STRUCTMUON: COMPACT CORE COMMUNICATION BY REUSING MUON HISTORY SUBSPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Muon improves large model training through momentum orthogonalization, but scaling requires efficient communication across devices. Low-rank compression can reduce communication costs, but existing methods compute their projection bases separately from the optimizer and must transmit these bases or recompute them with SVDs. Can the optimizer itself provide these bases? We find that Muon's momentum accumulates gradient history, while its orthogonalized update retains paired left and right singular directions. We extract history subspaces from these shared optimizer states, and each worker can reproduce them locally. We propose StructMuon, which uses these subspaces as communication bases and transmits a compact $r \times r$ core. A residual channel exploits concentrated residual energy to send information outside these subspaces, while local error feedback retains unsent information. With contractive residual compression, we prove communication error contraction without SVD safeguards and derive a nonconvex stationarity bound that tightens with better subspace capture. Experiments on GPT-2 Large and NanoGPT-124M show 92.0–97.5% less communication than Full Muon synchronized once per update at matched validation perplexity. These results suggest that communication can reuse the optimizer's own structure.

1 INTRODUCTION

Muon (Jordan et al., 2024) has demonstrated substantial gains in language model training efficiency, reaching a given loss with less computation than AdamW, and recent work (Liu et al., 2025; Kimi Team, 2025) has extended its use to larger models. Such models require training across multiple devices, where gradient synchronization (Agarwal et al., 2022) incurs recurring communication costs. Realizing Muon's efficiency gains at scale therefore calls for compression that reduces these costs without substantially compromising these benefits or adding much computation.

Low-rank compression reduces payloads but also incurs the cost of maintaining projection bases. PowerSGD (Vogels et al., 2019) uses subspace iteration and transmits its bases as rank-q factors, with $\mathcal{O}(q(m+n))$ values for an $m \times n$ matrix. PowerSGD+ (Xie et al., 2025) adds periodic SVD updates to establish convergence guarantees. TSR-Adam (Dang et al., 2026) reduces the dominant payload to an $r \times r$ core, but refreshing its left and right bases requires additional sketch communication and a reduced SVD. These compressors are designed independently of the optimizer, since SGD and Adam act entrywise and offer no matrix structure. Can an optimizer with such structure provide these bases, so that training achieves compact communication with provable error control?

Muon is such an optimizer. Its momentum retains historical gradient information, while its orthogonalized update carries paired left and right singular directions. Orthogonalization reduces the dominance of large singular values while preserving this pairing. Muon's own update thus maps right directions from the momentum to paired left directions, giving a pair of reusable history bases. Since these optimizer states are shared, each worker can extract the same bases locally. Our key idea is to communicate in these bases, enabling compact core transmission without a separate SVD or basis exchange. These history bases may still miss informative components of the current gradient, so reliable reuse requires control of the resulting communication error.

We propose StructMuon, combining Muon history subspaces with residual compression in two complementary channels. A compact $r \times r$ core encodes the component within the history subspaces exactly, while the residual channel applies Top-k outside these subspaces. Top-k exploits the observed

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

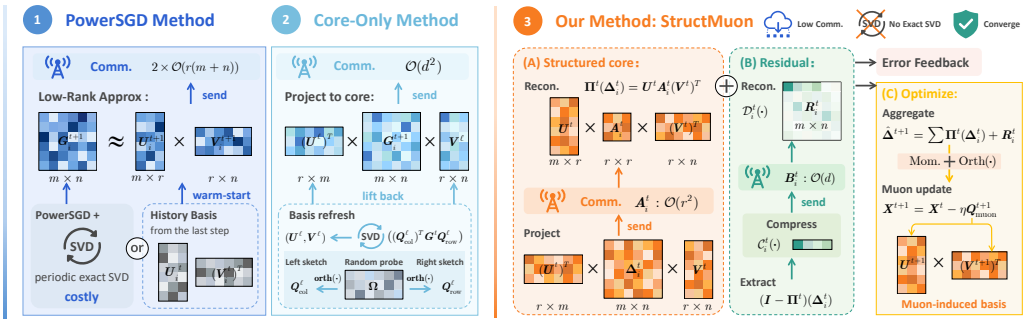


Figure 1: Comparison with factor and sketch-based compression. StructMuon reuses Muon history bases for compact core communication without separate SVDs or basis exchange.

concentration of residual energy to transmit large entries with little additional communication. Local error feedback (Karimireddy et al., 2019) preserves untransmitted information for subsequent projections, allowing transmission through either channel as bases evolve.

Our analysis explains how the two communication channels support optimization. We first bound the information retained by the paired history bases, accounting for differences between the current gradient and the historical structure. Residual compression and error feedback then yield a nonconvex stationarity bound. The bound identifies two ways to reduce communication error: improve the subspace capture or retain more residual information.

Experiments with GPT-2 Large and NanoGPT-124M on OpenWebText show that StructMuon reaches matched validation perplexity with 26.0–45.9% less communication than compressed Muon baselines. Ablations on WikiText-103 show that residual compression improves validation quality with little additional communication, while error feedback yields further gains at the same payload.

Our main contributions are summarized as follows:

- **Reusable subspaces.** We identify the paired structure in Muon's momentum and orthogonalized updates as a source of reusable history subspaces. We bound the energy retained by projection onto these paired subspaces and show that, under suitable spectral conditions, the resulting core can outperform the best rank-q reconstruction at comparable payloads.
- **StructMuon algorithm.** We propose StructMuon, reusing Muon's paired history subspaces as communication bases for $r \times r$ core communication, avoiding separate SVDs and basis exchange. Residual compression complements this reuse by selecting large residual entries, while error feedback retains unsent information across updates.
- **Convergence theory.** We establish communication error contraction and a nonconvex stationarity bound for StructMuon, using residual correction and Muon's orthogonalization properties. The result is expressed in blockwise nuclear gradient norms, tightens as the history subspaces capture more of the true gradient, and requires no exact SVD safeguards.

2 Related Work

Gradient compression reduces communication in distributed optimization through quantization, sign compression, and sparsification. Examples include 1-bit SGD (Seide et al., 2014), QSGD (Alistarh et al., 2017), and gradient sparsification methods such as Top-k (Wangni et al., 2018; Stich et al., 2018; Lin et al., 2018). For biased compressors, error feedback retains omitted information in memory and adds it to subsequent inputs. This mechanism improves convergence and training reliability for sign and Top-k compression (Seide et al., 2014; Stich et al., 2018; Karimireddy et al., 2019). EF21 (Richtárik et al., 2021) and its extensions (Gruntkowska et al., 2023) further develop error feedback and bidirectional compression. Simple biased compressors are useful for the residual channel because they require little computation and work with error feedback.

Low-rank compression exploits matrix structure to reduce payloads. ATOMO (Wang et al., 2018) uses SVD for atomic sparsification. PowerSGD (Vogels et al., 2019) replaces exact SVD with subspace iteration initialized from the previous bases. PowerSGD+ (Xie et al., 2025) addresses

PowerSGD's possible non-convergence through periodic SVD or other contractive safeguards. GaLore (Zhao et al., 2024) uses low-rank gradient projection to reduce memory usage. TSR-Adam (Dang et al., 2026) uses bases on both sides of a matrix to synchronize a compact core instead of low-rank factors. Constructing and maintaining these bases also incurs compression costs.

Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2025) also use matrix structure, but periodically recompute inverse roots or eigenbases. Muon (Jordan et al., 2024) orthogonalizes momentum with Newton–Schulz iterations, pairing left and right directions without decompositions. Scion (Pethick et al., 2025) and Gluon (Riabinin et al., 2025) place this idea in a linear minimization oracle (LMO) framework, and subsequent work studies its convergence properties (Li & Hong, 2025; Kovalev, 2025). MuLoCo (Thérien et al., 2025), Dion (Ahn et al., 2025), and EF21-Muon (Gruntkowska et al., 2025) explore Muon in distributed settings. EF21-Muon tracks local momentum using compressed estimator differences and allows general compressors, including low-rank ones. StructMuon keeps the full Muon update and compresses its inputs into a history core plus a residual.

3 The Proposed Algorithm

3.1 Muon history subspaces

StructMuon represents each corrected input in paired history bases. The right basis selects coordinates from momentum, and the Muon direction supplies the corresponding left coordinates. Workers form both bases locally from shared Muon states, so the bases themselves are never transmitted.

At optimizer update t, worker i compresses the corrected input in each communication round. Suppressing the microstep index, this input is $\boldsymbol{\Delta}_i^t = \boldsymbol{G}_i^t + \boldsymbol{H}_i^t$, where $\boldsymbol{G}_i^t$ is the stochastic gradient and $\boldsymbol{H}_i^t$ is the local error feedback memory. Compression precedes Muon's momentum and orthogonalization steps. The shared history bases $(\boldsymbol{U}^t, \boldsymbol{V}^t)$ have orthonormal columns and are fixed before the current gradients are sampled. Their two-sided projection is

$$\boldsymbol{\Pi}_t(\boldsymbol{X}) = \boldsymbol{U}^t(\boldsymbol{U}^t)^\top \boldsymbol{X} \boldsymbol{V}^t(\boldsymbol{V}^t)^\top.$$

We compare Muon, SGD, and AdamW history bases on the same corrected inputs from NanoGPT-124M. We measure cosine similarity and relative reconstruction error:

$$\mathrm{Cos} = \frac{\langle \boldsymbol{\Pi}_t(\boldsymbol{\Delta}_i^t), \boldsymbol{\Delta}_i^t \rangle}{\|\boldsymbol{\Pi}_t(\boldsymbol{\Delta}_i^t)\|_F \|\boldsymbol{\Delta}_i^t\|_F}, \qquad \mathrm{RelErr} = \frac{\|\boldsymbol{\Delta}_i^t - \boldsymbol{\Pi}_t(\boldsymbol{\Delta}_i^t)\|_F}{\|\boldsymbol{\Delta}_i^t\|_F}.$$

Higher Cos and lower RelErr indicate better reconstruction. Appendix A.3 defines their global aggregation across compressed hidden matrices, workers, and microsteps.

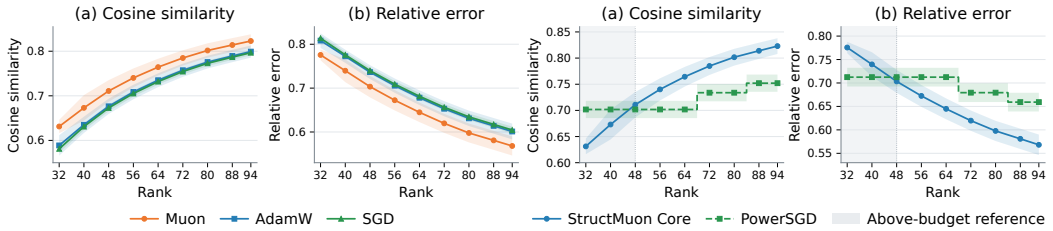


Figure 2: History bases on NanoGPT-124M at $r = 32$–94. Cosine similarity (left) and relative reconstruction error (right).

Figure 3: Core versus PowerSGD on NanoGPT-124M at ranks 32–94. The gray region marks above-budget references; ranks $r \geq 48$ satisfy the shared structured communication budget cap.

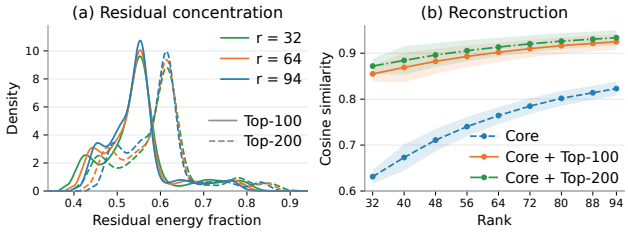
Muon history bases achieve higher Cos and lower RelErr than AdamW and SGD at every tested rank, 32–94 (Figure 2). The advantage is largest at lower ranks and narrows as the rank grows, supporting Muon history reuse when the core budget is tight.

Under a shared structured budget cap, PowerSGD transmits $q(m+n)$ factor coefficients for an $m \times n$ input, and the core transmits r^2 coefficients. PowerSGD receives the largest feasible rank (Appendix A.1). At $r = 32, 40$, even $q = 1$ exceeds the cap; comparisons at $r \geq 48$ respect it.

Both methods reconstruct the same $\boldsymbol{\Delta}_i^t$ using the metrics above. At $r = 48$, their structured payloads match, and the core has higher cosine similarity and lower error (Figure 3). This ordering persists across the budget-feasible ranks. At $r = 94$, the core attains cosine similarity 0.823 versus 0.752 and relative error 0.568 versus 0.659. Local basis reuse thus supports more representation coordinates within the shared budget cap, translating the larger rank into better reconstruction.

3.2 RESIDUAL COMPRESSION

The history subspaces provide compact coordinates, but their coverage of the current corrected input can be incomplete under a tight rank budget. We therefore examine the residual $\mathbf{\Delta}_i^t - \mathbf{\Pi}_t(\mathbf{\Delta}_i^t)$ to determine how to communicate information that the core misses.



Rank	k	Energy	Core	+Res.	Gain
32	100	54.9%	0.631	0.855	+0.224
32	200	59.8%	0.631	0.872	+0.241
64	100	54.6%	0.764	0.902	+0.138
64	200	59.5%	0.764	0.913	+0.149
94	100	54.5%	0.823	0.925	+0.102
94	200	59.6%	0.823	0.934	+0.111

Figure 4: Residual correction on NanoGPT-124M. Left panels: residual energy densities at $r = 32, 64, 94$ and reconstruction cosine similarity over $r = 32$–94. Solid and dashed density curves denote Top-100 and Top-200. The table reports retained residual energy and cosine gains; bands show ± 1 sample standard deviation over five seeds.

Figure 4 shows how residual correction improves reconstruction. Across ranks 32–94, Top-100 retains 54.5–54.9% of residual energy. At $r = 94$, Top-100 raises cosine similarity from 0.823 to 0.925. Selecting the largest residual entries retains the most Frobenius energy at a fixed sparsity budget, complementing the shared Muon history core.

StructMuon assigns the captured component and the residual to complementary communication channels. Worker i transmits the $r \times r$ core $\boldsymbol{A}_i^t = (\boldsymbol{U}^t)^\top \mathbf{\Delta}_i^t \boldsymbol{V}^t$ and applies a compressor to the residual. Combining the two channels reconstructs the corrected input as

$$\widehat{\mathbf{\Delta}}_i^t := \mathcal{K}_t(\mathbf{\Delta}_i^t) = \underbrace{\mathbf{\Pi}_t(\mathbf{\Delta}_i^t)}_{\text{core reconstruction}} + \underbrace{\mathcal{D}_t\mathcal{C}_t\big(\mathbf{\Delta}_i^t - \mathbf{\Pi}_t(\mathbf{\Delta}_i^t)\big)}_{\text{residual reconstruction}}.$$

Here $\mathcal{C}_t$ is a residual compressor such as Top-k, and $\mathcal{D}_t$ is its decoder. With Top-k, the message contains r^2 core values, k residual values, and residual indices. The core preserves the captured component exactly, so all communication error comes from residual compression. This separation yields communication error contraction for any orthonormal history bases whenever the residual compressor is contractive, as established in Lemma 1.

3.3 THE STRUCTMUON ALGORITHM

StructMuon combines core communication, residual compression, and local error feedback with Muon optimization. Workers aggregate their reconstructions for a shared Muon update, whose states supply the history bases for the next iteration. Algorithm 1 summarizes this procedure.

Each worker compresses the sum of its gradient and local error memory, $\mathbf{\Delta}_i^{t,s} = \boldsymbol{G}_i^{t,s} + \boldsymbol{H}_i^{t,s}$, into a core $\boldsymbol{A}_i^{t,s}$ and a residual message $\boldsymbol{B}_i^{t,s}$. Here $\mathcal{C}_t$ compresses the residual, and $\mathcal{D}_t$ decodes its message into a matrix approximation. Shared history bases give the core coefficients the same meaning across workers, so they can be averaged before reconstruction. The bases define the communication coordinates while remaining local to each worker.

For N workers and S microsteps per optimizer update, each processing one microbatch, we average the reconstructions while holding parameters and history bases fixed:

$$\boldsymbol{A}^{t,s} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{A}_i^{t,s}, \qquad \widehat{\mathbf{\Delta}}^{t,s} = \boldsymbol{U}^t \boldsymbol{A}^{t,s} (\boldsymbol{V}^t)^\top + \frac{1}{N}\sum_{i=1}^{N} \mathcal{D}_t(\boldsymbol{B}_i^{t,s}), \qquad \widehat{\mathbf{\Delta}}^{t} = \frac{1}{S}\sum_{s=0}^{S-1} \widehat{\mathbf{\Delta}}^{t,s}.$$

For Top-k, aggregation averages decoded residuals over their combined support, treating absent entries as zero; it does not apply Top-k again. All workers then compute the Muon update from the same reconstructed input, using momentum and Newton–Schulz (NS) orthogonalization. They consequently share the optimizer states from which the next bases are extracted, preserving common communication coordinates throughout distributed training.

Algorithm 1 StructMuon

Input: initial iterate $\boldsymbol{X}^0$, shared orthonormal bases $\boldsymbol{U}^0, \boldsymbol{V}^0$, momentum $\beta \in [0,1)$, S microsteps per optimizer update

initialize $\boldsymbol{H}_i^{0,0} \leftarrow \boldsymbol{0}$ for all $i \in [N]$, and $\boldsymbol{M}^0 \leftarrow \boldsymbol{0}$
for $t = 0, 1, \ldots, T-1$ **do**
  for $s = 0, 1, \ldots, S-1$ **do**
    for all workers $i \in [N]$ **in parallel do**
      $(\boldsymbol{A}_i^{t,s}, \boldsymbol{B}_i^{t,s}, \boldsymbol{H}_i^{t,s+1}) \leftarrow \textsc{Local-Compress}(\boldsymbol{U}^t, \boldsymbol{V}^t, \boldsymbol{X}^t, \boldsymbol{H}_i^{t,s})$
    end for
    $\boldsymbol{A}^{t,s} \leftarrow \mathrm{Aggregate}(\{\boldsymbol{A}_i^{t,s}\}_{i=1}^N), \quad \boldsymbol{B}^{t,s} \leftarrow \mathrm{Aggregate}(\{\boldsymbol{B}_i^{t,s}\}_{i=1}^N)$
    $\widehat{\boldsymbol{\Delta}}^{t,s} \leftarrow \boldsymbol{U}^t \boldsymbol{A}^{t,s} (\boldsymbol{V}^t)^\top + \mathcal{D}_t(\boldsymbol{B}^{t,s})$
  end for
  $\widehat{\boldsymbol{\Delta}}^t \leftarrow S^{-1} \sum_{s=0}^{S-1} \widehat{\boldsymbol{\Delta}}^{t,s}, \quad \boldsymbol{H}_i^{t+1,0} \leftarrow \boldsymbol{H}_i^{t,S}$ for all $i \in [N]$
  $\boldsymbol{M}^{t+1} \leftarrow \beta \boldsymbol{M}^t + \widehat{\boldsymbol{\Delta}}^t, \quad \widetilde{\boldsymbol{M}}^{t+1} \leftarrow \widehat{\boldsymbol{\Delta}}^t + \beta \boldsymbol{M}^{t+1}$
  $\boldsymbol{Q}^{t+1} \leftarrow \mathrm{NSOrth}(\widetilde{\boldsymbol{M}}^{t+1}), \quad \boldsymbol{X}^{t+1} \leftarrow \boldsymbol{X}^t - \eta_t \boldsymbol{Q}^{t+1}$
  $(\boldsymbol{U}^{t+1}, \boldsymbol{V}^{t+1}) \leftarrow \mathrm{Basis}(\widetilde{\boldsymbol{M}}^{t+1}, \boldsymbol{Q}^{t+1}, r)$
end for
Output: $\boldsymbol{X}^T$

Algorithm 2 LOCAL-COMPRESS at one microstep

function $\textsc{Local-Compress}(\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{X}, \boldsymbol{H}_i)$
  $\boldsymbol{G}_i \leftarrow \nabla F_i(\boldsymbol{X}, \xi_i), \quad \boldsymbol{\Delta}_i \leftarrow \boldsymbol{G}_i + \boldsymbol{H}_i$
  $\boldsymbol{A}_i \leftarrow \boldsymbol{U}^\top \boldsymbol{\Delta}_i \boldsymbol{V}$ ▷ structured core
  $\boldsymbol{B}_i \leftarrow \mathcal{C}(\boldsymbol{\Delta}_i - \boldsymbol{U}\boldsymbol{A}_i\boldsymbol{V}^\top)$ ▷ residual compression
  $\boldsymbol{H}_i \leftarrow \boldsymbol{\Delta}_i - (\boldsymbol{U}\boldsymbol{A}_i\boldsymbol{V}^\top + \mathcal{D}(\boldsymbol{B}_i))$ ▷ error feedback
  return $(\boldsymbol{A}_i, \boldsymbol{B}_i, \boldsymbol{H}_i)$
end function

The effective momentum $\widetilde{\boldsymbol{M}}^{t+1}$ supplies gradient history, while the Muon direction $\boldsymbol{Q}^{t+1}$ links the right and left subspaces. With $\mathcal{I}$ a shared set of r row indices fixed before training and qf the orthonormal factor of a thin QR factorization, the next bases are

$$\boldsymbol{V}^{t+1} = \mathrm{qf}\big((\widetilde{\boldsymbol{M}}_{\mathcal{I},:}^{t+1})^\top\big), \qquad \boldsymbol{U}^{t+1} = \mathrm{qf}\big(\boldsymbol{Q}^{t+1}\boldsymbol{V}^{t+1}\big).$$

The ideal polar map sets nonzero singular values to one, preserving paired left and right directions. It transfers selected right coordinates to the left without applying historical singular values again (Appendix A.1.3). Workers reproduce both bases from shared states, row indices, and the QR convention (Appendix A.1.2), enabling core communication without a separate SVD or basis exchange. In our main settings, basis extraction adds 1.20–2.92% of the FLOPs of five NS iterations (Appendix A.1.5), and Appendix A.1.6 compares the computation added by each compression method.

Residual compression improves the current reconstruction, while error feedback carries what remains uncommunicated into later inputs. Each memory records a worker's local reconstruction error before aggregation. Because it stays in the original matrix coordinates and is added before projection, information missed by the current bases can enter the core after a later refresh. Thus, updating the history subspaces does not discard information retained from earlier steps. With $\widehat{\boldsymbol{\Delta}}_i^{t,s}$ denoting the local reconstruction and $\boldsymbol{H}_i^{t+1,0} = \boldsymbol{H}_i^{t,S}$, the memory recursion yields

$$\sum_{t=0}^{T-1} \frac{1}{S} \sum_{s=0}^{S-1} \big(\widehat{\boldsymbol{\Delta}}_i^{t,s} - \boldsymbol{G}_i^{t,s}\big) = \frac{\boldsymbol{H}_i^{0,0} - \boldsymbol{H}_i^{T,0}}{S}.$$

With zero initial memory, the final memory accounts for the accumulated communication discrepancy, even as the bases evolve. Averaging over workers preserves this identity for the shared Muon input. The following analysis bounds the local memories and their effect on momentum tracking and Muon's progress toward stationarity under compressed communication.

4 THEORY

We study how communication error affects Muon's progress toward a stationary point. The bound shows how better history subspaces and residual correction reduce this error. We analyze one communication round per update ($S = 1$); Appendix A.2.7 covers any fixed number of rounds, and Appendix A.2.6 covers approximate orthogonalization with finitely many NS iterations.

4.1 SETUP AND ASSUMPTIONS

Let $f(\boldsymbol{X}) := N^{-1}\sum_{i=1}^{N} f_i(\boldsymbol{X})$, where $\boldsymbol{X} = (\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(B)})$ consists of B matrix blocks $\boldsymbol{X}^{(b)} \in \mathbb{R}^{m_b \times n_b}$. We assume $f^\star := \inf_{\boldsymbol{X}} f(\boldsymbol{X}) > -\infty$. For each $b \in [B]$, let $\boldsymbol{\Pi}_b^t$ be the orthogonal projector in the Frobenius inner product induced by the orthonormal history bases available at the start of iteration t. With residual compressor $\mathcal{C}_{b,t}$ and decoder $\mathcal{D}_{b,t}$, Algorithm 2 uses the following blockwise communication operator to reconstruct the corrected input

$$\mathcal{K}_{b,t}(\boldsymbol{Z}) := \boldsymbol{\Pi}_b^t(\boldsymbol{Z}) + \mathcal{D}_{b,t}\mathcal{C}_{b,t}\big((\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{Z}\big).$$

Let $\mathcal{F}_t$ denote the history before sampling the stochastic gradients at iteration t, and $\mathcal{G}_t$ the enlarged history after sampling these gradients but before drawing compression randomness. The iterate $\boldsymbol{X}^t$, local memories, and history bases are $\mathcal{F}_t$-measurable; the corrected inputs are $\mathcal{G}_t$-measurable.

We use the spectral norm $\|\cdot\|_{2\to 2}$ for parameter updates, its dual nuclear norm $\|\cdot\|_*$ for gradients, and the Frobenius norm for communication errors. With $\kappa_b := \sqrt{\min\{m_b, n_b\}}$ and $\kappa_{\max} := \max_{b\in[B]} \kappa_b$, we have $\|\boldsymbol{Z}^{(b)}\|_* \le \kappa_b \|\boldsymbol{Z}^{(b)}\|_F$. The analysis uses the ideal polar map for NSOrth.

Remark 1 (Muon descent direction) *For each block, omitting its index, the polar map satisfies $\|\boldsymbol{Q}^{t+1}\|_{2\to 2} \le 1$ and $\langle \widetilde{\boldsymbol{M}}^{t+1}, \boldsymbol{Q}^{t+1}\rangle = \|\widetilde{\boldsymbol{M}}^{t+1}\|_*$. These properties imply the alignment bound*

$$\big\langle \nabla f(\boldsymbol{X}^t), \boldsymbol{Q}^{t+1}\big\rangle \ge \big\|\nabla f(\boldsymbol{X}^t)\big\|_* - 2\big\|(1-\beta)\widetilde{\boldsymbol{M}}^{t+1} - \nabla f(\boldsymbol{X}^t)\big\|_*.$$

The factor $1-\beta$ normalizes momentum without changing the polar direction. Thus, the descent analysis reduces to controlling how accurately the effective momentum tracks the true gradient.

Assumption 1 (Residual contraction) *For each block $b \in [B]$, let $\alpha_b \in (0, 1]$ and set $\alpha := \min_{b\in[B]} \alpha_b$. For every iteration t and every $\mathcal{G}_t$-measurable residual matrix $\boldsymbol{R}$, assume*

$$\mathbb{E}\big[\|\boldsymbol{R} - \mathcal{D}_{b,t}\mathcal{C}_{b,t}(\boldsymbol{R})\|_F^2 \mid \mathcal{G}_t\big] \le (1-\alpha_b)\|\boldsymbol{R}\|_F^2.$$

Assumption 2 (Stochastic gradients) *For each worker $i \in [N]$ and block $b \in [B]$, the stochastic gradients satisfy the following conditional mean and variance bounds at every iteration t:*

$$\mathbb{E}\big[\boldsymbol{G}_i^{(b),t} \mid \mathcal{F}_t\big] = \nabla^{(b)} f_i(\boldsymbol{X}^t), \qquad \mathbb{E}\big[\|\boldsymbol{G}_i^{(b),t} - \nabla^{(b)} f_i(\boldsymbol{X}^t)\|_F^2 \mid \mathcal{F}_t\big] \le \sigma_b^2.$$

Assumption 3 (Bounded heterogeneity) *For every iteration t, the local gradients satisfy*

$$\frac{1}{N}\sum_{i=1}^{N}\sum_{b=1}^{B} \|\nabla^{(b)} f_i(\boldsymbol{X}^t) - \nabla^{(b)} f(\boldsymbol{X}^t)\|_F^2 \le \zeta^2.$$

For homogeneous local objectives, the bound holds with $\zeta = 0$.

Assumption 4 (Blockwise smoothness and gradient regularity) *For all block tuples $\boldsymbol{X}, \boldsymbol{Y}$ and perturbations $\boldsymbol{U}$, assume the following upper bound on the objective:*

$$f(\boldsymbol{X} + \boldsymbol{U}) \le f(\boldsymbol{X}) + \sum_{b=1}^{B} \langle \nabla^{(b)} f(\boldsymbol{X}), \boldsymbol{U}^{(b)}\rangle + \frac{1}{2}\sum_{b=1}^{B} L_b \|\boldsymbol{U}^{(b)}\|_{2\to 2}^2.$$

In addition, with $L_{\max} := \max_{b\in[B]} L_b$, the gradient map satisfies

$$\Big(\sum_{b=1}^{B} \|\nabla^{(b)} f(\boldsymbol{X}) - \nabla^{(b)} f(\boldsymbol{Y})\|_*^2\Big)^{1/2} \le L_{\max}\Big(\sum_{b=1}^{B} \|\boldsymbol{X}^{(b)} - \boldsymbol{Y}^{(b)}\|_{2\to 2}^2\Big)^{1/2}.$$

Energy retained by the history bases. Suppress block and time indices in $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{\Pi}$, and let $\boldsymbol{Q}_\star$ be the ideal orthogonalized direction of the historical effective momentum $\widetilde{\boldsymbol{M}}$. If the sampled rows have rank r, then $\boldsymbol{U} = \boldsymbol{Q}_\star \boldsymbol{V}$. Consider a reference matrix $\boldsymbol{Z}_0$ with the same polar factor $\boldsymbol{Q}_\star$. Let

$\lambda_- > 0$ and λ_+ be lower and upper bounds on its nonzero singular values. Then

$$\|\mathbf{\Pi}(\boldsymbol{Z}_0)\|_F^2 \geq \frac{4\lambda_-\lambda_+}{(\lambda_- + \lambda_+)^2}\|\boldsymbol{Z}_0\boldsymbol{V}\|_F^2.$$

For $\boldsymbol{Z}_0$, the two-sided projection thus retains a controlled fraction of the energy captured by the right projection. The guarantee is stronger when the reference matrix has more uniform nonzero singular values. At comparable structured payloads, $r\lambda_-^2 > q\lambda_+^2$ further guarantees smaller core error than the best rank-q reconstruction of $\boldsymbol{Z}_0$ (Corollary 2). Appendix A.1.4 extends this bound to current gradients that differ from the reference through a perturbation bound.

4.2 Nonconvex Convergence

We establish a nonconvex convergence bound for StructMuon in terms of the average gradient norm. The analysis controls communication error through residual compression and error feedback, then bounds its effect on momentum tracking and Muon's descent. Appendix A.2 contains all proofs.

Lemma 1 (Composite residual contraction) *Under Assumption 1, for every block $b \in [B]$, every iteration t, and any $\mathcal{G}_t$-measurable matrix $\boldsymbol{Z}$, we have*

$$\mathbb{E}\big[\|\boldsymbol{Z} - \mathcal{K}_{b,t}(\boldsymbol{Z})\|_F^2 \mid \mathcal{G}_t\big] \leq (1-\alpha_b)\|(\boldsymbol{I} - \mathbf{\Pi}_b^t)\boldsymbol{Z}\|_F^2 \leq (1-\alpha_b)\|\boldsymbol{Z}\|_F^2.$$

Lemma 1 ensures contraction throughout history reuse and bounds the next local memory by the residual after core reconstruction. The identity $\widehat{\boldsymbol{\Delta}}_i^t - \boldsymbol{G}_i^t = \boldsymbol{H}_i^t - \boldsymbol{H}_i^{t+1}$ connects successive memories to gradient reconstruction error, allowing us to control momentum tracking.

To state the stationarity guarantee and its dependence on subspace quality, define

$$H_T := \sum_{t=0}^{T-1}\eta_t, \qquad Q_T := \frac{1}{H_T}\sum_{t=0}^{T-1}\eta_t^2,$$
$$R_T := \frac{1}{H_T}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\Big[\Big(\sum_{b=1}^{B}\|(\boldsymbol{I} - \mathbf{\Pi}_b^t)\nabla^{(b)}f(\boldsymbol{X}^t)\|_F^2\Big)^{1/2}\Big].$$

Here R_T measures the expected magnitude of the true gradient outside the history subspaces, weighted by the stepsizes. The term Q_T collects the costs of smoothness and momentum tracking. Let $\mathcal{M}_0 := N^{-1}\sum_{i=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{H}_i^{(b),0}\|_F^2$ and $\Theta_1 := \sum_{b=1}^{B}\|(1-\beta)\boldsymbol{M}^{(b),1} - \nabla^{(b)}f(\boldsymbol{X}^0)\|_*^2$ denote the initial memory energy and momentum tracking error, respectively.

Theorem 1 (Nonconvex convergence) *Suppose Assumptions 1–4 hold. Let the stepsizes be deterministic, positive, nonincreasing, and common to all blocks. For every $T \geq 1$, the following holds with a constant $C > 0$ depending only on α, β, $\kappa_{\max}$, B, and $\{L_b\}_{b=1}^B$:*

$$\frac{1}{H_T}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\big[\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*\big]$$
$$\leq C\Bigg[\frac{f(\boldsymbol{X}^0) - f^\star + \eta_0\mathbb{E}[\sqrt{\mathcal{M}_0} + \sqrt{\Theta_1}]}{H_T} + \Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2} + Q_T\Bigg] + \frac{4\sqrt{B}\,\kappa_{\max}\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}(R_T + \zeta).$$

Remark 2 (Finite-horizon guarantee) For a fixed horizon T, take $\eta_t = T^{-1/2}$ for $0 \leq t < T$. Then $H_T = \sqrt{T}$ and $Q_T = T^{-1/2}$. With fixed problem parameters and uniformly bounded initial objective gap and expected initial errors, Theorem 1 gives

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\Big[\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*\Big] \leq \mathcal{O}(T^{-1/2}) + C\Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2} + \frac{4\sqrt{B}\,\kappa_{\max}\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}(R_T + \zeta).$$

The optimization term decreases as $T^{-1/2}$. The remaining terms account for stochastic noise, differences between workers, and the gradient component missed by the history subspaces. With $1-\beta = T^{-1/2}$ and $\eta_t = T^{-3/4}$, all three terms vanish, even if $\zeta > 0$ (Appendix A.2.5).

Appendix A.1.4 bounds R_T through two sources: information outside the right history subspace and further loss in the paired left projection. The latter depends on the reference matrix's singular-value spread and its difference from the current gradient. Better capture reduces R_T, while retaining more residual information reduces its coefficient. For fixed β, whether R_T decays remains open.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We test whether StructMuon improves training quality at comparable payloads and reaches a common quality with less communication. Main comparisons use OpenWebText; the main ablations use WikiText-103 to examine residual compression and error feedback. An additional OpenWebText residual ablation uses the main NanoGPT configuration. All runs use PyTorch on four NVIDIA A800 80GB GPUs with four logical data-parallel workers. Training results use five random seeds; their shaded bands show one sample standard deviation.

Full Muon is the uncompressed reference. PowerSGD communicates low-rank factors using momentum SGD with error feedback. EF-PowerMuon combines factor communication with Muon and error feedback. We also compare with EF21-Muon (Gruntkowska et al., 2025) using Top-K compression, where K is chosen to match the payload of the factor baselines (Appendix A.3).

The OpenWebText experiments use a 1.5B-token training subset and 2M tokens each for validation and testing. Both models are trained from scratch. Hidden weights use the learning rates in Table 7; all AdamW auxiliary groups use 2×10^{-4}. All Muon variants use hidden-weight peak learning rate 2×10^{-2}. Methods share weight decay 10^{-4}, a 25K-update schedule with four microsteps per update, and evaluation intervals. Muon uses momentum 0.95 and five NS iterations.

StructMuon uses $(r, k) = (276, 150)$ for GPT-2 Large and $(94, 100)$ for NanoGPT-124M; the factor baselines use $q = 15$ and $q = 3$, respectively. Top-k retains the k largest residual entries in magnitude per matrix, worker, and round. The compressed methods use fused QKV and approximately matched payloads. Communication is cumulative uplink logical payload per worker, including sparse indices and all parameter groups; Appendix A.3 specifies the accounting. Appendix A.5 varies the core rank r and the residual budget k, showing diminishing returns in both.

5.2 MAIN LANGUAGE MODEL TRAINING RESULTS

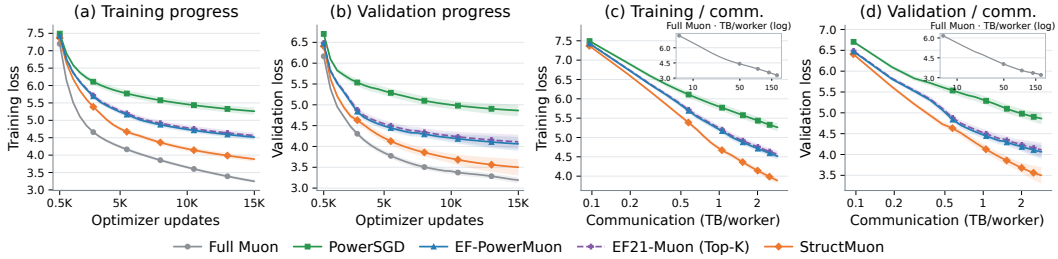


Figure 5: Five-method comparison on GPT-2 Large from 500 to 15K updates, including EF21-Muon. Communication panels use logarithmic TB/worker axes with Full Muon insets. Curves and shading show mean ± 1 SD over five seeds.

Table 1: Results at 10K optimizer updates; Comm. is cumulative uplink logical payload per worker. Cos. reports the global cosine similarity defined in Appendix A.3; Full Muon is uncompressed, so Cos. does not apply.

Method	GPT-2 Large, step = 10K					NanoGPT-124M, step = 10K				
	Train ↓	Val ↓	PPL ↓	Comm. ↓	Cos. ↑	Train ↓	Val ↓	PPL ↓	Comm. ↓	Cos. ↑
Full Muon	3.65	3.40	30.1	123.740 TB	–	4.72	4.12	61.8	19.853 TB	–
PowerSGD	5.46	5.00	148.4	1.917 TB	0.787	5.38	5.11	165.9	0.107 TB	0.752
EF-PowerMuon	4.74	4.21	67.1	1.917 TB	0.846	5.20	4.86	128.4	0.107 TB	0.827
EF21-Muon	4.78	4.25	70.4	1.917 TB	0.834	5.24	4.90	134.5	0.107 TB	0.814
StructMuon	4.18	3.71	40.9	1.909 TB	0.928	4.94	4.40	81.4	0.106 TB	0.925

At 10K updates, StructMuon achieves the lowest validation loss among the compressed methods (Table 1). Relative to EF-PowerMuon, validation loss falls from 4.21 to 3.71 on GPT-2 Large and from 4.86 to 4.40 on NanoGPT-124M, with slightly smaller payloads. The advantage also holds against EF21-Muon, supporting the proposed communication representation. Full Muon retains lower PPL, showing the remaining quality trade-off at fixed updates.

StructMuon improves reconstruction alignment over EF-PowerMuon: cosine similarity rises from 0.846 to 0.928 on GPT-2 Large and from 0.827 to 0.925 on NanoGPT-124M at 10K updates (Table 1). Figure 11 tracks the combined core and residual reconstruction throughout training.

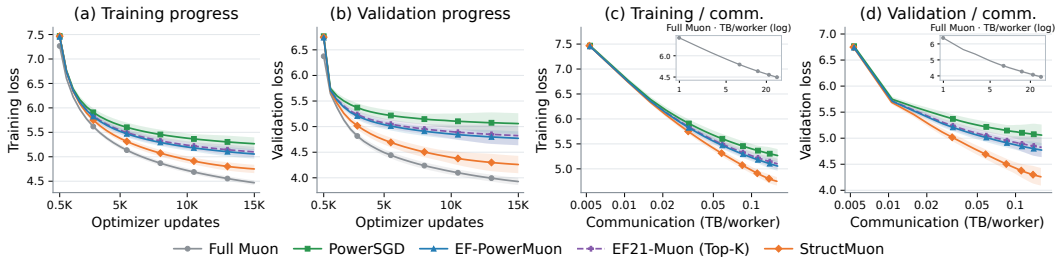


Figure 6: Five-method comparison on NanoGPT-124M from 500 to 15K updates, including EF21-Muon. Communication panels use logarithmic TB/worker axes with Full Muon insets. Curves and shading show mean ± 1 SD over five seeds.

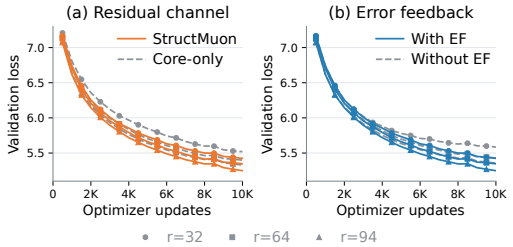
Table 2: Cumulative uplink logical payload per worker at target PPL. Saving is the relative reduction achieved by StructMuon against each baseline. NanoGPT PowerSGD reaches PPL 160 after 10K updates. Full Muon (1 round) synchronizes accumulated gradients once per update.

Method	GPT-2 Large, PPL $= 150$		GPT-2 Large, PPL $= 130$		NanoGPT-124M, PPL $= 180$		NanoGPT-124M, PPL $= 160$	
	Comm. $\downarrow$	Saving (%) $\uparrow$	Comm. $\downarrow$	Saving (%) $\uparrow$	Comm. $\downarrow$	Saving (%) $\uparrow$	Comm. $\downarrow$	Saving (%) $\uparrow$
StructMuon	0.3596 TB	–	0.4171 TB	–	0.0237 TB	–	0.0287 TB	–
EF-PowerMuon	0.5047 TB	28.7%	0.5634 TB	26.0%	0.0336 TB	29.4%	0.0474 TB	39.4%
EF21-Muon	0.5204 TB	30.9%	0.5834 TB	28.5%	0.0365 TB	35.0%	0.0531 TB	45.9%
PowerSGD	1.8682 TB	80.7%	2.8274 TB	85.2%	0.0665 TB	64.4%	0.1394 TB	79.4%
Full Muon (4 rounds)	17.9195 TB	98.0%	20.9879 TB	98.0%	3.7276 TB	99.4%	4.2846 TB	99.3%
Full Muon (1 round)	4.4799 TB	92.0%	5.2470 TB	92.1%	0.9319 TB	97.5%	1.0712 TB	97.3%

At matched validation PPL, StructMuon saves 26.0–39.4% over EF-PowerMuon and 28.5–45.9% over EF21-Muon (Table 2). With payloads within 1.3% per update, these gains mainly reflect fewer updates. Against Full Muon with gradients synchronized once per update, the saving is 92.0–97.5% (Table 2; Appendix A.3). StructMuon reaches each target 2.03–2.66$\times$ faster than four-round Full Muon and before all compressed baselines (Appendices A.4 and A.8).

5.3 Ablation Studies

We ablate StructMuon on WikiText-103. Core-only sets $k = 0$, and w/o EF removes error feedback.



Rank	Method	Comm. (TB)	Train Loss	Val Loss	PPL
64	**StructMuon**	0.070	**5.941**	**5.347**	**209.980**
64	Core-only	**0.068**	6.003	5.399	221.190
64	w/o EF	0.070	6.040	5.430	228.150
94	**StructMuon**	0.106	**5.844**	**5.249**	**190.380**
94	Core-only	**0.104**	5.941	5.330	206.440
94	w/o EF	0.106	5.974	5.357	212.090

Figure 7: Core-only and w/o EF comparisons on WikiText-103 at 10K updates. Left: validation loss at ranks 32, 64, and 94. Right: final losses, PPL, and cumulative uplink payload per worker at ranks 64 and 94.

At rank 94, StructMuon achieves validation loss 5.249, compared with 5.330 for Core-only, using 2.304 GB more communication (Figure 7). The w/o EF variant reaches 5.357 at the same payload as StructMuon. StructMuon also gives the lowest loss at rank 64. Residual correction improves quality with little extra communication, while error feedback adds gains at the same payload.

On OpenWebText at $r = 94$, adding the residual channel lowers PPL from 165.3 to 81.4 with 1.5% more communication (Appendix A.7). This loss reduction of 0.708 is much larger than the 0.081 on WikiText-103. Replacing only the Momentum-only left basis with the Muon direction lowers PPL from 183.2 to 81.4 at the same payload, while Random bases reach only 903.1 (Appendix A.6).

6 Conclusion

StructMuon reuses Muon's history subspaces as shared coordinates for compact core communication. A residual channel with error feedback preserves information outside these subspaces. We prove communication error contraction and a nonconvex stationarity bound without exact SVDs. At matched validation perplexity, StructMuon uses 26.0–45.9% less communication than compressed Muon baselines. Future work includes multi-node evaluation and sharded training, as well as combining the core with quantization or less frequent synchronization.

AI USE STATEMENT

In this work, we used generative AI tools to write and edit code for implementing our method and running experiments. We also used them to assist with writing, including revising text for clarity and grammar and formatting LaTeX. We did not use generative AI tools for any task that requires disclosure, such as generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, providing critical steps of proofs, proposing or refining hypotheses, designing research methodology or experiments, cleaning or reformatting datasets, performing thematic data analysis, or interpreting results. The authors reviewed all AI-assisted work for technical accuracy and consistency with our contributions. We take full responsibility for this paper.

REFERENCES

Saurabh Agarwal, Hongyi Wang, Shivaram Venkataraman, and Dimitris Papailiopoulos. On the utility of gradient compression in distributed training systems. In *Proceedings of Machine Learning and Systems*, volume 4, 2022.

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.

Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

Sizhe Dang, Jiaqi Shao, Xiaodong Zheng, Guang Dai, Yan Song, and Haishan Ye. From $o(mn)$ to $o(r^2)$: Two-sided low-rank communication for Adam in distributed training with memory efficiency. *arXiv preprint arXiv:2602.08007*, 2026.

Kaja Gruntkowska, Alexander Tyurin, and Peter Richtárik. EF21-P and friends: Improved theoretical communication complexity for distributed optimization with bidirectional compression. In *International conference on machine learning*, pp. 11761–11807. PMLR, 2023.

Kaja Gruntkowska, Alexander Gaponov, Zhirayr Tovmasyan, and Peter Richtárik. Error feedback for Muon and friends. *arXiv preprint arXiv:2510.00643*, 2025.

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

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

Sai Praneeth Karimireddy, Quentin Rebjock, Sebastian Stich, and Martin Jaggi. Error feedback fixes SignSGD and other gradient compression schemes. In *International conference on machine learning*, pp. 3252–3261. PMLR, 2019.

Kimi Team. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025. URL `https://arxiv.org/abs/2507.20534`.

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

Jiaxiang Li and Mingyi Hong. A note on the convergence of Muon and further. *arXiv preprint arXiv:2502.02900v1*, 2025. URL `https://arxiv.org/abs/2502.02900v1`.

Yujun Lin, Song Han, Huizi Mao, Yu Wang, and William J. Dally. Deep gradient compression: Reducing the communication bandwidth for distributed training. In *International Conference on Learning Representations*, 2018.

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

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.

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.

Peter Richtárik, Igor Sokolov, and Ilyas Fatkhullin. EF21: A new, simpler, theoretically better, and practically faster error feedback. *Advances in Neural Information Processing Systems*, 34: 4384–4396, 2021.

Frank Seide, Hao Fu, Jasha Droppo, Gang Li, and Dong Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns. In *Interspeech*, volume 2014, pp. 1058–1062. Singapore, 2014.

Sebastian U Stich, Jean-Baptiste Cordonnier, and Martin Jaggi. Sparsified SGD with memory. *Advances in neural information processing systems*, 31, 2018.

Benjamin Thérien, Xiaolong Huang, Irina Rish, and Eugene Belilovsky. MuLoCo: Muon is a practical inner optimizer for DiLoCo. *arXiv preprint arXiv:2505.23725v1*, 2025. URL https://arxiv.org/abs/2505.23725v1.

Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. PowerSGD: Practical low-rank gradient compression for distributed optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham M. Kakade. SOAP: Improving and stabilizing Shampoo using Adam for language modeling. In *International Conference on Learning Representations*, 2025.

Hongyi Wang, Scott Sievert, Shengchao Liu, Zachary Charles, Dimitris Papailiopoulos, and Stephen Wright. ATOMO: Communication-efficient learning via atomic sparsification. *Advances in neural information processing systems*, 31, 2018.

Jianqiao Wangni, Jialei Wang, Ji Liu, and Tong Zhang. Gradient sparsification for communication-efficient distributed optimization. *Advances in Neural Information Processing Systems*, 31, 2018.

Shengping Xie, Chuyan Chen, and Kun Yuan. From PowerSGD to PowerSGD+: Low-rank gradient compression for distributed optimization with convergence guarantees. *arXiv preprint arXiv:2509.11254*, 2025.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. *arXiv preprint arXiv:2403.03507*, 2024.

A Appendix

A.1 History bases and communication cost

This section explains how history bases support compact core communication. We compare structured payloads, define the basis construction in Algorithm 1, and bound the information retained by the core. We also quantify the additional computation and memory required for communication.

A.1.1 STRUCTURED COMMUNICATION BUDGET

For an $m \times n$ matrix block, the history bases $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times r}$ are stored locally. The core channel therefore transmits only the r^2 coefficients of the core. PowerSGD instead communicates two rank-q factors containing $q(m+n)$ coefficients (Vogels et al., 2019). At the same value precision, comparable structured payloads satisfy

$$r^2 \approx q(m+n), \qquad \frac{r}{q} \approx \frac{m+n}{r}.$$

The core can thus use larger left and right subspaces within the same structured budget. Figure 3 selects a common PowerSGD rank using the total structured payload per layer. The diagnostic counts four 768×768 attention matrices and two MLP matrices of shapes 3072×768 and 768×3072. The core sends $6r^2$ coefficients, while PowerSGD sends $13{,}824q$ coefficients, giving

$$q = \max\{1, \lfloor 6r^2/13{,}824 \rfloor\} = \max\{1, \lfloor r^2/2304 \rfloor\}.$$

Thus, $q = 1$ at $r = 32, 40, 48, 56, 64$, $q = 2$ at $r = 72, 80$, and $q = 3$ at $r = 88, 94$. The plotted staircase changes at integer ranks 68 and 84; markers denote only the nine evaluated ranks. At $r = 48$, both payloads equal 13,824 coefficients per layer. Above this rank, integer rounding can leave part of the core budget unused by PowerSGD. At $r = 32, 40$, the minimum rank-one payload exceeds the cap by factors 2.25 and 1.44, respectively; these points are shown only in the gray reference region. These diagnostic counts exclude residuals and other parameter groups.

The main training comparisons use four workers, with $(r, q, k) = (276, 15, 150)$ for GPT-2 Large and $(r, q, k) = (94, 3, 100)$ for NanoGPT-124M, as specified in Table 7. Main training compresses QKV jointly, giving four hidden matrices per layer: the fused QKV matrix, the attention output matrix, and the two MLP matrices. For hidden width h, their shapes are $3h \times h$, $h \times h$, $4h \times h$, and $h \times 4h$, up to transposition. The structured payloads per layer are therefore $4r^2$ for the core and $16hq$ for PowerSGD; Top-k adds $8k$ scalar entries per layer ($32k$ bytes with 32-bit values and indices). Table 3 gives the payloads for these blocks at the main ranks. The six-matrix grouping above is used only for the reconstruction diagnostic in Figure 3. Whole-model totals also include embeddings and parameters transmitted in full, as specified in Appendix A.3.

Table 3: Structured payload per matrix block and worker, measured in transmitted scalar values. The counts exclude residual messages and multipliers for collective communication. Main training compresses QKV jointly; the $h \times h$ block is the attention output matrix.

Model	Matrix size	r	q	Core: r^2	PowerSGD: $q(m+n)$
GPT-2 Large	1280×1280	276	15	76,176	38,400
GPT-2 Large	3840×1280 (fused)	276	15	76,176	76,800
GPT-2 Large	5120×1280	276	15	76,176	96,000
NanoGPT-124M	768×768	94	3	8,836	4,608
NanoGPT-124M	2304×768 (fused)	94	3	8,836	9,216
NanoGPT-124M	3072×768	94	3	8,836	11,520

A.1.2 CONSTRUCTION FROM EXISTING MUON STATES

Consider one $m \times n$ matrix block and a communication rank $1 \le r \le \min\{m, n\}$. Before training, choose a set $\mathcal{I} \subseteq [m]$ of r row indices uniformly without replacement using a shared seed, and keep it fixed. All workers use the same ordered set. The notation $\widetilde{\boldsymbol{M}}^{t+1}_{\mathcal{I},:}$ denotes the resulting $r \times n$ row submatrix. Here m and n are the dimensions supplied to the basis map, and $\mathcal{I}$ indexes its first dimension; the construction requires no ordering between m and n.

Definition 1 (History basis extraction) *For $\boldsymbol{Y} \in \mathbb{R}^{p\times r}$ with $p \ge r$, let $\mathrm{qf}(\boldsymbol{Y})$ denote the explicit $p \times r$ orthonormal factor from an unpivoted Householder QR factorization. Zero trailing columns use identity reflectors, and column signs are fixed so that the nonzero diagonal entries of the triangular factor are positive. This convention also defines the orthonormal completion for rank-deficient inputs. Set*

$$\boldsymbol{V}^{t+1} = \mathrm{qf}\big((\widetilde{\boldsymbol{M}}^{t+1}_{\mathcal{I},:})^\top\big), \qquad \boldsymbol{U}^{t+1} = \mathrm{qf}\big(\boldsymbol{Q}^{t+1}\boldsymbol{V}^{t+1}\big). \tag{1}$$

The map $\mathrm{Basis}(\widetilde{\boldsymbol{M}}^{t+1}, \boldsymbol{Q}^{t+1}, r)$ *returns* $(\boldsymbol{U}^{t+1}, \boldsymbol{V}^{t+1})$ *using this shared row set and QR convention.*

The momentum rows select the right subspace, and multiplication by $\boldsymbol{Q}^{t+1}$ gives corresponding left coordinates; the two thin QR factorizations make both bases orthonormal. This construction uses direction information already computed for optimization to define the coordinates of the next core. The full Muon direction continues to update the model.

For rank-deficient inputs, thin QR retains the available column space and completes it to r orthonormal columns. The first r columns of the corresponding identity matrices give shared initial bases. Shared momentum, Muon directions, row indices, and QR conventions allow workers to construct identical bases locally. These bases are fixed before the next stochastic gradients are sampled.

A.1.3 Relation to the ideal Muon direction

The construction pairs the left and right directions of the Muon update. The following proposition establishes this relation for the ideal polar map used in Section 4.1.

Proposition 1 (Paired directions under the polar map) *Suppress iteration indices and write* $\boldsymbol{S} = \widetilde{\boldsymbol{M}}^{t+1}$. *Let* $\boldsymbol{S} = \boldsymbol{L}\boldsymbol{\Sigma}\boldsymbol{R}^\top$ *be a compact SVD of rank* ℓ, *and let* $\boldsymbol{Q}_\star = \boldsymbol{L}\boldsymbol{R}^\top$ *be its polar factor. If* $\mathrm{rank}(\boldsymbol{S}_{\mathcal{I},:}) = r$ *(hence* $r \le \ell$*), then* $\boldsymbol{V} = \mathrm{qf}((\boldsymbol{S}_{\mathcal{I},:})^\top)$ *can be written as* $\boldsymbol{V} = \boldsymbol{R}\boldsymbol{W}$ *for some* $\boldsymbol{W} \in \mathbb{R}^{\ell \times r}$ *with* $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_r$. *With* $\boldsymbol{U} = \mathrm{qf}(\boldsymbol{Q}_\star\boldsymbol{V})$, *we have*

$$\boldsymbol{U} = \boldsymbol{L}\boldsymbol{W}, \qquad \boldsymbol{Q}_\star\boldsymbol{V} = \boldsymbol{U}, \qquad \boldsymbol{U}^\top\boldsymbol{Q}_\star\boldsymbol{V} = \boldsymbol{I}_r.$$

Consequently, the two-sided projection of $\boldsymbol{Q}_\star$ *onto these bases is* $\boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{Q}_\star\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{U}\boldsymbol{V}^\top$.

Proof. The columns of $(\boldsymbol{S}_{\mathcal{I},:})^\top$ lie in the column space of $\boldsymbol{R}$. The rank condition and thin QR therefore give $\boldsymbol{V} = \boldsymbol{R}\boldsymbol{W}$. Since $\boldsymbol{R}^\top\boldsymbol{R} = \boldsymbol{I}_\ell$ and $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_r$, it follows that $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_r$. Hence

$$\boldsymbol{Q}_\star\boldsymbol{V} = \boldsymbol{L}\boldsymbol{R}^\top\boldsymbol{R}\boldsymbol{W} = \boldsymbol{L}\boldsymbol{W}, \qquad (\boldsymbol{L}\boldsymbol{W})^\top(\boldsymbol{L}\boldsymbol{W}) = \boldsymbol{I}_r.$$

Thus $\boldsymbol{Q}_\star\boldsymbol{V}$ is already orthonormal. Its thin QR factorization with positive diagonal has triangular factor $\boldsymbol{I}_r$, so $\boldsymbol{U} = \boldsymbol{L}\boldsymbol{W}$. Substitution gives the remaining identities. □

The fixed row set does not guarantee that $\mathrm{rank}(\boldsymbol{S}_{\mathcal{I},:}) = r$, or that those rows remain informative throughout training. Proposition 1 is conditional on the stated rank requirement; orthonormal completion still preserves the contraction argument when it fails.

The proposition shows that the extracted bases are paired combinations of Muon's left and right directions. The SVD is used only to explain this relation; the construction in equation 1 never computes it. The momentum rows select the right coordinates, and Muon's polar map supplies their corresponding left directions. These coordinates need not be the leading singular vectors.

With finitely many NS iterations, the two QR factorizations still produce orthonormal history bases. Lemma 1 therefore applies without exact singular vectors. The memory recursion and aggregation error bounds in Appendix A.2 also use only the stated projection, compression, and gradient properties.

Theorem 1 uses the ideal polar map to analyze descent. Its spectral norm bound controls the change in the iterates, and its inner product with the effective momentum gives the descent alignment in Section 4.1. The quantity R_T measures the true gradient omitted by the history subspaces. Approximate orthogonalization introduces a separate alignment error, defined and analyzed in Appendix A.2.6.

A.1.4 Representation in history bases

We now quantify the information retained by the paired construction. The ideal Muon direction transfers the right history coordinates to the corresponding left directions without weighting them again by the historical singular values. The following result controls the additional loss in this transfer, and connects the basis construction to the omission term R_T in Theorem 1.

We use the notation and rank condition of Proposition 1. Thus $\boldsymbol{S} = \boldsymbol{L}\boldsymbol{\Sigma}\boldsymbol{R}^\top$ has rank ℓ, $\boldsymbol{Q}_\star = \boldsymbol{L}\boldsymbol{R}^\top$, and the extracted bases satisfy $\boldsymbol{V} = \boldsymbol{R}\boldsymbol{W}$ and $\boldsymbol{U} = \boldsymbol{L}\boldsymbol{W}$, with $\boldsymbol{W}^\top\boldsymbol{W} = \boldsymbol{I}_r$. Suppress the iteration

index and write $\mathbf{\Pi}(\boldsymbol{Z}) = \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^\top$. The representation space $\{\boldsymbol{U}\boldsymbol{A}\boldsymbol{V}^\top : \boldsymbol{A} \in \mathbb{R}^{r\times r}\}$ has dimension r^2 and contains matrices of rank up to r. Shared bases allow workers to transmit these core coefficients without exchanging the bases.

Consider a reference matrix $\boldsymbol{Z}_0 = \boldsymbol{L}\boldsymbol{D}\boldsymbol{R}^\top$, where $\boldsymbol{D} \in \mathbb{R}^{\ell\times\ell}$ is symmetric positive definite. It has the same polar factor as the historical momentum, but may have different singular values. The matrix $\boldsymbol{D}$ need not be diagonal: if $\boldsymbol{D} = \boldsymbol{O}\boldsymbol{\Lambda}\boldsymbol{O}^\top$, with $\boldsymbol{O}$ orthogonal and $\boldsymbol{\Lambda}$ positive diagonal, then $\boldsymbol{L}\boldsymbol{O}$ and $\boldsymbol{R}\boldsymbol{O}$ are paired singular vectors of $\boldsymbol{Z}_0$. For a general input, write $\boldsymbol{Z} = \boldsymbol{Z}_0 + \boldsymbol{E}$, where $\boldsymbol{E} \in \mathbb{R}^{m\times n}$ is the difference from the reference matrix.

Proposition 2 (Representation in history bases) *Let $\lambda_- > 0$ and λ_+ bound the eigenvalues of $\boldsymbol{D}$, so $\lambda_-\boldsymbol{I}_\ell \preceq \boldsymbol{D} \preceq \lambda_+\boldsymbol{I}_\ell$, where $\preceq$ denotes the positive semidefinite order. Under the preceding construction, the core satisfies*

$$\lambda_-\boldsymbol{I}_r \preceq \boldsymbol{U}^\top\boldsymbol{Z}_0\boldsymbol{V} = \boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W} \preceq \lambda_+\boldsymbol{I}_r, \qquad \operatorname{rank}(\mathbf{\Pi}(\boldsymbol{Z}_0)) = r. \tag{2}$$

Moreover, its retained energy and reconstruction error obey

$$\|\mathbf{\Pi}(\boldsymbol{Z}_0)\|_F^2 \geq \frac{4\lambda_-\lambda_+}{(\lambda_- + \lambda_+)^2}\|\boldsymbol{Z}_0\boldsymbol{V}\|_F^2, \tag{3}$$

$$\|\boldsymbol{Z}_0 - \mathbf{\Pi}(\boldsymbol{Z}_0)\|_F^2 \leq \|\boldsymbol{Z}_0(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top)\|_F^2 + \left(\frac{\lambda_+ - \lambda_-}{\lambda_+ + \lambda_-}\right)^2 \|\boldsymbol{Z}_0\boldsymbol{V}\|_F^2. \tag{4}$$

Proof. Set $\boldsymbol{A} = \boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W}$ and $\boldsymbol{B} = \boldsymbol{W}^\top\boldsymbol{D}^2\boldsymbol{W}$. Since $\|\boldsymbol{W}\boldsymbol{x}\| = \|\boldsymbol{x}\|$ for every $\boldsymbol{x} \in \mathbb{R}^r$, the spectral bounds on $\boldsymbol{D}$ give

$$\lambda_-\|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^\top\boldsymbol{A}\boldsymbol{x} \leq \lambda_+\|\boldsymbol{x}\|^2.$$

Thus $\boldsymbol{A}$ has r positive eigenvalues in $[\lambda_-, \lambda_+]$. Multiplication by the orthonormal factors $\boldsymbol{U}$ and $\boldsymbol{V}^\top$ preserves its nonzero singular values, proving equation 2. Orthonormality also gives

$$\|\mathbf{\Pi}(\boldsymbol{Z}_0)\|_F^2 = \|\boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W}\|_F^2 = \operatorname{tr}(\boldsymbol{A}^2), \qquad \|\boldsymbol{Z}_0\boldsymbol{V}\|_F^2 = \operatorname{tr}(\boldsymbol{B}).$$

The spectral bounds imply $(\boldsymbol{D} - \lambda_-\boldsymbol{I})(\lambda_+\boldsymbol{I} - \boldsymbol{D}) \succeq 0$, and hence

$$\boldsymbol{B} \preceq (\lambda_- + \lambda_+)\boldsymbol{A} - \lambda_-\lambda_+\boldsymbol{I}_r \preceq \frac{(\lambda_- + \lambda_+)^2}{4\lambda_-\lambda_+}\boldsymbol{A}^2.$$

The second inequality follows from the positive semidefinite square

$$\frac{(\lambda_- + \lambda_+)^2}{4\lambda_-\lambda_+}\boldsymbol{A}^2 - (\lambda_- + \lambda_+)\boldsymbol{A} + \lambda_-\lambda_+\boldsymbol{I}_r = \frac{((\lambda_- + \lambda_+)\boldsymbol{A} - 2\lambda_-\lambda_+\boldsymbol{I}_r)^2}{4\lambda_-\lambda_+} \succeq 0.$$

Taking traces proves equation 3. Finally, Frobenius orthogonality yields

$$\|\boldsymbol{Z}_0 - \mathbf{\Pi}(\boldsymbol{Z}_0)\|_F^2 = \|\boldsymbol{Z}_0\|_F^2 - \|\mathbf{\Pi}(\boldsymbol{Z}_0)\|_F^2, \qquad \|\boldsymbol{Z}_0\|_F^2 = \|\boldsymbol{Z}_0(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top)\|_F^2 + \|\boldsymbol{Z}_0\boldsymbol{V}\|_F^2.$$

Combining these identities with equation 3 proves equation 4. □

The core retains rank r under the paired structure, and equation 3 quantifies how much of the energy captured by the right basis remains after the left projection. If $\lambda_+/\lambda_- \leq 2$, this fraction is at least $8/9$. If $\boldsymbol{D} = c\boldsymbol{I}_\ell$ for $c > 0$, then $\boldsymbol{Z}_0 = c\boldsymbol{Q}_\star$ and the left projection loses none of that captured energy. These statements concern the energy captured by the right basis; the first term of equation 4 accounts for information outside its span.

More generally, the additional left projection error equals $\|(\boldsymbol{I} - \boldsymbol{W}\boldsymbol{W}^\top)\boldsymbol{D}\boldsymbol{W}\|_F^2$. It vanishes whenever $\boldsymbol{D}\boldsymbol{W}$ lies in the span of $\boldsymbol{W}$, even if the spectral interval is wide. Thus, for the reference matrix $\boldsymbol{Z}_0$, the loss from the left projection is governed by $\boldsymbol{D}$ and $\boldsymbol{W}$, while the historical singular values influence the selection of $\boldsymbol{V}$.

Corollary 1 (Deviation from the paired structure) *Under Proposition 2, let $\boldsymbol{Z} = \boldsymbol{Z}_0 + \boldsymbol{E}$ and $\delta = (\lambda_+ - \lambda_-)/(\lambda_+ + \lambda_-)$. Then*

$$\|\boldsymbol{Z} - \mathbf{\Pi}(\boldsymbol{Z})\|_F^2 \leq \|\boldsymbol{Z}(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top)\|_F^2 + \big(\delta\|\boldsymbol{Z}_0\boldsymbol{V}\|_F + \|\boldsymbol{E}\boldsymbol{V}\|_F\big)^2. \tag{5}$$

Proof. For any $\boldsymbol{Z}$, the projection residual splits into components with orthogonal right factors:

$$\boldsymbol{Z} - \boldsymbol{\Pi}(\boldsymbol{Z}) = \boldsymbol{Z}(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top) + (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top)\boldsymbol{Z}\boldsymbol{V}\boldsymbol{V}^\top.$$

Consequently,

$$\|\boldsymbol{Z} - \boldsymbol{\Pi}(\boldsymbol{Z})\|_F^2 = \|\boldsymbol{Z}(\boldsymbol{I} - \boldsymbol{V}\boldsymbol{V}^\top)\|_F^2 + \|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top)\boldsymbol{Z}\boldsymbol{V}\|_F^2.$$

Applying equation 4 to $\boldsymbol{Z}_0$ in this identity bounds its second term by $\delta^2\|\boldsymbol{Z}_0\boldsymbol{V}\|_F^2$. The triangle inequality and the contractivity of an orthogonal projection then give

$$\|(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^\top)\boldsymbol{Z}\boldsymbol{V}\|_F \le \delta\|\boldsymbol{Z}_0\boldsymbol{V}\|_F + \|\boldsymbol{E}\boldsymbol{V}\|_F.$$

Substitution proves the claim. □

Connection to the stationarity bound. For each block b and iteration t, choose a reference $\boldsymbol{Z}_0^{(b),t}$ satisfying Proposition 2 for the history bases available at that iteration, and set $\boldsymbol{E}^{(b),t} = \nabla^{(b)} f(\boldsymbol{X}^t) - \boldsymbol{Z}_0^{(b),t}$. Let $\delta_{b,t}$ be the value of δ in Corollary 1 for this reference. Define the right omission and a bound on the additional left loss by

$$e_{b,t}^{\mathrm{R}} := \|\nabla^{(b)} f(\boldsymbol{X}^t)(\boldsymbol{I} - \boldsymbol{V}^{(b),t}(\boldsymbol{V}^{(b),t})^\top)\|_F,$$

$$e_{b,t}^{\mathrm{L}} := \delta_{b,t}\|\boldsymbol{Z}_0^{(b),t}\boldsymbol{V}^{(b),t}\|_F + \|\boldsymbol{E}^{(b),t}\boldsymbol{V}^{(b),t}\|_F.$$

Applying equation 5 to each true gradient block and following the definition of R_T gives

$$R_T \le \frac{1}{H_T}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\Big[\Big(\sum_{b=1}^{B}\big((e_{b,t}^{\mathrm{R}})^2 + (e_{b,t}^{\mathrm{L}})^2\big)\Big)^{1/2}\Big]. \tag{6}$$

This bound separates information outside the right history subspaces from the additional loss in the paired left projection. The latter depends on the reference matrix's singular-value spread and its difference from the current gradient. These reference conditions are used only for the representation bound. Theorem 1 applies to orthonormal history bases and keeps R_T explicit.

Consequence for the communication budget. The core communicates r^2 coefficients while using r directions on each side. The retained-energy bound below gives a condition under which these additional directions reduce reconstruction error relative to any rank-q approximation with $q < r$.

Corollary 2 (Reconstruction with a larger core rank) *Under Proposition 2, write $\sigma_1(\boldsymbol{Z}_0) \ge \cdots \ge \sigma_\ell(\boldsymbol{Z}_0) > 0$ for the nonzero singular values. For $1 \le q < r$, define $\tau_q(\boldsymbol{Z}_0)^2 := \sum_{j=q+1}^{\ell}\sigma_j(\boldsymbol{Z}_0)^2$, the squared error of the best rank-q approximation. Then*

$$\tau_q(\boldsymbol{Z}_0)^2 - \|\boldsymbol{Z}_0 - \boldsymbol{\Pi}(\boldsymbol{Z}_0)\|_F^2 \ge r\lambda_-^2 - \sum_{j=1}^{q}\sigma_j(\boldsymbol{Z}_0)^2 \ge r\lambda_-^2 - q\lambda_+^2. \tag{7}$$

In particular, $r\lambda_-^2 > q\lambda_+^2$ implies $\|\boldsymbol{Z}_0 - \boldsymbol{\Pi}(\boldsymbol{Z}_0)\|_F < \tau_q(\boldsymbol{Z}_0)$, giving smaller error than every rank-q reconstruction of the same input.

Proof. Equation 2 gives $\|\boldsymbol{\Pi}(\boldsymbol{Z}_0)\|_F^2 \ge r\lambda_-^2$. The nonzero singular values of $\boldsymbol{Z}_0$ are the eigenvalues of $\boldsymbol{D}$ and are at most λ_+. Since $\boldsymbol{\Pi}$ is an orthogonal projector,

$$\tau_q(\boldsymbol{Z}_0)^2 - \|\boldsymbol{Z}_0 - \boldsymbol{\Pi}(\boldsymbol{Z}_0)\|_F^2 = \|\boldsymbol{\Pi}(\boldsymbol{Z}_0)\|_F^2 - \sum_{j=1}^{q}\sigma_j(\boldsymbol{Z}_0)^2.$$

Substitution proves equation 7. For any matrix $\boldsymbol{Y}$ of rank at most q, let $\boldsymbol{P}_Y$ project onto its column space. Then $\|\boldsymbol{Z}_0 - \boldsymbol{Y}\|_F^2 \ge \|(\boldsymbol{I} - \boldsymbol{P}_Y)\boldsymbol{Z}_0\|_F^2$. The captured energy $\mathrm{tr}(\boldsymbol{P}_Y\boldsymbol{Z}_0\boldsymbol{Z}_0^\top)$ is at most $\sum_{j=1}^{q}\sigma_j(\boldsymbol{Z}_0)^2$: in an eigenbasis, the projector diagonal entries lie in $[0,1]$ and sum to at most q. Thus $\|\boldsymbol{Z}_0 - \boldsymbol{Y}\|_F \ge \tau_q(\boldsymbol{Z}_0)$, with equality attained by truncating the SVD. □

For an $m \times n$ matrix, PowerSGD transmits $q(m+n)$ factor coefficients for a rank-q reconstruction (Vogels et al., 2019). Keeping both bases local allows StructMuon to use a larger rank at a comparable structured payload. Corollary 2 identifies a sufficient condition under which the additional directions retain more energy than the best rank-q approximation, and therefore more than a rank-q PowerSGD reconstruction of the same input. This comparison concerns the core and factor channels, with residual messages counted separately; it refers to one rank-q reconstruction rather than a sum of independently formed local factors.

For $\boldsymbol{Z}_0 = c\boldsymbol{Q}_\star$ with $c > 0$, the core retains exactly rc^2 energy and the best rank-q approximation retains qc^2, giving a squared error difference of $(r-q)c^2$. The history bases preserve the selected directions and remain local, allowing a larger rank at a comparable structured payload.

A.1.5 Incremental basis computation

We count one multiplication and one addition as two FLOPs and report leading arithmetic terms for one matrix block and one worker. All counts of additional computation exclude the underlying optimizer step, including Muon's momentum and NS computation. For basis extraction, the two QR inputs have sizes $n \times r$ and $m \times r$. For a $p \times r$ matrix, Householder factorization costs approximately $2pr^2 - \frac{2}{3}r^3$ FLOPs, and forming its explicit thin orthonormal factor costs another $2pr^2 - \frac{2}{3}r^3$. Including the product $\boldsymbol{Q}^{t+1}\boldsymbol{V}^{t+1}$, the additional cost is therefore

$$C_{\text{basis}} \approx 2mnr + 4(m+n)r^2 - \frac{8}{3}r^3 = \mathcal{O}\big(mnr + (m+n)r^2\big). \tag{8}$$

Selecting momentum rows requires copying nr entries and no dense matrix multiplication. Storing the two bases requires $(m+n)r$ values, in addition to temporary workspace. Both $\widetilde{\boldsymbol{M}}^{t+1}$ and $\boldsymbol{Q}^{t+1}$ are reused from the optimizer step; their computation is not repeated.

To compare this additional computation with the Muon update, set $d = \min\{m,n\}$ and $D = \max\{m,n\}$. The NS implementation arranges its matrix as $d \times D$. Each iteration multiplies this matrix by its transpose, squares the resulting $d \times d$ matrix, and applies the polynomial to the original matrix (Jordan et al., 2024). The respective leading costs are $2Dd^2$, $2d^3$, and $2Dd^2$. With s NS iterations,

$$C_{\text{NS}} \approx s(4Dd^2 + 2d^3), \qquad \frac{C_{\text{basis}}}{C_{\text{NS}}} \approx \frac{2mnr + 4(m+n)r^2 - \frac{8}{3}r^3}{s(4Dd^2 + 2d^3)}. \tag{9}$$

Here C_{NS} is the existing cost of Muon's orthogonalization, used to express the relative cost of basis extraction. For square matrices and $s = 5$, the ratio is $\frac{1}{15}(r/d) + \frac{4}{15}(r/d)^2 - \frac{4}{45}(r/d)^3$. At fixed rank, this fraction decreases as the matrix dimension grows.

For hidden width h, separate attention projections have shape $h \times h$, while the two MLP matrices have shapes $4h \times h$ and $h \times 4h$. A fused query, key, and value projection has shape $3h \times h$. Table 4 evaluates these shapes at $h = 768$ and $h = 1280$, corresponding to NanoGPT-124M and GPT-2 Large. The main training ranks are $r = 94$ and $r = 276$, respectively; $r = 256$ is included only as a supplementary arithmetic example. Main training uses fused QKV compression; the separate projection shapes also apply to the reconstruction diagnostic. Transposition leaves the arithmetic counts unchanged.

The listed extraction costs are 1.20%–1.42% of five NS iterations for $h = 768, r = 94$, 2.59%–2.92% for $h = 1280, r = 276$, and 2.33%–2.64% for the supplementary $h = 1280, r = 256$ examples. These fractions concern basis extraction alone. QR and NS can use different precisions and achieve different GPU throughput, so FLOP ratios do not specify runtime overhead.

A.1.6 Computation added for communication

We next include the remaining compression operations and compare methods under the same accounting convention. The counts are per matrix block and worker, excluding forward and backward passes, the underlying optimizer, and network collectives. Core encoding and each dense core reconstruction cost approximately $2mnr + 2dr^2$ FLOPs when matrix products are associated to minimize work. Each microstep performs one encoding, one local reconstruction for the residual and memory, and one global reconstruction. The bases are refreshed once after S microsteps, so the added dense

Table 4: FLOPs for basis extraction per block for the stated model widths and ranks, with $s = 5$. GFLOPs denotes 10^9 floating-point operations. The percentage is $100C_{\text{basis}}/C_{\text{NS}}$; only C_{basis} is additional computation. These analytical examples do not measure runtime.

Model width	Matrix size	r	Added GFLOPs	Fraction of NS
$h = 1280$	1280×1280	276	1.628	2.59%
$h = 1280$	3840×1280	276	4.217	2.87%
$h = 1280$	5120×1280	276	5.512	2.92%
$h = 768$	768×768	94	0.163	1.20%
$h = 768$	2304×768	94	0.439	1.38%
$h = 768$	3072×768	94	0.577	1.42%
$h = 1280$	1280×1280	256	1.465	2.33%
$h = 1280$	5120×1280	256	4.988	2.64%

arithmetic per optimizer update is

$$C_{\text{Struct,dense}} \approx C_{\text{basis}} + S(6mnr + 6dr^2), \qquad S = 4. \tag{10}$$

Residual selection, sparse decoding, and pointwise memory operations are also performed S times per optimizer update. The last are linear in mn; the selection cost depends on the Top-k implementation. They are not included in equation 10.

Table 5: Dense arithmetic added for communication per optimizer update, block, and worker, with S communication rounds and $d = \min\{m, n\}$. The underlying optimizer cost is excluded in every row. Pointwise memory operations and residual selection are described in the text. Ranks must be specified separately when comparing matched communication budgets.

Method	Additional dense arithmetic
Full Muon	0 (no compression)
PowerSGD, rank q	$S[(4 + 2\nu)mnq + \mathcal{O}(mq^2)]$
EF-PowerMuon, rank q	$S[(4 + 2\nu)mnq + \mathcal{O}(mq^2)]$
StructMuon, rank r	$C_{\text{basis}} + S(6mnr + 6dr^2)$
TSR-Adam, rank r	$S(4mnr + 4dr^2 + C_{\text{refresh}}/K)$

PowerSGD uses two products with the input matrix, one factor orthogonalization, and dense factor reconstruction (Vogels et al., 2019). Here ν is the number of reconstructions per communication round: $\nu = 1$ when one reconstruction is shared, and $\nu = 2$ when local error feedback and the global update are reconstructed separately. EF-PowerMuon has the same compression arithmetic for the same implementation; its Muon step is excluded. Standard PowerSGD already supports error feedback, whose pointwise memory operations are not a new cost of matrix factorization.

For TSR-Adam, K is the number of communication rounds between basis refreshes, and C_{refresh} is the cost of one refresh, including thin QR and the reduced SVD (Dang et al., 2026). With sketch width $\ell = r + p$, where $p \geq 0$ counts the extra sketch columns, one power iteration gives a leading refresh cost of $8mn\ell + \mathcal{O}((m + n)\ell^2 + \ell^3)$. The average refresh cost is C_{refresh}/K per round, or SC_{refresh}/K per optimizer update. These counts depend on the rank, number of reconstructions, and refresh interval; StructMuon refreshes its bases once per optimizer update.

Accumulating cores before global reconstruction. Since the bases remain fixed within an optimizer update,

$$\frac{1}{S}\sum_{s=0}^{S-1} \boldsymbol{U}^t \boldsymbol{A}^{t,s} (\boldsymbol{V}^t)^\top = \boldsymbol{U}^t \Big(\frac{1}{S}\sum_{s=0}^{S-1} \boldsymbol{A}^{t,s}\Big)(\boldsymbol{V}^t)^\top.$$

Thus an equivalent implementation can accumulate the aggregated cores and decoded residuals, reconstructing the global dense core only once. Local residual construction and error feedback still occur at every microstep. For this implementation, the additional dense arithmetic is $C_{\text{basis}} + (2S +$

$1)(2mnr + 2dr^2)$, saving $(S-1)(2mnr + 2dr^2)$ relative to Algorithm 1 as written. At $S = 4$, this reduces the core encoding and reconstruction arithmetic by 25%, excluding basis extraction and other operations. Table 5 describes the original per-microstep reconstruction; the runtime results below use the original per-microstep reconstruction.

Optimizer memory. For an $m \times n$ block, StructMuon stores dense local error feedback memory in addition to Muon's dense momentum. At equal precision, these two states require $2mn$ scalar entries, versus mn for the momentum alone in Full Muon. The two bases add $(m+n)r$ entries. The counts exclude model parameters, gradients, activations, and temporary workspace, so they do not imply a doubling of total GPU memory. They apply per worker in Algorithm 1, without splitting momentum or local error feedback memory across workers.

At matched structured payloads, the distinct ranks enter the arithmetic counts separately: products with the input matrix scale with mnr for the core and mnq for PowerSGD. StructMuon uses this local computation to construct and reuse more communication coordinates. Figure 3 selects the PowerSGD rank using the total core budget per layer. Table 2 separately evaluates communication during training at matched validation quality.

A.2 Proofs

A.2.1 Preliminaries

We first present the proof for one communication round per optimizer update ($S = 1$). Appendix A.2.7 gives the extension to fixed S. We work on the product space

$$\mathcal{S} = \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(B)}, \quad \mathcal{S}^{(b)} = \mathbb{R}^{m_b \times n_b}.$$

For $\boldsymbol{X}^t = \big(\boldsymbol{X}^{(1),t}, \ldots, \boldsymbol{X}^{(B),t}\big) \in \mathcal{S}$, the block gradients satisfy $\nabla^{(b)} f(\boldsymbol{X}^t) = N^{-1} \sum_{j=1}^N \nabla^{(b)} f_j(\boldsymbol{X}^t)$. Each block is equipped with the spectral norm $\|\cdot\|_{2\to 2}$ and its dual nuclear norm $\|\cdot\|_*$, as in the spectral norm geometry of Muon (Li & Hong, 2025; Pethick et al., 2025). Communication errors are measured in the Frobenius norm.

At the beginning of iteration t, the iterate $\boldsymbol{X}^t$, the local memories $\{\boldsymbol{H}_j^t\}_{j=1}^N$, and the history bases $\{(\boldsymbol{U}_b^t, \boldsymbol{V}_b^t)\}_{b=1}^B$ are $\mathcal{F}_t$-measurable. The bases define the projection

$$\boldsymbol{\Pi}_b^t(\boldsymbol{Z}) = \boldsymbol{U}_b^t (\boldsymbol{U}_b^t)^\top \boldsymbol{Z}\, \boldsymbol{V}_b^t (\boldsymbol{V}_b^t)^\top.$$

They are orthonormal and fixed before the current stochastic gradients are sampled. Algorithm 1 constructs the next bases through $\mathrm{Basis}(\widetilde{\boldsymbol{M}}, \boldsymbol{Q}, r)$, as specified in Definition 1.

Worker j samples a stochastic gradient and adds its local memory to form the corrected input $\boldsymbol{\Delta}_j^t = \boldsymbol{G}_j^t + \boldsymbol{H}_j^t$, where

$$\boldsymbol{G}_j^t = \big(\boldsymbol{G}_j^{(1),t}, \ldots, \boldsymbol{G}_j^{(B),t}\big), \qquad \boldsymbol{H}_j^t = \big(\boldsymbol{H}_j^{(1),t}, \ldots, \boldsymbol{H}_j^{(B),t}\big),$$
$$\boldsymbol{\Delta}_j^{(b),t} = \boldsymbol{G}_j^{(b),t} + \boldsymbol{H}_j^{(b),t}.$$

The corrected inputs are $\mathcal{G}_t$-measurable. Thus, the residual contraction in Assumption 1 applies conditionally on $\mathcal{G}_t$.

For a residual compressor $\mathcal{C}_{b,t}$ and decoder $\mathcal{D}_{b,t}$, the communication operator is

$$\mathcal{K}_{b,t}(\boldsymbol{Z}) = \boldsymbol{\Pi}_b^t(\boldsymbol{Z}) + \mathcal{D}_{b,t}\big(\mathcal{C}_{b,t}((\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{Z})\big).$$

The local reconstruction combines the core reconstruction with the decoded residual

$$\widehat{\boldsymbol{\Delta}}_j^{(b),t} = \mathcal{K}_{b,t}\big(\boldsymbol{\Delta}_j^{(b),t}\big) = \boldsymbol{\Pi}_b^t\big(\boldsymbol{\Delta}_j^{(b),t}\big) + \mathcal{D}_{b,t}\big(\mathcal{C}_{b,t}((\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{\Delta}_j^{(b),t})\big).$$

Each worker stores the reconstruction error, and the local reconstructions are averaged across workers

$$\boldsymbol{H}_j^{(b),t+1} = \boldsymbol{\Delta}_j^{(b),t} - \widehat{\boldsymbol{\Delta}}_j^{(b),t}, \quad \text{where } \widehat{\boldsymbol{\Delta}}^{(b),t} = \frac{1}{N} \sum_{j=1}^N \widehat{\boldsymbol{\Delta}}_j^{(b),t}.$$

The global reconstruction enters the momentum and parameter updates

$$\begin{aligned}\boldsymbol{M}^{(b),t+1} &= \beta \boldsymbol{M}^{(b),t} + \widehat{\boldsymbol{\Delta}}^{(b),t}, \quad \beta \in [0,1),\\ \widetilde{\boldsymbol{M}}^{(b),t+1} &= \widehat{\boldsymbol{\Delta}}^{(b),t} + \beta \boldsymbol{M}^{(b),t+1},\\ \boldsymbol{X}^{(b),t+1} &= \boldsymbol{X}^{(b),t} - \eta_t \,\mathrm{NSOrth}\big(\widetilde{\boldsymbol{M}}^{(b),t+1}\big),\end{aligned}$$

where $\eta_t > 0$ is shared across blocks.

We measure the memory energy and the projection residuals of the corrected inputs and the true gradient by

$$\mathcal{M}_t = \frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B} \|\boldsymbol{H}_j^{(b),t}\|_F^2, \qquad \mathcal{E}_t = \frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B} \|(\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{\Delta}_j^{(b),t}\|_F^2,$$
$$\mathcal{P}_t = \sum_{b=1}^{B} \big\|(\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\nabla^{(b)} f(\boldsymbol{X}^t)\big\|_F^2.$$

The aggregation and momentum tracking errors are measured in the dual norm

$$\Delta_t^{\mathrm{agg}} = \sum_{b=1}^{B} \big\|\widehat{\boldsymbol{\Delta}}^{(b),t} - \nabla^{(b)} f(\boldsymbol{X}^t)\big\|_*^2, \qquad \Theta_{t+1} = \sum_{b=1}^{B} \big\|(1-\beta)\boldsymbol{M}^{(b),t+1} - \nabla^{(b)} f(\boldsymbol{X}^t)\big\|_*^2.$$

Here $(1-\beta)\boldsymbol{M}^{(b),t+1}$ is the normalized momentum used to track the gradient at $\boldsymbol{X}^t$.

The following projection and norm properties will be used throughout the proofs.

Property 1 *For each block $b \in [B]$ and iteration t, let r_b be the number of columns in each history basis. The bases satisfy*

$$(\boldsymbol{U}_b^t)^\top \boldsymbol{U}_b^t = \boldsymbol{I}_{r_b}, \qquad (\boldsymbol{V}_b^t)^\top \boldsymbol{V}_b^t = \boldsymbol{I}_{r_b}.$$

Consequently, $\boldsymbol{\Pi}_b^t$ is the orthogonal projector onto the subspace represented by these bases, with respect to the Frobenius inner product.

Property 2 *Let $\kappa_b = \sqrt{\min\{m_b, n_b\}}$ and $\kappa_{\max} = \max_{b\in[B]} \kappa_b$. For every $\boldsymbol{Z}^{(b)} \in \mathcal{S}^{(b)}$,*

$$\|\boldsymbol{Z}^{(b)}\|_* \le \kappa_b \|\boldsymbol{Z}^{(b)}\|_F.$$

Property 3 *For a compact singular value decomposition $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, define the polar map $\mathrm{NSOrth}(\boldsymbol{A}) := \boldsymbol{U}\boldsymbol{V}^\top$, with $\mathrm{NSOrth}(\boldsymbol{0}) = \boldsymbol{0}$. Then*

$$\big\|\,\mathrm{NSOrth}(\boldsymbol{A})\big\|_{2\to 2} \le 1 \text{ and } \big\langle \boldsymbol{A}, \mathrm{NSOrth}(\boldsymbol{A})\big\rangle = \|\boldsymbol{A}\|_*.$$

A.2.2 Proof of Lemma 1

Proof. Fix a block $b \in [B]$, an iteration t, and a $\mathcal{G}_t$-measurable matrix $\boldsymbol{Z}^{(b)} \in \mathcal{S}^{(b)}$. Set $\boldsymbol{R}^{(b)} := (\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{Z}^{(b)}$. By the definition of $\mathcal{K}_{b,t}$,

$$\boldsymbol{Z}^{(b)} - \mathcal{K}_{b,t}(\boldsymbol{Z}^{(b)}) = \boldsymbol{R}^{(b)} - \mathcal{D}_{b,t}\big(\mathcal{C}_{b,t}(\boldsymbol{R}^{(b)})\big).$$

Since $\boldsymbol{\Pi}_b^t$ is $\mathcal{F}_t$-measurable and $\mathcal{F}_t \subseteq \mathcal{G}_t$, the residual $\boldsymbol{R}^{(b)}$ is $\mathcal{G}_t$-measurable. Assumption 1 therefore yields

$$\begin{aligned}\mathbb{E}\big[\|\boldsymbol{Z}^{(b)} - \mathcal{K}_{b,t}(\boldsymbol{Z}^{(b)})\|_F^2 \mid \mathcal{G}_t\big] &= \mathbb{E}\big[\|\boldsymbol{R}^{(b)} - \mathcal{D}_{b,t}(\mathcal{C}_{b,t}(\boldsymbol{R}^{(b)}))\|_F^2 \mid \mathcal{G}_t\big]\\ &\le (1-\alpha_b)\|\boldsymbol{R}^{(b)}\|_F^2 = (1-\alpha_b)\|(\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{Z}^{(b)}\|_F^2\\ &\le (1-\alpha_b)\|\boldsymbol{Z}^{(b)}\|_F^2.\end{aligned}$$

The last inequality follows because the orthogonal projection $\boldsymbol{I} - \boldsymbol{\Pi}_b^t$ cannot increase the Frobenius norm (Property 1). □

Exact residual energy for Top-k. For the implemented entrywise compressor without value quantization, let $\mathrm{Top}_{k_b}(\boldsymbol{R}^{(b)})$ retain the k_b largest-magnitude entries of $\boldsymbol{R}^{(b)} = (\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{Z}^{(b)}$ and set the others to zero. The retained and discarded entries have disjoint supports, so

$$\begin{aligned}\|\boldsymbol{Z}^{(b)} - \mathcal{K}_{b,t}(\boldsymbol{Z}^{(b)})\|_F^2 &= \|\boldsymbol{R}^{(b)}\|_F^2 - \|\,\mathrm{Top}_{k_b}(\boldsymbol{R}^{(b)})\|_F^2 \\ &= \|\boldsymbol{Z}^{(b)}\|_F^2 - \|\boldsymbol{\Pi}_b^t(\boldsymbol{Z}^{(b)})\|_F^2 - \|\,\mathrm{Top}_{k_b}(\boldsymbol{R}^{(b)})\|_F^2.\end{aligned}$$

The second equality uses the orthogonality of the core $\boldsymbol{\Pi}_b^t(\boldsymbol{Z}^{(b)})$ and residual $\boldsymbol{R}^{(b)}$. For $\boldsymbol{Z}^{(b)} = \boldsymbol{\Delta}_j^{(b),t}$, the left side is the next local memory energy $\|\boldsymbol{H}_j^{(b),t+1}\|_F^2$. Capturing more energy through either channel reduces this memory. Top-k_b retains at least $k_b\|\boldsymbol{R}^{(b)}\|_F^2/(m_b n_b)$, and can retain more at the same budget when residual energy is concentrated.

A.2.3 Auxiliary lemmas

Throughout the analysis, the stepsize sequence is deterministic. We first bound the local memory in terms of the true gradient omitted by the history subspaces, data heterogeneity, and stochastic noise.

Lemma 2 (Memory recursion) *Suppose that Assumptions 1, 2, and 3 hold. For every iteration t,*

$$\mathbb{E}[\sqrt{\mathcal{M}_{t+1}}] \le \sqrt{1-\alpha}\Big(\mathbb{E}[\sqrt{\mathcal{M}_t}] + \mathbb{E}[\sqrt{\mathcal{P}_t}] + \zeta + \Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2}\Big).$$

Proof. The corrected inputs are $\mathcal{G}_t$-measurable. The local memory update and Lemma 1 therefore give that

$$\begin{aligned}\mathbb{E}[\mathcal{M}_{t+1} \mid \mathcal{G}_t] &= \frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\mathbb{E}\big[\|\boldsymbol{\Delta}_j^{(b),t} - \mathcal{K}_{b,t}(\boldsymbol{\Delta}_j^{(b),t})\|_F^2 \,\big|\, \mathcal{G}_t\big] \\ &\le \frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}(1-\alpha_b)\|(\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{\Delta}_j^{(b),t}\|_F^2 \le (1-\alpha)\mathcal{E}_t.\end{aligned}$$

Conditional Jensen's inequality and the tower property therefore imply

$$\mathbb{E}[\sqrt{\mathcal{M}_{t+1}}] \le \mathbb{E}\big[\sqrt{\mathbb{E}[\mathcal{M}_{t+1} \mid \mathcal{G}_t]}\big] \le \sqrt{1-\alpha}\,\mathbb{E}[\sqrt{\mathcal{E}_t}].$$

To bound $\mathcal{E}_t$, write $\boldsymbol{\xi}_j^{(b),t} = \boldsymbol{G}_j^{(b),t} - \nabla^{(b)} f_j(\boldsymbol{X}^t)$. The corrected input satisfies

$$\begin{aligned}(\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{\Delta}_j^{(b),t} &= (\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{H}_j^{(b),t} + (\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\nabla^{(b)} f(\boldsymbol{X}^t) \\ &\quad + (\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\big(\nabla^{(b)} f_j(\boldsymbol{X}^t) - \nabla^{(b)} f(\boldsymbol{X}^t)\big) + (\boldsymbol{I} - \boldsymbol{\Pi}_b^t)\boldsymbol{\xi}_j^{(b),t}.\end{aligned}$$

By Property 1, each residual projector $\boldsymbol{I} - \boldsymbol{\Pi}_b^t$ is nonexpansive in the Frobenius norm. Applying the triangle inequality over the pairs of workers and blocks and then Assumption 3 yields

$$\begin{aligned}\sqrt{\mathcal{E}_t} &\le \sqrt{\mathcal{M}_t} + \sqrt{\mathcal{P}_t} + \Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\nabla^{(b)} f_j(\boldsymbol{X}^t) - \nabla^{(b)} f(\boldsymbol{X}^t)\|_F^2\Big)^{1/2} \\ &\quad + \Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{\xi}_j^{(b),t}\|_F^2\Big)^{1/2} \\ &\le \sqrt{\mathcal{M}_t} + \sqrt{\mathcal{P}_t} + \zeta + \Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{\xi}_j^{(b),t}\|_F^2\Big)^{1/2}.\end{aligned}$$

Assumption 2, the tower property, and Jensen's inequality give

$$\mathbb{E}\Big[\Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{\xi}_j^{(b),t}\|_F^2\Big)^{1/2}\Big] \le \Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\mathbb{E}[\|\boldsymbol{\xi}_j^{(b),t}\|_F^2]\Big)^{1/2} \le \Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2}.$$

Taking expectations in the bound on $\sqrt{\mathcal{E}_t}$ and combining these estimates proves the recursion. □

Error feedback expresses the aggregation error through two consecutive memory states and the averaged stochastic gradient noise.

Lemma 3 (Aggregation error) *Suppose that Assumption 2 holds. Then, for every iteration t,*

$$\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] \le \kappa_{\max}\Big(\mathbb{E}[\sqrt{\mathcal{M}_{t+1}}] + \mathbb{E}[\sqrt{\mathcal{M}_t}] + \Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2}\Big).$$

Proof. Write $\boldsymbol{\xi}_j^{(b),t} = \boldsymbol{G}_j^{(b),t} - \nabla^{(b)} f_j(\boldsymbol{X}^t)$. Rearranging the local memory update gives

$$\widehat{\boldsymbol{\Delta}}_j^{(b),t} = \boldsymbol{G}_j^{(b),t} + \boldsymbol{H}_j^{(b),t} - \boldsymbol{H}_j^{(b),t+1}.$$

Averaging over the workers and using $N^{-1}\sum_{j=1}^{N}\nabla^{(b)} f_j(\boldsymbol{X}^t) = \nabla^{(b)} f(\boldsymbol{X}^t)$, we obtain

$$\widehat{\boldsymbol{\Delta}}^{(b),t} - \nabla^{(b)} f(\boldsymbol{X}^t) = \frac{1}{N}\sum_{j=1}^{N}\big(\boldsymbol{H}_j^{(b),t} - \boldsymbol{H}_j^{(b),t+1} + \boldsymbol{\xi}_j^{(b),t}\big).$$

Property 2, convexity of the squared Frobenius norm, and the triangle inequality give

$$\begin{aligned}\sqrt{\Delta_t^{\mathrm{agg}}} &\le \kappa_{\max}\Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{H}_j^{(b),t} - \boldsymbol{H}_j^{(b),t+1} + \boldsymbol{\xi}_j^{(b),t}\|_F^2\Big)^{1/2}\\ &\le \kappa_{\max}\Big(\sqrt{\mathcal{M}_t} + \sqrt{\mathcal{M}_{t+1}} + \Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{\xi}_j^{(b),t}\|_F^2\Big)^{1/2}\Big).\end{aligned}$$

By Jensen's inequality and the variance bound in Assumption 2, we have

$$\mathbb{E}\Big[\Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\|\boldsymbol{\xi}_j^{(b),t}\|_F^2\Big)^{1/2}\Big] \le \Big(\frac{1}{N}\sum_{j=1}^{N}\sum_{b=1}^{B}\mathbb{E}[\|\boldsymbol{\xi}_j^{(b),t}\|_F^2]\Big)^{1/2} \le \Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2}.$$

Taking expectations in the preceding bound on $\sqrt{\Delta_t^{\mathrm{agg}}}$ proves the result. □

Independent worker sampling. If the gradient noises $\boldsymbol{\xi}_j^{(b),t} = \boldsymbol{G}_j^{(b),t} - \nabla^{(b)} f_j(\boldsymbol{X}^t)$ are additionally independent across workers conditional on $\mathcal{F}_t$, their conditional means vanish and

$$\mathbb{E}\Big[\Big\|\frac{1}{N}\sum_{j=1}^{N}\boldsymbol{\xi}_j^{(b),t}\Big\|_F^2 \;\Big|\; \mathcal{F}_t\Big] = \frac{1}{N^2}\sum_{j=1}^{N}\mathbb{E}[\|\boldsymbol{\xi}_j^{(b),t}\|_F^2 \mid \mathcal{F}_t] \le \frac{\sigma_b^2}{N}.$$

Separating the averaged noise from the two memory terms before applying the triangle inequality therefore sharpens Lemma 3 to

$$\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] \le \kappa_{\max}\Big(\mathbb{E}[\sqrt{\mathcal{M}_t}] + \mathbb{E}[\sqrt{\mathcal{M}_{t+1}}] + \Big(\frac{1}{N}\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2}\Big).$$

This refinement applies to the direct aggregation noise. The local memory recursion still contains noise from each worker, so it does not divide the entire stochastic term of Theorem 1 by $\sqrt{N}$.

The momentum tracking error depends on the aggregation error and the change in the true gradient between consecutive iterations.

Lemma 4 (Momentum tracking) *Suppose that Assumption 4 holds. Then, for every $t \ge 1$,*

$$\mathbb{E}[\sqrt{\Theta_{t+1}}] \le \beta\mathbb{E}[\sqrt{\Theta_t}] + (1-\beta)\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \beta L_{\max}\sqrt{B}\,\eta_{t-1}.$$

Proof. It follows from the momentum update that

$$\begin{aligned}&(1-\beta)\boldsymbol{M}^{(b),t+1}-\nabla^{(b)}f(\boldsymbol{X}^t)\\&=\beta\big((1-\beta)\boldsymbol{M}^{(b),t}-\nabla^{(b)}f(\boldsymbol{X}^{t-1})\big)+(1-\beta)\big(\widehat{\boldsymbol{\Delta}}^{(b),t}-\nabla^{(b)}f(\boldsymbol{X}^t)\big)\\&\quad+\beta\big(\nabla^{(b)}f(\boldsymbol{X}^{t-1})-\nabla^{(b)}f(\boldsymbol{X}^t)\big).\end{aligned}$$

Applying the triangle inequality across blocks gives

$$\sqrt{\Theta_{t+1}}\le\beta\sqrt{\Theta_t}+(1-\beta)\sqrt{\Delta_t^{\mathrm{agg}}}+\beta\Big(\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)-\nabla^{(b)}f(\boldsymbol{X}^{t-1})\|_*^2\Big)^{1/2}.$$

From Assumption 4, the parameter update, and Property 3, we bound the gradient difference by

$$\begin{aligned}\Big(\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)-\nabla^{(b)}f(\boldsymbol{X}^{t-1})\|_*^2\Big)^{1/2}&\le L_{\max}\Big(\sum_{b=1}^{B}\|\boldsymbol{X}^{(b),t}-\boldsymbol{X}^{(b),t-1}\|_{2\to2}^2\Big)^{1/2}\\&=L_{\max}\eta_{t-1}\Big(\sum_{b=1}^{B}\|\,\mathrm{NSOrth}(\widetilde{\boldsymbol{M}}^{(b),t})\|_{2\to2}^2\Big)^{1/2}\\&\le L_{\max}\sqrt{B}\,\eta_{t-1}.\end{aligned}$$

Substituting this estimate and taking expectations proves the claim. □

The following descent bound relates the objective change to the aggregation and momentum tracking errors.

Lemma 5 (One-step descent) *Suppose that Assumption 4 holds. Then, for every iteration t,*

$$f(\boldsymbol{X}^{t+1})\le f(\boldsymbol{X}^t)-\eta_t\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*+2\sqrt{B}\eta_t\big(\beta\sqrt{\Theta_{t+1}}+(1-\beta)\sqrt{\Delta_t^{\mathrm{agg}}}\big)+\frac{\eta_t^2}{2}\sum_{b=1}^{B}L_b.$$

Proof. For each block, write $\boldsymbol{Q}^{(b),t+1}=\mathrm{NSOrth}(\widetilde{\boldsymbol{M}}^{(b),t+1})$. It follows from Assumption 4, the parameter update, and the bound $\|\boldsymbol{Q}^{(b),t+1}\|_{2\to2}\le1$ that

$$f(\boldsymbol{X}^{t+1})\le f(\boldsymbol{X}^t)-\eta_t\sum_{b=1}^{B}\langle\nabla^{(b)}f(\boldsymbol{X}^t),\boldsymbol{Q}^{(b),t+1}\rangle+\frac{\eta_t^2}{2}\sum_{b=1}^{B}L_b.$$

Set $\boldsymbol{E}^{(b),t}=(1-\beta)\widetilde{\boldsymbol{M}}^{(b),t+1}-\nabla^{(b)}f(\boldsymbol{X}^t)$. Since $1-\beta>0$, from Property 3, duality between the spectral and nuclear norms, and the reverse triangle inequality, we have

$$\begin{aligned}\langle\nabla^{(b)}f(\boldsymbol{X}^t),\boldsymbol{Q}^{(b),t+1}\rangle&=\|(1-\beta)\widetilde{\boldsymbol{M}}^{(b),t+1}\|_*-\langle\boldsymbol{E}^{(b),t},\boldsymbol{Q}^{(b),t+1}\rangle\\&\ge\|(1-\beta)\widetilde{\boldsymbol{M}}^{(b),t+1}\|_*-\|\boldsymbol{E}^{(b),t}\|_*\ge\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*-2\|\boldsymbol{E}^{(b),t}\|_*.\end{aligned}$$

The definition of the effective momentum also gives

$$\boldsymbol{E}^{(b),t}=\beta\big((1-\beta)\boldsymbol{M}^{(b),t+1}-\nabla^{(b)}f(\boldsymbol{X}^t)\big)+(1-\beta)\big(\widehat{\boldsymbol{\Delta}}^{(b),t}-\nabla^{(b)}f(\boldsymbol{X}^t)\big).$$

Consequently, Cauchy–Schwarz and the triangle inequality give

$$\sum_{b=1}^{B}\|\boldsymbol{E}^{(b),t}\|_*\le\sqrt{B}\Big(\sum_{b=1}^{B}\|\boldsymbol{E}^{(b),t}\|_*^2\Big)^{1/2}\le\sqrt{B}\big(\beta\sqrt{\Theta_{t+1}}+(1-\beta)\sqrt{\Delta_t^{\mathrm{agg}}}\big).$$

Substituting these estimates into the smoothness bound proves the result. □

A.2.4 Proof of Theorem 1

Proof. Fix $T\ge1$. We first bound the accumulated local memory. From Lemma 2, we have

$$\mathbb{E}[\sqrt{\mathcal{M}_{t+1}}]\le\sqrt{1-\alpha}\Big(\mathbb{E}[\sqrt{\mathcal{M}_t}]+\mathbb{E}[\sqrt{\mathcal{P}_t}]+\zeta+\Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2}\Big).\tag{11}$$

Weighting this recursion by η_{t+1} and using $\eta_{t+1} \le \eta_t$, we obtain

$$\sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\mathcal{M}_t}] = \eta_0 \mathbb{E}[\sqrt{\mathcal{M}_0}] + \sum_{t=0}^{T-2} \eta_{t+1} \mathbb{E}[\sqrt{\mathcal{M}_{t+1}}]$$
$$\le \eta_0 \mathbb{E}[\sqrt{\mathcal{M}_0}] + \sqrt{1-\alpha} \sum_{t=0}^{T-2} \eta_{t+1} \Big(\mathbb{E}[\sqrt{\mathcal{M}_t}] + \mathbb{E}[\sqrt{\mathcal{P}_t}] + \zeta + \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2} \Big)$$
$$\le \eta_0 \mathbb{E}[\sqrt{\mathcal{M}_0}] + \sqrt{1-\alpha} \sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\mathcal{M}_t}] + \sqrt{1-\alpha}\, H_T \Big(R_T + \zeta + \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2} \Big),$$

where the last inequality uses the definitions of H_T and R_T. Since $\sqrt{1-\alpha} < 1$, rearranging yields

$$\sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\mathcal{M}_t}] \le \frac{\eta_0 \mathbb{E}[\sqrt{\mathcal{M}_0}]}{1-\sqrt{1-\alpha}} + \frac{\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}} H_T \Big(R_T + \zeta + \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2} \Big). \tag{12}$$

We next use this estimate to control the aggregation error. Lemma 3 and (11) imply

$$\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] \le \kappa_{\max} \Big(\mathbb{E}[\sqrt{\mathcal{M}_{t+1}}] + \mathbb{E}[\sqrt{\mathcal{M}_t}] + \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2} \Big)$$
$$\le \kappa_{\max} \Big[(1+\sqrt{1-\alpha}) \mathbb{E}[\sqrt{\mathcal{M}_t}] + \sqrt{1-\alpha}\, \mathbb{E}[\sqrt{\mathcal{P}_t}]$$
$$+ \sqrt{1-\alpha}\, \zeta + (1+\sqrt{1-\alpha}) \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2} \Big].$$

Multiplying by η_t and summing over $t = 0, \dots, T-1$ gives

$$\sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] \le \kappa_{\max}(1+\sqrt{1-\alpha}) \sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\mathcal{M}_t}]$$
$$+ \kappa_{\max} \sqrt{1-\alpha}\, H_T (R_T + \zeta) + \kappa_{\max}(1+\sqrt{1-\alpha}) H_T \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2}.$$

Substituting (12) and collecting terms, we conclude that

$$\sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] \le \frac{\kappa_{\max}(1+\sqrt{1-\alpha})}{1-\sqrt{1-\alpha}} \eta_0 \mathbb{E}[\sqrt{\mathcal{M}_0}]$$
$$+ \frac{2\kappa_{\max}\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}} H_T (R_T + \zeta) + \frac{\kappa_{\max}(1+\sqrt{1-\alpha})}{1-\sqrt{1-\alpha}} H_T \Big(\sum_{b=1}^{B} \sigma_b^2 \Big)^{1/2}. \tag{13}$$

For the momentum tracking error, Lemma 4 gives

$$\sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\Theta_{t+1}}] = \eta_0 \mathbb{E}[\sqrt{\Theta_1}] + \sum_{t=1}^{T-1} \eta_t \mathbb{E}[\sqrt{\Theta_{t+1}}]$$
$$\le \eta_0 \mathbb{E}[\sqrt{\Theta_1}] + \beta \sum_{t=1}^{T-1} \eta_t \mathbb{E}[\sqrt{\Theta_t}] + (1-\beta) \sum_{t=1}^{T-1} \eta_t \mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \beta L_{\max} \sqrt{B} \sum_{t=1}^{T-1} \eta_t \eta_{t-1}.$$

The nonincreasing stepsizes imply

$$\sum_{t=1}^{T-1} \eta_t \mathbb{E}[\sqrt{\Theta_t}] = \sum_{t=0}^{T-2} \eta_{t+1} \mathbb{E}[\sqrt{\Theta_{t+1}}] \le \sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\Theta_{t+1}}],$$
$$\sum_{t=1}^{T-1} \eta_t \eta_{t-1} \le \sum_{t=0}^{T-2} \eta_t^2 \le H_T Q_T.$$

Applying these bounds and extending the nonnegative aggregation error sum to include $t = 0$, we obtain

$$(1-\beta)\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Theta_{t+1}}] \le \eta_0\mathbb{E}[\sqrt{\Theta_1}] + (1-\beta)\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \beta L_{\max}\sqrt{B}\,H_T Q_T.$$

Dividing by $1-\beta > 0$ yields

$$\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Theta_{t+1}}] \le \frac{\eta_0\mathbb{E}[\sqrt{\Theta_1}]}{1-\beta} + \sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \frac{\beta L_{\max}\sqrt{B}}{1-\beta}H_T Q_T. \tag{14}$$

It remains to combine these estimates with the descent inequality. Taking expectations in Lemma 5 and summing over $t = 0, \dots, T-1$, the objective differences telescope to give

$$\begin{aligned}\sum_{t=0}^{T-1}\eta_t\mathbb{E}\Big[\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*\Big] &\le f(\boldsymbol{X}^0) - \mathbb{E}[f(\boldsymbol{X}^T)] + 2\sqrt{B}\,\beta\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Theta_{t+1}}] \\ &\quad + 2\sqrt{B}\,(1-\beta)\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \frac{1}{2}\Big(\sum_{b=1}^{B}L_b\Big)\sum_{t=0}^{T-1}\eta_t^2.\end{aligned}$$

Substituting (14) combines the two aggregation error terms into a single sum. Using $f(\boldsymbol{X}^T) \ge f^\star$ and $\sum_{t=0}^{T-1}\eta_t^2 = H_T Q_T$, we obtain

$$\begin{aligned}\sum_{t=0}^{T-1}\eta_t\mathbb{E}\Big[\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*\Big] &\le f(\boldsymbol{X}^0) - f^\star + \frac{2\sqrt{B}\,\beta}{1-\beta}\eta_0\mathbb{E}[\sqrt{\Theta_1}] + 2\sqrt{B}\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] \\ &\quad + \left(\frac{2B\beta^2 L_{\max}}{1-\beta} + \frac{1}{2}\sum_{b=1}^{B}L_b\right)H_T Q_T.\end{aligned} \tag{15}$$

Finally, inserting (13) into (15) gives

$$\begin{aligned}\sum_{t=0}^{T-1}\eta_t\mathbb{E}\Big[\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*\Big] &\le f(\boldsymbol{X}^0) - f^\star + \frac{2\sqrt{B}\,\kappa_{\max}(1+\sqrt{1-\alpha})}{1-\sqrt{1-\alpha}}\eta_0\mathbb{E}[\sqrt{\mathcal{M}_0}] \\ &\quad + \frac{2\sqrt{B}\,\beta}{1-\beta}\eta_0\mathbb{E}[\sqrt{\Theta_1}] + \frac{4\sqrt{B}\,\kappa_{\max}\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}H_T(R_T+\zeta) \\ &\quad + \frac{2\sqrt{B}\,\kappa_{\max}(1+\sqrt{1-\alpha})}{1-\sqrt{1-\alpha}}H_T\Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2} \\ &\quad + \left(\frac{2B\beta^2 L_{\max}}{1-\beta} + \frac{1}{2}\sum_{b=1}^{B}L_b\right)H_T Q_T.\end{aligned} \tag{16}$$

All displayed coefficients are finite and depend only on α, β, $\kappa_{\max}$, B, and $\{L_b\}_{b=1}^B$. Dividing equation 16 by H_T and choosing C to dominate the coefficients of the objective gap, initialization, stochastic noise, and Q_T, while retaining the explicit coefficient of $R_T + \zeta$, yields the bound in Theorem 1. □

A.2.5 HORIZON-DEPENDENT MOMENTUM

Theorem 1 holds for any fixed $\beta \in [0, 1)$ and bounds the momentum tracking error recursively, which leaves stochastic noise, heterogeneity, and R_T at order one. We now unroll the momentum recursion instead. Error feedback makes the memory contribution telescope inside the geometric momentum average, and the noise forms a martingale sum. A horizon-dependent momentum therefore makes all three terms vanish.

For block tuples $\boldsymbol{Z} \in \mathcal{S}$, write $\boldsymbol{\Phi}(\boldsymbol{Z}) := \sum_{b=1}^{B}\|\boldsymbol{Z}^{(b)}\|_*$ and $\|\boldsymbol{Z}\|_{\mathcal{S}} := (\sum_{b=1}^{B}\|\boldsymbol{Z}^{(b)}\|_F^2)^{1/2}$. Set $\sigma := (\sum_{b=1}^{B}\sigma_b^2)^{1/2}$, $c := \sqrt{1-\alpha}$, and $\gamma_\alpha := 1 - c$. Let $\bar{\boldsymbol{H}}^t := N^{-1}\sum_{j=1}^{N}\boldsymbol{H}_j^t$ and

$\bar{\boldsymbol{\xi}}^t := N^{-1}\sum_{j=1}^N \boldsymbol{\xi}_j^t$ with $\boldsymbol{\xi}_j^t = \boldsymbol{G}_j^t - \nabla f_j(\boldsymbol{X}^t)$. Property 2 and the Cauchy–Schwarz inequality give $\boldsymbol{\Phi}(\boldsymbol{Z}) \le \sqrt{B}\,\kappa_{\max}\|\boldsymbol{Z}\|_{\mathcal{S}}$. Jensen's inequality gives $\|\bar{\boldsymbol{H}}^t\|_{\mathcal{S}} \le \sqrt{\mathcal{M}_t}$, and together with Assumption 2 it gives $\mathbb{E}\|\bar{\boldsymbol{\xi}}^t\|_{\mathcal{S}}^2 \le \sigma^2$. As in Algorithm 1, $\boldsymbol{M}^0 = \boldsymbol{0}$, and $\boldsymbol{X}^0$ is deterministic.

Theorem 2 (Convergence with horizon-dependent momentum) *Suppose the assumptions of Theorem 1 hold. For every $T \ge 1$ and $\beta \in [0,1)$,*

$$\frac{1}{H_T}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}\big[\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^t))\big] \le \frac{f(\boldsymbol{X}^0)-f^\star}{H_T} + \frac{2\eta_0\,\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^0))}{(1-\beta)H_T} + \Big(\frac{1}{2}\sum_{b=1}^B L_b + \frac{2BL_{\max}}{1-\beta}\Big)Q_T$$
$$+ \frac{10\sqrt{B}\,\kappa_{\max}\,\eta_0\,\mathbb{E}[\sqrt{\mathcal{M}_0}]}{\gamma_\alpha H_T} + 4\sqrt{B}\,\kappa_{\max}\sqrt{1-\beta}\,\sigma$$
$$+ \frac{8\sqrt{B}\,\kappa_{\max}\sqrt{1-\alpha}}{\gamma_\alpha}(1-\beta)\big(R_T+\zeta+\sigma\big).$$

Compared with Theorem 1, the memory, heterogeneity, and omission terms carry the factor $1-\beta$, since error feedback telescopes inside the momentum average. Their coefficient retains $\sqrt{1-\alpha}$, so they vanish when the residual compressor is exact. The initialization enters only through $f(\boldsymbol{X}^0) - f^\star$, $\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^0))$, and $\mathcal{M}_0$. Unlike the tracking error Θ_1, none of these depends on β, which varies with T below.

Corollary 3 (Rate without non-vanishing terms) *Take $\eta_t = T^{-3/4}$ and $1-\beta = T^{-1/2}$. Suppose the problem parameters and the initial quantities $f(\boldsymbol{X}^0) - f^\star$, $\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^0))$, and $\mathbb{E}[\sqrt{\mathcal{M}_0}]$ are fixed. Then*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\big[\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^t))\big] \le \mathcal{O}\big(T^{-1/4}\big)\big(1+\sigma\big) + \mathcal{O}\big(T^{-1/2}\big)\big(R_T+\zeta\big).$$

Since R_T is at most the left side, it is absorbed once $T \ge (16\sqrt{B}\,\kappa_{\max}\sqrt{1-\alpha}/\gamma_\alpha)^2$. This leaves $\mathcal{O}(T^{-1/4})(1+\sigma) + \mathcal{O}(T^{-1/2})\zeta$.

The rates follow from $H_T = T^{1/4}$ and $Q_T = T^{-3/4}$. For the absorption, constant stepsizes give R_T and the left side the same weights, and $\sqrt{\mathcal{P}_t} \le \boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^t))$ since the residual projectors are nonexpansive and $\|\cdot\|_F \le \|\cdot\|_*$. The schedule $1-\beta = \Theta(T^{-1/2})$ is standard in analyses of Muon-type methods; our experiments use $\beta = 0.95$. With independent worker noise, σ in the $\sqrt{1-\beta}$ term becomes $\sigma/\sqrt{N}$. Appendix A.2.7 extends the result to S microsteps.

Proof of Theorem 2. Error decomposition. Define, blockwise,

$$\boldsymbol{\Omega}^t := (1-\beta)\widetilde{\boldsymbol{M}}^{t+1} - \nabla f(\boldsymbol{X}^t), \qquad \boldsymbol{\omega}^{t+1} := (1-\beta)\boldsymbol{M}^{t+1} - \nabla f(\boldsymbol{X}^t).$$

Thus $\boldsymbol{\Omega}^t$ collects the blocks $\boldsymbol{E}^{(b),t}$ in the proof of Lemma 5. That proof, before the Cauchy–Schwarz step, gives

$$\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}[\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^t))] \le f(\boldsymbol{X}^0) - f^\star + 2\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}[\boldsymbol{\Phi}(\boldsymbol{\Omega}^t)] + \frac{1}{2}\Big(\sum_{b=1}^B L_b\Big)H_T Q_T. \tag{17}$$

The proof of Lemma 3 shows $\widehat{\boldsymbol{\Delta}}^t - \nabla f(\boldsymbol{X}^t) = \boldsymbol{\delta}^t := \bar{\boldsymbol{\xi}}^t + \bar{\boldsymbol{H}}^t - \bar{\boldsymbol{H}}^{t+1}$. The definition of the effective momentum therefore gives $\boldsymbol{\Omega}^t = \beta\boldsymbol{\omega}^{t+1} + (1-\beta)\boldsymbol{\delta}^t$.

Set $\boldsymbol{\omega}^0 := -\nabla f(\boldsymbol{X}^0)$, $\mathbf{D}^0 := \boldsymbol{0}$, and $\mathbf{D}^s := \nabla f(\boldsymbol{X}^{s-1}) - \nabla f(\boldsymbol{X}^s)$ for $s \ge 1$. Since $\boldsymbol{M}^0 = \boldsymbol{0}$, we have $\boldsymbol{\omega}^1 = (1-\beta)\widehat{\boldsymbol{\Delta}}^0 - \nabla f(\boldsymbol{X}^0) = \beta\boldsymbol{\omega}^0 + (1-\beta)\boldsymbol{\delta}^0$. Hence the momentum update gives

$$\boldsymbol{\omega}^{t+1} = \beta\boldsymbol{\omega}^t + (1-\beta)\boldsymbol{\delta}^t + \beta\mathbf{D}^t \qquad \text{for every } t \ge 0.$$

Unrolling this recursion yields

$$\boldsymbol{\omega}^{t+1} = -\beta^{t+1}\nabla f(\boldsymbol{X}^0) + \underbrace{(1-\beta)\sum_{s=0}^{t}\beta^{t-s}\bar{\boldsymbol{\xi}}^s}_{=:\boldsymbol{\Xi}^t} + \underbrace{(1-\beta)\sum_{s=0}^{t}\beta^{t-s}\big(\bar{\boldsymbol{H}}^s - \bar{\boldsymbol{H}}^{s+1}\big)}_{=:\boldsymbol{\Psi}^t}$$
$$+ \sum_{s=1}^{t}\beta^{t-s+1}\mathbf{D}^s.$$

Noise. For $s < s'$, $\bar{\boldsymbol{\xi}}^s$ is $\mathcal{F}_{s'}$-measurable and $\mathbb{E}[\bar{\boldsymbol{\xi}}^{s'} \mid \mathcal{F}_{s'}] = \mathbf{0}$, so the cross terms vanish. Hence

$$\mathbb{E}\|\boldsymbol{\Xi}^t\|_{\mathcal{S}}^2 = (1-\beta)^2 \sum_{s=0}^{t} \beta^{2(t-s)} \mathbb{E}\|\bar{\boldsymbol{\xi}}^s\|_{\mathcal{S}}^2 \le \frac{1-\beta}{1+\beta}\sigma^2,$$

and therefore $\mathbb{E}[\boldsymbol{\Phi}(\boldsymbol{\Xi}^t)] \le \sqrt{B}\,\kappa_{\max}\sqrt{1-\beta}\,\sigma$.

Memory. Summation by parts gives

$$\boldsymbol{\Psi}^t = (1-\beta)\Big[(1-\beta)\sum_{s=0}^{t} \beta^{t-s}\bar{\boldsymbol{H}}^s + \beta^{t+1}\bar{\boldsymbol{H}}^0 - \bar{\boldsymbol{H}}^{t+1}\Big],$$

so the memory enters only with a factor $1-\beta$. Since the stepsizes are nonincreasing, $\sum_{t=s}^{T-1} \eta_t(1-\beta)\beta^{t-s} \le \eta_s$ and $\sum_{t\ge 0} \eta_t\beta^{t+1} \le \eta_0\beta/(1-\beta)$. With $S_0 := \sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\mathcal{M}_t}]$ and $S_1 := \sum_{t=0}^{T-1} \eta_t \mathbb{E}[\sqrt{\mathcal{M}_{t+1}}]$, it follows that

$$\sum_{t=0}^{T-1} \eta_t\, \mathbb{E}\|\boldsymbol{\Psi}^t\|_{\mathcal{S}} \le (1-\beta)(S_0+S_1) + \beta\eta_0 \mathbb{E}[\sqrt{\mathcal{M}_0}]. \tag{18}$$

Let $A := \gamma_\alpha^{-1}\big(\eta_0\mathbb{E}[\sqrt{\mathcal{M}_0}] + c\,H_T(R_T+\zeta+\sigma)\big)$. Equation equation 12 gives $S_0 \le A$. Weighting equation 11 by η_t and then applying equation 12 gives

$$S_1 \le c\,S_0 + c\,H_T(R_T+\zeta+\sigma) \le \frac{c\,\eta_0\mathbb{E}[\sqrt{\mathcal{M}_0}]}{\gamma_\alpha} + \Big(\frac{c^2}{\gamma_\alpha} + c\Big)H_T(R_T+\zeta+\sigma) \le A,$$

because $c^2/\gamma_\alpha + c = c/\gamma_\alpha$ and $c \le 1$.

Drift. The proof of Lemma 4 gives $\boldsymbol{\Phi}(\mathbf{D}^s) \le BL_{\max}\eta_{s-1}$ for $s \ge 1$. Since the stepsizes are nonincreasing,

$$\sum_{t=1}^{T-1} \eta_t \sum_{s=1}^{t} \beta^{t-s+1}\eta_{s-1} \le \sum_{s=1}^{T-1} \eta_{s-1}^2 \sum_{t\ge s} \beta^{t-s+1} \le \frac{\beta}{1-\beta} H_T Q_T.$$

Combination. For every $t \ge 0$, the triangle inequality and $\boldsymbol{\Phi} \le \sqrt{B}\,\kappa_{\max}\|\cdot\|_{\mathcal{S}}$ give

$$\begin{aligned}\boldsymbol{\Phi}(\boldsymbol{\Omega}^t) \le \beta^{t+2}\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^0)) + \beta\,\boldsymbol{\Phi}(\boldsymbol{\Xi}^t) + \beta\sqrt{B}\,\kappa_{\max}\|\boldsymbol{\Psi}^t\|_{\mathcal{S}} + \sum_{s=1}^{t}\beta^{t-s+2}\boldsymbol{\Phi}(\mathbf{D}^s)\\ + (1-\beta)\sqrt{B}\,\kappa_{\max}\big(\|\bar{\boldsymbol{\xi}}^t\|_{\mathcal{S}} + \sqrt{\mathcal{M}_t} + \sqrt{\mathcal{M}_{t+1}}\big).\end{aligned}$$

We weight by η_t, take expectations, sum over t, and insert the four estimates above. This uses $\sum_{t\ge 0}\eta_t\beta^{t+2} \le \eta_0/(1-\beta)$, $\beta\sqrt{1-\beta}+(1-\beta) \le 2\sqrt{1-\beta}$, $(1+\beta)(1-\beta)(S_0+S_1) \le 4(1-\beta)A$, and $\beta^2 \le 1$. The result is

$$\begin{aligned}\sum_{t=0}^{T-1}\eta_t\,\mathbb{E}[\boldsymbol{\Phi}(\boldsymbol{\Omega}^t)] \le \frac{\eta_0\,\boldsymbol{\Phi}(\nabla f(\boldsymbol{X}^0))}{1-\beta} + \frac{BL_{\max}}{1-\beta}H_TQ_T + 2\sqrt{B}\,\kappa_{\max}\sqrt{1-\beta}\,\sigma H_T\\ + \sqrt{B}\,\kappa_{\max}\eta_0\mathbb{E}[\sqrt{\mathcal{M}_0}] + 4\sqrt{B}\,\kappa_{\max}(1-\beta)A.\end{aligned}$$

Substitute A and bound $1 + 4(1-\beta)/\gamma_\alpha \le 5/\gamma_\alpha$ in the coefficient of $\eta_0\mathbb{E}[\sqrt{\mathcal{M}_0}]$. Inserting the result into equation 17 and dividing by H_T proves the theorem. □

A.2.6 Approximate orthogonalization

We extend the analysis to a computed direction $\widehat{\boldsymbol{Q}}^{(b),t+1}$, with $\boldsymbol{X}^{(b),t+1} = \boldsymbol{X}^{(b),t} - \eta_t\widehat{\boldsymbol{Q}}^{(b),t+1}$ and the same communication and momentum recursions. All quantities below, including R_T, are evaluated along this trajectory. Suppose a deterministic constant $\gamma \ge 1$ satisfies, almost surely,

$$\|\widehat{\boldsymbol{Q}}^{(b),t+1}\|_{2\to 2} \le \gamma, \qquad b \in [B], \quad t \ge 0. \tag{19}$$

Define the alignment error by

$$\varepsilon_{b,t} := (1-\beta)\big[\|\widetilde{\boldsymbol{M}}^{(b),t+1}\|_* - \langle \widetilde{\boldsymbol{M}}^{(b),t+1}, \widehat{\boldsymbol{Q}}^{(b),t+1}\rangle\big]_+, \tag{20}$$

where $[x]_+ := \max\{x, 0\}$. This error measures the loss of alignment with the effective momentum relative to the ideal Muon direction.

Lemma 6 (Descent with approximate orthogonalization) *Under Assumption 4 and equation 19, for every iteration t,*

$$f(\boldsymbol{X}^{t+1}) \le f(\boldsymbol{X}^t) - \eta_t \sum_{b=1}^{B} \|\nabla^{(b)} f(\boldsymbol{X}^t)\|_* + (1+\gamma)\sqrt{B}\,\eta_t\big(\beta\sqrt{\Theta_{t+1}} + (1-\beta)\sqrt{\Delta_t^{\mathrm{agg}}}\big) + \eta_t \sum_{b=1}^{B} \varepsilon_{b,t} + \frac{\gamma^2\eta_t^2}{2}\sum_{b=1}^{B} L_b.$$

Proof. Set $\boldsymbol{E}^{(b),t} := (1-\beta)\widetilde{\boldsymbol{M}}^{(b),t+1} - \nabla^{(b)} f(\boldsymbol{X}^t)$. By equation 20, duality between the spectral and nuclear norms, and the reverse triangle inequality,

$$\begin{aligned}\langle \nabla^{(b)} f(\boldsymbol{X}^t), \widehat{\boldsymbol{Q}}^{(b),t+1}\rangle &\ge (1-\beta)\|\widetilde{\boldsymbol{M}}^{(b),t+1}\|_* - \gamma\|\boldsymbol{E}^{(b),t}\|_* - \varepsilon_{b,t} \\ &\ge \|\nabla^{(b)} f(\boldsymbol{X}^t)\|_* - (1+\gamma)\|\boldsymbol{E}^{(b),t}\|_* - \varepsilon_{b,t}.\end{aligned}$$

The effective momentum satisfies

$$\boldsymbol{E}^{(b),t} = \beta\big((1-\beta)\boldsymbol{M}^{(b),t+1} - \nabla^{(b)} f(\boldsymbol{X}^t)\big) + (1-\beta)\big(\widehat{\boldsymbol{\Delta}}^{(b),t} - \nabla^{(b)} f(\boldsymbol{X}^t)\big).$$

Hence, the triangle inequality and Cauchy–Schwarz give

$$\sum_{b=1}^{B} \|\boldsymbol{E}^{(b),t}\|_* \le \sqrt{B}\Big(\sum_{b=1}^{B} \|\boldsymbol{E}^{(b),t}\|_*^2\Big)^{1/2} \le \sqrt{B}\big(\beta\sqrt{\Theta_{t+1}} + (1-\beta)\sqrt{\Delta_t^{\mathrm{agg}}}\big).$$

Substituting these estimates into the blockwise smoothness bound and using $\|\widehat{\boldsymbol{Q}}^{(b),t+1}\|_{2\to 2} \le \gamma$ proves the claim. □

Corollary 4 (Stationarity with approximate orthogonalization) *Suppose Assumptions 1–4 and equation 19 hold. Let the stepsizes be deterministic, positive, nonincreasing, and common to all blocks. For every $T \ge 1$, there is a constant $C_\gamma > 0$ depending only on α, β, γ, $\kappa_{\max}$, B, and $\{L_b\}_{b=1}^B$ such that*

$$\begin{aligned}&\frac{1}{H_T}\sum_{t=0}^{T-1} \eta_t \mathbb{E}\big[\sum_{b=1}^{B} \|\nabla^{(b)} f(\boldsymbol{X}^t)\|_*\big] \\ &\le C_\gamma\left[\frac{f(\boldsymbol{X}^0) - f^\star + \eta_0\mathbb{E}[\sqrt{\mathcal{M}_0} + \sqrt{\Theta_1}]}{H_T} + \Big(\sum_{b=1}^{B}\sigma_b^2\Big)^{1/2} + Q_T\right] \\ &\quad + \frac{2(1+\gamma)\sqrt{B}\,\kappa_{\max}\sqrt{1-\alpha}}{1-\sqrt{1-\alpha}}(R_T + \zeta) + \frac{1}{H_T}\sum_{t=0}^{T-1}\eta_t\mathbb{E}\big[\sum_{b=1}^{B}\varepsilon_{b,t}\big].\end{aligned}$$

Proof. The QR construction preserves orthonormal history bases, so Lemma 1 and the weighted aggregation bound equation 13 remain applicable. Each block now moves by at most $\gamma\eta_{t-1}$ in spectral norm. Accordingly, Assumption 4 and the argument of Lemma 4 give, for $t \ge 1$,

$$\mathbb{E}[\sqrt{\Theta_{t+1}}] \le \beta\mathbb{E}[\sqrt{\Theta_t}] + (1-\beta)\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \beta\gamma L_{\max}\sqrt{B}\,\eta_{t-1}.$$

Weighting by η_t and using the nonincreasing stepsizes as in equation 14, we obtain

$$\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Theta_{t+1}}] \le \frac{\eta_0\mathbb{E}[\sqrt{\Theta_1}]}{1-\beta} + \sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}] + \frac{\beta\gamma L_{\max}\sqrt{B}}{1-\beta}H_T Q_T.$$

Taking expectations in Lemma 6, summing over t, and substituting this estimate yields

$$\sum_{t=0}^{T-1}\eta_t\mathbb{E}\big[\sum_{b=1}^{B}\|\nabla^{(b)}f(\boldsymbol{X}^t)\|_*\big] \le f(\boldsymbol{X}^0)-f^\star+\frac{(1+\gamma)\sqrt{B}\,\beta}{1-\beta}\eta_0\mathbb{E}[\sqrt{\Theta_1}] + (1+\gamma)\sqrt{B}\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sqrt{\Delta_t^{\mathrm{agg}}}]+\sum_{t=0}^{T-1}\eta_t\mathbb{E}[\sum_{b=1}^{B}\varepsilon_{b,t}] + \left(\frac{\gamma(1+\gamma)B\beta^2L_{\max}}{1-\beta}+\frac{\gamma^2}{2}\sum_{b=1}^{B}L_b\right)H_TQ_T.$$

Substituting equation 13 and dividing by H_T gives the stated coefficient of $R_T+\zeta$. Collecting the remaining coefficients into C_γ completes the proof. □

Application to finite NS iterations. The quintic iteration in Jordan et al. (2024) applies the following polynomial to each normalized singular value:

$$\varphi(x)=3.4445x-4.7750x^3+2.0315x^5,\qquad \varphi([0,1.21])\subseteq[0,1.21].$$

The interval inclusion follows by checking the endpoints and critical points of φ. Frobenius normalization, optionally with a nonnegative stabilizer, places the initial singular values in $[0,1]$, with zero input mapped to zero. Induction therefore gives $\|\widehat{\boldsymbol{Q}}^{(b),t+1}\|_{2\to2}\le 1.21$ for any finite number of iterations in exact arithmetic. Thus, Corollary 4 applies with $\gamma=1.21$ and the alignment error in equation 20. For the ideal polar direction, $\gamma=1$ and $\varepsilon_{b,t}=0$, recovering Theorem 1.

Alignment error after finite NS iterations. For a nonzero effective momentum block, let $\mu_1,\dots,\mu_\ell>0$ be its nonzero singular values. Let $a\ge\|\widetilde{\boldsymbol{M}}^{(b),t+1}\|_F$ be the denominator used to normalize the block. Define $p_0(x)=x$ and $p_{s+1}(x)=\varphi(p_s(x))$. In exact arithmetic, NS preserves the singular vectors and maps each normalized singular value μ_j/a to $p_s(\mu_j/a)$. After s iterations, the alignment error is

$$\varepsilon_{b,t}=(1-\beta)\Big[\sum_{j=1}^{\ell}\mu_j\big(1-p_s(\mu_j/a)\big)\Big]_+\le(1-\beta)\sum_{j=1}^{\ell}\mu_j\big[1-p_s(\mu_j/a)\big]_+.$$

For a zero block, the alignment error is zero. Otherwise, it depends on the NS iteration count and normalized singular values. Neither $\gamma=1.21$ nor five NS iterations guarantees that its weighted average vanishes. Equation 20 includes rounding errors in the computed direction.

A.2.7 Fixed microstep accumulation

We now connect the notation for a single communication round to Algorithm 1 with a fixed number $S\ge1$ of microsteps. The iterate and bases stay fixed within each optimizer update. Assumptions 1 and 2 apply at each microstep, conditional on the history before compression and gradient sampling, respectively. No independence between workers or microsteps is needed for the bounds below.

At each optimizer update, average the gradients and reconstructions over the S microsteps and divide the local memories at the start and end by S:

$$\boldsymbol{G}_i^t:=\frac{1}{S}\sum_{s=0}^{S-1}\boldsymbol{G}_i^{t,s},\qquad\widehat{\boldsymbol{\Delta}}_i^t:=\frac{1}{S}\sum_{s=0}^{S-1}\widehat{\boldsymbol{\Delta}}_i^{t,s},\qquad\boldsymbol{H}_i^t:=\frac{\boldsymbol{H}_i^{t,0}}{S},\qquad\boldsymbol{H}_i^{t+1}:=\frac{\boldsymbol{H}_i^{t,S}}{S}.$$

The microstep error feedback identities telescope to $\widehat{\boldsymbol{\Delta}}_i^t=\boldsymbol{G}_i^t+\boldsymbol{H}_i^t-\boldsymbol{H}_i^{t+1}$. Thus the aggregation identity used in Lemma 3 is unchanged.

To verify the memory estimate, write $\mathcal{M}_{t,s}:=N^{-1}\sum_{i,b}\|\boldsymbol{H}_i^{(b),t,s}\|_F^2$ and $c:=\sqrt{1-\alpha}$. Lemma 2, applied at a microstep with fixed $\boldsymbol{X}^t$ and $\boldsymbol{\Pi}_b^t$, gives

$$\mathbb{E}[\sqrt{\mathcal{M}_{t,s+1}}]\le c\left(\mathbb{E}[\sqrt{\mathcal{M}_{t,s}}]+\mathbb{E}[\sqrt{\mathcal{P}_t}]+\zeta+\Big(\sum_b\sigma_b^2\Big)^{1/2}\right).$$

Since $\mathcal{M}_t = \mathcal{M}_{t,0}/S^2$ and $\mathcal{M}_{t+1} = \mathcal{M}_{t,S}/S^2$, iteration over the S microsteps yields

$$\mathbb{E}[\sqrt{\mathcal{M}_{t+1}}] \le c^S \mathbb{E}[\sqrt{\mathcal{M}_t}] + \frac{c}{S}\sum_{s=0}^{S-1} c^s \left(\mathbb{E}[\sqrt{\mathcal{P}_t}] + \zeta + \Big(\sum_b \sigma_b^2\Big)^{1/2}\right)$$
$$\le c\left(\mathbb{E}[\sqrt{\mathcal{M}_t}] + \mathbb{E}[\sqrt{\mathcal{P}_t}] + \zeta + \Big(\sum_b \sigma_b^2\Big)^{1/2}\right).$$

Here $c^S \le c$ and $S^{-1}\sum_{s=0}^{S-1} c^s \le 1$. The averaged gradient is conditionally unbiased by the tower property, and convexity gives

$$\mathbb{E}\big[\|\boldsymbol{G}_i^{(b),t} - \nabla^{(b)} f_i(\boldsymbol{X}^t)\|_F^2 \mid \mathcal{F}_t\big] \le \frac{1}{S}\sum_{s=0}^{S-1} \mathbb{E}\big[\|\boldsymbol{G}_i^{(b),t,s} - \nabla^{(b)} f_i(\boldsymbol{X}^t)\|_F^2 \mid \mathcal{F}_t\big] \le \sigma_b^2.$$

The memory and aggregation bounds retain their constants. Momentum and parameters update once using $\widehat{\boldsymbol{\Delta}}^t$, so Theorem 1 and Corollary 4 apply with this notation.

Theorem 2 and Corollary 3 also carry over. Besides the aggregation identity and the memory recursion verified above, their proof uses three properties of the averaged noise $\bar{\boldsymbol{\xi}}^t = N^{-1}\sum_{i=1}^N (\boldsymbol{G}_i^t - \nabla f_i(\boldsymbol{X}^t))$: it is $\mathcal{F}_{t+1}$-measurable, it satisfies $\mathbb{E}[\bar{\boldsymbol{\xi}}^t \mid \mathcal{F}_t] = \mathbf{0}$, and it satisfies $\mathbb{E}\|\bar{\boldsymbol{\xi}}^t\|_S^2 \le \sigma^2$. The first two follow from the tower property over microsteps. The third follows from the variance bound above and Jensen's inequality over workers.

The quantities T, R_T, and η_t refer to optimizer updates. The initial memory is $\boldsymbol{H}_i^{0,0}/S$, which is zero in Algorithm 1.

A.3 EXPERIMENTAL CONFIGURATIONS

Tables 6 and 7 give the main OpenWebText training settings. Table 8 lists the settings used in each figure and table, including the separate reconstruction experiments and ablations.

Table 6: Shared training settings for the main OpenWebText experiments with GPT-2 Large and NanoGPT-124M.

Setting	Both models
Logical workers	4
Micro-batch per worker	8 sequences of length 512
Gradient accumulation	4 microsteps per optimizer update
Effective global batch	128 sequences (65,536 tokens)
Schedule horizon	100,000 microsteps; 25,000 optimizer updates
Random seeds	5; matched initialization and data order
Training subset	1.5B tokens
Validation / test sets	2M tokens each
AdamW auxiliary peak learning rate	2×10^{-4}
Weight decay	10^{-4}
Muon momentum / NS iterations	0.95 / 5
Validation interval	500 optimizer updates

Training and communication steps. Each optimizer update contains four microsteps, each processing eight sequences per worker. Parameters and communication bases remain fixed during these microsteps. Forward and backward computation, compression, communication, reconstruction, and error feedback each occur four times; StructMuon and the factor baselines average the reconstructions over the microsteps before one momentum, NS, and parameter update. The bases are then refreshed once for the next update. Thus, 10K steps means 10,000 optimizer updates and 40,000 microsteps per worker, with effective batch size $4 \times 4 \times 8 = 128$ sequences. Accordingly, 10K optimizer updates process 0.65536B tokens, and the full 25K-update schedule processes 1.6384B tokens. The 1.5B-token quantity specifies the training subset, from which samples can be reused. The 25K-update horizon is distinct from the reporting checkpoints: Table 1 evaluates 10K updates, while Figures 5 and 6 show all five methods, including EF21-Muon, from 500 to 15K updates on GPT-2 Large and NanoGPT-124M, respectively.

Table 7: Communication and optimizer settings for the main OpenWebText comparisons. Optimizer and peak learning rate (LR) refer to hidden weight matrices; the AdamW auxiliary rate is listed separately in Table 6.

Method	Optimizer	Peak LR	EF	GPT-2 Large	NanoGPT-124M
Full Muon	Muon	2×10^{-2}	–	Full transmission	Full transmission
PowerSGD	Momentum SGD	2×10^{-2}	On	$q = 15$	$q = 3$
EF-PowerMuon	Muon	2×10^{-2}	On	$q = 15$	$q = 3$
EF21-Muon	Muon	2×10^{-2}	EF21	Top-K, $K = 38{,}400$	Top-K, $K = 4{,}608$
StructMuon	Muon	2×10^{-2}	On	$r = 276,\ k = 150$	$r = 94,\ k = 100$

EF21-Muon baseline. For the NanoGPT training comparison, EF21-Muon uses Top-K compression, retaining $K = 4{,}608$ entries per hidden matrix, worker, and communication round. It maintains the EF21 estimate of local momentum through compressed differences (Gruntkowska et al., 2025); it does not form StructMuon's corrected inputs. The server-side compressor is the identity. Token and position embeddings retain rank-two factor compression, and bias and LayerNorm parameters are transmitted in full, matching the other compressed NanoGPT methods. With 32-bit values and indices, its hidden-matrix payload is $12 \times 4 \times 4608 \times 8 = 1{,}769{,}472$ bytes per round. Including $904{,}192$ auxiliary bytes gives 0.107 TB/worker over 10K updates, matching the factor baselines. For GPT-2 Large, EF21-Muon uses $K = 38{,}400$ entries per hidden matrix, matching the factor

payload of 1.917 TB/worker at 10K updates. Its full trajectory is shown through 15K updates in Figure 5, and its 10K endpoint and target-PPL payloads appear in Tables 1 and 2. Payloads in the main comparisons count the worker-to-server direction; they exclude downstream broadcasts, collective amplification, and protocol overhead.

Optimizer and compression scope. PowerSGD uses momentum SGD with local error feedback for the hidden weight matrices; methods using Muon set momentum to 0.95 and use five NS iterations. PowerSGD adds its local error memory before factor compression and retains the difference between that corrected input and its local reconstruction for the next communication round. All methods update the remaining parameters with AdamW at peak learning rate 2×10^{-4}, independently of the hidden weight learning rates in Table 7. For the main training comparisons, query, key, and value weights are compressed jointly as one fused QKV matrix. This matrix, the attention output matrix, and the two MLP weight matrices use the communication rule in Table 7. In all compressed methods, token and position embeddings use PowerSGD with $q = 6$ for GPT-2 Large and $q = 2$ for NanoGPT-124M, while biases and LayerNorm parameters are transmitted in full. Full Muon transmits all parameters in full. Embedding handling is kept fixed within each ablation of residual compression or error feedback.

Quantization baselines. Value quantization is bounded by a $32\times$ reduction relative to FP32, attained only at one bit per coordinate. Our matched hidden-matrix budgets correspond to $64\times$ compression for GPT-2 Large ($q = 15$) and $192\times$ for NanoGPT-124M ($q = 3$), so quantization alone cannot meet these budgets and is not directly comparable. It is orthogonal to StructMuon: the transmitted $r \times r$ core and the Top-k residual values can themselves be quantized.

Here r is the side length of the core, q is the PowerSGD factor rank, and k is the absolute number of retained residual entries per matrix, worker, and communication round. The reconstruction diagnostic in Figure 4 uses fixed residual budgets $k = 100$ and $k = 200$. The OpenWebText Core-only + EF ablation sets $(r, k) = (94, 0)$ and otherwise retains the main NanoGPT configuration. The independent WikiText-103 w/o EF variant retains residual compression and disables error feedback.

Table 8: Correspondence between experimental settings and reported results. Rank and residual budget sweeps use the values shown in their panels.

Results	Setting	Comparison
Figures 5–6; Table 1	OpenWebText, both models	Tables 6–7; Table 1 evaluates 10K optimizer updates.
Table 2; Figure 8	OpenWebText, both models	Main runs at PPL 150/130 (GPT-2 Large) and 180/160 (NanoGPT-124M).
Figure 11	OpenWebText, both models	Main configurations; reconstruction quality through 10K optimizer updates.
Figure 9	OpenWebText, both models	Vary r; report validation loss and payload at 10K optimizer updates.
Figure 13	OpenWebText, NanoGPT-124M	Core-only + EF versus StructMuon at $r = 94$, $k = 0/100$, with error feedback enabled; 10K updates.
Figures 10 and 7	WikiText-103, NanoGPT-124M	10K updates. Figure 10: $r = 32, 94$. Figure 7: $r = 32, 64, 94$; Core-only $k = 0$ and StructMuon $k = 150$ retain EF; w/o EF uses $k = 150$.
Figures 2–3	NanoGPT-124M	History bases at matched ranks; core versus PowerSGD with q selected from the total structured payload per layer, using separate attention matrices.
Figure 4	Residual diagnostic	Core reconstruction over $r = 32$–94, with Top-100 and Top-200 corrections; density panels use $r = 32, 64, 94$.
Figure 12; Table 11	OpenWebText, NanoGPT-124M	Learning curves for different communication bases and a 10K-update comparison with $r = 94$, $k = 100$, and error feedback enabled.

Communication accounting. Communication volumes sum logical payloads per worker over all parameter groups and communication rounds. With 32-bit values and sparse indices, an $m \times n$ block costs $4(r^2 + 2k)$ bytes for core plus Top-k, $4q(m + n)$ for PowerSGD, or $4mn$ for full transmission, following the rules above. For a compressed hidden matrix, $B_{\text{round}} = 4(r^2 + 2k)$ bytes and $B_{\text{update}} = 4B_{\text{round}} = 16(r^2 + 2k)$ bytes. The $2k$ term counts k values and k indices. Whole-model totals sum the appropriate rule over hidden matrices, embeddings, and parameters transmitted in full. Bases remain local. Counts exclude traffic multipliers imposed by collectives and protocol overhead; GB and TB denote 10^9 and 10^{12} bytes. Equal payload sizes across the four workers make their combined payload four times the reported volume, preserving all communication ratios. Appendix A.8 separately accounts for downstream broadcasts under an explicit coordinator protocol.

For a model with L layers, hidden width h, vocabulary size V, P position rows, and embedding factor rank q_e, the four fused hidden blocks contribute $16Lr^2 + 32Lk$ bytes per round for StructMuon and $64Lhq$ for factor compression. Compressed embeddings add $4q_e(V + P + 2h)$ bytes, with tied token/output weights counted once; biases and LayerNorm add $4(13L + 2)h$. Full transmission costs $4[12Lh^2 + (13L + 2)h + (V + P)h]$ bytes per round. We use $(L, h, V, P, q_e) = (36, 1280, 50257, 512, 6)$ for GPT-2 Large and $(12, 768, 50304, 512, 2)$ for NanoGPT-124M. At 10K updates, StructMuon sends 1.909 and 0.106 TB per worker, compared with 1.917 and 0.107 TB for the factor methods.

Residual aggregation interface. The residual aggregation in Algorithm 1 is defined by its decoded value:

$$\mathcal{D}_t(\boldsymbol{B}^{t,s}) = \frac{1}{N}\sum_{i=1}^{N} \mathcal{D}_t(\boldsymbol{B}_i^{t,s}).$$

For Top-k, concatenate the workers' value-index pairs, sum entries with the same index, and divide their values by N. The aggregated message may contain more than k entries. This gives the same matrix as decoding before averaging. In Algorithm 2, $\mathcal{C}$ and $\mathcal{D}$ denote the current round's compressor and decoder, with iteration indices suppressed.

Collective scaling. The core can be averaged with an AllReduce on r^2 values. Sparse supports can differ across workers; one exact implementation gathers their value-index pairs and performs a local scatter-add. With N workers and k entries per worker, this implementation receives $\mathcal{O}(Nk)$ entries per worker, and the combined support has size at most $\min\{Nk, mn\}$. Core payload and sparse payload therefore have different scaling with worker count. The logical volumes reported here exclude these collective-dependent traffic and workspace costs; the four-worker measurements do not establish scaling to large clusters.

Comparison with once-per-update dense synchronization. Full Muon can accumulate its S microstep gradients locally and synchronize their average once per update, producing the same update in exact arithmetic. Its logical payload is then divided by $S = 4$. If s is the saving against four-round Full Muon, the saving against this accumulated reference is $1 - 4(1 - s)$. Using unrounded target payloads gives 92.0–97.5% savings for StructMuon under this alternative accounting. Table 9 reports both protocols. At 10K updates, once-per-update Full Muon sends 30.935 TB for GPT-2 Large and 4.963 TB for NanoGPT-124M. This rescaling concerns payload, not measured runtime; moving compression after accumulation would change the compressed input sequence.

Table 9: Analytical Full Muon payloads per worker at the target locations in Table 2. Once-per-update values divide the unrounded four-round payloads by four; displayed values use three decimal places.

Model	Target PPL	Four rounds (TB)	One round (TB)
GPT-2 Large	150	17.920	4.480
GPT-2 Large	130	20.988	5.247
NanoGPT-124M	180	3.728	0.932
NanoGPT-124M	160	4.285	1.071

Matching validation perplexity. For the reported aggregate validation-PPL trajectories in Table 2 and Figure 8, target locations are obtained by interpolating between adjacent checkpoints that bracket the target PPL. Cumulative payload is evaluated at each interpolated location, including both EF21-Muon comparisons. The NanoGPT PowerSGD crossing at PPL 160 occurs at approximately 13.03K updates in the extended trajectory; this target is not reached within 10K. Core-only + EF is evaluated in Appendix A.7 and does not reach PPL 160 or 140 within 10K. Appendix A.4 separately reports measured runtime at a fixed budget of 10K optimizer updates.

Global reconstruction metrics. At a given optimizer update, let j index the compressed hidden matrices, workers, and all four microsteps. Write $\boldsymbol{D}_j$ for the local reconstruction target and $\widehat{\boldsymbol{D}}_j$ for its approximation, before averaging across workers. For StructMuon, these are the corrected input and the combined core and residual reconstruction. Parameters transmitted in full and embeddings are excluded from these diagnostics of hidden matrices. We compute

$$\mathrm{Cos}_{\mathrm{global}} = \frac{\sum_j \langle \boldsymbol{D}_j, \widehat{\boldsymbol{D}}_j \rangle}{\sqrt{\sum_j \|\boldsymbol{D}_j\|_F^2}\sqrt{\sum_j \|\widehat{\boldsymbol{D}}_j\|_F^2}}, \qquad \mathrm{RelErr}_{\mathrm{global}} = \sqrt{\frac{\sum_j \|\boldsymbol{D}_j - \widehat{\boldsymbol{D}}_j\|_F^2}{\sum_j \|\boldsymbol{D}_j\|_F^2}}.$$

Equivalently, with $w_j = \|\boldsymbol{D}_j\|_F^2 / \sum_\ell \|\boldsymbol{D}_\ell\|_F^2$, global relative error is $\sqrt{\sum_j w_j \mathrm{RelErr}_j^2}$. Global cosine is computed from the summed inner products and energies above, rather than by averaging local cosines with these weights. The core diagnostics use $\widehat{\boldsymbol{D}}_j = \boldsymbol{\Pi}_t(\boldsymbol{D}_j)$ and report both metrics; training curves show cosine similarity. EF21-Muon's 10K relative errors are reported in Appendix A.5.

Repeated runs. Training comparisons use five independent random seeds, with matched initialization, data order, and evaluation settings across methods. Tables and curves use the same runs and checkpoints. In Table 1, PPL is the exponential of the mean validation loss. Training bands show mean $\pm$ one sample standard deviation: for run values $z_1, \ldots, z_5$, the mean is $\bar{z} = \frac{1}{5}\sum_{a=1}^5 z_a$ and the SD is $[\frac{1}{4}\sum_{a=1}^5 (z_a - \bar{z})^2]^{1/2}$. The gray region in Figure 3 marks above-budget references.

A.4 Measured Training Runtime

We report NanoGPT-124M runtime over 10K optimizer updates using the main configurations in Table 7. Each update comprises four microsteps. Table 10 separates per-update time from total run time, which also includes initialization and final data saving. These one-off costs are excluded from per-update timing, accounting for the difference between total time and the sum of update times. Runtimes are measured on the export-compliant A800 variant, whose inter-GPU bandwidth is restricted; in our measurements, dense gradient synchronization dominates the Full Muon update time. This bandwidth-limited regime resembles PCIe or cross-node links, where communication compression matters most. With high-bandwidth interconnects such as full NVLink, the runtime gap to Full Muon would be smaller.

Table 10: Measured NanoGPT-124M runtime at 10K optimizer updates. The two columns report per-update time and total run time, at a fixed update budget.

Method	Time per update (ms)	Total time at 10K (min)
Full Muon	1861.1 ± 5.4	315.60 ± 1.27
EF-PowerMuon	597.0 ± 1.6	99.90 ± 0.54
StructMuon	630.7 ± 2.1	107.77 ± 0.41

StructMuon reduces total time from 315.60 to 107.77 minutes relative to Full Muon, a $2.93\times$ speedup. Compared with EF-PowerMuon, it takes 7.87 additional minutes (7.9%) over 10K updates while lowering validation loss from 4.86 to 4.40 and PPL from 128.4 to 81.4 (Table 1). Thus the structured representation improves quality at a modest runtime cost relative to factor compression, while both compressed Muon methods substantially reduce runtime relative to full communication. Table 2 compares communication at matched PPL, and Appendix A.8 reports the corresponding time-to-target comparison.

A.5 Additional Experimental Results

This section compares communication at matched validation PPL, examines the core rank and residual budget, and tracks reconstruction during training. These experiments show how the two channels affect training quality at different communication costs.

Reconstruction targets. The history basis comparison in Figure 2 evaluates Muon, SGD, and AdamW bases on the same corrected inputs $\boldsymbol{\Delta}_i^t = \boldsymbol{G}_i^t + \boldsymbol{H}_i^t$. The PowerSGD comparison in Figure 3 uses the same inputs and metrics, with q selected from the total core budget per layer. The history bases are fixed before the current gradients are sampled. The core reconstruction is $\mathbf{\Pi}_t(\boldsymbol{\Delta}_i^t)$, and adding the transmitted residual gives $\widehat{\boldsymbol{\Delta}}_i^t$. Core comparisons report cosine similarity and relative error. StructMuon's residual and training comparisons measure cosine similarity between the corrected input and the complete reconstruction.

These measurements assess how accurately the corrected input is reconstructed. The Muon direction $\boldsymbol{Q}^{t+1}$ is computed afterward, through aggregation, momentum, and orthogonalization. In the analysis, averaging the unnormalized squared projection residuals over workers and summing over blocks gives $\mathcal{E}_t$. The quantity R_T measures the component of the true gradient omitted by the history subspaces. The memory recursion connects these two quantities, relating the residual of the corrected input to the gradient omission in the stationarity bound.

Communication at matched validation quality. The main results compare methods at a fixed number of optimizer updates. Here we examine the communication required to reach a common validation perplexity. This comparison accounts for both the payload of each update and the number of updates needed to reach the target, connecting message compression to progress in training.

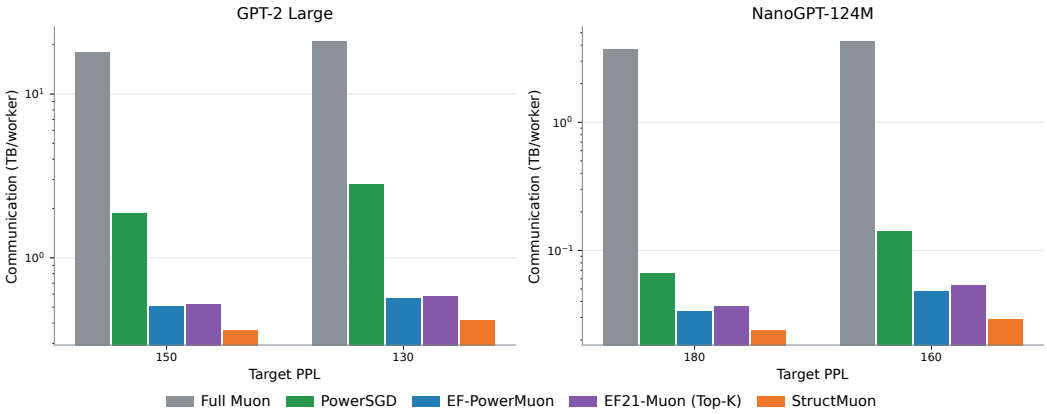


Figure 8: Cumulative uplink logical payload per worker at matched validation PPL, with four rounds per update and logarithmic vertical axes. EF21-Muon target results are included on both models. The hatched PowerSGD bar uses the PPL 160 crossing at approximately 13.03K updates, beyond the 10K reporting checkpoint.

Figure 8 shows that StructMuon requires the least communication at every reported target on both models. EF-PowerMuon and EF21-Muon have similar payloads per update, so the savings mainly reflect fewer updates to reach the target (Table 2).

Sensitivity to the core rank. We next vary the rank of the core channel. Increasing r gives the core more coordinates in the left and right history subspaces, allowing it to represent more of the corrected input. The core payload grows as r^2, so the experiment examines how much validation quality improves as this communication cost increases.

Figure 9 shows lower validation loss at larger ranks on both models, accompanied by higher communication cost. The loss reduction becomes smaller as the rank grows, so each further increase in core capacity yields less improvement in validation quality. The main settings lie in this transition. This behavior supports using a compact core that captures much of the observed training benefit while limiting the cost of additional coordinates.

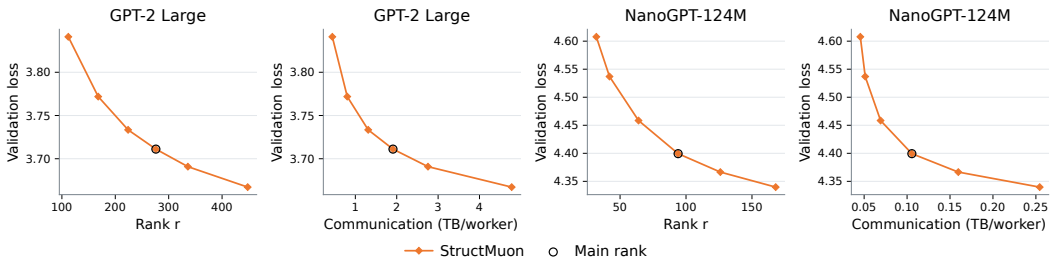


Figure 9: Core-rank sensitivity at 10K updates on OpenWebText. For each model, validation loss is plotted against rank and cumulative TB/worker in separate panels. Open circles mark the main settings $r = 276$ (GPT-2 Large) and $r = 94$ (NanoGPT-124M).

Sensitivity to the residual Top-k budget. The structured core can leave part of the corrected input outside its history subspaces, particularly at small ranks. We therefore vary the residual budget k at a fixed core rank to examine the value of transmitting this information. Here, $k = 0$ is the Core-only variant with error feedback retained; increasing k transmits more residual entries.

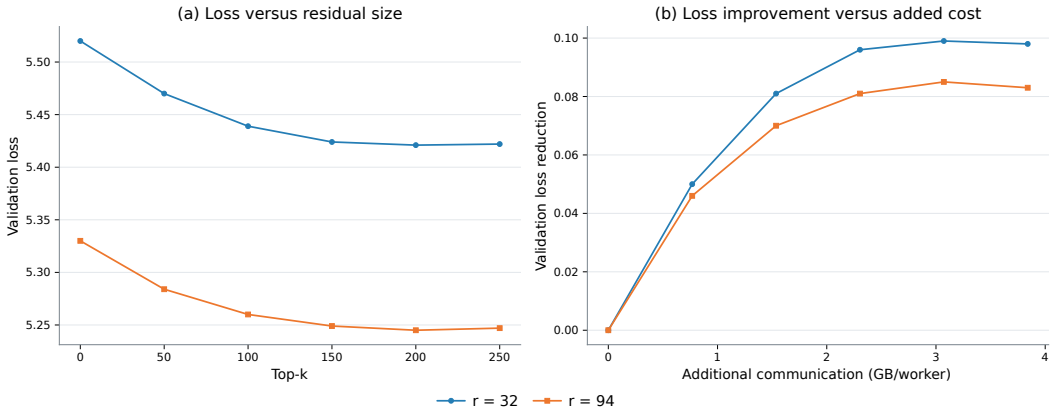


Figure 10: Residual Top-k sensitivity on WikiText-103 with NanoGPT-124M at $r = 32, 94$ and 10K updates. Left: validation loss versus k. Right: loss reduction versus additional GB/worker, both relative to $k = 0$ at the same rank. Error feedback remains enabled throughout.

Figure 10 shows diminishing returns from the residual channel. At $k = 150$, validation loss falls from 5.520 to 5.424 at $r = 32$ and from 5.330 to 5.249 at $r = 94$, reductions of 0.096 and 0.081. Increasing k to 200 gives only a further 0.003 and 0.004 reduction, respectively; at $k = 250$, loss increases slightly. These values support a compact residual correction, without establishing statistical significance for the small differences at large k.

The right panel shows the added communication and loss reduction relative to $k = 0$ at each fixed rank. Let $C(k)$ and $\ell_{\text{val}}(k)$ denote cumulative communication and validation loss at residual budget k; the plotted values are $C(k) - C(0)$ and $\ell_{\text{val}}(0) - \ell_{\text{val}}(k)$. Each additional 50 entries costs 0.768 GB/worker over 10K updates, so $k = 150$ adds 2.304 GB/worker at either rank. The total payloads still differ: at $k = 0$, they are 0.044 and 0.104 TB/worker for $r = 32$ and $r = 94$. This WikiText-103 sweep is separate from the main NanoGPT setting $k = 100$.

Reconstruction quality during training. Finally, we track reconstruction alignment as training progresses. For StructMuon, cosine similarity compares the complete reconstruction from both channels $\widehat{\mathbf{\Delta}}_i^t$ with its input $\mathbf{\Delta}_i^t$. This global metric aggregates compressed hidden matrices, workers, and microsteps as defined in Appendix A.3, including both the structured core and the transmitted residual. Higher cosine similarity indicates better directional alignment with the corrected input.

Figure 11 shows higher reconstruction alignment for StructMuon, reaching cosine similarity 0.928 on GPT-2 Large and 0.925 on NanoGPT-124M at 10K updates. EF21-Muon attains cosine similarity 0.834 and relative error 0.558 on GPT-2 Large, and 0.814 and 0.586 on NanoGPT-124M.

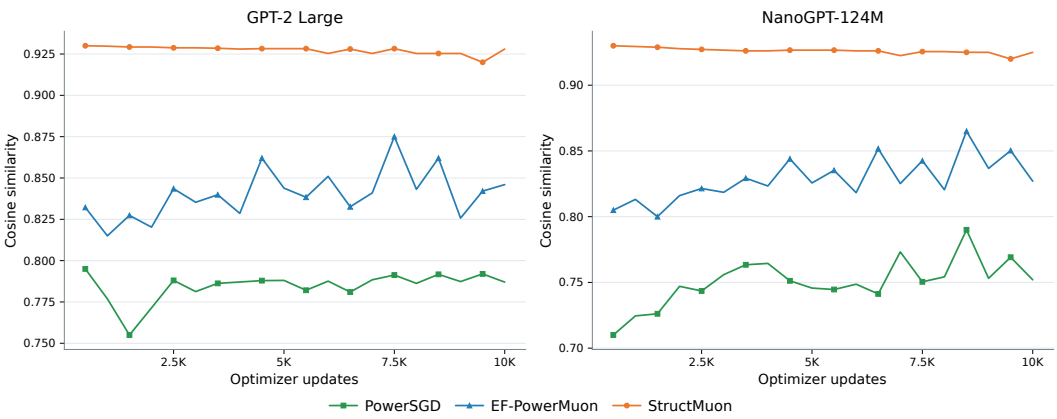


Figure 11: Global reconstruction cosine similarity during training on GPT-2 Large and NanoGPT-124M. Curves show means over five independent seeds; values at 10K optimizer updates match Table 1.

A.6 Communication Basis Ablation

We compare Muon history bases with Random and Momentum-only bases. The comparison uses NanoGPT-124M on OpenWebText with the main configuration $r = 94$ and $k = 100$, retaining Muon updates, local error feedback, and the remaining training settings. All variants therefore have the same core dimension, residual budget, and payload per update, allowing us to assess the role of the basis source at fixed communication capacity.

Basis construction and refresh. Only the communication bases change. All variants use the same Muon optimizer, residual compressor, error feedback, and refresh schedule. For the states obtained at the end of update t, the next bases are

Bases	Right basis $\boldsymbol{V}^{t+1}$	Left basis $\boldsymbol{U}^{t+1}$
Random	$\mathrm{qf}(\boldsymbol{Z}_V^{t+1})$	$\mathrm{qf}(\boldsymbol{Z}_U^{t+1})$
Momentum-only	$\mathrm{qf}((\widetilde{\boldsymbol{M}}_{\mathcal{I},:}^{t+1})^\top)$	$\mathrm{qf}(\widetilde{\boldsymbol{M}}^{t+1}\boldsymbol{V}^{t+1})$
Muon history	$\mathrm{qf}((\widetilde{\boldsymbol{M}}_{\mathcal{I},:}^{t+1})^\top)$	$\mathrm{qf}(\boldsymbol{Q}^{t+1}\boldsymbol{V}^{t+1})$

Here $\boldsymbol{Z}_V^{t+1} \in \mathbb{R}^{n\times r}$ and $\boldsymbol{Z}_U^{t+1} \in \mathbb{R}^{m\times r}$ are independent standard Gaussian matrices, resampled after each optimizer update using a seed shared by all workers. The ordered row set $\mathcal{I}$ is sampled before training and held fixed. Momentum-only and Muon history bases therefore use the same construction for the right basis; the left basis uses momentum in the former and the Muon direction in the latter. Momentum-only refers to basis construction, not to a different optimizer.

Each new pair is used for all four microsteps of the next optimizer update. All variants start from the same initial bases and use the deterministic QR sign and completion conventions in Definition 1, including for zero or deficient rank. This keeps basis dimension, sampling rows, and refresh frequency fixed across the two history constructions.

Table 11: Mean basis comparison over five seeds at 10K optimizer updates with $r = 94$ and $k = 100$. All variants use the same communication budget (0.106 TB per worker). The StructMuon row agrees with Table 1.

Communication bases	Train loss	Val loss	Val PPL	Cos.
Random	6.97	6.81	903.1	0.134
Momentum-only	5.49	5.21	183.2	0.850
Muon history (StructMuon)	**4.94**	**4.40**	**81.4**	**0.925**

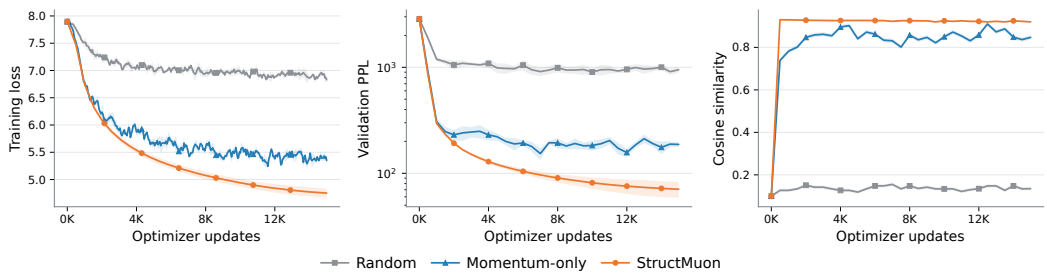
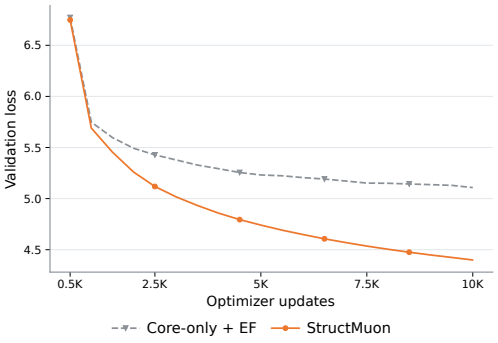


Figure 12: Communication basis ablation on NanoGPT-124M with $r = 94$ and $k = 100$, through 15K optimizer updates. Panels show training loss, validation PPL, and global cosine similarity. Lines show means over five independent seeds; shading shows ± 1 SD. The PPL axis is logarithmic. All variants share the initialization, and 10K values agree with Tables 1 and 11.

Momentum-only bases improve reconstruction and validation quality over Random bases. Muon history bases further reduce validation loss from 5.21 to 4.40 and improve global cosine similarity from 0.850 to 0.925 at 10K updates (Table 11). The two constructions use the same rule for the right basis. In the notation of Proposition 1, $\boldsymbol{V} = \boldsymbol{RW}$: the ideal polar map gives $\boldsymbol{Q}_\star \boldsymbol{V} = \boldsymbol{LW}$, whereas the momentum map gives $\widetilde{\boldsymbol{M}}\boldsymbol{V} = \boldsymbol{L\Sigma W}$. The latter reweights the selected coordinates by historical singular values before QR. Removing this reweighting preserves the paired coordinates and provides a structural explanation consistent with the observed improvement. It does not prove smaller current-gradient projection error for every input.

A.7 Residual-Channel Ablation on OpenWebText

We isolate the residual channel on OpenWebText by comparing StructMuon with Core-only + EF at $r = 94$. Both retain error feedback and the main training settings; only the residual budget changes from $k = 100$ to $k = 0$.



Metric	Core-only + EF	StructMuon
Residual k	0	100
Train loss	5.385	**4.940**
Val loss	5.108	**4.399**
PPL	165.313	**81.400**
Comm. (TB/worker)	**0.104**	0.106

Figure 13: OpenWebText residual ablation on NanoGPT-124M at $r = 94$, with EF enabled. Left: validation loss; right: 10K results.

Adding the residual channel lowers validation loss from 5.108 to 4.399 and PPL from 165.31 to 81.40 (Figure 13), for only 1.536 GB/worker more communication (1.48%). With error feedback kept, the 0.708 loss reduction isolates the value of transmitting information outside the subspaces.

A.8 Time to Target and Bidirectional Communication

Time to target perplexity. Table 12 reports cumulative wall-clock time to the first evaluated checkpoint with validation PPL at or below the target, including validation through that checkpoint. Evaluation occurs every 500 updates, with a 15K-update limit. We report one mean time per method and target. NR marks targets not reached in every run; the count in parentheses gives the number that reached the target. We do not substitute a mean over successful runs.

Table 12: Mean wall-clock time to target validation PPL, in minutes. Columns show PPL thresholds; NR denotes incomplete attainment within 15K updates, with the reached count out of five. Compressed methods use four uplink rounds per update; Full Muon uses four rounds or one round after accumulation.

	GPT-2 Large		NanoGPT-124M			
Method	150	130	200	180	160	140
Full Muon (4 rounds)	450.37	563.28	64.07	64.07	79.71	85.96
Full Muon (1 round)	210.89	265.74	29.38	29.38	37.29	43.26
PowerSGD	795.90	NR (2/5)	43.80	NR (4/5)	NR (2/5)	NR (2/5)
EF-PowerMuon	262.33	309.17	28.02	35.93	49.81	79.80
EF21-Muon	242.03	266.28	30.01	39.27	58.77	NR (3/5)
StructMuon	203.22	252.42	24.11	27.33	32.63	42.32

StructMuon has shorter mean times than both Full Muon variants at all six targets, with a 2.03–2.66$\times$ speedup over four-round Full Muon. Against once-per-update Full Muon, the time reduction is 3.6–5.0% on GPT-2 Large and 2.2–17.9% on NanoGPT-124M. Among compressed methods, it is fastest at all six targets, saving 16.0% and 5.2% over EF21-Muon on GPT-2 Large. These checkpoint-based times complement the interpolated communication results in Table 2.

Bidirectional communication. The main comparisons report uplink payload. To make the downstream cost explicit, Table 13 considers a coordinator protocol with four uplink rounds and one full-model FP32 broadcast per optimizer update. It also includes intermediate PowerSGD factor broadcasts: all compressed methods use these for embeddings, and PowerSGD and EF-PowerMuon additionally use them for hidden matrices. Bases remain local. The once-per-update Full Muon reference accumulates gradients before synchronization, as described in Appendix A.3.

Table 13: Logical payload at 10K optimizer updates under the coordinator protocol described here, in TB/worker. Downlink includes intermediate factors and the final dense broadcast; total is uplink plus downlink. These are analytical payload counts, excluding collective amplification and protocol overhead.

	GPT-2 Large			NanoGPT-124M		
Method	Up	Down	Total	Up	Down	Total
Full Muon (4 rounds)	123.740	30.935	154.675	19.853	4.963	24.817
Full Muon (1 round)	30.935	30.935	61.870	4.963	4.963	9.927
PowerSGD	1.917	31.979	33.896	0.107	5.019	5.126
EF-PowerMuon	1.917	31.979	33.896	0.107	5.019	5.126
EF21-Muon	1.917	30.984	32.901	0.107	4.980	5.087
StructMuon	1.909	30.984	32.893	0.106	4.980	5.085

The dense broadcast contributes 30.935 TB/worker for GPT-2 Large and 4.963 TB/worker for NanoGPT-124M, dominating the compressed methods' downstream traffic. StructMuon's total-payload savings relative to four-round Full Muon are 78.7% and 79.5%; against once-per-update Full Muon, they are 46.8% and 48.8%. Thus uplink savings do not translate directly into the same bidirectional savings.