{U} \in \mathbb{R}^{a\times r}$ and $\mathbf{V} \in \mathbb{R}^{b\times r}$ contain the left and right singular vectors and $\mathbf{\Sigma} \in \mathbb{R}^{r\times r}$ contains the singular values. The spectral-norm LMO returns $-\mathbf{U}\mathbf{V}^\top$, the idealized Muon update direction. Under the max-ℓ_1 norm pair, the LMO returns $-\operatorname{sign}(\mathbf{B})$, corresponding to Signum (Bernstein et al., 2018). This geometry is also more aligned with AdamW, whose adaptive rescaling is defined elementwise rather than at the matrix level.

Distributed optimization. Local SGD (Stich, 2018) reduces communication overhead by allowing each worker to perform p consecutive local steps before synchronization. There have been several recent advances for distributed variants of Muon. Moonlight (Liu et al., 2025) introduces weight decay and per-parameter update scaling for distributed Muon training, and demonstrates its scalability with a memory-optimal and communication-efficient distributed implementation. Dion (Ahn et al., 2025) replaces the NS iteration with amortized power iteration on a momentum buffer that works directly on sharded matrices. DeMuon (He et al., 2025) extends Muon to decentralized optimization over arbitrary communication graphs via gradient tracking, and establishes iteration complexity guarantees matching centralized bounds. FedMuon (Takezawa et al., 2025) extends bias-corrected LMO-based optimization to the federated setting. MuonBP (Khaled et al., 2025) reduces communication through block-periodic orthogonalization, but uses the orthogonal LMO for all hidden local matrices during local steps. This can introduce partition-induced error with a norm-comparison factor $\sqrt{M}$, whereas elementwise updates such as Signum or AdamW are unaffected by partitioning. This motivates our mixed-LMO framework, in which each layer selects the most appropriate LMO for its geometry.

2.2 Preliminaries

Problem setup. We consider the stochastic optimization problem $\min_{W\in\mathcal{W}} f(W) = \mathbb{E}_{\xi\sim\mathcal{D}}\big[F(W;\xi)\big]$, where $W \in \mathcal{W}$ denotes the model parameters, $\xi \sim \mathcal{D}$ is a random mini-batch, and $F(\cdot;\xi)$ is the stochastic loss associated with ξ. At each step t, fresh mini-batches are used to form the stochastic gradient. After the standard training collectives, each device holds its shard of this gradient. Devices holding different parameter shards need not use independent mini-batches.

Notation. We consider a transformer model with H hidden weight matrices. We partition these matrices into two groups: the set $\mathcal{N}$ of matrices that are described by spectral geometry, and the set $\mathcal{S}$ of matrices whose gradient geometry is better captured by max norm. We denote a weight matrix as $\mathbf{W} \in \mathbb{R}^{a\times b}$ and its mini-batch stochastic gradient as $\mathbf{G} \in \mathbb{R}^{a\times b}$. We let $\|\mathbf{G}\|_{\mathrm{sp}}$ be the spectral norm (largest singular value), $\|\mathbf{G}\|_*^{\mathrm{nuc}}$ be the nuclear norm (sum of singular values), $\|\mathbf{G}\|_\infty$ be the max norm and $\|\mathbf{G}\|_*^1$ be the ℓ_1 norm. The SVD of the update matrix produced by SGD-momentum is denoted $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$. In the distributed training setup, we use $M = rc$ devices. The matrices are partitioned into r row blocks and c column blocks, and each device processes one such partition. Let p be the synchronization period, with $p-1$ local block steps between full steps. The learning rate is η and the momentum coefficient is μ or β (see later). For layer ℓ, $\mathbf{G}_t^\ell$, $\mathbf{B}_t^\ell$, and $\mathbf{x}_t^\ell$ denote the full layer gradient, momentum, and model weights at iteration t, and $[\mathbf{G}_t^\ell]^{(m)}$, $[\mathbf{B}_t^\ell]^{(m)}$, and $[\mathbf{x}_t^\ell]^{(m)}$ denote the corresponding shards on device m. The layerwise full-step and local block-step learning rates are denoted by $\eta_{\mathrm{full}}^\ell$ and $\eta_{\mathrm{block}}^\ell$, respectively.

Linear minimization oracle. Given a primal norm $\|\cdot\|$ on a parameter space, the associated LMO returns the direction that minimizes the linearized loss over the unit ball of that norm. Formally, the LMO is defined as $\mathrm{LMO}(u) = \arg\min_{\|X\|\le 1} \langle u, X\rangle$. Different choices of primal norm yield different update rules with qualitatively different properties, as summarized in Table 1. Under the spectral norm $\|\cdot\|_{\mathrm{sp}}$ as the primal norm, $\mathrm{LMO}(\mathbf{B}) = -\mathbf{U}\mathbf{V}^\top$, which is the orthogonalized update used by Muon (Jordan et al., 2024b). Under the max norm $\|\cdot\|_\infty$ as the primal norm, $\mathrm{LMO}(\mathbf{B}) = -\operatorname{sign}(\mathbf{B})$, the elementwise sign update used by Signum.

Muon optimizer. Muon is a geometry-aware optimizer for hidden weight matrices, combining SGD-momentum with a NS orthogonalization step. At iteration t, given a stochastic gradient matrix $\mathbf{G}_t$, the momentum matrix $\mathbf{B}_t \in \mathbb{R}^{a\times b}$ and weight matrix $\mathbf{W}_t$ are updated as $\mathbf{B}_t = \mu\mathbf{B}_{t-1} + \mathbf{G}_t$ and $\mathbf{W}_{t+1} = \mathbf{W}_t - \eta\,\mathrm{NS}_T(\mathbf{B}_t)$, where $\mathrm{NS}_T(\cdot)$ denotes T steps of the NS iteration. Let $\mathcal{Q}$ denote the set of semi-orthogonal matrices with the same shape as $\mathbf{B}$. $\mathbf{U}\mathbf{V}^\top$ is a closest such matrix in Frobenius norm: $\mathrm{Ortho}(\mathbf{B}) = \mathbf{U}\mathbf{V}^\top \in \arg\min_{Q\in\mathcal{Q}} \|Q - \mathbf{B}\|_F$. For full-rank $\mathbf{B}$, NS approximates this semi-orthogonal factor. In practice, typically, $T = 5$ NS steps are used. Muon is applied to hidden matrix parameters, while the embedding layer, the language modeling head, and scalar or vector parameters are optimized with AdamW (Jordan et al., 2024b).

Table 1: Different optimizer directions viewed as linear minimization oracles. For a vector $x \in \mathbb{R}^d$ and a matrix $\mathbf{B} \in \mathbb{R}^{a\times b}$ with singular value decomposition $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, different choices of constraint geometry induce different normalized update directions.

Update direction	Constraint geometry	LMO	Method
Normalized gradient	Euclidean unit ball for $z \in \mathbb{R}^d$, $\{z : \|z\|_2 \le 1\}$	$-x/\|x\|_2$	Normalized SGD
Elementwise sign	Max norm unit ball for $Z \in \mathbb{R}^{a\times b}$, $\{Z : \|Z\|_\infty \le 1\}$	$-\operatorname{sign}(\mathbf{B})$	Signum
Newton–Schulz Orthogonalization	Spectral norm unit ball for $Z \in \mathbb{R}^{a\times b}$, $\{Z : \|Z\|_{\mathrm{sp}} \le 1\}$	$-\mathbf{U}\mathbf{V}^\top$	Muon

Distributed Muon with block-periodic communication (MuonBP). MuonBP (Khaled et al., 2025) extends Muon to distributed training through block-periodic orthogonalization. During off-period iterations, each device orthogonalizes its local tensor shard independently, while a full orthogonalization is performed only once every p steps after gathering the corresponding shards. MuonBP applies the spectral-norm LMO uniformly to all hidden layers with local updates. While this design reduces the extra communication cost associated with orthogonalization relative to Moonlight (Liu et al., 2025), it can significantly increase the approximation error of the orthogonalization. The proposed MuonMix shows that assigning max-norm LMO to certain layers can indeed reduce this approximation error with local updates and lower the all-gather/scatter communication cost for orthogonalization.

3 Proposed Algorithm

3.1 Algorithm Description

We consider a model with H layers, partitioned into two groups. A subset of N layers indexed by $\ell \in \{1, \dots, N\}$ are updated via MuonBP, using the spectral norm as primal norm and the nuclear norm as its dual. The remaining $S = H - N$ layers indexed by $\ell \in \{1, \dots, S\}$ are updated via elementwise sign under the ℓ_∞–ℓ_1 norm pair. The layer selection is determined by the per-layer optimizer selection criterion in Section 3.3. Each device $m \in \{1, \dots, M\}$ holds a shard of every layer. Let $p \ge 1$ be the synchronization period, $S_{\mathrm{op}} = \{t : t \bmod p = 0\}$ be the set of full synchronization steps (global LMO with all-gather or scatter), and $S_B = \{t : t \bmod p \neq 0\}$ the set of local block steps. At every step, each device updates its local momentum matrix, with $\beta^\ell = \mu$ for $\ell \in \mathcal{N}$ and $\beta^\ell = \beta$ for $\ell \in \mathcal{S}$. The full update rule is presented in Algorithm 1.

Communication cost. Our method incurs no additional all-gather/scatter communication overhead compared to MuonBP at the same synchronization period p. On a local block step, no optimizer-specific communication is required for either layer group beyond the standard training collectives. On a full step, MuonBP layers perform one all-gather and one scatter to assemble the full matrix update for global NS orthogonalization. For elementwise sign layers, the all-gather and scatter on full steps is unnecessary. The elementwise sign operation satisfies $\operatorname{sign}(\operatorname{concat}_m(\{[\mathbf{B}_t^\ell]^{(m)}\}_m)) = \operatorname{concat}_m(\{\operatorname{sign}([\mathbf{B}_t^\ell]^{(m)})\}_m)$, so each device can compute its local portion of the sign update directly from its local update without gathering the full matrix. On local block steps, the max-norm LMO applies elementwise sign operations rather than NS iterations, which also reduces per-step NS computation for those layers.

3.2 Convergence Analysis

In order to characterize the convergence upper bound of the exact-LMO idealization of Algorithm 1, we make the following assumptions (Riabinin et al., 2025) and (Hao et al., 2025).

Assumption 1 (Layerwise Smoothness). *For each layer ℓ, f is L_ℓ-smooth with respect to a layer-specific norm $\|\cdot\|_\ell$ for all $\mathbf{x}, \mathbf{y}$, $\|\nabla_\ell f(\mathbf{x}) - \nabla_\ell f(\mathbf{y})\|_{\ell*} \le L_\ell \|\mathbf{x}^\ell - \mathbf{y}^\ell\|_\ell$. The norm $\|\cdot\|_\ell$ and smoothness constant L_ℓ depend on both the layer group and the step type. For each MuonBP layer $\ell \in \{1, \dots, N\}$, $\|\cdot\|_\ell$ is the spectral norm $\|\cdot\|_{\mathrm{sp}}$ with $L_{\ell,\mathrm{mu}}^{\mathrm{op}}$ on full steps, and the block spectral*

Algorithm 1: MuonMix

// Initialize: device m holds parameter shard $[\mathbf{x}_0^\ell]^{(m)}$, computes $[\mathbf{G}_0^\ell]^{(m)}$ at $\mathbf{x}_0$, and sets $[\mathbf{B}_0^\ell]^{(m)} \leftarrow [\mathbf{G}_0^\ell]^{(m)}$ for all ℓ.

for $t \leftarrow 0$ **to** $T-1$ **do**

  // Local momentum update on every device

  if $t \geq 1$ **then**

    for *each device* m in parallel **do**

      Compute local gradient shard $[\mathbf{G}_t^\ell]^{(m)}$ for each layer ℓ;

      $[\mathbf{B}_t^\ell]^{(m)} \leftarrow \beta^\ell [\mathbf{B}_{t-1}^\ell]^{(m)} + (1-\beta^\ell)[\mathbf{G}_t^\ell]^{(m)}$;

  if $t \bmod p = 0$ **then**

    // All-gather and scatter only for spectral-norm layers

    $\mathbf{B}_t^\ell \leftarrow \mathrm{concat}_m\big(\{[\mathbf{B}_t^\ell]^{(m)}\}_m\big)$ for each $\ell \in \mathcal{N}$;

    $O_t^\ell \leftarrow -\mathrm{NS}_5(\mathbf{B}_t^\ell)$ for each $\ell \in \mathcal{N}$; scatter its shards;

    $[O_t^\ell]^{(m)} \leftarrow -\operatorname{sign}([\mathbf{B}_t^\ell]^{(m)})$ locally for each $\ell \in \mathcal{S}$ and device m;

    for *each device* m in parallel **do**

      $[\mathbf{x}_{t+1}^\ell]^{(m)} \leftarrow [\mathbf{x}_t^\ell]^{(m)} + \eta_{\text{full}}^\ell\, [O_t^\ell]^{(m)}$ for each layer ℓ;

  else

    // Local updates, no extra optimizer communication

    for *each device* m in parallel **do**

      $[O_t^\ell]^{(m)} \leftarrow \mathrm{LMO}^\ell\big([\mathbf{B}_t^\ell]^{(m)}\big)$ for each layer ℓ ; // LMO^ℓ: $-\mathrm{NS}_5$ if $\ell \in \mathcal{N}$; $-\operatorname{sign}$ if $\ell \in \mathcal{S}$;

      $[\mathbf{x}_{t+1}^\ell]^{(m)} \leftarrow [\mathbf{x}_t^\ell]^{(m)} + \eta_{\text{block}}^\ell\, [O_t^\ell]^{(m)}$ for each layer ℓ;

return $\{[\mathbf{x}_T^\ell]^{(m)}\}_{\ell,m}$;

norm $B_\ell^{\mathrm{mu}}(\cdot) = \max_m \|[\cdot]^{(m)}\|_{\mathrm{sp}}$ *with* $L_{\ell,\mathrm{mu}}^B$ *on local block steps. For each layer* $\ell \in \{1,\dots,S\}$ *updated by elementwise sign,* $\|\cdot\|_\ell$ *is the max norm* $\|\cdot\|_\infty$ *with* $L_{\ell,\mathrm{sg}}$.

Assumption 2 (Layerwise Bounded Stochastic Gradient Noise)**.** *For each layer* ℓ*, the stochastic gradient oracle is unbiased with* $\mathbb{E}[\mathbf{G}^\ell(\mathbf{x}_t) \mid \mathbf{x}_t] = \nabla_\ell f(\mathbf{x}_t)$*, and the stochastic gradient noise satisfies* $\|\mathbf{G}^\ell(\mathbf{x}_t) - \nabla_\ell f(\mathbf{x}_t)\|_{\ell*} \leq \sigma_\ell$ *almost surely in the dual norm, where* $\sigma_\ell \geq 0$.

For $g \in \{\mathrm{mu}, \mathrm{sg}\}$, let $C_1^g, C_2^g > 0$ satisfy $C_1^g \|A\|_{\ell*} \leq \|A\|_F \leq C_2^g \|A\|_{\ell*}$ uniformly over the layers and both dual norms in group g. Write $\eta_{\max}^g = \max\{\eta_{\text{full}}^g, \eta_{\text{block}}^g\}$ and define $\gamma^{\mathrm{mix}} = \frac{\min\{\eta_{\text{full}}^{\mathrm{mu}}, \eta_{\text{full}}^{\mathrm{sg}}\} + (p-1)\min\{\eta_{\text{block}}^{\mathrm{mu}}, \eta_{\text{block}}^{\mathrm{sg}}\}}{p}$, $\kappa^g = \eta_{\max}^g / \gamma^{\mathrm{mix}}$.

Theorem 1 (Convergence of MuonMix)**.** *Suppose Assumptions 1 and 2 hold. Consider exact LMOs with* $\mathbf{B}_0^\ell = \mathbf{G}^\ell(\mathbf{x}_0)$*,* $N, S > 0$*,* $p \mid T$*, positive smoothness constants, and positive step sizes uniform within each group. Assume* $f \geq f^* > -\infty$*,* $\Delta = f(\mathbf{x}_0) - f^* > 0$*, and* $\Big(\sum_{\ell=1}^N L_{\ell,\mathrm{mu}}^{\mathrm{op}}\Big)\eta_{\text{full}}^{\mathrm{mu}} = \Big(\sum_{\ell=1}^N L_{\ell,\mathrm{mu}}^B\Big)\eta_{\text{block}}^{\mathrm{mu}}$*,* $\eta_{\text{full}}^{\mathrm{sg}} = \eta_{\text{block}}^{\mathrm{sg}}$*. Let* $\eta_{\max}^{\mathrm{mu}} = \sqrt{\Delta(1-\mu)/(T\widehat{L}_{BP}^{\mathrm{mu}})}$*,* $\eta_{\max}^{\mathrm{sg}} = \sqrt{\Delta(1-\beta)/(T\widehat{L}_{BP}^{\mathrm{sg}})}$*,* $1-\mu = \min\Big(\sqrt{\Delta\widehat{L}_{BP}^{\mathrm{mu}}}/\big((\sum_{\ell=1}^N \sigma_\ell)\sqrt{T}\big), 1\Big)$*, and* $1-\beta = \min\Big(\sqrt{\Delta\widehat{L}_{BP}^{\mathrm{sg}}}/\big((\sum_{\ell=1}^S \sigma_\ell)\sqrt{T}\big), 1\Big)$*. If a group has zero total noise, set its momentum coefficient to zero. Then for any* $\delta \in (0,1)$*, with probability at least* $1-\delta$*,*

$$\min_{0\leq t\leq T-1}\left(\sum_{\ell=1}^N \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + \sum_{\ell=1}^S \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right)$$

$$\leq \underbrace{\left(10\kappa^{\mathrm{sg}} + \frac{8\kappa^{\mathrm{sg}} C_2^{\mathrm{sg}}\sqrt{\log\frac{2HT}{\delta}}}{C_1^{\mathrm{sg}}}\right)\left(\frac{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}}}{\sqrt{T}} + \frac{\sqrt{\sum_{\ell=1}^S \sigma_\ell}\,(\Delta\,\widehat{L}_{BP}^{\mathrm{sg}})^{1/4}}{T^{1/4}}\right)}_{\text{max-norm LMO terms in } \mathcal{S}} + Z^{\mathrm{sg}}$$

$$+\underbrace{\left(10\sqrt{rc}\,\kappa^{\mathrm{mu}}+\frac{8\kappa^{\mathrm{mu}}C_2^{\mathrm{mu}}\sqrt{\log\frac{2HT}{\delta}}}{C_1^{\mathrm{mu}}}\right)\left(\frac{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}}}{\sqrt{T}}+\frac{\sqrt{\sum_{\ell=1}^{N}\sigma_\ell}(\Delta\,\widehat{L}_{BP}^{\mathrm{mu}})^{1/4}}{T^{1/4}}\right)}_{\textit{spectral-norm LMO terms in }\mathcal{N}}+Z^{\mathrm{mu}}, \quad (1)$$

where $Z^{\mathrm{mu}} = 4\kappa^{\mathrm{mu}}\left(\sum_{\ell=1}^{N}\sigma_\ell\right)^2/\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\,T}$, $Z^{\mathrm{sg}} = 4\kappa^{\mathrm{sg}}\left(\sum_{\ell=1}^{S}\sigma_\ell\right)^2/\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\,T}$, $(\widehat{L}_{BP}^{\mathrm{mu}})^{-1} = p^{-1}(\sum_{\ell=1}^{N}L_{\ell,\mathrm{mu}}^{\mathrm{op}})^{-1} + (p-1)p^{-1}(\sum_{\ell=1}^{N}L_{\ell,\mathrm{mu}}^{B})^{-1}$ *and* $\widehat{L}_{BP}^{\mathrm{sg}} = \sum_{\ell=1}^{S}L_{\ell,\mathrm{sg}}$.

Discussions. First, the convergence upper bound in Theorem 1 separates the contributions of spectral-norm LMO and max-norm LMO derived from Theorem B.1 in Appendix B.4. When $N = 0$ or $S = 0$, (1) reduces to the "all max-norm LMO" or "all-MuonBP" bound, respectively. The overall convergence rate is $\widetilde{\mathcal{O}}(T^{-1/2} + (\sum_\ell \sigma_\ell)^{1/2}T^{-1/4})$. Second, the spectral-norm LMO terms in (1) include a factor of $\sqrt{rc}$ from the worst-case comparison between full and block spectral norms. The max-norm LMO terms in (1) do not have such an explicit factor. The max norm satisfies $\|\mathbf{G}\|_\infty = \max_m \|[\mathbf{G}]^{(m)}\|_\infty$ by definition, implying that the elementwise sign of each local shard matches that of the corresponding entries in the full matrix. The block-step and full-step update directions are therefore identical, and no block approximation error arises. As the number of devices M grows, the convergence error for MuonBP layers increases while Signum layers remain unaffected.

3.3 Per-Layer Optimizer Selection

Theorem 1 holds for any fixed partition of the layers into N layers updated by the spectral-norm LMO and S layers updated by the max-norm LMO. We now develop a principled criterion for assigning each layer independently to one of the two LMOs

Average per-step descent lower bounds. The convergence bound for spectral-norm LMO and max-norm LMO involves different dual norms and cannot be compared directly. We instead compare both methods through the *average per-step descent lower bound* that each one guarantees along its own trajectory in this noiseless setting. As shown in Proposition 1 in Appendix C, using MuonBP for all H layers, we get $(f(\mathbf{x}_0) - f^*)/T \geq \Phi^{\mathrm{mu}} = \sum_{s=1}^{H} B_s^{\mathrm{mu}} - \sum_{s=1}^{H} E_s^{\mathrm{mu}}$ which is defined as the average per-step descent lower bound for MuonBP, where $B_s^{\mathrm{mu}} = \frac{1}{T}\big(\sum_{k\in S_{\mathrm{op}}}\eta_{\mathrm{full}}^{\mathrm{mu}}\|\nabla_s f(\mathbf{x}_k)\|_{s,*}^{\mathrm{nuc}} + \sum_{k\in S_B}\eta_{\mathrm{block}}^{\mathrm{mu}}\|\nabla_s f(\mathbf{x}_k)\|_{s,*}^{\mathrm{nuc}}\big)$ and E_s^{mu} is given by (2). Using MuonMix, where layer s is updated by max-norm LMO and all other $H-1$ layers are updated by MuonBP, we get $(f(\mathbf{x}_0) - f^*)/T \geq \Phi_s^{\mathrm{mix}} = \sum_{\ell=1,\ell\neq s}^{H} B_\ell^{\mathrm{mu}} + B_s^{\mathrm{sg}} - \sum_{\ell=1,\ell\neq s}^{H} E_\ell^{\mathrm{mu}} - E_s^{\mathrm{sg}}$. Therefore, a larger value for Φ^{mu} or Φ_s^{mix} gives a larger descent lower bound along the corresponding trajectory. Ordering these bounds alone does not order the actual decreases.

Average per-step descent advantage. We define the *average per-step descent advantage* as $\Phi_s^{\mathrm{diff}} = \Phi_s^{\mathrm{mix}} - \Phi^{\mathrm{mu}}$. On the same fixed reference trajectory, a positive Φ_s^{diff} favors Signum under this surrogate. Here we use Signum as an example for the max-norm LMO implementation. As shown in Proposition 2 in Appendix C, the descent advantage is $\Phi_s^{\mathrm{diff}} = \big[B_s^{\mathrm{sg}} - E_s^{\mathrm{sg}}\big] - \big[B_s^{\mathrm{mu}} - E_s^{\mathrm{mu}}\big]$, where B_s^{sg} is defined in the same way as B_s^{mu} with the ℓ_1 norm as the dual norm. Let $\eta_{\mathrm{full}}^{\mathrm{sg}} = \eta_{\mathrm{block}}^{\mathrm{sg}}$, E_s^{mu} and E_s^{sg} are given by (see (B.70) in Appendix B.4)

$$E_s^{\mathrm{mu}} = \frac{L_{s,\mathrm{mu}}^{\mathrm{op}}(\eta_{\mathrm{full}}^{\mathrm{mu}})^2 + (p-1)L_{s,\mathrm{mu}}^{B}(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2p} + \frac{2}{(1-\mu)p}\Big(L_{s,\mathrm{mu}}^{\mathrm{op}}\,\eta_{\mathrm{full}}^{\mathrm{mu}}\max\{\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{mu}}\sqrt{rc}\} + (p-1)L_{s,\mathrm{mu}}^{B}\,\eta_{\mathrm{block}}^{\mathrm{mu}}\,\eta_{\mathrm{max}}^{\mathrm{mu}}\Big), \quad (2)$$

$$E_s^{\mathrm{sg}} = \frac{L_{s,\mathrm{sg}}(\eta_{\mathrm{full}}^{\mathrm{sg}})^2 + (p-1)L_{s,\mathrm{sg}}(\eta_{\mathrm{block}}^{\mathrm{sg}})^2}{2p} + \frac{2}{(1-\beta)p}\Big(L_{s,\mathrm{sg}}\,\eta_{\mathrm{full}}^{\mathrm{sg}}\,\eta_{\mathrm{max}}^{\mathrm{sg}} + (p-1)L_{s,\mathrm{sg}}\,\eta_{\mathrm{block}}^{\mathrm{sg}}\,\eta_{\mathrm{max}}^{\mathrm{sg}}\Big). \quad (3)$$

Geometric interpretation. The factor $\sqrt{rc}$ in E_s^{mu} grows with the number of devices M and is present whenever $p > 1$. No such term appears in E_s^{sg}. For layers where the gradient structure is well captured by the max norm, Φ_s^{diff} is more likely to be positive when M is large, making the elementwise sign the preferred choice.

4 EXPERIMENTS

We conduct experiments in two stages. We first run a profiling phase on the 160M model to estimate $\widehat{\Phi}_s^{\text{diff}}$ for each weight matrix and determine the per-layer LMO selection. We then evaluate MuonMix against Moonlight (Liu et al., 2025) and MuonBP (Khaled et al., 2025) on language model pretraining at three model sizes, 160M parameters, 960M parameters, and 1.2B parameters.[3] The code for our experiments is available at: `https://anonymous.4open.science/r/MuonMix-ADB3/`.

4.1 EXPERIMENTAL SETUP

Codebase and architecture. Using the Dion codebase (Ahn et al., 2025), we train Modded-NanoGPT (Jordan et al., 2024a) with rotary positional embeddings (Su et al., 2024), pre-norm RMSNorm (Zhang & Sennrich, 2019), QK normalization (Dehghani et al., 2023), squared ReLU activations (So et al., 2022), and bias-free linear layers. All models use a 50304-token vocabulary (GPT-2 tokenizer (Radford et al., 2019)). The three models train for 5000, 15000, and 20000 steps, respectively, with a global batch of 1024 sequences of 1024 tokens, consuming approximately 5.2B, 15.7B, and 20.9B tokens from the FineWeb dataset (Penedo et al., 2024). We do not target Chinchilla-optimal training (Hoffmann et al., 2022). The horizon is chosen to observe optimizer convergence behavior over the same number of training steps. All runs use bfloat16 mixed precision and a warmup-stable-decay (WSD) learning-rate schedule with no warmup and 20% linear cooldown.

8-GPU configuration. We run the 160M experiment on one 8-GPU B200 node with data-parallel (DP) degree 1 (Li et al., 2020), FSDP degree 4 (Zhao et al., 2023), and TP degree 2. Under column-row TP (Shoeybi et al., 2019), query, key, and value attention projections and first MLP linear layer are column-sharded across TP ranks. Attention output projection and second MLP linear layer are row-sharded. FSDP shards each transformer block's parameters with bfloat16 parameter all-gather and float32 gradient reduction. It shards the attention output projection and second MLP layer along their output dimension and all other linear layers along their input dimension.

Virtual simulation configuration. To evaluate scalability beyond the 8-GPU setup, we conduct virtual sharding simulation experiments. During each local block step, each physical GPU's local shard is partitioned into smaller virtual sub-shards, and the NS iteration is applied to each sub-shard. The sub-shard updates are then concatenated and used for the global matrix update. This reproduces the orthogonalized update direction induced by 128-GPU parallelism for the 960M model and 256-GPU parallelism for the 1.2B model on 4 physical NVIDIA B200 GPUs.

Methods. Moonlight (Liu et al., 2025) performs a full all-gather and global NS orthogonalization every step, typically yielding stable convergence at higher per-step communication cost. MuonBP (Khaled et al., 2025) uses block-periodic orthogonalization ($p = 5$)[4]: NS on local shards, with a full all-gather every p-th step. *MuonMix* retains MuonBP at the same p for all hidden weights except those with positive $\widehat{\Phi}_s^{\text{diff}}$ in Section 4.2, which use Signum. Signum matches the max-norm LMO in our theoretical analysis and requires no all-gather/scatter communication. We approximate spectral-norm LMOs with five NS iterations, following (Jordan et al., 2024b). Signum also updates the embedding layer, language modeling head, and all scalar and vector parameters. Given AdamW's popularity and superior empirical performance, we evaluate it as an alternative elementwise optimizer. Additional results appear in Appendix A.3. Following (Ahn et al., 2025), we scale the MuonBP learning rate by $\sqrt{a/b}$ for each pre-sharding weight matrix $W \in \mathbb{R}^{a\times b}$. To ensure a fair comparison, all three methods at each model size are initialized from the same initial weights and trained with the same seed, so that any difference in validation loss or throughput is attributable solely to the choice of optimizer. Model details and selected learning rates appear in Table A.1 of Appendix A.1.

4.2 PER-LAYER PROFILING

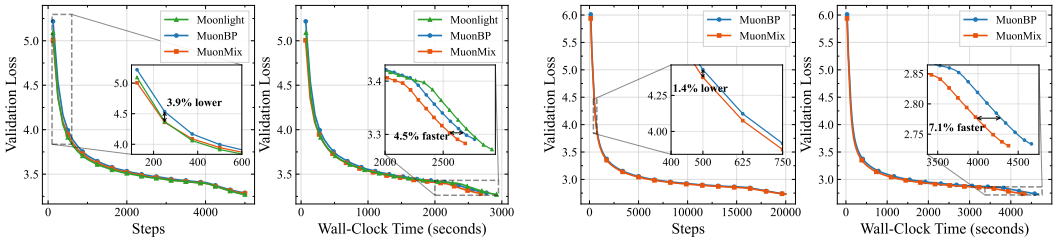
Setup. We profile the 160M model for 500 steps on 4 B200 GPUs, using Signum as the elementwise optimizer corresponding exactly to the max-norm LMO. At each step, we record per-shard gradient

[3] See Appendix A.2 that we also validate the profiling on 960M and 1.2B models and it achieves the same results as that for 160M model.

[4] MuonBP (Khaled et al., 2025) notes that the period should ideally minimize the wall-clock time to reach a target accuracy. They find that $p = 5$ balances this tradeoff well in most experiments.

Table 2: Per-layer profiling results for the 160M model on 4 B200 GPUs. $W_Q, W_K, W_V, W_O, W_1, W_2$ denote the query, key, value, attention output, first MLP, and second MLP weight matrices, respectively. Training step time is measured over 500 profiling steps.

Weight matrix	$\widehat{L}^{\mathrm{op}}_{s,\mathrm{mu}}$	$\widehat{L}_{s,\mathrm{sg}}$	$\|G\|^{\mathrm{nuc}}_{s,*}$	$\|G\|^{1}_{s,*}$	$\widehat{B}^{\mathrm{sg}}_s - \widehat{B}^{\mathrm{mu}}_s$	$\widehat{E}^{\mathrm{mu}}_s - \widehat{E}^{\mathrm{sg}}_s$	$\widehat{\Phi}^{\mathrm{diff}}_s$	Step (ms)	Assign
MuonBP	—	—	—	—	—	—	—	21777	—
W_Q	5.76	1066	0.056	0.963	$+9.50\times10^{-4}$	$+3.78\times10^{-3}$	$\mathbf{+4.73\times10^{-3}}$	18781	Signum
W_K	7.66	1414	0.072	1.215	$+1.18\times10^{-3}$	$+6.38\times10^{-3}$	$\mathbf{+7.56\times10^{-3}}$	18855	Signum
W_V	18.23	6012	0.170	6.786	$+8.65\times10^{-3}$	-5.53×10^{-2}	-4.67×10^{-2}	19535	MuonBP
W_O	15.70	6762	0.150	7.361	$+9.69\times10^{-3}$	-7.29×10^{-2}	-6.32×10^{-2}	19642	MuonBP
W_1	42.85	9456	0.395	10.334	$+1.20\times10^{-2}$	-4.27×10^{-2}	-3.07×10^{-2}	19337	MuonBP
W_2	154.1	7116	1.624	31.915	$+3.32\times10^{-2}$	-2.52×10^{-1}	-2.19×10^{-1}	20258	MuonBP



(a) 160M model (8GPUs): validation loss vs. steps (left) and wall-clock time (right).

(b) 1.2B model (simulated 256 GPUs): validation loss vs. steps (left) and wall-clock time (right).

Figure 1: Comparison of Moonlight, MuonBP, and MuonMix across training steps and wall-clock time.

and parameter differences and compute the smoothness estimates $\widehat{L}^{\mathrm{op}}_{s,\mathrm{mu}}$, $\widehat{L}^{B}_{s,\mathrm{mu}}$, and $\widehat{L}_{s,\mathrm{sg}}$, along with per-step gradient dual norms, to obtain $\widehat{B}^{\mathrm{mu}}_s$ and $\widehat{B}^{\mathrm{sg}}_s$ following Section 3.3. At the end of profiling, we assign each weight's optimizer once based on the sign of $\widehat{\Phi}^{\mathrm{diff}}_s$ defined in Section 3.3 and keep the assignments fixed for all subsequent training.

Results. Table 2 presents profiling results averaged across 12 transformer blocks. Only W_Q and W_K have positive $\widehat{\Phi}^{\mathrm{diff}}_s$. For these query and key projection layers, $\widehat{L}_{s,\mathrm{sg}}$ is 1066 and 1414, respectively, while for all remaining layers it ranges from 6012 to 9456. For W_V through W_2, the larger $\widehat{L}_{s,\mathrm{sg}}$ increases $\widehat{E}^{\mathrm{sg}}_s$, making $\widehat{\Phi}^{\mathrm{diff}}_s$ negative and assigning these layers to MuonBP. The first two non-MuonBP rows, for W_Q and W_K, show that replacing them with Signum reduces average per-step time by 13.8% and 13.4%, respectively, despite the overhead of estimating smoothness at every step. These step times combine computation and communication and do not isolate communication savings. Accordingly, here we use Signum for W_Q and W_K and MuonBP for all remaining layers. Tables A.3 and A.4 of Appendix A.2 give the per-layer profiling results for the 960M and 1.2B models.

4.3 Experimental Results

Table 3 presents the minimum validation loss, average throughput, and per-step training time for 160M model. Figure 1 shows validation loss against training steps and wall-clock time for different methods and model sizes. Training curves for the 960M model are provided in the Appendix A.2.

160M model (8 GPUs). MuonMix reaches 243,628 tokens/s/GPU, 3.0% above MuonBP (236,592) and 8.6% above Moonlight (224,438). It also converges faster early on, with 3.4% lower average validation loss than MuonBP over the first 400 training steps. Following (Khaled et al., 2025), we assess two related metrics: wall-clock time to a target loss and validation loss at a fixed step count. MuonMix reaches each target faster than MuonBP: loss 3.40 in 2,098 versus 2,247 seconds (6.6% less), and loss 3.30 in 2,572 versus 2,692 seconds (4.5% less; Figure 1a). At fixed step counts, MuonMix consistently achieves lower validation loss. Its end-of-training minimum is 3.283 versus 3.291 for MuonBP. Replacing W_Q and W_K with Signum thus partially recovers the convergence behavior of full orthogonalization on those layers.

Table 3: Minimum validation loss, average validation loss over the first 400 training steps (early convergence), average throughput, and per-step training time for 160M model on 8 GPUs.

Model	Method	Min Val Loss	Early Val Loss	Step Time (ms)	Tokens/s/GPU
160M	Moonlight	**3.272**	4.508	584	224,438
	MuonBP	3.291	4.640	554	236,592
	MuonMix	3.283	**4.484**	**538**	**243,628**

Table 4: Minimum validation loss, average validation loss over the first 1,000 training steps (early convergence), estimated per-step computation, communication, and total step times, and throughput under virtual simulation (128 GPUs for 960M; 256 GPUs for 1.2B) conducted on 4 physical GPUs.

Model	Method	Validation Loss		Comp. Time	Comm. Time	Step Time (ms)	Tokens/s/GPU
		Min	Early				
960M	MuonBP	**2.796**	4.279	126.84	141.19	268.03	30,564
	MuonMix	2.798	**4.201**	**114.50**	141.19	**255.69**	**32,039**
1.2B	MuonBP	2.734	4.410	104.47	128.79	233.26	17,560
	MuonMix	**2.731**	**4.361**	**90.21**	128.79	**219.00**	**18,703**

Step time estimation. To further assess scalability at larger model sizes and higher parallelism degrees, we simulate training on 128 and 256 GPUs using virtual sharding on 4 physical GPUs. Table 4 reports estimated step times for the target configurations with 128 or 256 GPUs. We measure the computation times of global and blockwise NS orthogonalization with five iterations (NS5) and elementwise updates on a B200 using the target matrix shapes, and amortize orthogonalization costs over the synchronization period p. We use Calculon (Isaev et al., 2023) to estimate forward and backward computation times and TP communication times. For FSDP communication, we use Calculon's network cost model to estimate the time of parameter all-gathers during the forward and backward passes and gradient reduce-scatter. All larger-model wall-clock curves and time-to-target values use these estimated step times. These non-optimizer costs are identical for both methods. Detailed computation and communication times are provided in Table A.2 of Appendix A.2. We exclude communication time for global NS orthogonalization. To the best of our knowledge, prior work does not model this cost at the target scale. We conservatively report the gains from reduced computation, excluding potential savings from NS all-gather and scatter communication operations. Details of the estimation using Calculon are provided in Appendix A.5.

960M (128 GPUs virtual simulation) and 1.2B model (256 GPUs virtual simulation). Table 4 and Figure 1 summarize the simulation results. Throughput in the table is tokens per step divided by step time and the number of simulated GPUs. At 256 simulated GPUs, MuonMix reaches 18,703 tokens/s/GPU, 6.5% above MuonBP (17,560), and converges faster early on: average validation loss over the first 1,000 steps is 4.361 versus 4.410, a 1.1% reduction. MuonMix also reaches target losses sooner in wall-clock time: 3.20 in 598 versus 672 seconds (11.0% less), and 2.78 in 3,982 versus 4,286 seconds (7.1% less; Figure 1b). At a fixed number of training steps, MuonMix's minimum validation loss is 2.731 versus 2.734 for MuonBP. At 960M, its minimum is slightly higher (2.798 versus 2.796), which concludes that the optimization performance of our approach strengthens further at larger model and higher parallelism degrees. [5]

5 Conclusion

In this paper, we proposed MuonMix, a distributed Muon framework that improves both convergence behavior and training efficiency over MuonBP. By assigning selected layers to the max-norm LMO, MuonMix eliminates the need for Newton–Schulz iterations and the associated all-gather and scatter communication operations for those layers, while introducing no block approximation error. Experimental results show higher throughput and faster early convergence than MuonBP in the tested settings. Our findings suggest that geometry-aware optimizer design is a promising direction for scalable distributed training.

[5] We note that the step time and throughput in Table 4 are estimates for the target configurations with 128 or 256 GPUs and are not directly comparable to Table 3.

AI USE STATEMENT

In this work, we used generative AI tools for mathematical check and code implementation assist. We have not used generative AI tools for the algorithm design and theoretical development, the problem formulation, the convergence analysis or its proofs, and synthetic data generation is not applicable to this work. Additionally, we used generative AI tools for figure and table generation, code editing, and readability editing of the manuscript. We have reviewed all AI-assisted work. The implementation was verified against an independent reference implementation of the same algorithms, every reported number is produced by the released code from stored per-seed records, and all AI-assisted text was read and revised by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work contains no personally identifiable information and no human subjects. We see no direct ethical concerns arising from the method itself.

REPRODUCIBILITY STATEMENT

Algorithm 1 describes MuonMix. Sections A.1 and A.5 provide the model configurations, selected learning rates, and timing model. The assumptions and proofs appear in Section 3.2 and Appendices B and C. The experiment code is linked in Section 4.

REFERENCES

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

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International conference on machine learning*, pp. 560–569. PMLR, 2018.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pp. 177–186. Springer, 2010.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International conference on machine learning*, pp. 7480–7512. PMLR, 2023.

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

Kevin Frans, Pieter Abbeel, and Sergey Levine. What really matters in matrix-whitening optimizers? *arXiv preprint arXiv:2510.25000*, 2025.

Kaja Gruntkowska, Yassine Maziane, Zheng Qu, and Peter Richtárik. Drop-muon: Update less, converge faster. *arXiv preprint arXiv:2510.02239*, 2025.

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

Jie Hao, Xiaochuan Gong, Jie Xu, Zhengdao Wang, and Mingrui Liu. Noise-adaptive layerwise learning rates: Accelerating geometry-aware optimization for deep neural network training. *arXiv preprint arXiv:2510.14009*, 2025.

Chuan He, Shuyi Ren, Jingwei Mao, and Erik G Larsson. Demuon: A decentralized muon for matrix optimization over graphs. *arXiv preprint arXiv:2510.01377*, 2025.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, DDL Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 10, 2022.

Mikhail Isaev, Nic McDonald, Larry Dennison, and Richard Vuduc. Calculon: a methodology and tool for high-level co-design of systems and large language models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–14, 2023.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pp. 427–435. PMLR, 2013.

Xiaowen Jiang, Andrei Semenov, and Sebastian U Stich. Enhancing llm training via spectral clipping. *arXiv preprint arXiv:2603.14315*, 2026.

Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024a. URL `https://github.com/KellerJordan/modded-nanogpt`.

Keller Jordan, Yuchen Jin, Vlado Boza, You Jiacheng, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024. *URL https://kellerjordan. github. io/posts/muon*, 6(3):4, 2024b.

Ahmed Khaled, Kaan Ozkara, Tao Yu, Mingyi Hong, and Youngsuk Park. Muonbp: Faster muon via block-periodic orthogonalization. *arXiv preprint arXiv:2510.16981*, 2025.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, Pritam Damania, et al. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*, 2020.

Zichong Li, Liming Liu, Chen Liang, Weizhu Chen, and Tuo Zhao. Normuon: Making muon more efficient and scalable. *arXiv preprint arXiv:2510.05491*, 2025.

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

Zijian Liu, Srikanth Jagabathula, and Zhengyuan Zhou. Near-optimal non-convex stochastic optimization under generalized smoothness. *arXiv preprint arXiv:2302.06032*, 2023.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International conference on machine learning*, pp. 2408–2417. PMLR, 2015.

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

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

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

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

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). *arXiv preprint arXiv:2505.13416*, 2025.

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

Egor Shulgin, Sultan AlRashed, Francesco Orabona, and Peter Richtárik. Beyond the ideal: Analyzing the inexact muon update. *arXiv preprint arXiv:2510.19933*, 2025.

David R. So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. Primer: Searching for efficient transformers for language modeling, 2022. URL `https://arxiv.org/abs/2109.08668`.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Yuki Takezawa, Anastasia Koloskova, Xiaowen Jiang, and Sebastian U Stich. Fedmuon: Federated learning with bias-corrected lmo-based optimization. *arXiv preprint arXiv:2509.26337*, 2025.

Xuan Tang, Jichu Li, and Difan Zou. A convergence analysis of adaptive optimizers under floating-point quantization. *arXiv preprint arXiv:2510.21314*, 2025.

Ajay Tirumala and Raymond Wong. Nvidia blackwell platform: Advancing generative ai and accelerated computing. In *2024 IEEE Hot Chips 36 Symposium (HCS)*, pp. 1–33. IEEE Computer Society, 2024.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Mujin Kwun, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. Soap: Improving and stabilizing shampoo using adam. *arXiv preprint arXiv:2409.11321*, 2024.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*, 2023.

Appendix

A Additional Results of Experiments **15**

A.1 Additional Details 15

A.2 Additional Results with Signum 15

A.3 Additional Results with AdamW 16

A.4 Ablation Study on Layer Selection 19

A.5 Step Time Estimation 19

B Proof of Theorem 1 **22**

B.1 Analysis for Max-Norm LMO on Local Block Step 24

B.2 Layerwise Descent Analysis for Spectral-Norm LMO 26

B.3 Layerwise Descent Analysis for Max-Norm LMO 28

B.4 MuonMix: Layerwise Descent Analysis for Mixed-Norm LMOs 30

C Per-Layer Optimizer Selection **36**

C.1 Per-Step Descent Lower Bound for MuonBP 36

C.2 Per-Step Descent Lower Bound for MuonMix 37

C.3 Average Per-Step Descent Lower Bounds 38

C.4 Empirical Smoothness Constant Estimation 40

A Additional Results of Experiments

A.1 Additional Details

Hyperparameter selection. For the 160M model we perform a grid search over the Muon learning rate $\eta \in \{0.0075, 0.01, 0.013, 0.015, 0.02, 0.03, 0.04\}$ and the ratio of the elementwise updates learning rate to η over the values $\{0.05, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50, 0.60\}$. For the 960M and 1.2B models, we scale the optimal 160M learning rates by factors in $[0.1, 0.8]$, allowing different factors for the Muon and elementwise learning rates, and select the best pair based on validation loss after 500 steps as shown in Table A.1.

Table A.1: Model architecture and selected learning rates for each method and model with AdamW or Signum elementwise updates. Let η denote the Muon learning rate, η_{a} the AdamW learning rate, and η_{s} the Signum learning rate.

Model	Method	Layers	Heads	Dim	AdamW		Signum	
					η	η_{a}	η	η_{s}
160M	Moonlight	12	6	768	0.020	0.004	0.020	0.001
	MuonBP				0.015	0.003	0.015	0.0015
	MuonMix				0.010	0.005	0.013	0.0013
960M	Moonlight	24	16	1664	0.010	0.002	0.012	0.0006
	MuonBP				0.009	0.002	0.009	0.0009
	MuonMix				0.006	0.003	0.007	0.0007
1.2B	Moonlight	28	16	1792	0.009	0.0009	0.010	0.0005
	MuonBP				0.008	0.002	0.008	0.0008
	MuonMix				0.004	0.002	0.006	0.0006

Profiling cost. The reported training comparisons do not establish a net saving after profiling. For example, the W_Q row in Table 2 corresponds to $500 \times 18.781 = 9{,}390.5$ seconds, whereas the reported 160M step-time difference saves about $5{,}000 \times (0.554 - 0.538) = 80$ seconds over training. These runs use different GPU counts. The training-time gains therefore do not imply an end-to-end saving for a single profiled run. The profiling overhead can be amortized when the selected layer assignment is reused across multiple training runs.

A.2 Additional Results with Signum

Table A.2: Computation and communication times under virtual simulation (128 GPUs for 960M; 256 GPUs for 1.2B). Elementwise updates use Signum.

		Computation Time (ms)					Communication Time (ms)	
Model	Method	FW	BW	Global NS5	Block NS5	Elem. Update	TP	FSDP
960M	MuonBP	29.86	53.99	58.38	39.09	0.04	92.57	48.62
	MuonMix	29.86	53.99	45.41	25.99	0.78	92.57	48.62
1.2B	MuonBP	19.47	35.10	68.35	45.24	0.04	63.81	64.98
	MuonMix	19.47	35.10	53.43	30.09	0.88	63.81	64.98

960M model (128 GPUs virtual simulation). We simulate 128 GPUs training on 4 physical GPUs based on virtual sharding. Results are summarized in Table 4 and Figure A.1. MuonMix achieves a throughput of 32,039 tokens/s/GPU, 4.8% higher than MuonBP (30,564) under the simulated 128 GPUs configuration. Compared to MuonBP, MuonMix also exhibits faster early convergence. The average validation loss over the first 1,000 training steps is 4.201, versus 4.279 for MuonBP, a 1.8% reduction. Given a target validation loss, MuonMix reaches it faster in wall-clock time than MuonBP. To reach a validation loss of 3.20, MuonMix requires 698 s against 759 s for MuonBP, a 8.0% reduction. In the late-training regime, to reach a validation loss of 2.83, MuonMix requires 3,541 s against 3,719 s for MuonBP, a 4.8% reduction. Given a fixed number of training steps, MuonMix completes 15,000 training steps in 3,835 s, approximately 185 s earlier than MuonBP (4,020 s), a 4.6% reduction. At the end of training, MuonBP achieves a minimum validation loss of 2.796, compared to 2.798 for MuonMix, giving a small validation loss gap of 0.002.

Table A.3: Per-layer profiling results for the 960M model on 4 B200 GPUs. W_Q, W_K, W_V, W_O, W_1, W_2 denote the query, key, value, attention output, first MLP, and second MLP weight matrices, respectively. Training step time is measured over 500 profiling steps.

Weight matrix	$\widehat{L}^{\mathrm{op}}_{s,\mathrm{mu}}$	$\widehat{L}_{s,\mathrm{sg}}$	$\|G\|^{\mathrm{nuc}}_{s,*}$	$\|G\|^{1}_{s,*}$	$\widehat{B}^{\mathrm{sg}}_s - \widehat{B}^{\mathrm{mu}}_s$	$\widehat{E}^{\mathrm{mu}}_s - \widehat{E}^{\mathrm{sg}}_s$	$\widehat{\Phi}^{\mathrm{diff}}_s$	Step (ms)	Assign
MuonBP	—	—	—	—	—	—	—	36459	—
W_Q	8.44	1636	0.052	0.974	$+5.98\times10^{-4}$	$+8.32\times10^{-4}$	$\mathbf{+1.43\times10^{-3}}$	30540	Signum
W_K	10.59	1940	0.062	1.174	$+7.22\times10^{-4}$	$+1.89\times10^{-3}$	$\mathbf{+2.61\times10^{-3}}$	30527	Signum
W_V	25.16	7658	0.149	9.599	$+7.84\times10^{-3}$	-5.30×10^{-2}	-4.52×10^{-2}	31739	MuonBP
W_O	22.27	9060	0.137	10.249	$+8.49\times10^{-3}$	-6.36×10^{-2}	-5.51×10^{-2}	32788	MuonBP
W_1	51.30	12399	0.304	11.788	$+8.97\times10^{-3}$	-5.38×10^{-2}	-4.48×10^{-2}	30829	MuonBP
W_2	170.5	10124	1.148	21.496	$+1.33\times10^{-2}$	-9.29×10^{-2}	-7.96×10^{-2}	35661	MuonBP

Table A.4: Per-layer profiling results for the 1.2B model on 4 B200 GPUs. W_Q, W_K, W_V, W_O, W_1, W_2 denote the query, key, value, attention output, first MLP, and second MLP weight matrices, respectively. Training step time is measured over 500 profiling steps.

Weight matrix	$\widehat{L}^{\mathrm{op}}_{s,\mathrm{mu}}$	$\widehat{L}_{s,\mathrm{sg}}$	$\|G\|^{\mathrm{nuc}}_{s,*}$	$\|G\|^{1}_{s,*}$	$\widehat{B}^{\mathrm{sg}}_s - \widehat{B}^{\mathrm{mu}}_s$	$\widehat{E}^{\mathrm{mu}}_s - \widehat{E}^{\mathrm{sg}}_s$	$\widehat{\Phi}^{\mathrm{diff}}_s$	Step (ms)	Assign
MuonBP	—	—	—	—	—	—	—	48023	—
W_Q	8.96	1760	0.049	0.978	$+5.46\times10^{-4}$	$+2.75\times10^{-4}$	$\mathbf{+8.21\times10^{-4}}$	40276	Signum
W_K	11.13	2258	0.059	1.173	$+6.56\times10^{-4}$	$+6.73\times10^{-4}$	$\mathbf{+1.33\times10^{-3}}$	40391	Signum
W_V	26.51	9621	0.143	10.310	$+7.57\times10^{-3}$	-5.92×10^{-2}	-5.16×10^{-2}	41830	MuonBP
W_O	23.67	10085	0.132	11.263	$+8.38\times10^{-3}$	-6.54×10^{-2}	-5.70×10^{-2}	43365	MuonBP
W_1	52.93	14692	0.285	11.597	$+7.92\times10^{-3}$	-5.24×10^{-2}	-4.45×10^{-2}	40746	MuonBP
W_2	162.7	9007	0.965	17.317	$+9.29\times10^{-3}$	-3.14×10^{-2}	-2.21×10^{-2}	47824	MuonBP

A.3 Additional Results with AdamW

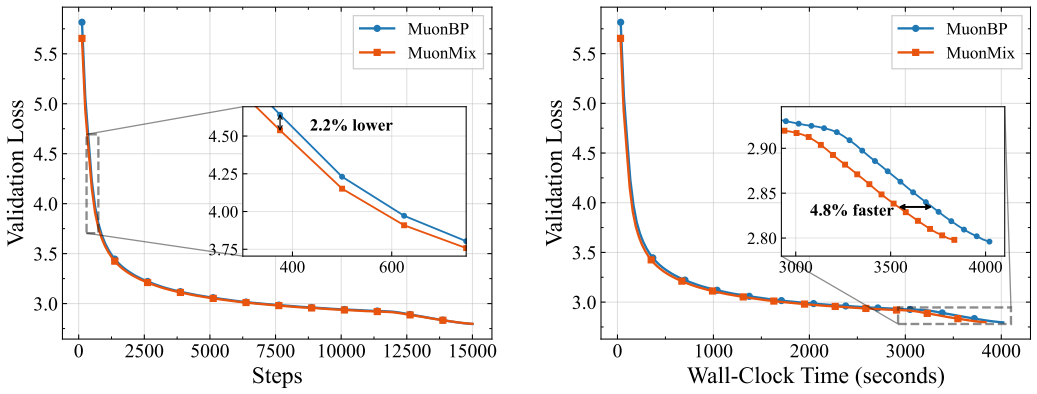
160M model (8 GPUs). MuonMix achieves a throughput of 245,453 tokens/s/GPU, 2.6% higher than MuonBP (239,182) and 7.3% higher than Moonlight (228,747). Compared to MuonBP, MuonMix achieves not only higher throughput but also faster early convergence, as evidenced by a 2.3% lower average validation loss over the first 400 training steps. We characterize performance with respect to two related metrics following (Khaled et al., 2025). Given a target validation loss, MuonMix reaches each target faster in wall-clock time than MuonBP. To reach a validation loss of 3.40, MuonMix requires 2,070 seconds while MuonBP requires 2,216 seconds, a 6.6% wall-clock time reduction. To reach 3.34, MuonMix requires 2,393 seconds against 2,494 seconds for MuonBP, a 4.0% wall-clock time reduction, as illustrated in Figure A.2a. Given a fixed number of training steps, MuonMix consistently achieves lower validation loss than MuonBP. At the end of training, MuonMix achieves a minimum validation loss of 3.284, compared to 3.293 for MuonBP. It indicates that replacing W_Q and W_K with AdamW partially recovers the convergence behavior of full orthogonalization on those layers.

Table A.5: Minimum validation loss, average validation loss over the first 400 training steps (early convergence), average throughput, and per-step training time for 160M model on 8 GPUs. Elementwise updates use AdamW.

Model	Method	Min Val Loss	Early Val Loss	Step Time (ms)	Tokens/s/GPU
160M	Moonlight	**3.268**	**4.479**	573	228,747
	MuonBP	3.293	4.600	548	239,182
	MuonMix	3.284	4.496	**534**	**245,453**

Step time estimation. We use Calculon (Isaev et al., 2023) to estimate step times for the 960M and 1.2B simulations. We follow the same method as for Signum (Appendix A.5), with AdamW for elementwise updates. Communication for global Newton–Schulz orthogonalization is excluded. Estimated step times and detailed computation and communication times are reported in Tables A.6 and A.7, respectively.

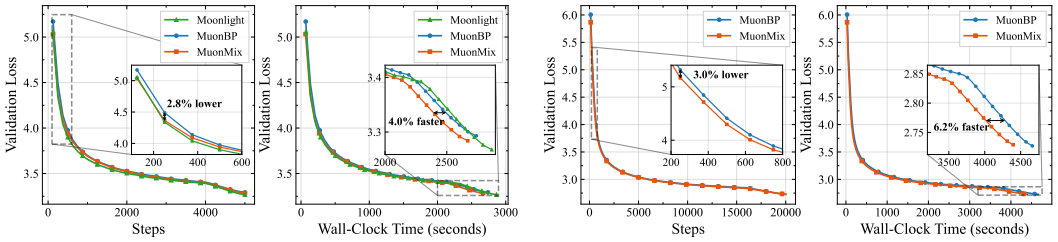
960M model (128 GPUs virtual simulation). We simulate 128 GPUs training on 4 physical GPUs based on virtual sharding. Results are summarized in Table A.6 and Figure A.3. MuonMix achieves a throughput of 31,974 tokens/s/GPU, 4.4% higher than MuonBP (30,613) under the simulated



(a) Validation loss vs. training steps.

(b) Validation loss vs. wall-clock time.

Figure A.1: 960M model training curves with 128GPUs simulation. Comparison of MuonBP and MuonMix across training steps and wall-clock time. Elementwise updates use Signum.



(a) 160M model (8GPUs): validation loss vs. steps (left) and wall-clock time (right).

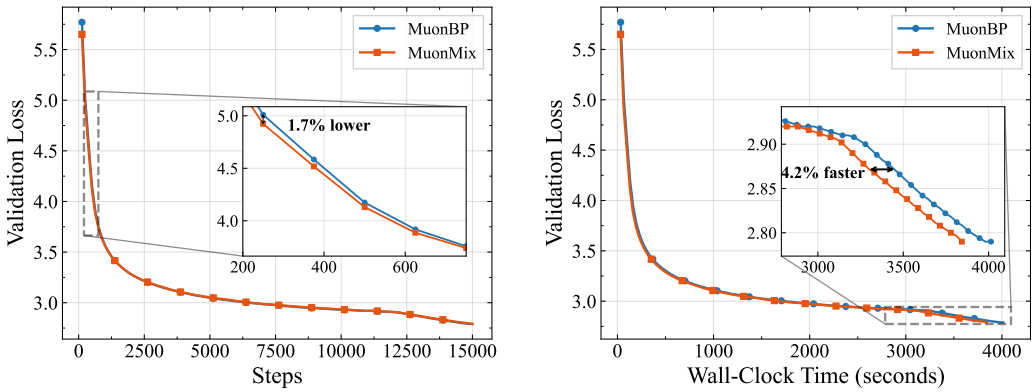
(b) 1.2B model (simulated 256 GPUs): validation loss vs. steps (left) and wall-clock time (right).

Figure A.2: Comparison of Moonlight, MuonBP, and MuonMix across training steps and wall-clock time. Elementwise updates use AdamW.

128 GPUs configuration. Compared to MuonBP, MuonMix also exhibits faster early convergence. The average validation loss over the first 1,000 training steps is 4.183, versus 4.228 for MuonBP, a 1.1% reduction. Given a target validation loss, MuonMix reaches it faster in wall-clock time than MuonBP. To reach a validation loss of 3.20, MuonMix requires 683 s against 707 s for MuonBP, a 3.4% reduction. In the late-training regime, to reach a validation loss of 2.87, MuonMix requires 3,302 s against 3,448 s for MuonBP, a 4.2% reduction. Given a fixed number of training steps, MuonMix completes 15,000 training steps in 3,843 s, approximately 171 s earlier than MuonBP (4,014 s), a 4.3% reduction. At the end of training, MuonBP achieves a minimum validation loss of 2.788, compared to 2.794 for MuonMix, giving a small validation loss gap of 0.006.

1.2B model (256 GPUs virtual simulation). The throughputs in Table A.6 are computed as tokens per step divided by step time and the number of simulated GPUs. MuonMix achieves a throughput of 18,634 tokens/s/GPU, 6.0% higher than MuonBP (17,571) under the simulated 256 GPUs configuration. Compared to MuonBP, MuonMix also exhibits faster early convergence. The average validation loss over the first 1,000 training steps is 4.294, versus 4.394 for MuonBP, a 2.3% reduction. Given a target validation loss, MuonMix reaches it faster in wall-clock time than MuonBP. To reach a validation loss of 3.20, MuonMix requires 577 seconds against 638 seconds for MuonBP, a 9.6% wall-clock time reduction. As illustrated in Figure A.2b, at a validation loss of 2.77, MuonMix requires 4,016 seconds compared to 4,283 seconds for MuonBP, a 6.2% wall-clock time reduction. Given a fixed number of training steps, MuonMix achieves the same minimum validation loss of 2.730 as MuonBP. [6]

[6]We note that the step time and throughput in Table A.6 are estimates for the target configurations with 128 or 256 GPUs and are not directly comparable to Table A.5.



(a) Validation loss vs. training steps.

(b) Validation loss vs. wall-clock time.

Figure A.3: 960M model training curves with 128GPUs simulation. Comparison of MuonBP and MuonMix across training steps and wall-clock time. Elementwise updates use AdamW.

Table A.6: Minimum validation loss, average validation loss over the first 1,000 training steps, estimated per-step computation, communication, and total step times, and throughput under virtual simulation (128 GPUs for 960M; 256 GPUs for 1.2B) conducted on 4 physical GPUs. Elementwise updates use AdamW.

Model	Method	Validation Loss		Comp. Time	Comm. Time	Step Time (ms)	Tokens/s/GPU
		Min	Early				
960M	MuonBP	**2.788**	4.228	126.41	141.19	267.60	30,613
	MuonMix	2.794	**4.183**	**115.02**	141.19	**256.21**	**31,974**
1.2B	MuonBP	**2.730**	4.394	104.32	128.79	233.11	17,571
	MuonMix	**2.730**	**4.294**	**91.02**	128.79	**219.81**	**18,634**

Table A.7: Computation and communication times under virtual simulation (128 GPUs for 960M; 256 GPUs for 1.2B). Elementwise updates use AdamW.

		Computation Time (ms)					Communication Time (ms)	
Model	Method	FW	BW	Global NS5	Block NS5	Elem. Update	TP	FSDP
960M	MuonBP	29.86	53.99	58.53	38.48	0.08	92.57	48.62
	MuonMix	29.86	53.99	45.54	25.61	1.57	92.57	48.62
1.2B	MuonBP	19.47	35.10	68.38	45.01	0.07	63.81	64.98
	MuonMix	19.47	35.10	53.43	29.93	1.82	63.81	64.98

Discussions. In the real 8 GPUs setting, model local shards remain relatively large, so the MuonBP block approximation error may not be clearly reflected in validation loss. We therefore use virtual sharding at higher parallelism degrees to study the behavior of MuonMix and MuonBP. Our analysis gives a partition-dependent worst-case bound for MuonBP. MuonMix avoids local orthogonalization error on its max-norm layers, while retaining MuonBP updates on its spectral-norm layers. In the Signum experiments, at the 128 GPUs virtual simulation, MuonBP retains a slight advantage in minimum validation loss (2.796 vs. 2.798), whereas at the 256 GPUs virtual simulation MuonMix achieves strictly lower minimum validation loss (2.731 vs. 2.734). We note that 4 physical GPUs cannot reproduce the communication latency of a true cluster with 128 or 256 GPUs, so the reported step times are estimates for the target configurations and exclude communication for global Newton–Schulz orthogonalization. In a real large-scale distributed deployment, where orthogonalization costs are larger, the communication benefit of our approach may become more pronounced, depending on the implementation and communication overlap.

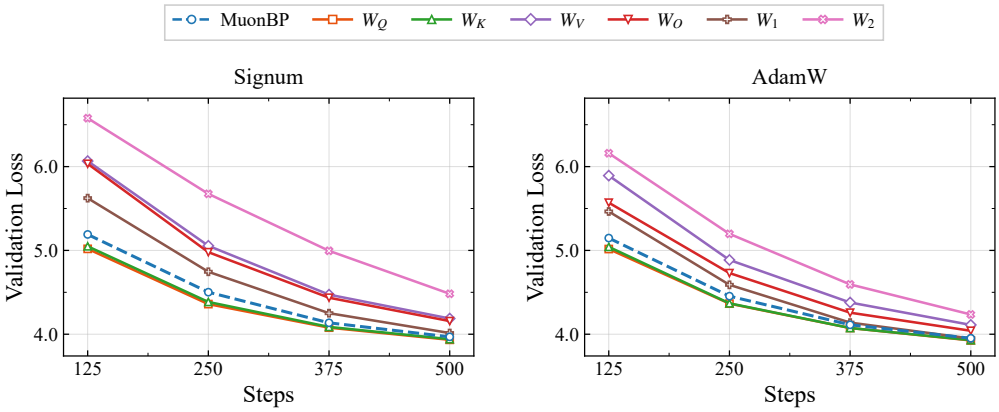


Figure A.4: Early validation-loss curves for the 160M layer-selection ablation on 4 B200 GPUs.

A.4 Ablation Study on Layer Selection

To evaluate the layer selection heuristic with different elementwise optimizers, we conduct an ablation study on the 160M model using 4 B200 GPUs. In each run, we update one hidden weight matrix with Signum or AdamW and retain MuonBP for all remaining hidden weight matrices. Each run lasts 3,000 training steps. These results are not directly comparable to the 8-GPU experiments in Tables 3 and A.5. Figure A.4 compares the validation loss during the first 500 training steps, while Figure A.5 presents the complete validation-loss curves over 3,000 training steps. Table A.8 summarizes the results. Under both Signum and AdamW, only W_Q and W_K reduce both the minimum validation loss over 3,000 steps and the average validation loss over the first 500 steps relative to the corresponding MuonBP baseline. These are also the two matrices with positive $\widehat{\Phi}_s^{\text{diff}}$ in Table 2. The results therefore support the same selection of W_Q and W_K under both optimizers when considering both loss metrics. All assignments reduce the step time. The elementwise updates avoid Newton–Schulz orthogonalization and its associated all-gather and scatter operations for the assigned matrix.

Table A.8: Ablation study of the assigned hidden weight matrix on the 160M model using 4 B200 GPUs. The profiling scores are from Table 2. Min val denotes the minimum validation loss over 3,000 steps, and Early val denotes the average validation loss over the first 500 steps.

		Signum			AdamW		
Assigned matrix	$\widehat{\Phi}_s^{\text{diff}}$	Min val	Early val	Step (ms)	Min val	Early val	Step (ms)
none (MuonBP)	—	3.445	4.449	1128	3.444	4.416	1049
W_Q	$\mathbf{+4.73 \times 10^{-3}}$	**3.440**	**4.347**	1073	**3.438**	**4.345**	1027
W_K	$\mathbf{+7.56 \times 10^{-3}}$	**3.442**	**4.365**	1027	**3.441**	**4.354**	1021
W_V	-4.67×10^{-2}	3.478	4.945	1054	3.451	4.817	1029
W_O	-6.32×10^{-2}	3.461	4.901	1038	3.448	4.649	1023
W_1	-3.07×10^{-2}	3.458	4.659	1063	3.416	4.535	1022
W_2	-2.19×10^{-1}	3.460	5.433	1062	3.453	5.047	1036

A.5 Step Time Estimation

We use Calculon (Isaev et al., 2023) to estimate forward and backward computation and tensor parallel (TP) communication for the target configurations with 128 or 256 GPUs. For fully sharded data parallel (FSDP) communication, we apply its network model to parameter all-gathers and gradient reduce-scatter. Optimizer computation is timed separately on a single NVIDIA B200 GPU using the matrix shapes of the target configurations. For each model, the Signum and AdamW experiments share the model dimensions, batch sizes, parallelism degrees, and hardware parameters. Optimizer times are obtained separately for each method with Signum and AdamW updates.

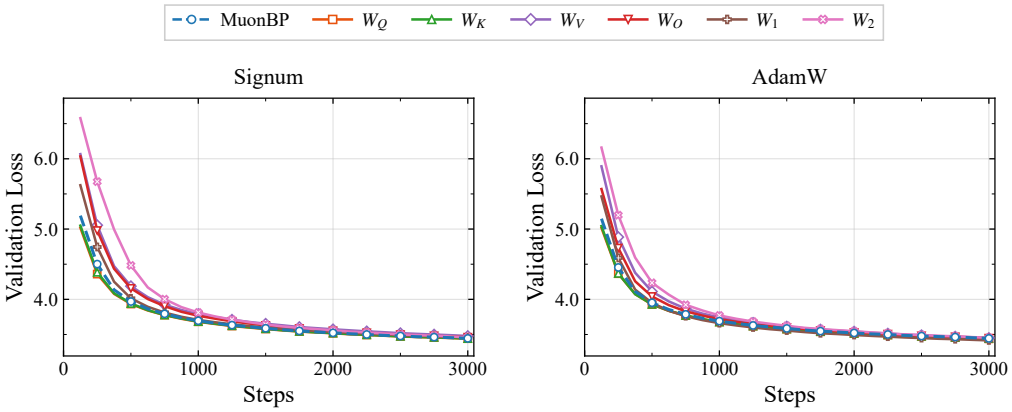


Figure A.5: Validation-loss curves for the 160M layer-selection ablation on 4 B200 GPUs over 3,000 training steps. Each curve assigns the indicated hidden weight matrix to Signum or AdamW.

Model and parallel configuration. Table A.9 lists the model and parallelism settings. With global batch size B and local microbatch size b, the number of microbatches per GPU per training step is $M = B/(bP_{\mathrm{FSDP}})$. GPUs within a TP group process the same microbatch. We use Calculon's float16 setting to represent BF16 storage, as both formats use two bytes per element.

Table A.9: Model and parallel configurations used in Calculon.

Parameter	960M	1.2B
Transformer blocks L	24	28
Hidden width d	1,664	1,792
Feedforward width f	6,656	7,168
Attention heads	16	16
Head dimension	104	112
Sequence length S	1,024	1,024
Simulated GPUs N	128	256
TP degree P_{TP}	8	8
FSDP degree P_{FSDP}	16	32
Global batch size B	1,024	1,024
Local microbatch size b	16	8
Microbatches per step M	4	4

B200 hardware configuration. Table A.10 gives the GPU specifications from the NVIDIA HGX B200 datasheet (Tirumala & Wong, 2024). We use 2,250 TFLOP/s for BF16 matrix multiplications, which is half of the 4.5 PFLOP/s sparse Tensor Core rate in the datasheet. For elementwise computation, including normalization and softmax, we use the 75 TFLOP/s FP32 rate. Matrix and elementwise computation therefore use separate peak rates in Calculon. Memory accesses use the 7,700 GB/s memory bandwidth. The efficiency factors below adjust both the computation rates and memory bandwidth for each operator.

Table A.10: NVIDIA B200 hardware parameters per GPU.

Parameter	Value
Dense BF16 matrix throughput P_{mat}	2,250 TFLOP/s
FP32 arithmetic throughput P_{vec}	75 TFLOP/s
Memory bandwidth B_{mem}	7,700 GB/s
Memory capacity	180 GB

Forward and backward computation. Calculon computes the operation count and memory access volume of each operator using the model dimensions and TP partitioning. For a matrix product of shapes $m \times n$ and $n \times k$, the forward operation count is $2mnk$. With two bytes per element, Calculon models the memory access volume as $2(mn + mk + nk)$ bytes. For the first feedforward matrix,

$m = bS$, $n = d$, and $k = f/P_{\mathrm{TP}}$. Calculon's released A100 and H100 configurations use the same efficiency factors. We retain these factors as a modeling assumption and replace the peak computation rates and memory bandwidth with the B200 values. We select efficiency factors from Table A.11 based on each operator's operation count and memory access volume. For the first feedforward matrix multiplication in the 1.2B forward pass, 26.307 GFLOP and 47.25 MB give factors of 0.90 and 0.75, which scale the peak computation rate and memory bandwidth, respectively.

Table A.11: Efficiency factors from the released Calculon configurations.

Resource	Lower thresholds	Efficiency factors
Matrix computation	$(0, 1, 16, 128)$ GFLOP	$(0.10, 0.60, 0.90, 0.95)$
Vector computation	$(0, 1, 16)$ GFLOP	$(0.10, 0.50, 0.95)$
Memory access	$(0, 1, 10, 100)$ MB	$(0.30, 0.60, 0.75, 0.90)$

For operator o at stage s, let $F_{o,s}$ be its floating point operation count and $V_{o,s}$ its memory access volume in bytes. We divide the operation count and memory access volume by the corresponding effective rates

$$t_{o,s}^{\mathrm{comp}} = \frac{F_{o,s}}{P_o \eta_o(F_{o,s})}, \qquad t_{o,s}^{\mathrm{mem}} = \frac{V_{o,s}}{B_{\mathrm{mem}} \eta_{\mathrm{mem}}(V_{o,s})}. \tag{A.1}$$

Here P_o is the matrix or vector peak rate, and η_o and η_{mem} are selected from Table A.11. Rates are expressed in operations or bytes per second. We assume no overlap between computation and memory access and add the two times as $t_{o,s} = t_{o,s}^{\mathrm{comp}} + t_{o,s}^{\mathrm{mem}}$. We sum the operator times within each block and multiply each stage time by LM to obtain its time per training step. Backward time is the sum of the activation gradient and weight gradient computation times. Table A.12 reports the times per block and microbatch and the resulting forward and backward times per step. TP communication is counted separately.

Table A.12: Estimated computation times per transformer block and microbatch and per training step.

Model	Per block and microbatch (μs)			LM	Per step (ms)	
	Forward	Activation gradient	Weight gradient		Forward	Backward
960M	311.0674	400.6456	161.7455	96	29.86	53.99
1.2B	173.8263	221.5445	91.8725	112	19.47	35.10

Optimizer computation. We run the optimizer operations on a single NVIDIA B200 GPU. Global Newton–Schulz orthogonalization uses full matrices. Blockwise orthogonalization and elementwise updates use the local matrix shapes required by the target TP and FSDP partitioning. We use five Newton–Schulz iterations (NS5). Table A.13 reports the resulting computation times. Global and block times are the medians over 1000 steps. For every $p = 5$ training steps, one step uses global orthogonalization and four use blockwise orthogonalization. The average optimizer time is

$$T_{\mathrm{opt}} = \frac{T_{\mathrm{global}} + (p-1)T_{\mathrm{block}}}{p} + T_{\mathrm{elem}}. \tag{A.2}$$

Elementwise updates run every step. They cover the embedding and output matrices for both methods and additionally W_Q and W_K for MuonMix.

Table A.13: Optimizer computation times on a single NVIDIA B200 GPU in milliseconds.

Model	Update	Method	Global NS5	Block NS5	Elem. Update	Time
960M	Signum	MuonBP	58.375	39.095	0.041	42.992
		MuonMix	45.415	25.988	0.776	30.650
	AdamW	MuonBP	58.527	38.475	0.076	42.562
		MuonMix	45.536	25.611	1.573	31.169
1.2B	Signum	MuonBP	68.353	45.237	0.039	49.900
		MuonMix	53.430	30.088	0.881	35.638
	AdamW	MuonBP	68.378	45.005	0.074	49.754
		MuonMix	53.431	29.933	1.821	36.454

TP and FSDP communication. Table A.14 gives the network parameters. TP communication is modeled within each node, while FSDP communication spans nodes. Calculon sums the TP collectives in the forward and backward passes over all transformer blocks and microbatches.

Table A.14: Network parameters used for communication time estimation.

Communication	Bandwidth (GB/s)	Efficiency	Latency (μs)
TP	900	0.65	10
FSDP	50	0.90	20

For FSDP, we use Calculon's network cost model with the collective settings from its released configuration. The estimated time for a parameter all-gather or gradient reduce-scatter with volume x bytes is

$$t(x) = \alpha + \frac{x}{\beta\eta}, \tag{A.3}$$

where α is latency, β is bandwidth, and η is bandwidth efficiency. For the six linear layers in one block, the parameter volume assigned to one TP rank before FSDP sharding is $W = 2(4d^2 + 2df)/P_{\mathrm{TP}}$ bytes in BF16. The FP32 gradient volume is $2W$. For each block, we count parameter all-gathers before both the forward and backward passes of every microbatch and one gradient reduce-scatter per training step

$$T_{\mathrm{FSDP}} = L\left[M\, t_{\mathrm{ag}}(W) + M\, t_{\mathrm{ag}}(W) + t_{\mathrm{rs}}(2W)\right]. \tag{A.4}$$

The first two terms correspond to forward and backward parameter all-gathers. The last term is gradient reduce-scatter. Table A.15 reports the resulting times.

Table A.15: Estimated communication times per training step in milliseconds. Parameter volume W is reported in MB. AG and RS denote all-gather and reduce-scatter.

Model	W (MB)	FW AG	BW AG	Gradient RS	FSDP	TP
960M	8.307	19.641	19.641	9.340	48.622	92.570
1.2B	9.634	26.217	26.217	12.549	64.984	63.806

Total step time. We add computation and communication times without overlap

$$T_{\mathrm{step}} = T_{\mathrm{FW}} + T_{\mathrm{BW}} + T_{\mathrm{opt}} + T_{\mathrm{TP}} + T_{\mathrm{FSDP}}. \tag{A.5}$$

Global Newton–Schulz communication is excluded because we lack a reliable estimate at the target scale. The forward, backward, TP, and FSDP times are shared by MuonBP and MuonMix for the same model configuration.

B Proof of Theorem 1

There are H layers in total, indexed by $\ell \in \{1, \dots, H\}$. Each layer ℓ is associated with its own smoothness constant L_ℓ, gradient noise level σ_ℓ, per-layer step-size η_t^ℓ, and a layer-specific norm $\|\cdot\|_\ell$ whose dual is $\|\cdot\|_{\ell*}$. The full parameter vector decomposes as $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(H)}]$, the gradient as $\nabla f(\mathbf{x}) = [\nabla_1 f(\mathbf{x}), \dots, \nabla_H f(\mathbf{x})]$, and the stochastic gradient as $\mathbf{G}_t = [\mathbf{G}_t^{(1)}, \dots, \mathbf{G}_t^{(H)}]$. We partition the $H = N + S$ layers into two groups according to their optimizer type and primal–dual norm pair. Layers $\ell \in \{1, \dots, N\}$ are updated by the *MuonBP* (Newton–Schulz orthogonalization) rule, which operates under the spectral–nuclear norm pair. Layers $\ell \in \{1, \dots, S\}$ are updated via the *elementwise sign*, which operates under the max–ℓ_1 norm pair. Let ℓ denote the layer index within each group. When $S = H$ and $N = 0$, the MuonBP group is empty and the analysis reduces to the elementwise sign group alone. When $N = H$ and $S = 0$, we recover all-MuonBP layers update.

Notation. Let T and p be positive integers with $p \mid T$. We define

$$S_{\mathrm{op}} = \big\{t \in \{0, \dots, T-1\} : t \bmod p = 0\big\}, \quad |S_{\mathrm{op}}| = T/p,$$
$$S_B = \big\{t \in \{0, \dots, T-1\} : t \bmod p \neq 0\big\}, \quad |S_B| = T(p-1)/p.$$

(A) MuonBP layers (N layers, spectral–nuclear norm). The analysis treats MuonBP as an exact spectral-norm LMO, whose objective value is the negative nuclear norm. For each MuonBP layer $\ell \in \{1, \dots, N\}$, the primal norm for full synchronization steps is the spectral norm $\|\cdot\|_{\mathrm{sp}}$, and the primal norm for local block steps is the per-layer block spectral norm

$$B_\ell^{\mathrm{mu}}(\cdot) = \max_m \|[\cdot]^{(m)}\|_{\mathrm{sp}}, \qquad \ell \in \{1, \dots, N\}. \tag{B.1}$$

The corresponding dual norms are the per-layer nuclear norm $\|\cdot\|_{\ell,*}^{\mathrm{nuc}}$ for full steps and $(B_\ell^{\mathrm{mu}})^*(\cdot) = \sum_m \|[\cdot]^{(m)}\|_{\ell,*}^{\mathrm{nuc}}$ for local block steps, where $(\|\cdot\|_{\mathrm{sp}})^* = \|\cdot\|_{\ell,*}^{\mathrm{nuc}}$ applied blockwise. For the row–column partition,

$$B_\ell^{\mathrm{mu}}(X) \le \|X\|_{\mathrm{sp}} \le \sqrt{rc}\, B_\ell^{\mathrm{mu}}(X), \qquad \|X\|_{\ell,*}^{\mathrm{nuc}} \le (B_\ell^{\mathrm{mu}})^*(X).$$

The first two inequalities follow from block restriction and Cauchy–Schwarz, respectively; the last follows by duality. The per-layer smoothness constants are $L_{\ell,\mathrm{mu}}^{\mathrm{op}} > 0$ (under $\|\cdot\|_{\mathrm{sp}}$, full steps) and $L_{\ell,\mathrm{mu}}^{B} > 0$ (under $B_\ell^{\mathrm{mu}}(\cdot)$, local block steps), for $\ell \in \{1, \dots, N\}$. The per-layer step-sizes are $\eta_{\mathrm{full},\ell}^{\mathrm{mu}}$ and $\eta_{\mathrm{block},\ell}^{\mathrm{mu}}$, with

$$\eta_{\mathrm{max},\ell}^{\mathrm{mu}} = \max\{\eta_{\mathrm{full},\ell}^{\mathrm{mu}}, \eta_{\mathrm{block},\ell}^{\mathrm{mu}}\}, \quad \alpha_\ell^{\mathrm{mu}} = \frac{\eta_{\mathrm{full},\ell}^{\mathrm{mu}}}{p} + \frac{(p-1)\,\eta_{\mathrm{block},\ell}^{\mathrm{mu}}}{p}, \quad \ell \in \{1, \dots, N\}. \tag{B.2}$$

The momentum parameter for MuonBP layers is $\mu \in [0, 1)$. Norm-equivalence constants $C_1^{\mathrm{mu}}, C_2^{\mathrm{mu}} > 0$ satisfy $C_1^{\mathrm{mu}}\|A\|_{\ell,*}^{\mathrm{nuc}} \le \|A\|_F \le C_2^{\mathrm{mu}}\|A\|_{\ell,*}^{\mathrm{nuc}}$ for all matrices A and all $\ell \in \{1, \dots, N\}$ (see Lemma A.3 of (Hao et al., 2025)). These constants are chosen uniformly over all MuonBP layers and both the full and block nuclear dual norms. We choose C_1^{mu} no larger than any corresponding lower norm-equivalence constant and C_2^{mu} no smaller than any corresponding upper constant. The harmonic-average smoothness constant for the MuonBP group is

$$\left(\widehat{L}_{BP}^{\mathrm{mu}}\right)^{-1} = \frac{1}{p}\left(\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}\right)^{-1} + \frac{p-1}{p}\left(\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}\right)^{-1}. \tag{B.3}$$

(B) Layers updated by elementwise sign (S layers, max–ℓ_1 norm).

For each layer $\ell \in \{1, \dots, S\}$, the primal norm for full synchronization steps is the ℓ_∞ max norm $\|\cdot\|_\infty$, and the primal norm for local block steps is the per-layer block max norm

$$B_\ell^{\mathrm{sg}}(\cdot) = \max_m \|[\cdot]^{(m)}\|_\infty, \qquad \ell \in \{1, \dots, S\}. \tag{B.4}$$

The corresponding dual norms are the per-layer ℓ_1 norm $\|\cdot\|_{\ell,*}^1$ for full steps and $(B_\ell^{\mathrm{sg}})^*(\cdot) = \sum_m \|[\cdot]^{(m)}\|_{\ell,*}^1$ for local block steps, where $(\|\cdot\|_\infty)^* = \|\cdot\|_{\ell,*}^1$ applied blockwise. In Assumption 1, we denoted that $\|\cdot\|_\ell$ is the max norm $\|\cdot\|_\infty$ with $L_{\ell,\mathrm{sg}}$. But here we denote $L_{\ell,\mathrm{sg}}^{\mathrm{op}} > 0$ (under $\|\cdot\|_\infty$, full steps) and $L_{\ell,\mathrm{sg}}^{B} > 0$ (under $B_\ell^{\mathrm{sg}}(\cdot)$, local block steps), for $\ell \in \{1, \dots, S\}$ just for notational consistency with MuonBP terms. They are equal to $L_{\ell,\mathrm{sg}}$. The per-layer step-sizes are $\eta_{\mathrm{full},\ell}^{\mathrm{sg}}$ and $\eta_{\mathrm{block},\ell}^{\mathrm{sg}}$, with:

$$\eta_{\mathrm{max},\ell}^{\mathrm{sg}} = \max\{\eta_{\mathrm{full},\ell}^{\mathrm{sg}}, \eta_{\mathrm{block},\ell}^{\mathrm{sg}}\}, \quad \alpha_\ell^{\mathrm{sg}} = \frac{\eta_{\mathrm{full},\ell}^{\mathrm{sg}}}{p} + \frac{(p-1)\,\eta_{\mathrm{block},\ell}^{\mathrm{sg}}}{p}, \quad \ell \in \{1, \dots, S\}. \tag{B.5}$$

The momentum parameter is $\beta \in [0, 1)$. Norm-equivalence constants $C_1^{\mathrm{sg}}, C_2^{\mathrm{sg}} > 0$ satisfy $C_1^{\mathrm{sg}}\|A\|_{\ell,*}^1 \le \|A\|_F \le C_2^{\mathrm{sg}}\|A\|_{\ell,*}^1$ for all $\ell \in \{1, \dots, S\}$. The harmonic-average smoothness constant for the elementwise sign group is

$$\left(\widehat{L}_{BP}^{\mathrm{sg}}\right)^{-1} = \frac{1}{p}\left(\sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}}\right)^{-1} + \frac{p-1}{p}\left(\sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B}\right)^{-1}. \tag{B.6}$$

We first state the notation in a per-layer form, since the layerwise descent argument applies to layer-dependent step sizes. Here, we specialize to uniform full and local block step sizes. That is, for all MuonBP layers $\ell \in \{1, \dots, N\}$, $\eta_{\mathrm{full},\ell}^{\mathrm{mu}} = \eta_{\mathrm{full}}^{\mathrm{mu}}$, $\eta_{\mathrm{block},\ell}^{\mathrm{mu}} = \eta_{\mathrm{block}}^{\mathrm{mu}}$, and for all S layers $\ell \in \{1, \dots, S\}$, $\eta_{\mathrm{full},\ell}^{\mathrm{sg}} = \eta_{\mathrm{full}}^{\mathrm{sg}}$, and $\eta_{\mathrm{block},\ell}^{\mathrm{sg}} = \eta_{\mathrm{block}}^{\mathrm{sg}}$.

(C) Per-layer momentum errors.

For $t \geq 1$, the momentum is maintained locally on each device m

$$\mathbf{B}_t^{(\ell,m)} = \mu\,\mathbf{B}_{t-1}^{(\ell,m)} + (1-\mu)\big[\mathbf{G}^\ell(\mathbf{x}_t)\big]^{(m)}, \qquad \ell \in \{1,\dots,N\}, \tag{B.7}$$

$$\mathbf{B}_t^{(\ell,m)} = \beta\,\mathbf{B}_{t-1}^{(\ell,m)} + (1-\beta)\big[\mathbf{G}^\ell(\mathbf{x}_t)\big]^{(m)}, \qquad \ell \in \{1,\dots,S\}. \tag{B.8}$$

The per-layer momentum errors are

$$\hat{\boldsymbol{\epsilon}}_t^\ell = \mathbf{B}_t^\ell - \nabla_\ell f(\mathbf{x}_t), \qquad \ell \in \{1,\dots,N\} \text{ (MuonBP)}, \tag{B.9}$$

$$\hat{\boldsymbol{\epsilon}}_t^\ell = \mathbf{B}_t^\ell - \nabla_\ell f(\mathbf{x}_t), \qquad \ell \in \{1,\dots,S\} \text{ (elementwise sign)}. \tag{B.10}$$

Joint momentum-error event. Let $e_i^\ell = \mathbf{G}^\ell(\mathbf{x}_i) - \nabla_\ell f(\mathbf{x}_i)$ and let $q = \mu$ or β for the corresponding group g. For each fixed layer and $1 \leq t \leq T-1$, the matrices $(1-q)q^{t-i}e_i^\ell$, $1 \leq i \leq t$, are martingale differences in Frobenius space by Assumption 2. Their squared almost-sure bounds sum to at most $(C_2^g\sigma_\ell)^2(1-q)^2\sum_{i=1}^t q^{2(t-i)} \leq (C_2^g\sigma_\ell)^2(1-q)$. Apply Lemma 2.4 of (Liu et al., 2023) with failure probability $\delta/(HT)$ to each layer–time pair and take a union bound. On a single event of probability at least $1-\delta$,

$$\left\|(1-q)\sum_{i=1}^t q^{t-i}e_i^\ell\right\|_{\ell*} \leq \frac{4C_2^g}{C_1^g}\sigma_\ell\sqrt{(1-q)\log(2HT/\delta)}$$

simultaneously for all layers, times, and the full and block dual norms. Indeed the same Frobenius event controls both norms. The recursion is

$$\hat{\epsilon}_t^\ell = q^t e_0^\ell + (1-q)\sum_{i=1}^t q^{t-i}e_i^\ell + \sum_{i=1}^t q^{t-i+1}\big(\nabla_\ell f(\mathbf{x}_{i-1}) - \nabla_\ell f(\mathbf{x}_i)\big).$$

Assumption 1 bounds the last term by $L_\ell A_\ell/(1-q)$ whenever $\|\mathbf{x}_i^\ell - \mathbf{x}_{i-1}^\ell\|_\ell \leq A_\ell$. The initial term is at most $q^t\sigma_\ell$ in every required norm. All subsequent probability bounds use this common event; $q^0 = 1$, including when $q = 0$.

Proof strategy and adaptation. The following proof follows the strategy of Hao et al. (Hao et al., 2025, Appendix C) for the common layerwise LMO descent and momentum error analysis. We adapt it to the block-periodic distributed setting by introducing separate full-step and block-step norms and by separating the MuonBP and elementwise sign layer contributions.

B.1 Analysis for Max-Norm LMO on Local Block Step

Remark 1. *Here we consider the all-max-norm case ($S = H$); the mixed-group contributions are treated in the following subsections. Although the local block step is* physically executed *entirely at the shard level on each layer ℓ with no inter-device communication, the descent inequality for layer ℓ must be* established *in the full-matrix space. We explain why this is necessary and how the step levels are rigorously connected for each layer ℓ.*

Shard-level analysis for layer ℓ. *Each device m executes independently on layer ℓ:*

$$[\mathbf{x}_{t+1}^\ell]^{(m)} = [\mathbf{x}_t^\ell]^{(m)} + \eta_{\text{block}}^\ell\,\mathrm{LMO}_{B_\ell}\big([\mathbf{B}_t^\ell]^{(m)}\big). \tag{B.11}$$

However, the objective function f takes the full *parameter tensor (all layers, all shards) as input*

$$f(\mathbf{x}_{t+1}) = f\big(\big\{[\mathbf{x}_{t+1}^\ell]^{(m)}\big\}_{\ell,m}\big). \tag{B.12}$$

Therefore, $f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t)$ depends on the joint effect of all shard-level updates (B.11) *across all layers ℓ and all devices m. We define $B_\ell^*(\cdot) = \sum_m \|[\cdot]^{(m)}\|_1$ is the dual of $B_\ell(\cdot) = \max_m \|[\cdot]^{(m)}\|_\infty$.*

Equivalence to full-matrix update for layer ℓ. *Define the per-layer blockwise LMO operator $\mathrm{LMO}_{B_\ell} : \mathbb{R}^{d_1\times d_2} \to \mathbb{R}^{d_1\times d_2}$ by specifying its action on each shard*

$$\big[\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big]^{(m)} = \mathrm{LMO}_{B_\ell}\big([\mathbf{B}_t^\ell]^{(m)}\big), \qquad \forall\, m \in \{1,\dots,M\}. \tag{B.13}$$

That is, $\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)$ is the full layer-ℓ matrix whose m-th shard is exactly the result of applying the shard-level LMO to $[\mathbf{B}_t^\ell]^{(m)}$.

We verify that the M independent shard updates (B.11) *are jointly equivalent to a single full-matrix update for layer ℓ. For any shard index m,*

$$\begin{aligned}\big[\mathbf{x}_t^\ell + \eta_{\text{block}}^\ell \,\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big]^{(m)} &= [\mathbf{x}_t^\ell]^{(m)} + \eta_{\text{block}}^\ell \,\big[\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big]^{(m)} \\ &\overset{\text{(B.13)}}{=} [\mathbf{x}_t^\ell]^{(m)} + \eta_{\text{block}}^\ell \,\mathrm{LMO}_{B_\ell}\big([\mathbf{B}_t^\ell]^{(m)}\big) \\ &\overset{\text{(B.11)}}{=} [\mathbf{x}_{t+1}^\ell]^{(m)}. \end{aligned} \tag{B.14}$$

Since (B.14) *holds for all m, defining*

$$\mathbf{x}_{t+1}^\ell = \mathbf{x}_t^\ell + \eta_{\text{block}}^\ell \,\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell) \tag{B.15}$$

is mathematically equivalent to the collection of shard updates (B.11)*. We state that $\mathbf{x}_{t+1}^\ell$ as a full matrix of layer ℓ* never physically exists *on any single device. Then,* (B.15) *is a mathematical identity introduced solely to enable the application of L_ℓ^B-smoothness.*

By definition of the per-layer block max norm $B_\ell(\cdot) = \max_m \|[\cdot]^{(m)}\|_\infty$,

$$\begin{aligned} B_\ell\big(\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big) &= \max_m \big\|\big[\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big]^{(m)}\big\|_\infty \\ &\overset{\text{(B.13)}}{=} \max_m \big\|\mathrm{LMO}_{B_\ell}\big([\mathbf{B}_t^\ell]^{(m)}\big)\big\|_\infty \;\le\; 1, \end{aligned} \tag{B.16}$$

where the last inequality holds because each shard-level LMO satisfies $\|\mathrm{LMO}_{B_\ell}([\mathbf{B}_t^\ell]^{(m)})\|_\infty \le 1$ by definition (the LMO output lies in the unit ball of $B_\ell(\cdot)$, which implies $\|[\cdot]^{(m)}\|_\infty \le 1$ for each shard). Therefore,

$$\|\mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell\|_{B_\ell} = \eta_{\text{block}}^\ell \, B_\ell\big(\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big) \overset{\text{(B.16)}}{\le} \eta_{\text{block}}^\ell. \tag{B.17}$$

Since $\mathbf{x}_{t+1}^\ell$ in (B.15) *is a full matrix for layer ℓ and $\|\mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell\|_{B_\ell} \le \eta_{\text{block}}^\ell$ by* (B.17)*, Assumption 1, integrated along the joint update segment, gives*

$$\begin{aligned} f(\mathbf{x}_{t+1}) &\le f(\mathbf{x}_t) + \sum_{\ell=1}^H \big\langle \nabla_\ell f(\mathbf{x}_t),\, \mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell \big\rangle + \sum_{\ell=1}^H \frac{L_\ell^B}{2} \,\|\mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell\|_{B_\ell}^2 \\ &\le f(\mathbf{x}_t) + \sum_{\ell=1}^H \big\langle \nabla_\ell f(\mathbf{x}_t),\, \mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell \big\rangle + \sum_{\ell=1}^H \frac{L_\ell^B}{2} \big(\eta_{\text{block}}^\ell\big)^2. \end{aligned} \tag{B.18}$$

By the definition of LMO_{B_ℓ} and the dual norm: $\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell) \in \arg\min_{\|D\|_{B_\ell} \le 1} \langle \mathbf{B}_t^\ell, D\rangle$,

$$\big\langle \mathbf{B}_t^\ell,\, \mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big\rangle = -B_\ell^*(\mathbf{B}_t^\ell) = -\sum_{m=1}^M \big\|[\mathbf{B}_t^\ell]^{(m)}\big\|_1. \tag{B.19}$$

By Hölder's inequality $\langle u, v\rangle \le \|u\|_{B_\ell^*}\|v\|_{B_\ell}$ and (B.16)*,*

$$\begin{aligned} \big\langle \nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell,\, \mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big\rangle &\le B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell\big) \cdot B_\ell\big(\mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big) \\ &\overset{\text{(B.16)}}{\le} B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell\big). \end{aligned} \tag{B.20}$$

$$\big\langle \nabla_\ell f(\mathbf{x}_t),\, \mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big\rangle \le -B_\ell^*(\mathbf{B}_t^\ell) + B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell\big). \tag{B.21}$$

By the reverse triangle inequality for $B_\ell^*(\cdot)$,

$$B_\ell^*(\mathbf{B}_t^\ell) \ge B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t)\big) - B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell\big). \tag{B.22}$$

$$\big\langle \nabla_\ell f(\mathbf{x}_t),\, \mathrm{LMO}_{B_\ell}(\mathbf{B}_t^\ell)\big\rangle \le -B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t)\big) + 2B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell\big). \tag{B.23}$$

Summing over $\ell = 1, \dots, H$,

$$f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) - \sum_{\ell=1}^{H} \eta_{\text{block}}^{\ell} \Big[B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t)\big) - 2B_\ell^*\big(\nabla_\ell f(\mathbf{x}_t) - \mathbf{B}_t^\ell\big) - \tfrac{1}{2} L_\ell^B \eta_{\text{block}}^{\ell} \Big]. \tag{B.24}$$

$$B_\ell^*(X^\ell) = \sum_m \|[X^\ell]^{(m)}\|_1 = \|X^\ell\|_1 = \|X^\ell\|_*, \qquad \forall\, X^\ell \in \mathbb{R}^{d_1 \times d_2}. \tag{B.25}$$

Therefore,

$$f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) - \sum_{\ell=1}^{H} \eta_{\text{block}}^{\ell} \|\nabla_\ell f(\mathbf{x}_t)\|_* + 2\sum_{\ell=1}^{H} \eta_{\text{block}}^{\ell} \|\mathbf{B}_t^\ell - \nabla_\ell f(\mathbf{x}_t)\|_* + \sum_{\ell=1}^{H} \frac{L_\ell^B \big(\eta_{\text{block}}^{\ell}\big)^2}{2}. \tag{B.26}$$

B.2 Layerwise Descent Analysis for Spectral-Norm LMO

Since f depends on the parameters of *all* $H = N + S$ layers simultaneously, the smoothness-based quadratic upper bound must be applied to the full parameter space. For any step t,

$$\begin{aligned} f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) &+ \sum_{\ell=1}^{N} \big\langle \nabla_\ell f(\mathbf{x}_t),\, \mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell \big\rangle + \sum_{\ell=1}^{N} \frac{L_{\ell,\text{mu}}^{\text{op}}}{2} \big\|\mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell\big\|_{\text{sp}}^2 \\ &+ \sum_{\ell=1}^{S} \big\langle \nabla_\ell f(\mathbf{x}_t),\, \mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell \big\rangle + \sum_{\ell=1}^{S} \frac{L_{\ell,\text{sg}}^{\text{op}}}{2} \big\|\mathbf{x}_{t+1}^\ell - \mathbf{x}_t^\ell\big\|_\infty^2. \end{aligned} \tag{B.27}$$

In the following two subsections we bound each group's contribution to the right-hand side of (B.27) separately. The complete one-step descent inequality is then assembled in Section B.4.

Remark 2. *The bounds derived in this subsection for the MuonBP group and in Section B.3 for the elementwise sign group are* contributions to the right-hand side of (B.27)*, not independent descent inequalities. Write D_t^{mu} and D_t^{sg} for the respective sums of linear and quadratic terms, using the full-step or block-step norms and smoothness constants as appropriate. Then $f(\mathbf{x}_{t+1}) \le f(\mathbf{x}_t) + D_t^{\text{mu}} + D_t^{\text{sg}}$. The complete one-step descent inequality is obtained only after adding these two contributions, and is given in* (B.63) *and* (B.64)*.*

(A) Full step for MuonBP layers (spectral norm $\|\cdot\|_{\text{sp}}$, $t \in S_{\text{op}}$). On a full synchronization step, all devices gather their local per-layer buffer shards to form the complete per-layer momentum $\mathbf{B}_t^\ell = \text{concat}_m(\{\mathbf{B}_t^{(\ell,m)}\}_m)$, apply the global per-layer LMO under $\|\cdot\|_{\text{sp}}$ (the exact spectral-norm oracle), and update each MuonBP layer

$$\mathbf{x}_{t+1}^\ell = \mathbf{x}_t^\ell + \eta_{\text{full},\ell}^{\text{mu}}\, \text{LMO}_{\text{sp}}^\ell(\mathbf{B}_t^\ell), \qquad \forall\, \ell \in \{1, \dots, N\}. \tag{B.28}$$

By layerwise $L_{\ell,\text{mu}}^{\text{op}}$-smoothness under $\|\cdot\|_{\text{sp}}$ and the per-layer LMO descent argument, with all $\|\cdot\|_*$ replaced by $\|\cdot\|_{\ell,*}^{\text{nuc}}$ and $\|\cdot\|_\infty$ replaced by $\|\cdot\|_{\text{sp}}$, the full-step MuonBP contribution satisfies

$$D_t^{\text{mu}} \le -\sum_{\ell=1}^{N} \eta_{\text{full},\ell}^{\text{mu}} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\text{nuc}} + 2\sum_{\ell=1}^{N} \eta_{\text{full},\ell}^{\text{mu}} \|\hat{\boldsymbol{\epsilon}}_t^\ell\|_{\ell,*}^{\text{nuc}} + \frac{1}{2}\sum_{\ell=1}^{N} L_{\ell,\text{mu}}^{\text{op}} \big(\eta_{\text{full},\ell}^{\text{mu}}\big)^2. \tag{B.29}$$

From the per-layer momentum error recursion for MuonBP layers, with $L_{\ell,\text{mu}}^{\text{op}}$-smoothness under $\|\cdot\|_{\text{sp}}$, momentum parameter μ, effective step-size bound $\eta_{\text{op},\tau,\ell}^{\text{mu}} = \max\{\eta_{\text{full}}^{\text{mu}},\, \eta_{\text{block}}^{\text{mu}}\sqrt{rc}\}$ derived in (Khaled et al., 2025), with probability at least $1 - \delta$ for all $0 \le t < T$, we apply Lemma 2.4 of (Liu et al., 2023) to get

$$\big\|\hat{\boldsymbol{\epsilon}}_t^\ell\big\|_{\ell,*}^{\text{nuc}} \le \mu^t\, \sigma_\ell + \frac{4C_2^{\text{mu}}}{C_1^{\text{mu}}}(1-\mu)\,\sigma_\ell \sqrt{\frac{\log(2HT/\delta)}{1-\mu}} + \frac{L_{\ell,\text{mu}}^{\text{op}}\, \eta_{\text{max},\ell}^{\text{mu}} \sqrt{rc}}{1-\mu}, \quad \ell \in \{1, \dots, N\}, \tag{B.30}$$

where $\sqrt{rc} = \sqrt{M}$ and M is the number of devices.

Summing (B.30) over $\ell = 1, \dots, N$,

$$\sum_{\ell=1}^{N} \big\|\hat{\boldsymbol{\epsilon}}_t^{\ell}\big\|_{\ell,*}^{\mathrm{nuc}} \le \mu^t \sum_{\ell=1}^{N} \sigma_\ell + \frac{4C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}(1-\mu)\sqrt{\frac{\log(2HT/\delta)}{1-\mu}} \sum_{\ell=1}^{N} \sigma_\ell + \frac{\sqrt{rc}}{1-\mu} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}\, \eta_{\max,\ell}^{\mathrm{mu}}. \tag{B.31}$$

Then,

$$\begin{aligned} D_t^{\mathrm{mu}} \le -\eta_{\mathrm{full}}^{\mathrm{mu}} \sum_{\ell=1}^{N} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + 2\eta_{\mathrm{full}}^{\mathrm{mu}}\, \mu^t \sum_{\ell=1}^{N} \sigma_\ell + \frac{2\eta_{\mathrm{full}}^{\mathrm{mu}}\, \eta_{\max}^{\mathrm{mu}}}{1-\mu} \sqrt{rc} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} \\ + \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\, \eta_{\mathrm{full}}^{\mathrm{mu}}\, (1-\mu)\sqrt{\frac{\log(2HT/\delta)}{1-\mu}} \sum_{\ell=1}^{N} \sigma_\ell + \frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}. \end{aligned} \tag{B.32}$$

(B) Local Block step for MuonBP layers (block spectral norm $B_\ell^{\mathrm{mu}}(\cdot)$, $t \in S_B$). On a local block step, each device m executes independently for each MuonBP layer $\ell \in \{1, \dots, N\}$,

$$[\mathbf{x}_{t+1}^{\ell}]^{(m)} = [\mathbf{x}_t^{\ell}]^{(m)} + \eta_{\mathrm{block},\ell}^{\mathrm{mu}}\, \mathrm{LMO}_{B_\ell^{\mathrm{mu}}}\big([\mathbf{B}_t^{\ell}]^{(m)}\big), \tag{B.33}$$

where $\mathrm{LMO}_{B_\ell^{\mathrm{mu}}}([\mathbf{B}_t^{\ell}]^{(m)}) \in \arg\min_{\|D\|_{\mathrm{sp}} \le 1} \langle [\mathbf{B}_t^{\ell}]^{(m)}, D \rangle$ is the per-shard LMO under $B_\ell^{\mathrm{mu}}(\cdot)$.

Similarly, the block-step MuonBP contribution satisfies

$$D_t^{\mathrm{mu}} \le \sum_{\ell=1}^{N} \Big[-\eta_{\mathrm{block},\ell}^{\mathrm{mu}} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + 2\eta_{\mathrm{block},\ell}^{\mathrm{mu}} (B_\ell^{\mathrm{mu}})^*(\hat{\epsilon}_t^{\ell}) + \tfrac{1}{2} L_{\ell,\mathrm{mu}}^{B} \big(\eta_{\mathrm{block},\ell}^{\mathrm{mu}}\big)^2 \Big]. \tag{B.34}$$

From the effective step-size bound $\eta_{\mathrm{block},\tau,\ell}^{\mathrm{mu}} = \eta_{\max,\ell}^{\mathrm{mu}}$, with probability at least $1-\delta$ for all $0 \le t < T$,

$$(B_\ell^{\mathrm{mu}})^*(\hat{\epsilon}_t^{\ell}) \le \mu^t\, \sigma_\ell + \frac{4C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}(1-\mu)\, \sigma_\ell \sqrt{\frac{\log(2HT/\delta)}{1-\mu}} + \frac{L_{\ell,\mathrm{mu}}^{B}\, \eta_{\max,\ell}^{\mathrm{mu}}}{1-\mu}, \quad \ell \in \{1, \dots, N\}. \tag{B.35}$$

Summing (B.35) over $\ell = 1, \dots, N$,

$$\sum_{\ell=1}^{N} (B_\ell^{\mathrm{mu}})^*(\hat{\epsilon}_t^{\ell}) \le \mu^t \sum_{\ell=1}^{N} \sigma_\ell + \frac{4C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}(1-\mu)\sqrt{\frac{\log(2HT/\delta)}{1-\mu}} \sum_{\ell=1}^{N} \sigma_\ell + \frac{1}{1-\mu} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}\, \eta_{\max,\ell}^{\mathrm{mu}}. \tag{B.36}$$

Substituting (B.35) into (B.34),

$$\begin{aligned} D_t^{\mathrm{mu}} \le -\eta_{\mathrm{block}}^{\mathrm{mu}} \sum_{\ell=1}^{N} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + 2\eta_{\mathrm{block}}^{\mathrm{mu}}\, \mu^t \sum_{\ell=1}^{N} \sigma_\ell + \frac{2\eta_{\mathrm{block}}^{\mathrm{mu}}\, \eta_{\max}^{\mathrm{mu}}}{1-\mu} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \\ + \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\, \eta_{\mathrm{block}}^{\mathrm{mu}}\, (1-\mu)\sqrt{\frac{\log(2HT/\delta)}{1-\mu}} \sum_{\ell=1}^{N} \sigma_\ell + \frac{(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}. \end{aligned} \tag{B.37}$$

Layerwise MuonBP error terms. We define the layerwise error applied to the MuonBP layers,

$$\begin{aligned} \mathcal{E}_t^{\mathrm{full,mu}} = 2\eta_{\mathrm{full}}^{\mathrm{mu}}\, \mu^t \sum_{\ell=1}^{N} \sigma_\ell + \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\, \eta_{\mathrm{full}}^{\mathrm{mu}}\, (1-\mu)\sqrt{\frac{\log(2HT/\delta)}{1-\mu}} \sum_{\ell=1}^{N} \sigma_\ell \\ + \frac{2\eta_{\mathrm{full}}^{\mathrm{mu}}\, \eta_{\max}^{\mathrm{mu}}}{1-\mu} \sqrt{rc} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}, \end{aligned} \tag{B.38}$$

$$\mathcal{E}_t^{\text{block,mu}} = 2\eta_{\text{block}}^{\text{mu}}\,\mu^t \sum_{\ell=1}^{N}\sigma_\ell + \frac{8C_2^{\text{mu}}}{C_1^{\text{mu}}}\,\eta_{\text{block}}^{\text{mu}}\,(1-\mu)\sqrt{\frac{\log(2HT/\delta)}{1-\mu}}\sum_{\ell=1}^{N}\sigma_\ell$$
$$+ \frac{2\eta_{\text{block}}^{\text{mu}}\,\eta_{\max}^{\text{mu}}}{1-\mu}\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{B} + \frac{(\eta_{\text{block}}^{\text{mu}})^2}{2}\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{B}. \tag{B.39}$$

B.3 Layerwise Descent Analysis for Max-Norm LMO

From Lemma 7 of (Khaled et al., 2025), $\|Z_\tau^\ell\|\,\eta_\tau^\ell \le A^\ell$ holds for all $\tau \le k$, where

$$Z_\tau^\ell = \frac{\mathbf{x}_{\tau+1}^\ell - \mathbf{x}_\tau^\ell}{\eta_\tau^\ell} \tag{B.40}$$

is the normalized per-layer update direction at step τ. Since both the full step and the local block step use LMOs under the ℓ_∞-max-norm ($\|\cdot\|_\infty$ and $B_\ell^{\text{sg}}(\cdot)$ respectively), and

$$B_\ell^{\text{sg}}(Z^\ell) = \max_m \|[Z^\ell]^{(m)}\|_\infty \le \|Z^\ell\|_\infty \tag{B.41}$$

holds for any layer-ℓ matrix Z^ℓ, the update direction for both step types satisfies $\|Z_\tau^\ell\|_\infty \le 1$ by definition of the respective LMOs.

Effective Step-size bound for max-norm of layer ℓ ($\|\cdot\|_\infty$). *Full steps* ($\tau \bmod p = 0$): The per-layer update direction is $Z_\tau^\ell = \text{LMO}_\infty^\ell(\mathbf{B}_\tau^\ell)$, which by definition of the LMO under $\|\cdot\|_\infty$ satisfies $\|Z_\tau^\ell\|_\infty \le 1$. Hence:

$$\|Z_\tau^\ell\|_\infty\,\eta_\tau^\ell \le 1\cdot\eta_{\text{full},\ell}^{\text{sg}} = \eta_{\text{full},\ell}^{\text{sg}}. \tag{B.42}$$

Local Block steps ($\tau \bmod p \ne 0$): The per-layer update direction satisfies $B_\ell^{\text{sg}}(Z_\tau^\ell) \le 1$ by definition of the block-norm LMO ($B_\ell^{\text{sg}}(Z_\tau^\ell) = \max_m \|[Z_\tau^\ell]^{(m)}\|_\infty \le 1$). Since $\|Z\|_\infty = \max_m \|Z^{(m)}\|_\infty = B_\ell^{\text{sg}}(Z)$, we have $\|Z_\tau^\ell\|_\infty \le 1$. Hence, we have

$$\|Z_\tau^\ell\|_\infty\,\eta_\tau^\ell \le 1\cdot\eta_{\text{block},\ell}^{\text{sg}} = \eta_{\text{block},\ell}^{\text{sg}}. \tag{B.43}$$

Combining (B.42) and (B.43), in all cases

$$\|Z_\tau^\ell\|_\infty\,\eta_\tau^\ell \le \max\{\eta_{\text{full},\ell}^{\text{sg}},\,\eta_{\text{block},\ell}^{\text{sg}}\} = \eta_{\max}^{\text{sg}}. \tag{B.44}$$

We therefore set for the max-norm analysis of layer ℓ:

$$\eta_{\text{op},\tau,\ell}^{\text{sg}} = \eta_{\max}^{\text{sg}} = \max\{\eta_{\text{full},\ell}^{\text{sg}},\,\eta_{\text{block},\ell}^{\text{sg}}\}. \tag{B.45}$$

Effective bound for block-norm analysis of layer ℓ ($B_\ell^{\text{sg}}(\cdot)$). *Full steps* ($\tau \bmod p = 0$): Since $B_\ell^{\text{sg}}(Z^\ell) = \max_m \|[Z^\ell]^{(m)}\|_\infty \le \|Z^\ell\|_\infty$ for any layer-ℓ matrix Z^ℓ, we have $B_\ell^{\text{sg}}(Z_\tau^\ell) \le 1$. Then,

$$B_\ell^{\text{sg}}(Z_\tau^\ell)\,\eta_\tau^\ell \le 1\cdot\eta_{\text{full},\ell}^{\text{sg}} = \eta_{\text{full},\ell}^{\text{sg}}. \tag{B.46}$$

Local Block steps ($\tau \bmod p \ne 0$): The per-layer update direction satisfies $B_\ell^{\text{sg}}(Z_\tau^\ell) \le 1$ by definition of the block-norm LMO. We have

$$B_\ell^{\text{sg}}(Z_\tau^\ell)\,\eta_\tau^\ell \le 1\cdot\eta_{\text{block},\ell}^{\text{sg}} = \eta_{\text{block},\ell}^{\text{sg}}. \tag{B.47}$$

Combining (B.46) and (B.47), in all cases

$$B_\ell^{\text{sg}}(Z_\tau^\ell)\,\eta_\tau^\ell \le \max\{\eta_{\text{full},\ell}^{\text{sg}},\,\eta_{\text{block},\ell}^{\text{sg}}\}. \tag{B.48}$$

We therefore set for the block-norm analysis of layer ℓ:

$$\eta_{\text{block},\tau,\ell}^{\text{sg}} = \eta_{\max}^{\text{sg}}. \tag{B.49}$$

(A) Full step for S layers (max norm $\|\cdot\|_\infty$, $t \in S_{\text{op}}$).

By layerwise $L_{\ell,\text{sg}}^{\text{op}}$-smoothness under $\|\cdot\|_\infty$ and the per-layer LMO descent argument, we can get

$$D_t^{\text{sg}} \le -\sum_{\ell=1}^{S}\eta_{\text{full},\ell}^{\text{sg}}\,\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^1 + 2\sum_{\ell=1}^{S}\eta_{\text{full},\ell}^{\text{sg}}\,\|\hat{\boldsymbol{\epsilon}}_t^\ell\|_{\ell,*}^1 + \frac{1}{2}\sum_{\ell=1}^{S}L_{\ell,\text{sg}}^{\text{op}}\big(\eta_{\text{full},\ell}^{\text{sg}}\big)^2. \tag{B.50}$$

From the effective step-size bound $\eta^{\mathrm{sg}}_{\mathrm{op},\tau,\ell} = \eta^{\mathrm{sg}}_{\max,\ell}$ (from (B.45)), with probability at least $1-\delta$ for all $0 \le t < T$,

$$\big\|\hat{\boldsymbol{\epsilon}}^{\ell}_t\big\|^1_{\ell,*} \le \beta^t\,\sigma_\ell + \frac{4C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}(1-\beta)\,\sigma_\ell\sqrt{\frac{\log(2HT/\delta)}{1-\beta}} + \frac{L^{\mathrm{op}}_{\ell,\mathrm{sg}}\,\eta^{\mathrm{sg}}_{\max,\ell}}{1-\beta}, \quad \ell \in \{1,\dots,S\}. \tag{B.51}$$

Summing (B.51) over $\ell = 1,\dots,S$,

$$\sum_{\ell=1}^{S}\big\|\hat{\boldsymbol{\epsilon}}^{\ell}_t\big\|^1_{\ell,*} \le \beta^t\sum_{\ell=1}^{S}\sigma_\ell + \frac{4C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}(1-\beta)\sqrt{\frac{\log(2HT/\delta)}{1-\beta}}\sum_{\ell=1}^{S}\sigma_\ell + \frac{1}{1-\beta}\sum_{\ell=1}^{S}L^{\mathrm{op}}_{\ell,\mathrm{sg}}\,\eta^{\mathrm{sg}}_{\max,\ell}. \tag{B.52}$$

Substituting (B.51) into (B.50),

$$\begin{aligned} D_t^{\mathrm{sg}} \le\; & -\eta^{\mathrm{sg}}_{\mathrm{full}}\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|^1_{\ell,*} + 2\eta^{\mathrm{sg}}_{\mathrm{full}}\,\beta^t\sum_{\ell=1}^{S}\sigma_\ell \\ & + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\,\eta^{\mathrm{sg}}_{\mathrm{full}}\,(1-\beta)\sqrt{\frac{\log(2HT/\delta)}{1-\beta}}\sum_{\ell=1}^{S}\sigma_\ell + \frac{2\eta^{\mathrm{sg}}_{\mathrm{full}}\,\eta^{\mathrm{sg}}_{\max}}{1-\beta}\sum_{\ell=1}^{S}L^{\mathrm{op}}_{\ell,\mathrm{sg}} + \frac{(\eta^{\mathrm{sg}}_{\mathrm{full}})^2}{2}\sum_{\ell=1}^{S}L^{\mathrm{op}}_{\ell,\mathrm{sg}}. \end{aligned} \tag{B.53}$$

(B) Local Block step for S layers (per-layer block max norm $B^{\mathrm{sg}}_\ell(\cdot)$, $t \in S_B$). Similarly, we get

$$D_t^{\mathrm{sg}} \le -\sum_{\ell=1}^{S}\eta^{\mathrm{sg}}_{\mathrm{block},\ell}\Big[\|\nabla_\ell f(\mathbf{x}_t)\|^1_{\ell,*} - 2(B^{\mathrm{sg}}_\ell)^*(\hat{\epsilon}^{\ell}_t) - \tfrac{1}{2}L^{B}_{\ell,\mathrm{sg}}\eta^{\mathrm{sg}}_{\mathrm{block},\ell}\Big]. \tag{B.54}$$

$$(B^{\mathrm{sg}}_\ell)^*(\hat{\epsilon}^{\ell}_t) \le \beta^t\,\sigma_\ell + \frac{4C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}(1-\beta)\,\sigma_\ell\sqrt{\frac{\log(2HT/\delta)}{1-\beta}} + \frac{L^{B}_{\ell,\mathrm{sg}}\,\eta^{\mathrm{sg}}_{\max,\ell}}{1-\beta}, \quad \ell \in \{1,\dots,S\}. \tag{B.55}$$

Summing (B.55) over $\ell = 1,\dots,S$,

$$\sum_{\ell=1}^{S}(B^{\mathrm{sg}}_\ell)^*(\hat{\epsilon}^{\ell}_t) \le \beta^t\sum_{\ell=1}^{S}\sigma_\ell + \frac{4C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}(1-\beta)\sqrt{\frac{\log(2HT/\delta)}{1-\beta}}\sum_{\ell=1}^{S}\sigma_\ell + \frac{1}{1-\beta}\sum_{\ell=1}^{S}L^{B}_{\ell,\mathrm{sg}}\,\eta^{\mathrm{sg}}_{\max,\ell}. \tag{B.56}$$

Substituting (B.55) into (B.54),

$$\begin{aligned} D_t^{\mathrm{sg}} \le\; & -\eta^{\mathrm{sg}}_{\mathrm{block}}\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|^1_{\ell,*} + 2\eta^{\mathrm{sg}}_{\mathrm{block}}\,\beta^t\sum_{\ell=1}^{S}\sigma_\ell + \frac{2\eta^{\mathrm{sg}}_{\mathrm{block}}\,\eta^{\mathrm{sg}}_{\max}}{1-\beta}\sum_{\ell=1}^{S}L^{B}_{\ell,\mathrm{sg}} \\ & + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\,\eta^{\mathrm{sg}}_{\mathrm{block}}\,(1-\beta)\sqrt{\frac{\log(2HT/\delta)}{1-\beta}}\sum_{\ell=1}^{S}\sigma_\ell + \frac{(\eta^{\mathrm{sg}}_{\mathrm{block}})^2}{2}\sum_{\ell=1}^{S}L^{B}_{\ell,\mathrm{sg}}. \end{aligned} \tag{B.57}$$

Layerwise error terms.

$$\begin{aligned} \mathcal{E}_t^{\mathrm{full,sg}} =\; & 2\eta^{\mathrm{sg}}_{\mathrm{full}}\,\beta^t\sum_{\ell=1}^{S}\sigma_\ell + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\,\eta^{\mathrm{sg}}_{\mathrm{full}}\,(1-\beta)\sqrt{\frac{\log(2HT/\delta)}{1-\beta}}\sum_{\ell=1}^{S}\sigma_\ell \\ & + \frac{2\eta^{\mathrm{sg}}_{\mathrm{full}}\,\eta^{\mathrm{sg}}_{\max}}{1-\beta}\sum_{\ell=1}^{S}L^{\mathrm{op}}_{\ell,\mathrm{sg}} + \frac{(\eta^{\mathrm{sg}}_{\mathrm{full}})^2}{2}\sum_{\ell=1}^{S}L^{\mathrm{op}}_{\ell,\mathrm{sg}}, \end{aligned} \tag{B.58}$$

$$\begin{aligned} \mathcal{E}_t^{\mathrm{block,sg}} =\; & 2\eta^{\mathrm{sg}}_{\mathrm{block}}\,\beta^t\sum_{\ell=1}^{S}\sigma_\ell + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\,\eta^{\mathrm{sg}}_{\mathrm{block}}\,(1-\beta)\sqrt{\frac{\log(2HT/\delta)}{1-\beta}}\sum_{\ell=1}^{S}\sigma_\ell \\ & + \frac{2\eta^{\mathrm{sg}}_{\mathrm{block}}\,\eta^{\mathrm{sg}}_{\max}}{1-\beta}\sum_{\ell=1}^{S}L^{B}_{\ell,\mathrm{sg}} + \frac{(\eta^{\mathrm{sg}}_{\mathrm{block}})^2}{2}\sum_{\ell=1}^{S}L^{B}_{\ell,\mathrm{sg}}. \end{aligned} \tag{B.59}$$

B.4 MuonMix: Layerwise Descent Analysis for Mixed-Norm LMOs

Theorem B.1 (Mixed-Norm Layerwise Distributed Convergence)**.** *Suppose Assumptions 1 and 2 hold, $f \geq f^\star > -\infty$, $N, S > 0$, and $p \mid T$. Consider exact LMOs, $\mathbf{B}_0^\ell = \mathbf{G}_0^\ell$, fixed momenta $\mu, \beta \in [0,1)$, and positive step sizes uniform within each group satisfying $\left(\sum_{\ell=1}^N L_{\ell,\mathrm{mu}}^{\mathrm{op}}\right)\eta_{\mathrm{full}}^{\mathrm{mu}} = \left(\sum_{\ell=1}^N L_{\ell,\mathrm{mu}}^{B}\right)\eta_{\mathrm{block}}^{\mathrm{mu}}, \eta_{\mathrm{full}}^{\mathrm{sg}} = \eta_{\mathrm{block}}^{\mathrm{sg}}$. Let $\Delta = f(\mathbf{x}_0) - f^\star$. Define $\gamma^{\mathrm{mix}} = p^{-1}\min(\eta_{\mathrm{full}}^{\mathrm{mu}}, \eta_{\mathrm{full}}^{\mathrm{sg}}) + (p-1)p^{-1}\min(\eta_{\mathrm{block}}^{\mathrm{mu}}, \eta_{\mathrm{block}}^{\mathrm{sg}})$. Then, with probability at least $1-\delta$,*

$$\min_{0\leq t\leq T-1}\left(\sum_{\ell=1}^N \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + \sum_{\ell=1}^S \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right) \leq \mathcal{R}^{\mathrm{mu}} + \mathcal{R}^{\mathrm{sg}}, \tag{B.60}$$

where $\kappa^{\mathrm{mu}} = \eta_{\max}^{\mathrm{mu}}/\gamma^{\mathrm{mix}}$, $\kappa^{\mathrm{sg}} = \eta_{\max}^{\mathrm{sg}}/\gamma^{\mathrm{mix}}$,

$$\begin{aligned}\mathcal{R}^{\mathrm{mu}} = \frac{\kappa^{\mathrm{mu}}\Delta}{2\eta_{\max}^{\mathrm{mu}}T} + \frac{4\kappa^{\mathrm{mu}}}{(1-\mu)T}\sum_{\ell=1}^N\sigma_\ell + \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\kappa^{\mathrm{mu}}\sqrt{1-\mu}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^N\sigma_\ell \\ + \frac{2\sqrt{rc}\,\kappa^{\mathrm{mu}}\widehat{L}_{BP}^{\mathrm{mu}}\eta_{\max}^{\mathrm{mu}}}{1-\mu} + \frac{\kappa^{\mathrm{mu}}\widehat{L}_{BP}^{\mathrm{mu}}\eta_{\max}^{\mathrm{mu}}}{2},\end{aligned} \tag{B.61}$$

$$\begin{aligned}\mathcal{R}^{\mathrm{sg}} = \frac{\kappa^{\mathrm{sg}}\Delta}{2\eta_{\max}^{\mathrm{sg}}T} + \frac{4\kappa^{\mathrm{sg}}}{(1-\beta)T}\sum_{\ell=1}^S\sigma_\ell + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\kappa^{\mathrm{sg}}\sqrt{1-\beta}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^S\sigma_\ell \\ + \frac{2\kappa^{\mathrm{sg}}\widehat{L}_{BP}^{\mathrm{sg}}\eta_{\max}^{\mathrm{sg}}}{1-\beta} + \frac{\kappa^{\mathrm{sg}}\widehat{L}_{BP}^{\mathrm{sg}}\eta_{\max}^{\mathrm{sg}}}{2}.\end{aligned} \tag{B.62}$$

Proof. **Full step ($t \in S_{\mathrm{op}}$):**

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \eta_{\mathrm{full}}^{\mathrm{mu}}\sum_{\ell=1}^N \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} - \eta_{\mathrm{full}}^{\mathrm{sg}}\sum_{\ell=1}^S \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1} + \mathcal{E}_t^{\mathrm{full,mu}} + \mathcal{E}_t^{\mathrm{full,sg}}. \tag{B.63}$$

Local Block step ($t \in S_B$):

$$f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t) - \eta_{\mathrm{block}}^{\mathrm{mu}}\sum_{\ell=1}^N \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} - \eta_{\mathrm{block}}^{\mathrm{sg}}\sum_{\ell=1}^S \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1} + \mathcal{E}_t^{\mathrm{block,mu}} + \mathcal{E}_t^{\mathrm{block,sg}}. \tag{B.64}$$

Mixed-Norm Summing Over All Steps

Rearranging (B.63) and (B.64) and summing over $t = 0, \ldots, T-1$,

$$\begin{aligned}&\sum_{t\in S_{\mathrm{op}}}\left(\eta_{\mathrm{full}}^{\mathrm{mu}}\sum_{\ell=1}^N \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + \eta_{\mathrm{full}}^{\mathrm{sg}}\sum_{\ell=1}^S \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right) \\ &\quad + \sum_{t\in S_B}\left(\eta_{\mathrm{block}}^{\mathrm{mu}}\sum_{\ell=1}^N \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + \eta_{\mathrm{block}}^{\mathrm{sg}}\sum_{\ell=1}^S \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right) \\ &\leq \underbrace{\sum_{t=0}^{T-1}(f(\mathbf{x}_t) - f(\mathbf{x}_{t+1}))}_{\leq \Delta} + \sum_{t\in S_{\mathrm{op}}}\left(\mathcal{E}_t^{\mathrm{full,mu}} + \mathcal{E}_t^{\mathrm{full,sg}}\right) + \sum_{t\in S_B}\left(\mathcal{E}_t^{\mathrm{block,mu}} + \mathcal{E}_t^{\mathrm{block,sg}}\right).\end{aligned} \tag{B.65}$$

We bound each error contribution separately.

1) MuonBP momentum-dependent error.

$$\sum_{t\in S_{\mathrm{op}}} 2\eta_{\mathrm{full}}^{\mathrm{mu}}\,\mu^t\sum_{\ell=1}^N\sigma_\ell + \sum_{t\in S_B} 2\eta_{\mathrm{block}}^{\mathrm{mu}}\,\mu^t\sum_{\ell=1}^N\sigma_\ell$$

$$\leq\ 4\eta_{\max}^{\mathrm{mu}} \sum_{t=0}^{T-1} \mu^t \sum_{\ell=1}^{N} \sigma_\ell\ \leq\ \frac{4\eta_{\max}^{\mathrm{mu}}}{1-\mu} \sum_{\ell=1}^{N} \sigma_\ell. \tag{B.66}$$

2) Elementwise sign momentum-dependent error.

$$\begin{aligned}
&\sum_{t\in S_{\mathrm{op}}} 2\eta_{\mathrm{full}}^{\mathrm{sg}}\, \beta^t \sum_{\ell=1}^{S} \sigma_\ell + \sum_{t\in S_B} 2\eta_{\mathrm{block}}^{\mathrm{sg}}\, \beta^t \sum_{\ell=1}^{S} \sigma_\ell \\
&\leq\ 4\eta_{\max}^{\mathrm{sg}} \sum_{t=0}^{T-1} \beta^t \sum_{\ell=1}^{S} \sigma_\ell\ \leq\ \frac{4\eta_{\max}^{\mathrm{sg}}}{1-\beta} \sum_{\ell=1}^{S} \sigma_\ell.
\end{aligned} \tag{B.67}$$

$$\begin{aligned}
&\sum_{t\in S_{\mathrm{op}}} \left(\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\, \eta_{\mathrm{full}}^{\mathrm{mu}} \sqrt{1-\mu} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{N} \sigma_\ell + \frac{2\eta_{\mathrm{full}}^{\mathrm{mu}}\, \eta_{\max}^{\mathrm{mu}}}{1-\mu} \sqrt{rc} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} \right) \\
&+ \sum_{t\in S_B} \left(\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\, \eta_{\mathrm{block}}^{\mathrm{mu}} \sqrt{1-\mu} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{N} \sigma_\ell + \frac{2\eta_{\mathrm{block}}^{\mathrm{mu}}\, \eta_{\max}^{\mathrm{mu}}}{1-\mu} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} + \frac{(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \right) \\
&= T \left[\frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2p} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2p} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \right] \\
&\quad + \frac{2T\, \eta_{\max}^{\mathrm{mu}}}{(1-\mu)p} \left(\sqrt{rc}\eta_{\mathrm{full}}^{\mathrm{mu}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + (p-1)\eta_{\mathrm{block}}^{\mathrm{mu}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \right) \\
&\quad + \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\, T \sqrt{1-\mu} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{N} \sigma_\ell \underbrace{\left(\frac{\eta_{\mathrm{full}}^{\mathrm{mu}}}{p} + \frac{(p-1)\eta_{\mathrm{block}}^{\mathrm{mu}}}{p} \right)}_{=\alpha^{\mathrm{mu}}}.
\end{aligned} \tag{B.68}$$

$$\begin{aligned}
&\sum_{t\in S_{\mathrm{op}}} \left(\frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\, \eta_{\mathrm{full}}^{\mathrm{sg}} \sqrt{1-\beta} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{S} \sigma_\ell + \frac{2\eta_{\mathrm{full}}^{\mathrm{sg}}\, \eta_{\max}^{\mathrm{sg}}}{1-\beta} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} + \frac{(\eta_{\mathrm{full}}^{\mathrm{sg}})^2}{2} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} \right) \\
&+ \sum_{t\in S_B} \left(\frac{8C_2^{\mathrm{sg}}\eta_{\mathrm{block}}^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\, \sqrt{1-\beta} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{S} \sigma_\ell + \frac{2\eta_{\mathrm{block}}^{\mathrm{sg}}\, \eta_{\max}^{\mathrm{sg}}}{1-\beta} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} + \frac{(\eta_{\mathrm{block}}^{\mathrm{sg}})^2}{2} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} \right) \\
&= T \left[\frac{(\eta_{\mathrm{full}}^{\mathrm{sg}})^2}{2p} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{sg}})^2}{2p} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} \right] \\
&\quad + \frac{2T\, \eta_{\max}^{\mathrm{sg}}}{(1-\beta)p} \left(\eta_{\mathrm{full}}^{\mathrm{sg}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} + (p-1)\eta_{\mathrm{block}}^{\mathrm{sg}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} \right) \\
&\quad + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\, T \sqrt{1-\beta} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{S} \sigma_\ell \underbrace{\left(\frac{\eta_{\mathrm{full}}^{\mathrm{sg}}}{p} + \frac{(p-1)\eta_{\mathrm{block}}^{\mathrm{sg}}}{p} \right)}_{=\alpha^{\mathrm{sg}}}.
\end{aligned} \tag{B.69}$$

Substituting (B.66)–(B.69) into (B.65),

$$\begin{aligned}
&\sum_{t\in S_{\mathrm{op}}} \left(\eta_{\mathrm{full}}^{\mathrm{mu}} \sum_{\ell=1}^{N} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + \eta_{\mathrm{full}}^{\mathrm{sg}} \sum_{\ell=1}^{S} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1} \right) \\
&+ \sum_{t\in S_B} \left(\eta_{\mathrm{block}}^{\mathrm{mu}} \sum_{\ell=1}^{N} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}} + \eta_{\mathrm{block}}^{\mathrm{sg}} \sum_{\ell=1}^{S} \|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1} \right) \\
&\leq\ \Delta + \frac{4\eta_{\max}^{\mathrm{mu}}}{1-\mu} \sum_{\ell=1}^{N} \sigma_\ell + \frac{4\eta_{\max}^{\mathrm{sg}}}{1-\beta} \sum_{\ell=1}^{S} \sigma_\ell + T \left[\frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2p} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2p} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \right. \\
&\quad \left. + \frac{(\eta_{\mathrm{full}}^{\mathrm{sg}})^2}{2p} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{sg}})^2}{2p} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} \right]
\end{aligned}$$

$$
\begin{aligned}
&+ \frac{2T\,\eta_{\max}^{\mathrm{mu}}}{(1-\mu)p}\left(\sqrt{rc}\eta_{\mathrm{full}}^{\mathrm{mu}}\sum_{\ell=1}^{N}L_{\ell,\mathrm{mu}}^{\mathrm{op}}+(p-1)\eta_{\mathrm{block}}^{\mathrm{mu}}\sum_{\ell=1}^{N}L_{\ell,\mathrm{mu}}^{B}\right)\\
&+ \frac{2T\,\eta_{\max}^{\mathrm{sg}}}{(1-\beta)p}\left(\eta_{\mathrm{full}}^{\mathrm{sg}}\sum_{\ell=1}^{S}L_{\ell,\mathrm{sg}}^{\mathrm{op}}+(p-1)\eta_{\mathrm{block}}^{\mathrm{sg}}\sum_{\ell=1}^{S}L_{\ell,\mathrm{sg}}^{B}\right)\\
&+ \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\,T\,\alpha^{\mathrm{mu}}\sqrt{1-\mu}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{N}\sigma_\ell+\frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\,T\,\alpha^{\mathrm{sg}}\sqrt{1-\beta}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{S}\sigma_\ell. \qquad \text{(B.70)}
\end{aligned}
$$

The left-hand side of (B.70) satisfies

$$
\begin{aligned}
\sum_{t=0}^{T-1}\Bigg(&\mathbf{1}_{t\in S_{\mathrm{op}}}\left[\eta_{\mathrm{full}}^{\mathrm{mu}}\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\eta_{\mathrm{full}}^{\mathrm{sg}}\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right]\\
&+\mathbf{1}_{t\in S_B}\left[\eta_{\mathrm{block}}^{\mathrm{mu}}\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\eta_{\mathrm{block}}^{\mathrm{sg}}\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right]\Bigg). \qquad \text{(B.71)}
\end{aligned}
$$

For each $t\in S_{\mathrm{op}}$, since $\eta_{\mathrm{full}}^{\mathrm{mu}}\ge\min(\eta_{\mathrm{full}}^{\mathrm{mu}},\eta_{\mathrm{full}}^{\mathrm{sg}})$ and $\eta_{\mathrm{full}}^{\mathrm{sg}}\ge\min(\eta_{\mathrm{full}}^{\mathrm{mu}},\eta_{\mathrm{full}}^{\mathrm{sg}})$,

$$
\begin{aligned}
&\eta_{\mathrm{full}}^{\mathrm{mu}}\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\eta_{\mathrm{full}}^{\mathrm{sg}}\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\\
&\ge\ \min\big(\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{full}}^{\mathrm{sg}}\big)\left(\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right). \qquad \text{(B.72)}
\end{aligned}
$$

Similarly, for each $t\in S_B$,

$$
\begin{aligned}
&\eta_{\mathrm{block}}^{\mathrm{mu}}\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\eta_{\mathrm{block}}^{\mathrm{sg}}\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\\
&\ge\ \min\big(\eta_{\mathrm{block}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{sg}}\big)\left(\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right). \qquad \text{(B.73)}
\end{aligned}
$$

Define

$$
\gamma^{\mathrm{mix}}=\frac{\min(\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{full}}^{\mathrm{sg}})}{p}+\frac{(p-1)\min(\eta_{\mathrm{block}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{sg}})}{p}. \qquad \text{(B.74)}
$$

For notational simplicity, define

$$
G_t^{\mathrm{mix}}=\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}. \qquad \text{(B.75)}
$$

Substituting (B.72)–(B.73) into (B.71), we obtain

$$
\begin{aligned}
\text{(B.71)}\ &\ge\ \min\big(\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{full}}^{\mathrm{sg}}\big)\sum_{t\in S_{\mathrm{op}}}G_t^{\mathrm{mix}}+\min\big(\eta_{\mathrm{block}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{sg}}\big)\sum_{t\in S_B}G_t^{\mathrm{mix}}\\
&\ge\ \big[|S_{\mathrm{op}}|\min\big(\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{full}}^{\mathrm{sg}}\big)+|S_B|\min\big(\eta_{\mathrm{block}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{sg}}\big)\big]\min_{0\le t\le T-1}G_t^{\mathrm{mix}}\\
&=T\,\gamma^{\mathrm{mix}}\cdot\min_{0\le t\le T-1}\left(\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right). \qquad \text{(B.76)}
\end{aligned}
$$

We define $\kappa^{\mathrm{mu}}=\frac{\eta_{\max}^{\mathrm{mu}}}{\gamma^{\mathrm{mix}}},\kappa^{\mathrm{sg}}=\frac{\eta_{\max}^{\mathrm{sg}}}{\gamma^{\mathrm{mix}}}$. Combining (B.70) and (B.76) and dividing by $T\gamma^{\mathrm{mix}}$,

$$
\min_{0\le t\le T-1}\left(\sum_{\ell=1}^{N}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{\mathrm{nuc}}+\sum_{\ell=1}^{S}\|\nabla_\ell f(\mathbf{x}_t)\|_{\ell,*}^{1}\right)
$$

$$
\begin{aligned}
&\leq \frac{\Delta}{\gamma^{\mathrm{mix}} T} + \frac{4\eta_{\max}^{\mathrm{mu}}}{(1-\mu)\,\gamma^{\mathrm{mix}} T} \sum_{\ell=1}^{N} \sigma_\ell + \frac{4\eta_{\max}^{\mathrm{sg}}}{(1-\beta)\,\gamma^{\mathrm{mix}} T} \sum_{\ell=1}^{S} \sigma_\ell \\
&\quad + \frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}} \frac{\alpha^{\mathrm{mu}}}{\gamma^{\mathrm{mix}}} \sqrt{1-\mu} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{N} \sigma_\ell + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}} \frac{\alpha^{\mathrm{sg}}}{\gamma^{\mathrm{mix}}} \sqrt{1-\beta} \sqrt{\log(2HT/\delta)} \sum_{\ell=1}^{S} \sigma_\ell \\
&\quad + \frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} + \frac{(\eta_{\mathrm{full}}^{\mathrm{sg}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} \\
&\quad + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{sg}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} + \frac{2\sqrt{rc}\,\eta_{\mathrm{full}}^{\mathrm{mu}}\,\eta_{\max}^{\mathrm{mu}}}{(1-\mu)p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{2(p-1)\eta_{\mathrm{block}}^{\mathrm{mu}}\,\eta_{\max}^{\mathrm{mu}}}{(1-\mu)p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \\
&\quad + \frac{2\eta_{\mathrm{full}}^{\mathrm{sg}}\,\eta_{\max}^{\mathrm{sg}}}{(1-\beta)p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} + \frac{2(p-1)\eta_{\mathrm{block}}^{\mathrm{sg}}\,\eta_{\max}^{\mathrm{sg}}}{(1-\beta)p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B}. \qquad \text{(B.77)}
\end{aligned}
$$

By the assumed balance condition, define

$$
\left(\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}\right) \eta_{\mathrm{full}}^{\mathrm{mu}} = \left(\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}\right) \eta_{\mathrm{block}}^{\mathrm{mu}} =: c^{\mathrm{mu}}. \qquad \text{(B.78)}
$$

Then $\eta_{\mathrm{full}}^{\mathrm{mu}} = c^{\mathrm{mu}} / \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}$ and $\eta_{\mathrm{block}}^{\mathrm{mu}} = c^{\mathrm{mu}} / \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}$. Under (B.78), we get

$$
\alpha^{\mathrm{mu}} = \frac{c^{\mathrm{mu}}}{p} \left(\frac{1}{\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}} + \frac{p-1}{\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}} \right) = \frac{c^{\mathrm{mu}}}{\widehat{L}_{BP}^{\mathrm{mu}}}. \qquad \text{(B.79)}
$$

Then, we define

$$
\left(\sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}}\right) \eta_{\mathrm{full}}^{\mathrm{sg}} = \left(\sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B}\right) \eta_{\mathrm{block}}^{\mathrm{sg}} =: c^{\mathrm{sg}}. \qquad \text{(B.80)}
$$

Then $\eta_{\mathrm{full}}^{\mathrm{sg}} = c^{\mathrm{sg}} / \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}}$ and $\eta_{\mathrm{block}}^{\mathrm{sg}} = c^{\mathrm{sg}} / \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B}$. We get

$$
\alpha^{\mathrm{sg}} = \frac{c^{\mathrm{sg}}}{p} \left(\frac{1}{\sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}}} + \frac{p-1}{\sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B}} \right) = \frac{c^{\mathrm{sg}}}{\widehat{L}_{BP}^{\mathrm{sg}}}, \qquad \text{(B.81)}
$$

where $L_{\ell,\mathrm{sg}}^{\mathrm{op}} = L_{\ell,\mathrm{sg}}$, $L_{\ell,\mathrm{sg}}^{B} = L_{\ell,\mathrm{sg}}$.

Simplifying the MuonBP related term in (B.77).

$$
\begin{aligned}
&\frac{(\eta_{\mathrm{full}}^{\mathrm{mu}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{mu}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B} \\
&= \frac{(c^{\mathrm{mu}})^2}{2p\,\gamma^{\mathrm{mix}}} \left(\frac{1}{\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{\mathrm{op}}} + \frac{p-1}{\sum_{\ell=1}^{N} L_{\ell,\mathrm{mu}}^{B}} \right) = \frac{(c^{\mathrm{mu}})^2}{2\gamma^{\mathrm{mix}}\,\widehat{L}_{BP}^{\mathrm{mu}}} = \frac{c^{\mathrm{mu}} \alpha^{\mathrm{mu}}}{2\gamma^{\mathrm{mix}}}. \qquad \text{(B.82)}
\end{aligned}
$$

Since $\alpha^{\mathrm{mu}} \leq \eta_{\max}^{\mathrm{mu}}$, we get

$$
\frac{c^{\mathrm{mu}} \alpha^{\mathrm{mu}}}{2\gamma^{\mathrm{mix}}} \leq \frac{\widehat{L}_{BP}^{\mathrm{mu}}\,\kappa^{\mathrm{mu}}\,\eta_{\max}^{\mathrm{mu}}}{2}. \qquad \text{(B.83)}
$$

Simplifying the related elementwise sign term in (B.77).

$$
\frac{(\eta_{\mathrm{full}}^{\mathrm{sg}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{\mathrm{op}} + \frac{(p-1)(\eta_{\mathrm{block}}^{\mathrm{sg}})^2}{2p\,\gamma^{\mathrm{mix}}} \sum_{\ell=1}^{S} L_{\ell,\mathrm{sg}}^{B} = \frac{c^{\mathrm{sg}} \alpha^{\mathrm{sg}}}{2\gamma^{\mathrm{mix}}} \leq \frac{\widehat{L}_{BP}^{\mathrm{sg}}\,\kappa^{\mathrm{sg}}\,\eta_{\max}^{\mathrm{sg}}}{2}. \qquad \text{(B.84)}
$$

Simplifying the MuonBP related term in (B.77).

$$\begin{aligned}
&\frac{2\sqrt{rc}\,\eta_{\text{full}}^{\text{mu}}\,\eta_{\max}^{\text{mu}}}{(1-\mu)p\,\gamma^{\text{mix}}}\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{\text{op}}+\frac{2(p-1)\eta_{\text{block}}^{\text{mu}}\,\eta_{\max}^{\text{mu}}}{(1-\mu)p\,\gamma^{\text{mix}}}\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{B}\\
&=\frac{2\eta_{\max}^{\text{mu}}}{(1-\mu)p\,\gamma^{\text{mix}}}\left(\frac{\sqrt{rc}\,c^{\text{mu}}}{\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{\text{op}}}\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{\text{op}}+(p-1)\frac{c^{\text{mu}}}{\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{B}}\sum_{\ell=1}^{N}L_{\ell,\text{mu}}^{B}\right)\\
&\le\frac{2\sqrt{rc}\,\eta_{\max}^{\text{mu}}}{(1-\mu)p\,\gamma^{\text{mix}}}\cdot p\,c^{\text{mu}}=\frac{2\sqrt{rc}\,\eta_{\max}^{\text{mu}}\,c^{\text{mu}}}{(1-\mu)\gamma^{\text{mix}}}\le\frac{2\sqrt{rc}\,\kappa^{\text{mu}}\,\widehat{L}_{BP}^{\text{mu}}\,\eta_{\max}^{\text{mu}}}{1-\mu}.
\end{aligned}\tag{B.85}$$

Simplifying the related elementwise sign term in (B.77).

$$\begin{aligned}
(2)^{\text{sg}}&=\frac{2\eta_{\text{full}}^{\text{sg}}\,\eta_{\max}^{\text{sg}}}{(1-\beta)p\,\gamma^{\text{mix}}}\sum_{\ell=1}^{S}L_{\ell,\text{sg}}^{\text{op}}+\frac{2(p-1)\eta_{\text{block}}^{\text{sg}}\,\eta_{\max}^{\text{sg}}}{(1-\beta)p\,\gamma^{\text{mix}}}\sum_{\ell=1}^{S}L_{\ell,\text{sg}}^{B}\\
&=\frac{2\eta_{\max}^{\text{sg}}}{(1-\beta)p\,\gamma^{\text{mix}}}\cdot p\,c^{\text{sg}}=\frac{2\kappa^{\text{sg}}\,\widehat{L}_{BP}^{\text{sg}}\,\eta_{\max}^{\text{sg}}}{1-\beta}.
\end{aligned}\tag{B.86}$$

Bounding the noise terms. Since $\alpha^{\text{mu}}\le\eta_{\max}^{\text{mu}}$, we have $\alpha^{\text{mu}}/\gamma^{\text{mix}}\le\eta_{\max}^{\text{mu}}/\gamma^{\text{mix}}=\kappa^{\text{mu}}$, and similarly $\alpha^{\text{sg}}/\gamma^{\text{mix}}\le\kappa^{\text{sg}}$. Therefore,

$$\frac{8C_2^{\text{mu}}}{C_1^{\text{mu}}}\frac{\alpha^{\text{mu}}}{\gamma^{\text{mix}}}\sqrt{1-\mu}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{N}\sigma_\ell\ \le\ \frac{8C_2^{\text{mu}}}{C_1^{\text{mu}}}\kappa^{\text{mu}}\sqrt{1-\mu}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{N}\sigma_\ell,\tag{B.87}$$

$$\frac{8C_2^{\text{sg}}}{C_1^{\text{sg}}}\frac{\alpha^{\text{sg}}}{\gamma^{\text{mix}}}\sqrt{1-\beta}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{S}\sigma_\ell\ \le\ \frac{8C_2^{\text{sg}}}{C_1^{\text{sg}}}\kappa^{\text{sg}}\sqrt{1-\beta}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{S}\sigma_\ell.\tag{B.88}$$

Since $1/\gamma^{\text{mix}}=\kappa^{\text{mu}}/\eta_{\max}^{\text{mu}}=\kappa^{\text{sg}}/\eta_{\max}^{\text{sg}}$, we have

$$\frac{\Delta}{\gamma^{\text{mix}}T}=\frac{\Delta}{2\gamma^{\text{mix}}T}+\frac{\Delta}{2\gamma^{\text{mix}}T}=\frac{\kappa^{\text{mu}}\,\Delta}{2\eta_{\max}^{\text{mu}}\,T}+\frac{\kappa^{\text{sg}}\,\Delta}{2\eta_{\max}^{\text{sg}}\,T}.\tag{B.89}$$

Substituting (B.83)–(B.89) into (B.77), we obtain the upper bound of Theorem B.1. □

We set

$$\eta_{\max}^{\text{mu}}=\sqrt{\frac{\Delta(1-\mu)}{T\,\widehat{L}_{BP}^{\text{mu}}}}.\tag{B.90}$$

Substituting (B.90) into $\mathcal{R}^{\text{mu}}$:

$$\frac{\kappa^{\text{mu}}\,\Delta}{2\eta_{\max}^{\text{mu}}\,T}=\frac{\kappa^{\text{mu}}}{2}\sqrt{\frac{\Delta\,\widehat{L}_{BP}^{\text{mu}}}{(1-\mu)\,T}},\tag{B.91}$$

$$\frac{\kappa^{\text{mu}}\,\widehat{L}_{BP}^{\text{mu}}\,\eta_{\max}^{\text{mu}}}{2}\ \le\ \frac{\kappa^{\text{mu}}}{2}\sqrt{\frac{\Delta\,\widehat{L}_{BP}^{\text{mu}}}{(1-\mu)\,T}},\tag{B.92}$$

$$\frac{2\sqrt{rc}\,\kappa^{\text{mu}}\,\widehat{L}_{BP}^{\text{mu}}\,\eta_{\max}^{\text{mu}}}{1-\mu}=\frac{2\sqrt{rc}\,\kappa^{\text{mu}}}{\sqrt{1-\mu}}\sqrt{\frac{\Delta\,\widehat{L}_{BP}^{\text{mu}}}{T}}.\tag{B.93}$$

$$T\cdot\mathcal{R}^{\text{mu}}\ \le\ \frac{6\sqrt{rc}\,\kappa^{\text{mu}}}{\sqrt{1-\mu}}\sqrt{\Delta\,\widehat{L}_{BP}^{\text{mu}}\,T}+\frac{4\kappa^{\text{mu}}}{1-\mu}\sum_{\ell=1}^{N}\sigma_\ell$$

$$+\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\kappa^{\mathrm{mu}}T\sqrt{1-\mu}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{N}\sigma_\ell, \tag{B.94}$$

Step-size choice for elementwise sign layers. Set

$$\eta_{\max}^{\mathrm{sg}}=\sqrt{\frac{\Delta(1-\beta)}{T\,\widehat{L}_{BP}^{\mathrm{sg}}}}. \tag{B.95}$$

$$T\cdot\mathcal{R}^{\mathrm{sg}} \;\le\; \frac{6\kappa^{\mathrm{sg}}}{\sqrt{1-\beta}}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\,T}+\frac{4\kappa^{\mathrm{sg}}}{1-\beta}\sum_{\ell=1}^{S}\sigma_\ell+\frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\kappa^{\mathrm{sg}}T\sqrt{1-\beta}\sqrt{\log(2HT/\delta)}\sum_{\ell=1}^{S}\sigma_\ell. \tag{B.96}$$

Momentum choice for MuonBP layers. We choose

$$1-\mu=\min\left(\frac{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}}}{\left(\sum_{\ell=1}^{N}\sigma_\ell\right)\sqrt{T}},\;1\right). \tag{B.97}$$

MuonBP, Case 1: $1-\mu=\dfrac{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}}}{\left(\sum_{\ell=1}^{N}\sigma_\ell\right)\sqrt{T}}$.

$$\begin{aligned}T\cdot\mathcal{R}^{\mathrm{mu}} \;\le\; & 6\sqrt{rc}\,\kappa^{\mathrm{mu}}\left(\sum_{\ell=1}^{N}\sigma_\ell\right)^{1/2}T^{3/4}\left(\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\right)^{1/4}+\frac{4\kappa^{\mathrm{mu}}\left(\sum_{\ell=1}^{N}\sigma_\ell\right)^{2}\sqrt{T}}{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}}}\\ & +\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\kappa^{\mathrm{mu}}\left(\sum_{\ell=1}^{N}\sigma_\ell\right)^{1/2}T^{3/4}\left(\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\right)^{1/4}\sqrt{\log(2HT/\delta)}.\end{aligned} \tag{B.98}$$

MuonBP, Case 2: $1-\mu=1$, $\mu=0$, holding when $\sum_{\ell=1}^{N}\sigma_\ell\le\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}/T}$.

$$T\cdot\mathcal{R}^{\mathrm{mu}} \;\le\; 6\sqrt{rc}\,\kappa^{\mathrm{mu}}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\,T}+4\kappa^{\mathrm{mu}}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\,T}+\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\kappa^{\mathrm{mu}}\sqrt{\log(2HT/\delta)}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\,T}. \tag{B.99}$$

Taking the uniform bound covering both MuonBP cases, we get

$$\begin{aligned}T\cdot\mathcal{R}^{\mathrm{mu}} \;\le\; & \left(10\sqrt{rc}\,\kappa^{\mathrm{mu}}+\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\kappa^{\mathrm{mu}}\sqrt{\log\left(\frac{2HT}{\delta}\right)}\right)\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\,T}\\ & +\frac{4\kappa^{\mathrm{mu}}\left(\sum_{\ell=1}^{N}\sigma_\ell\right)^{2}\sqrt{T}}{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}}}\\ & +\kappa^{\mathrm{mu}}\left(6\sqrt{rc}+\frac{8C_2^{\mathrm{mu}}}{C_1^{\mathrm{mu}}}\sqrt{\log\left(\frac{2HT}{\delta}\right)}\right)\left(\sum_{\ell=1}^{N}\sigma_\ell\right)^{1/2}T^{3/4}\left(\Delta\,\widehat{L}_{BP}^{\mathrm{mu}}\right)^{1/4}.\end{aligned} \tag{B.100}$$

Momentum choice for the elementwise sign terms. Set

$$1-\beta=\min\left(\frac{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}}}{\left(\sum_{\ell=1}^{S}\sigma_\ell\right)\sqrt{T}},\;1\right). \tag{B.101}$$

Elementwise sign terms, Case 1: $1-\beta = \dfrac{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}}}{\left(\sum_{\ell=1}^{S}\sigma_\ell\right)\sqrt{T}}$.

$$T\cdot\mathcal{R}^{\mathrm{sg}} \;\le\; 6\kappa^{\mathrm{sg}}\left(\sum_{\ell=1}^{S}\sigma_\ell\right)^{1/2} T^{3/4}\left(\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\right)^{1/4} + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\kappa^{\mathrm{sg}}\left(\sum_{\ell=1}^{S}\sigma_\ell\right)^{1/2} T^{3/4}\left(\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\right)^{1/4}\sqrt{\log(2HT/\delta)} + \frac{4\kappa^{\mathrm{sg}}\left(\sum_{\ell=1}^{S}\sigma_\ell\right)^2\sqrt{T}}{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}}}. \tag{B.102}$$

Elementwise sign terms, Case 2: $1-\beta=1$, $\beta=0$, holding when $\sum_{\ell=1}^{S}\sigma_\ell \le \sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}/T}$.

$$T\cdot\mathcal{R}^{\mathrm{sg}} \;\le\; 6\kappa^{\mathrm{sg}}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\,T} + 4\kappa^{\mathrm{sg}}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\,T} + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\kappa^{\mathrm{sg}}\sqrt{\log(2HT/\delta)}\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\,T}. \tag{B.103}$$

Taking the uniform bound, we have

$$\begin{aligned} T\cdot\mathcal{R}^{\mathrm{sg}} \;\le\; & \left(10\kappa^{\mathrm{sg}} + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\kappa^{\mathrm{sg}}\sqrt{\log\!\left(\frac{2HT}{\delta}\right)}\right)\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\,T} \\ & + \frac{4\kappa^{\mathrm{sg}}\left(\sum_{\ell=1}^{S}\sigma_\ell\right)^2\sqrt{T}}{\sqrt{\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}}} \\ & + \kappa^{\mathrm{sg}}\left(6 + \frac{8C_2^{\mathrm{sg}}}{C_1^{\mathrm{sg}}}\sqrt{\log\!\left(\frac{2HT}{\delta}\right)}\right)\left(\sum_{\ell=1}^{S}\sigma_\ell\right)^{1/2} T^{3/4}\left(\Delta\,\widehat{L}_{BP}^{\mathrm{sg}}\right)^{1/4}. \end{aligned} \tag{B.104}$$

Therefore, we get the convergence bound of Theorem 1 combining (B.100) and (B.104). When $N=0$ or $S=0$, omit the empty group and use the remaining group's average step size for γ^{mix}. In Theorem B.1, replace the remaining group's initial-gap term $\kappa^g\Delta/(2\eta^g_{\max}T)$ by $\kappa^g\Delta/(\eta^g_{\max}T)$. The coefficients 6 and 10 used in the subsequent bounds still dominate the resulting terms, so (1) holds with only the active group's contribution. For a zero-noise group the corresponding momentum is zero and only Case 2 is needed. For fixed problem parameters, if both groups have positive total noise, $\gamma^{\mathrm{mix}}=\Theta(T^{-3/4})$ and both $\kappa^g=O(1)$. If only one group is noisy, the noiseless group's κ^g can grow as $T^{1/4}$, but it multiplies only a $T^{-1/2}$ term, so the total rate is still $\widetilde{O}(T^{-1/4})$. If both groups are noiseless, both $\kappa^g=O(1)$ and the rate is $O(T^{-1/2})$.

C Per-Layer Optimizer Selection

In this section we develop the noiseless descent surrogate motivating the per-layer LMO selection heuristic. We compare the **per-step descent lower bound**. In this section, we work in the **noiseless setting** $\sigma_\ell = 0$ for all layers, with exact LMOs, $p \mid T$, and fixed positive candidate step sizes, momenta in $[0,1)$, and $\mathbf{B}_0^\ell = \nabla_\ell f(\mathbf{x}_0)$. The descent bounds in Sections C.1–C.3 refer to the trajectory generated by each method separately. They do not require the optimized rate choices of Theorem 1.

C.1 Per-Step Descent Lower Bound for MuonBP

We first establish the per-step descent for the baseline in which *all* H layers are updated by MuonBP (spectral–nuclear norm pair), corresponding to $S=0$ and $N=H$ in the notation of Section B.4. In the noiseless setting ($\sigma_\ell=0$), the momentum error recursion used in (B.30) and (B.35) gives

$$\left\|\hat{\epsilon}_k^\ell\right\|_{\ell,*}^{\mathrm{nuc}} \le \frac{L_{\ell,\mathrm{mu}}^{\mathrm{op}}\,\max\{\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{mu}}\sqrt{rc}\}}{1-\mu}, \quad k\in S_{\mathrm{op}},$$

$$(B_\ell^{\mathrm{mu}})^*(\hat{\epsilon}_k^\ell) \le \frac{L_{\ell,\mathrm{mu}}^B\,\eta_{\max}^{\mathrm{mu}}}{1-\mu}, \quad k \in S_B, \tag{C.1}$$

for each layer $\ell \in \{1,\dots,H\}$.

Full step ($k \in S_{\mathrm{op}}$) from (B.29).

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\ge \sum_{\ell=1}^{H}\Big[\eta_{\mathrm{full}}^{\mathrm{mu}}\,\|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\mathrm{nuc}} - 2\eta_{\mathrm{full}}^{\mathrm{mu}}\,\big\|\hat{\epsilon}_k^\ell\big\|_{\ell,*}^{\mathrm{nuc}} - \frac{L_{\ell,\mathrm{mu}}^{\mathrm{op}}}{2}\big(\eta_{\mathrm{full}}^{\mathrm{mu}}\big)^2\Big] \\ &\ge \sum_{\ell=1}^{H}\underbrace{\eta_{\mathrm{full}}^{\mathrm{mu}}\,\|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\mathrm{nuc}}}_{\mathcal{B}_{k,\ell}^{\mathrm{full,mu}}} \\ &\quad - \sum_{\ell=1}^{H}\underbrace{\left[\frac{L_{\ell,\mathrm{mu}}^{\mathrm{op}}}{2}\big(\eta_{\mathrm{full}}^{\mathrm{mu}}\big)^2 + \frac{2\,L_{\ell,\mathrm{mu}}^{\mathrm{op}}\,\eta_{\mathrm{full}}^{\mathrm{mu}}\,\max\{\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{mu}}\sqrt{rc}\}}{1-\mu}\right]}_{\mathcal{C}_{k,\ell}^{\mathrm{full,mu}}}. \end{aligned} \tag{C.2}$$

Local Block step ($k \in S_B$) from (B.34).

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\ge \sum_{\ell=1}^{H}\Big[\eta_{\mathrm{block}}^{\mathrm{mu}}\,\|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\mathrm{nuc}} - 2\eta_{\mathrm{block}}^{\mathrm{mu}}\,(B_\ell^{\mathrm{mu}})^*(\hat{\epsilon}_k^\ell) - \frac{L_{\ell,\mathrm{mu}}^B}{2}\big(\eta_{\mathrm{block}}^{\mathrm{mu}}\big)^2\Big] \\ &\ge \sum_{\ell=1}^{H}\underbrace{\eta_{\mathrm{block}}^{\mathrm{mu}}\,\|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\mathrm{nuc}}}_{\mathcal{B}_{k,\ell}^{\mathrm{block,mu}}} - \sum_{\ell=1}^{H}\underbrace{\left[\frac{L_{\ell,\mathrm{mu}}^B}{2}\big(\eta_{\mathrm{block}}^{\mathrm{mu}}\big)^2 + \frac{2\,L_{\ell,\mathrm{mu}}^B\,\eta_{\mathrm{block}}^{\mathrm{mu}}\,\eta_{\max}^{\mathrm{mu}}}{1-\mu}\right]}_{\mathcal{C}_{k,\ell}^{\mathrm{block,mu}}}. \end{aligned} \tag{C.3}$$

C.2 Per-Step Descent Lower Bound for MuonMix

The proposed method assigns a single target layer $s \in \{1,\dots,H\}$ to max–ℓ_1 norm pair while all remaining $H-1$ layers $\ell \in \{1,\dots,H\} \setminus \{s\}$ continue to use MuonBP.

$$\big\|\hat{\epsilon}_k^{(s)}\big\|_{s,*}^1 \le \frac{L_{s,\mathrm{sg}}^{\mathrm{op}}\,\eta_{\max}^{\mathrm{sg}}}{1-\beta}, \quad k \in S_{\mathrm{op}}, \qquad \big\|\hat{\epsilon}_k^{(s)}\big\|_{s,*}^1 \le \frac{L_{s,\mathrm{sg}}^B\,\eta_{\max}^{\mathrm{sg}}}{1-\beta}, \quad k \in S_B. \tag{C.4}$$

The error bounds for the remaining $H-1$ MuonBP layers are as in (C.1) with $\ell \ne s$.

Full step ($k \in S_{\mathrm{op}}$) from (B.63).

$$\begin{aligned} f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\ge \sum_{\substack{\ell=1\\ \ell\ne s}}^{H}\Big[\eta_{\mathrm{full}}^{\mathrm{mu}}\,\|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\mathrm{nuc}} - 2\eta_{\mathrm{full}}^{\mathrm{mu}}\,\big\|\hat{\epsilon}_k^\ell\big\|_{\ell,*}^{\mathrm{nuc}} - \frac{L_{\ell,\mathrm{mu}}^{\mathrm{op}}}{2}\big(\eta_{\mathrm{full}}^{\mathrm{mu}}\big)^2\Big] \\ &\quad + \Big[\eta_{\mathrm{full}}^{\mathrm{sg}}\,\|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1 - 2\eta_{\mathrm{full}}^{\mathrm{sg}}\,\big\|\hat{\epsilon}_k^{(s)}\big\|_{s,*}^1 - \frac{L_{s,\mathrm{sg}}^{\mathrm{op}}}{2}\big(\eta_{\mathrm{full}}^{\mathrm{sg}}\big)^2\Big] \\ &\ge \sum_{\substack{\ell=1\\ \ell\ne s}}^{H}\underbrace{\eta_{\mathrm{full}}^{\mathrm{mu}}\,\|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\mathrm{nuc}}}_{\mathcal{B}_{k,\ell}^{\mathrm{full,mu}}} \\ &\quad - \sum_{\substack{\ell=1\\ \ell\ne s}}^{H}\underbrace{\left[\frac{L_{\ell,\mathrm{mu}}^{\mathrm{op}}}{2}\big(\eta_{\mathrm{full}}^{\mathrm{mu}}\big)^2 + \frac{2\,L_{\ell,\mathrm{mu}}^{\mathrm{op}}\,\eta_{\mathrm{full}}^{\mathrm{mu}}\,\max\{\eta_{\mathrm{full}}^{\mathrm{mu}},\,\eta_{\mathrm{block}}^{\mathrm{mu}}\sqrt{rc}\}}{1-\mu}\right]}_{\mathcal{C}_{k,\ell}^{\mathrm{full,mu}}} \\ &\quad + \underbrace{\eta_{\mathrm{full}}^{\mathrm{sg}}\,\|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1}_{\mathcal{B}_{k,s}^{\mathrm{full,sg}}} - \underbrace{\left[\frac{L_{s,\mathrm{sg}}^{\mathrm{op}}}{2}\big(\eta_{\mathrm{full}}^{\mathrm{sg}}\big)^2 + \frac{2\,L_{s,\mathrm{sg}}^{\mathrm{op}}\,\eta_{\mathrm{full}}^{\mathrm{sg}}\,\eta_{\max}^{\mathrm{sg}}}{1-\beta}\right]}_{\mathcal{C}_{k,s}^{\mathrm{full,sg}}}. \end{aligned} \tag{C.5}$$

Local Block step ($k \in S_B$) from (B.64).

$$
\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}_{k+1}) &\geq \sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} \Big[\eta_{\text{block}}^{\text{mu}} \, \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} - 2\eta_{\text{block}}^{\text{mu}} \, (B_\ell^{\text{mu}})^*(\hat{\epsilon}_k^\ell) - \frac{L_{\ell,\text{mu}}^B}{2} \big(\eta_{\text{block}}^{\text{mu}}\big)^2 \Big] \\
&\quad + \Big[\eta_{\text{block}}^{\text{sg}} \, \|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1 - 2\eta_{\text{block}}^{\text{sg}} (B_s^{\text{sg}})^*(\hat{\epsilon}_k^{(s)}) - \frac{L_{s,\text{sg}}^B}{2} \big(\eta_{\text{block}}^{\text{sg}}\big)^2 \Big] \\
&\geq \sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} \underbrace{\eta_{\text{block}}^{\text{mu}} \, \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}}}_{\mathcal{B}_{k,\ell}^{\text{block,mu}}} - \sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} \underbrace{\left[\frac{L_{\ell,\text{mu}}^B}{2} \big(\eta_{\text{block}}^{\text{mu}}\big)^2 + \frac{2\, L_{\ell,\text{mu}}^B \, \eta_{\text{block}}^{\text{mu}} \, \eta_{\max}^{\text{mu}}}{1-\mu} \right]}_{\mathcal{C}_{k,\ell}^{\text{block,mu}}} \\
&\quad + \underbrace{\eta_{\text{block}}^{\text{sg}} \, \|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1}_{\mathcal{B}_{k,s}^{\text{block,sg}}} - \underbrace{\left[\frac{L_{s,\text{sg}}^B}{2} \big(\eta_{\text{block}}^{\text{sg}}\big)^2 + \frac{2\, L_{s,\text{sg}}^B \, \eta_{\text{block}}^{\text{sg}} \, \eta_{\max}^{\text{sg}}}{1-\beta} \right]}_{\mathcal{C}_{k,s}^{\text{block,sg}}}. \qquad \text{(C.6)}
\end{aligned}
$$

C.3 Average Per-Step Descent Lower Bounds

We define the **average per-step descent lower bound** by summing the per-step descent inequalities over $k = 0, \ldots, T-1$ and dividing by T.

Definition 1 (Φ^{mu}: all-MuonBP average descent)**.**

$$
\begin{aligned}
\Phi^{\text{mu}} &= \frac{1}{T} \sum_{k=0}^{T-1} \sum_{\ell=1}^{H} \Big[\eta_k^{\text{mu}} \, \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} - \mathcal{C}_{k,\ell}^{\text{mu}} \Big] \\
&= \underbrace{\frac{1}{T} \left(\sum_{k \in S_{\text{op}}} \sum_{\ell=1}^{H} \eta_{\text{full}}^{\text{mu}} \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} + \sum_{k \in S_B} \sum_{\ell=1}^{H} \eta_{\text{block}}^{\text{mu}} \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} \right)}_{B^{\text{mu}}} - \sum_{\ell=1}^{H} E_\ell^{\text{mu}}, \qquad \text{(C.7)}
\end{aligned}
$$

$$
\begin{aligned}
E_\ell^{\text{mu}} &= \frac{L_{\ell,\text{mu}}^{\text{op}} \big(\eta_{\text{full}}^{\text{mu}}\big)^2}{2p} + \frac{(p-1)\, L_{\ell,\text{mu}}^B \big(\eta_{\text{block}}^{\text{mu}}\big)^2}{2p} \\
&\quad + \frac{2}{1-\mu} \left(\frac{L_{\ell,\text{mu}}^{\text{op}} \, \eta_{\text{full}}^{\text{mu}} \, \max\{\eta_{\text{full}}^{\text{mu}}, \, \eta_{\text{block}}^{\text{mu}} \sqrt{rc}\}}{p} + \frac{(p-1)\, L_{\ell,\text{mu}}^B \, \eta_{\text{block}}^{\text{mu}} \, \eta_{\max}^{\text{mu}}}{p} \right). \qquad \text{(C.8)}
\end{aligned}
$$

Definition 2 (Φ_s^{mix}: MuonMix average descent with layer s assigned to Signum)**.**

$$
\begin{aligned}
\Phi_s^{\text{mix}} &= \frac{1}{T} \sum_{k=0}^{T-1} \left[\sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} \Big(\eta_k^{\text{mu}} \, \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} - \mathcal{C}_{k,\ell}^{\text{mu}} \Big) + \Big(\eta_k^{\text{sg}} \, \|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1 - \mathcal{C}_{k,s}^{\text{sg}} \Big) \right] \\
&= \underbrace{\frac{1}{T} \left(\sum_{k \in S_{\text{op}}} \sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} \eta_{\text{full}}^{\text{mu}} \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} + \sum_{k \in S_B} \sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} \eta_{\text{block}}^{\text{mu}} \|\nabla_\ell f(\mathbf{x}_k)\|_{\ell,*}^{\text{nuc}} \right)}_{B_{\neq s}^{\text{mu}}} - \sum_{\substack{\ell=1 \\ \ell \neq s}}^{H} E_\ell^{\text{mu}} \\
&\quad + \underbrace{\frac{1}{T} \left(\sum_{k \in S_{\text{op}}} \eta_{\text{full}}^{\text{sg}} \|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1 + \sum_{k \in S_B} \eta_{\text{block}}^{\text{sg}} \|\nabla_s f(\mathbf{x}_k)\|_{s,*}^1 \right)}_{B_s^{\text{sg}}} - E_s^{\text{sg}}. \qquad \text{(C.9)}
\end{aligned}
$$

$$
E_s^{\text{sg}} = \frac{L_{s,\text{sg}}^{\text{op}} \big(\eta_{\text{full}}^{\text{sg}}\big)^2}{2p} + \frac{(p-1)\, L_{s,\text{sg}}^B \big(\eta_{\text{block}}^{\text{sg}}\big)^2}{2p}
$$

$$+\frac{2}{1-\beta}\left(\frac{L^{\mathrm{op}}_{s,\mathrm{sg}}\,\eta^{\mathrm{sg}}_{\mathrm{full}}\,\eta^{\mathrm{sg}}_{\max}}{p}+\frac{(p-1)\,L^{B}_{s,\mathrm{sg}}\,\eta^{\mathrm{sg}}_{\mathrm{block}}\,\eta^{\mathrm{sg}}_{\max}}{p}\right). \tag{C.10}$$

Proposition 1. *Under Assumption 1, exact gradients and LMOs, $f \geq f^* > -\infty$, and the fixed rates and momenta specified above, evaluate each score along the trajectory generated by its corresponding method. Let $\Delta = f(\mathbf{x}_0) - f^*$ for a common starting point. Then*

$$\Phi^{\mathrm{mu}} \leq \frac{\Delta}{T}, \qquad \Phi^{\mathrm{mix}}_s \leq \frac{\Delta}{T} \quad \textit{for all } s \in \{1,\dots,H\}, \tag{C.11}$$

Proof. For the all-MuonBP baseline, summing (C.2)–(C.3) over $k = 0,\dots,T-1$ and telescoping

$$\sum_{k=0}^{T-1}\sum_{\ell=1}^{H}\Big[\eta^{\mathrm{mu}}_k\,\|\nabla_\ell f(\mathbf{x}_k)\|^{\mathrm{nuc}}_{\ell,*} - \mathcal{C}^{\mathrm{mu}}_{k,\ell}\Big] \leq \sum_{k=0}^{T-1}\big[f(\mathbf{x}_k) - f(\mathbf{x}_{k+1})\big] = f(\mathbf{x}_0) - f(\mathbf{x}_T) \leq \Delta. \tag{C.12}$$

Dividing by T yields $\Phi^{\mathrm{mu}} \leq \Delta/T$.

For MuonMix, summing (C.5)–(C.6) over $k = 0,\dots,T-1$ and telescoping

$$\sum_{k=0}^{T-1}\left[\sum_{\substack{\ell=1\\ \ell\neq s}}^{H}\Big(\eta^{\mathrm{mu}}_k\,\|\nabla_\ell f(\mathbf{x}_k)\|^{\mathrm{nuc}}_{\ell,*} - \mathcal{C}^{\mathrm{mu}}_{k,\ell}\Big) + \Big(\eta^{\mathrm{sg}}_k\,\|\nabla_s f(\mathbf{x}_k)\|^{1}_{s,*} - \mathcal{C}^{\mathrm{sg}}_{k,s}\Big)\right] \leq \Delta. \tag{C.13}$$

Dividing by T yields $\Phi^{\mathrm{mix}}_s \leq \Delta/T$. □

Per-Layer Comparison. The key observation enabling a per-layer comparison is that, when both score expressions are evaluated on the same fixed reference trajectory with the common rates and constants held fixed, the $H-1$ common MuonBP layers cancel algebraically in Φ^{diff}_s. Thus, Φ^{diff}_s is used as a warm-up fixed trajectory surrogate for layer selection.

Definition 3 (Per-layer optimizer selection criterion)**.** *For each layer $s \in \{1,\dots,H\}$, define the* descent advantage *of assigning layer s to max-norm LMO as,*

$$\Phi^{\mathrm{diff}}_s = \Phi^{\mathrm{mix}}_s - \Phi^{\mathrm{mu}}. \tag{C.14}$$

Proposition 2. *On a fixed trajectory $\{\mathbf{x}_k\}_{k=0}^{T-1}$ with common rates and constants held fixed, the $H-1$ layers $\ell \neq s$ that use MuonBP in both Φ^{mu} and Φ^{mix}_s cancel exactly,*

$$\Phi^{\mathrm{diff}}_s = \big[B^{\mathrm{sg}}_s - E^{\mathrm{sg}}_s\big] - \big[B^{\mathrm{mu}}_s - E^{\mathrm{mu}}_s\big], \tag{C.15}$$

where

$$B^{\mathrm{mu}}_s = \frac{1}{T}\left(\sum_{k\in S_{\mathrm{op}}}\eta^{\mathrm{mu}}_{\mathrm{full}}\|\nabla_s f(\mathbf{x}_k)\|^{\mathrm{nuc}}_{s,*} + \sum_{k\in S_B}\eta^{\mathrm{mu}}_{\mathrm{block}}\|\nabla_s f(\mathbf{x}_k)\|^{\mathrm{nuc}}_{s,*}\right), \tag{C.16}$$

$$B^{\mathrm{sg}}_s = \frac{1}{T}\left(\sum_{k\in S_{\mathrm{op}}}\eta^{\mathrm{sg}}_{\mathrm{full}}\|\nabla_s f(\mathbf{x}_k)\|^{1}_{s,*} + \sum_{k\in S_B}\eta^{\mathrm{sg}}_{\mathrm{block}}\|\nabla_s f(\mathbf{x}_k)\|^{1}_{s,*}\right). \tag{C.17}$$

Proof.

$$\begin{aligned}
\Phi^{\mathrm{diff}}_s &= \Phi^{\mathrm{mix}}_s - \Phi^{\mathrm{mu}} \\
&= \left[B^{\mathrm{mu}}_{\neq s} + B^{\mathrm{sg}}_s - \sum_{\substack{\ell=1\\ \ell\neq s}}^{H} E^{\mathrm{mu}}_\ell - E^{\mathrm{sg}}_s\right] - \left[B^{\mathrm{mu}}_{\neq s} + B^{\mathrm{mu}}_s - \sum_{\substack{\ell=1\\ \ell\neq s}}^{H} E^{\mathrm{mu}}_\ell - E^{\mathrm{mu}}_s\right] \\
&= \big[B^{\mathrm{sg}}_s - E^{\mathrm{sg}}_s\big] - \big[B^{\mathrm{mu}}_s - E^{\mathrm{mu}}_s\big].
\end{aligned} \tag{C.18}$$

□

C.4 Empirical Smoothness Constant Estimation

The smoothness constants $L_{s,\cdot}^{\mathrm{op}}$ and $L_{s,\cdot}^{B}$ are approximated by empirical secant ratios during a short warm-up phase. These ratios are proxies, not certified upper bounds for Assumption 1. In practice, the smoothness constants $L_{s,\cdot}^{\mathrm{op}}$ and $L_{s,\cdot}^{B}$ are not known in advance. We estimate each smoothness constant from the gradient differences and parameter updates observed during a short warm-up phase of training, and then substitute the estimates directly into (C.15) to evaluate the empirical score $\widehat{\Phi}_s^{\mathrm{diff}}$ for each layer s. For MuonBP, we define

$$\widehat{L}_{s,\mathrm{mu}}^{\mathrm{op}} = \frac{1}{|S_{\mathrm{op}}|} \sum_{k \in S_{\mathrm{op}}} \frac{\|\mathbf{G}_{k+1}^{s} - \mathbf{G}_{k}^{s}\|_{s,*}^{\mathrm{nuc}}}{\|\mathbf{x}_{k+1}^{(s)} - \mathbf{x}_{k}^{(s)}\|_{\mathrm{sp}}}, \tag{C.19}$$

$$\widehat{L}_{s,\mathrm{mu}}^{B} = \frac{1}{|S_B|} \sum_{k \in S_B} \frac{\sum_m \|\Delta g_k^{(s,m)}\|_{s,*}^{\mathrm{nuc}}}{\max_m \|\Delta w_k^{(s,m)}\|_{\mathrm{sp}}}, \tag{C.20}$$

where $\Delta g_k^{(s,m)} = [\mathbf{G}_{k+1}^{s}]^{(m)} - [\mathbf{G}_{k}^{s}]^{(m)}$ and $\Delta w_k^{(s,m)} = \mathbf{x}_{k+1}^{(s,m)} - \mathbf{x}_{k}^{(s,m)}$ denote the per-shard stochastic gradient and weight differences. The formulas require nonempty averaging sets and nonzero denominators; when $p = 1$, no block estimate is needed. Gradient differences from different mini-batches include sampling noise, and for coupled objectives may also reflect changes in other layers. Even with exact gradients, an average secant ratio need not upper-bound a global smoothness constant.

For Signum, the counterparts $L_{s,\mathrm{sg}}^{\mathrm{op}}$ and $L_{s,\mathrm{sg}}^{B}$ are estimated analogously, with the dual–primal pair $(\|\cdot\|_{s,*}^{1},\ \|\cdot\|_{\infty})$ replacing $(\|\cdot\|_{s,*}^{\mathrm{nuc}},\ \|\cdot\|_{\mathrm{sp}})$,

$$\widehat{L}_{s,\mathrm{sg}}^{\mathrm{op}} = \frac{1}{|S_{\mathrm{op}}|} \sum_{k \in S_{\mathrm{op}}} \frac{\|\mathbf{G}_{k+1}^{s} - \mathbf{G}_{k}^{s}\|_{s,*}^{1}}{\|\mathbf{x}_{k+1}^{(s)} - \mathbf{x}_{k}^{(s)}\|_{\infty}}, \tag{C.21}$$

$$\widehat{L}_{s,\mathrm{sg}}^{B} = \frac{1}{|S_B|} \sum_{k \in S_B} \frac{\sum_m \|\Delta g_k^{(s,m)}\|_{s,*}^{1}}{\max_m \|\Delta w_k^{(s,m)}\|_{\infty}}. \tag{C.22}$$

Although the full and block max norms coincide, estimates averaged over S_{op} and S_B need not be equal. Using a common empirical $\widehat{L}_{s,\mathrm{sg}}$ in the simplified penalty (3) is an additional profiling choice. Equality of the norms does not imply equality of these two sample averages.