# FROM SPECTRAL MEASUREMENT TO FEEDBACK COLLAPSE: THE LIMITS OF ADAPTIVE ORTHOGONAL OPTIMIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Low-rank orthogonal optimizers reduce computation and communication but require a rank hyperparameter that is costly to tune. We ask whether this rank can be inferred directly from the momentum spectrum. We develop a spectral diagnostic based on random matrix theory, using corrected residual-energy and tail-quantile estimators to identify structured directions above the estimated noise edge. Experiments on GPT-style models from 60M to 350M parameters show that momentum spectra are gapless and approximately power-law, while sampling noise accounts for under $11\%$ of the estimated noise floor. The diagnostic remains stable across scales, selecting a rank fraction of $0.353$–$0.361$ as model size grows sixfold, but validation loss continues to improve beyond the estimated rank, indicating that it provides a lower bound on useful rank rather than an optimization optimum. When used online with error feedback, the criterion induces a positive feedback loop that drives $83$–$88\%$ of weight matrices toward rank one. A pool-wide subtraction modification eliminates this collapse across all tested scales. These results distinguish spectral rank measurement from adaptive rank control in low-rank orthogonal optimization.

## 1 INTRODUCTION

Orthogonalized optimizers such as Muon (Jordan et al., 2024) update weight matrices along orthogonalized momentum directions, but full-rank orthogonalization can be costly in computation and distributed communication. Low-rank variants reduce this cost by restricting each update to $r$ dominant directions (He et al., 2026; Ahn et al., 2025). The rank $r$ therefore controls a basic trade-off between computational efficiency and optimization quality. In practice, however, selecting an effective rank typically requires grid sweeps over full training runs, making rank selection increasingly expensive as model scale grows.

This raises a fundamental question: *does the momentum spectrum contain a natural rank cutoff?* The approximately low-rank structure of gradients and momentum (Gur-Ari et al., 2018; Refael et al., 2025; Shen et al., 2026) suggests that much of the update may be concentrated in a relatively small subspace, but this does not determine what lies outside that subspace. If the discarded directions are dominated by sampling noise, random matrix theory (RMT) provides a principled criterion: pure-noise singular values concentrate below the Marchenko–Pastur edge (Marčenko & Pastur, 1967; Bai & Silverstein, 2010), while sufficiently strong signal directions separate above it (Baik et al., 2005). Momentum averaging should further suppress stochastic perturbations and enlarge this separation (Li et al., 2026). Under this picture, counting singular values above the noise edge could determine the rank without manual tuning. If the spectral tail instead contains weak but persistent structure, however, no sharp signal-noise cutoff should exist.

We test this hypothesis by turning the RMT noise edge into a spectral diagnostic for transformer pretraining. Because the low-rank optimizers considered here observe only part of the momentum spectrum, applying the criterion requires estimating an unobserved noise floor $\hat{\sigma}$. A naive residual-energy estimator assumes that noise energy is distributed uniformly across singular directions and consequently underestimates the noise level even for pure-noise matrices. We address this problem with a corrected estimator that combines a residual estimate adjusted by the Marchenko–Pastur energy



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.

distribution with a tail-quantile estimate following Gavish & Donoho (2014). The resulting diagnostic is robust to partial spectral observation and provides a measurable structured-rank fraction.

Applying this diagnostic to GPT-style language models from 60M to 350M parameters shows that pretraining momentum does not exhibit the signal-plus-noise separation required by a natural spectral cutoff. The singular-value spectrum is approximately gapless and power-law, while variance decomposition using batch-size scaling and momentum dynamics indicates that sampling noise accounts for only a small fraction of the estimated noise floor (under 11% at 60M and under 6% at 350M). These observations complement evidence that non-dominant gradient directions can retain useful information (Song et al., 2025; Zhang et al., 2025; Huang et al., 2026). Consistently, validation loss improves monotonically with rank across all six fixed-rank configurations that we examine. In the settings studied here, the noise edge therefore identifies a lower bound on structured rank rather than an optimization-optimal rank: low-rank orthogonal updates are better viewed as lossy compression of useful update information than as spectral denoising.

We next examine whether the same measurement can be used for online rank control. In an optimizer with selective error feedback such as Dion (Ahn et al., 2025), the rank decision changes the momentum state on which subsequent decisions are based. Directions that are not selected remain in the buffer and accumulate, increasing the estimated noise floor; the resulting higher threshold then selects fewer directions at the next step. In our language-model experiments, this positive feedback loop drives 83–88% of weight matrices toward rank one, a failure mode we call **Feedback Collapse**. A direct intervention that subtracts residual momentum across the full candidate pool prevents the collapse at all three model scales while leaving the rank-selection criterion and weight update unchanged. This isolates the failure to the interaction between adaptive selection and selectively retained residuals rather than to the spectral estimator itself.

These findings separate two uses of the same spectral criterion. As an offline measurement, it provides a stable estimate of structured rank that can guide a fixed-rank run and reduce the total cost of rank selection and training by about $2.2\times$ relative to a three-level grid sweep. As an online controller, it can interact with optimizer state and become unstable under selective error feedback. The distinction is important because accurate measurement alone does not guarantee stable closed-loop control.

**Contributions.**

- **A spectral diagnostic for structured momentum rank.** We develop a corrected noise-floor estimator for partially observed momentum spectra and use it to obtain a stable structured-rank measurement across the language-model scales studied, reducing the total cost of rank selection and training by about $2.2\times$ compared to a grid sweep.
- **A characterization of the discarded spectrum.** We show that pretraining momentum is approximately gapless and power-law, that over 89% of the sub-threshold component is not sampling noise, and that validation loss continues to improve beyond the measured rank. The noise edge therefore provides a lower bound on structured rank rather than an optimization optimum.
- **Feedback Collapse and its resolution.** We identify a positive feedback loop between adaptive spectral selection and selective error feedback that drives 83–88% of weight matrices toward rank one, and show through a pool-wide subtraction intervention that this loop can be eliminated across all three language-model scales.

## 2 Background and Related Work

### 2.1 Orthogonalized Optimizers and Low-Rank Error Feedback

For a weight matrix $\boldsymbol{W} \in \mathbb{R}^{m\times n}$ with gradient $\boldsymbol{G}_t$, orthogonalized optimizers maintain a momentum buffer:

$$\boldsymbol{M}_t = \mu \boldsymbol{M}_{t-1} + \boldsymbol{G}_t. \tag{1}$$

Instead of applying this buffer directly, they replace it with its nearest orthogonal matrix. Specifically, if the singular value decomposition is $\boldsymbol{M}_t = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, Muon (Jordan et al., 2024) updates the parameters as:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta\,\gamma\,\boldsymbol{U}\boldsymbol{V}^\top, \qquad \gamma = \sqrt{m/n}. \tag{2}$$

In practice, this orthogonal update is computed via an efficient Newton–Schulz iteration rather than an explicit SVD.

Bernstein & Newhouse (2024) interpret equation 2 as steepest descent under the spectral norm. Because the singular values $\mathbf{\Sigma}$ drop out, every retained direction receives identical weight regardless of its original magnitude. Consequently, any direction that survives selection is amplified to the magnitude of the strongest singular component. Within a single step, a discarded direction contributes nothing, whereas a kept one is raised to full strength. This amplification property makes rank selection more consequential in orthogonal optimizers than in methods that preserve relative singular magnitudes.

**Low-rank variants and error feedback.** Full-rank orthogonalization becomes computationally prohibitive as matrices grow large. To alleviate this cost, low-rank variants orthogonalize only $r \ll \min(m, n)$ dominant directions. Let $\boldsymbol{Q}_t \in \mathbb{R}^{m\times r}$ denote an orthonormal basis for the leading column space of $\boldsymbol{M}_t$, and let $\boldsymbol{P}_t = \boldsymbol{M}_t^\top \boldsymbol{Q}_t$. The low-rank update is then formulated as:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta\,\gamma\,\boldsymbol{Q}_t\ \mathrm{colnorm}(\boldsymbol{P}_t)^\top . \tag{3}$$

This formulation substantially reduces computation and distributed communication, although $\boldsymbol{M}_t$ is still stored in full memory. In our experiments, we examine two representative host implementations. Low-rank Muon (He et al., 2026) generates $\boldsymbol{Q}_t$ independently at each step via random projection (Halko et al., 2011); its analysis also covers an adaptive variant that raises the rank until the approximation error falls below a tolerance. In contrast, Dion (Ahn et al., 2025) maintains the basis across steps using power iteration, and retains unapplied updates via error feedback:

$$\boldsymbol{M}_t \ \leftarrow\ \boldsymbol{M}_t \ -\ \alpha\,\boldsymbol{Q}_t[:,\mathcal{S}]\left(\boldsymbol{P}_t[:,\mathcal{S}]\right)^\top, \tag{4}$$

where $\mathcal{S}$ indexes the specific directions updated at the current step.

Under error feedback, unapplied residuals accumulate through equation 1 over subsequent iterations. As a result, weak but persistent directions eventually gain sufficient energy to be selected (Ahn et al., 2025). When the rank $r$ is fixed, the subset $\mathcal{S}$ contains all $r$ columns and remains independent of the instantaneous buffer $\boldsymbol{M}_t$. However, when the rank is chosen adaptively, the selection criterion directly determines $\mathcal{S}$.

Other low-rank methods locate the update subspace through different mechanisms, such as periodically refreshed SVD in GaLore (Zhao et al., 2024), Frobenius energy preservation in AdaRankGrad (Refael et al., 2025), or fine-tuning compression in MLorc (Shen et al., 2026). In distributed settings, error feedback guarantees convergence for contractive compressors (Richtárik et al., 2021) and finite compressor families (Makarenko et al., 2023), and has been extended to compressed optimizer states in LDAdam (Robert et al., 2025). However, these theoretical guarantees often fail when the compressor dynamically depends on the state it compresses (Chen et al., 2026). In Section 5, we show that this state dependence can induce a severe positive feedback loop. Note that parameter-level adaptation like LoRA (Hu et al., 2022) falls outside our scope.

### 2.2 Spectral Thresholds

For an $m \times n$ matrix with independent entries of standard deviation $\sigma$, singular values concentrate below the Marchenko–Pastur edge:

$$\tau_{\mathrm{MP}}\ =\ \sigma\left(\sqrt{m}+\sqrt{n}\right) \tag{5}$$

as dimensions grow at a fixed aspect ratio (Marčenko & Pastur, 1967; Bai & Silverstein, 2010). Sufficiently strong signal directions separate above this boundary (Baik et al., 2005).

Optimal hard thresholds for matrix denoising (Gavish & Donoho, 2014; Donoho et al., 2023) minimize the reconstruction error of a single matrix and assume a noise spectrum with compact support. Where a spiked model is a reasonable description, spectral thresholding has been applied effectively, for instance to separate consensus updates in federated LoRA (Ramesh & Dass, 2026). For optimizers, Li et al. (2026) prove, under a signal-plus-perturbation model of the gradient, that momentum averaging equation 1 suppresses the perturbation and enlarges the spectral gap. In that model a spectral cutoff is a natural rule for adaptive rank control, and its usefulness in practice depends on how much of real momentum the perturbation term accounts for.

**What the discarded spectrum contains.** For gradients, the evidence is mixed. The non-dominant part of the gradient spectrum can carry useful signal (Song et al., 2025; Zhang et al., 2025; Huang et al., 2026), while late in training an SVD subspace of the stochastic gradient can itself become dominated by noise (He et al., 2025). Concurrently, Thomas (2026) find that the top-$r$ gradient subspace of GaLore-type optimizers is reproducible across minibatches only in a small core, and that the gradient spectrum is gapless, with a tail that shrinks under averaging as $N^{-1/4}$ rather than $N^{-1/2}$. Their reproducible rank $k^\star$ plays a role similar to our lower bound. We study the momentum of orthogonalized optimizers with a random-matrix threshold instead, and show that using this measurement for control under error feedback creates a feedback loop. Analyses of momentum in Muon adopt a signal-plus-perturbation model (Li et al., 2026); our measurements quantify, for pretraining momentum, how much of the spectrum below the noise edge that perturbation accounts for.

# 3 The Spectral Diagnostic Instrument

This section turns the noise edge (Equation (5)) into a measurement instrument: the criterion, a corrected noise-floor estimator, and the protocol that aggregates layer-wise measurements.

## 3.1 Form of the noise-edge criterion

Our rank-selection procedure follows a straightforward, three-step pipeline. First, we estimate the background noise magnitude $\hat{\sigma}$ within the current momentum buffer $\boldsymbol{M}_t$. Second, we establish the cutoff threshold by scaling the theoretical noise edge:

$$\tau \;=\; c\,\hat{\sigma}\left(\sqrt{m}+\sqrt{n}\right), \tag{6}$$

where $c > 0$ is an explicit scaling coefficient. Third, we select the subspace of directions whose singular values strictly exceed this threshold:

$$k \;=\; \max\Big(1,\; \big|\{\, i : \sigma_i > \tau \,\}\big|\Big). \tag{7}$$

Here, enforcing a floor of $k \geq 1$ is essential to prevent vanishing updates. Without this floor, the criterion could return $k = 0$ whenever no singular value clears the boundary, which would stall optimization completely.

Setting $c = 1$ aligns equation 6 exactly with the theoretical Marchenko–Pastur edge. This value serves as our default standard for all main experiments. Nevertheless, we retain $c$ as an explicit hyperparameter rather than absorbing it into a constant. Sweeping $c$ allows us to observe how validation loss responds when the threshold shifts across the spectrum (Section E).

## 3.2 Correcting the noise-floor estimator

In equation 6, the matrix dimensions $m$ and $n$ are fixed, so only the noise floor $\hat{\sigma}$ must be estimated from data. A naive approach divides the residual Frobenius energy, $\|\boldsymbol{M}\|_F^2 - \sum_{i\leq q}\sigma_i^2$, evenly among the unobserved directions, since each host observes only the leading $q$ directions (Ahn et al., 2025; He et al., 2026). However, this estimator implicitly assumes a flat noise spectrum. In reality, pure noise matrices concentrate disproportionate energy in their leading singular values according to the Marchenko–Pastur distribution (Marčenko & Pastur, 1967; Bai & Silverstein, 2010). As a result, the naive formula systematically underestimates $\hat{\sigma}$, causing the criterion to retain far too many spurious directions.

This failure mode is particularly insidious because it produces no numerical errors. Instead, its signature is that the selected rank $k$ drifts wildly with the number of observed singular values $q$. For example, on a synthetic rank-5 matrix corrupted by noise, the naive estimator returns values of $k$ ranging from 15 to 358 as $q$ increases. To resolve this instability, we take the larger of two complementary estimators: a residual estimate corrected for the Marchenko–Pastur energy distribution, and a tail-quantile estimate following Gavish & Donoho (2014). Because these two estimators fail in non-overlapping regimes of $q$, taking their maximum guarantees a robust and conservative bound. On the same planted rank-5 problem, this corrected estimator reliably returns $k = 5$ across almost all observation depths. Complete mathematical derivations and algorithmic pseudocode appear in Section C, while robustness to non-Gaussian noise is discussed in Section A.15.

### 3.3 THE OFFLINE DIAGNOSTIC PROTOCOL

Because neural networks contain weight matrices of diverse shapes, we aggregate per-matrix measurements into a single parameter-weighted rank fraction:

$$\bar{\rho} = \frac{\sum_\ell m_\ell n_\ell \cdot \bar{k}_\ell / \min(m_\ell, n_\ell)}{\sum_\ell m_\ell n_\ell}, \tag{8}$$

where $\bar{k}_\ell$ denotes the time-averaged selected rank for weight matrix $\ell$. In practice, the measured rank fraction gradually declines over the course of training. Consequently, evaluating $\bar{\rho}$ over a short initial window yields a slightly higher estimate than measuring over a full run. For instance, in our 60M model, $\bar{\rho}$ averages $0.370$ over the first $800$ steps, compared to $0.353$ over $5000$ steps.

To balance diagnostic fidelity against computational overhead, we adopt an $800$-step profiling window throughout our study. Crucially, this diagnostic procedure is designed to be executed offline. We run the criterion once during early training to identify the structured rank fraction, and then fix that rank for the remainder of the run. In Section 5, we show why allowing this criterion to update adaptively at every step triggers severe training collapse.

## 4 THE SPECTRUM OF PRETRAINING MOMENTUM

We now use this instrument to characterize the momentum spectrum along three axes: spectral shape, batch-size scaling and momentum dynamics. Section 4.1 first states what a noise-edge rule would need to find.

### 4.1 WHAT A NOISE-EDGE RULE REQUIRES

A noise-edge rule works best under a spiked model (Baik et al., 2005), $\boldsymbol{M}_t = \boldsymbol{M}_{\text{signal}} + \boldsymbol{M}_{\text{noise}}$, with a low-rank signal and independent noise entries. Discarding singular values below $\tau_{\text{MP}}$ would then remove mostly sampling noise, and validation loss should level off once the rank covers the super-threshold components. The three probes below measure how closely pretraining momentum matches this picture.

### 4.2 EMPIRICAL SPECTRAL PROBES

To determine whether sub-threshold singular values represent uncorrelated sampling noise or persistent components, we design three diagnostic tests.

**Probe 1: Spectral shape.** Pure random noise matrices exhibit compact spectral support (Bai & Silverstein, 2010), which yields a theoretical peak-to-median singular value ratio of approximately $1.6$. To test this property in practice, we examine $464$ empirical momentum spectra from attention projections across $48$ training runs. As illustrated in Figure 1, the empirical momentum spectrum shows neither compact support nor a visible spectral gap. Instead, it follows an unbroken power law, $\sigma_i \propto i^{-\gamma}$, across its entire middle range, with a median exponent of $\gamma = 0.75$ ($R^2 \geq 0.91$). Moreover, the ratio of largest to median singular values has a median of $35$, with $80\%$ of spectra between $20$ and $101$, an order of magnitude above the random-noise prediction. The spectrum is therefore a smooth continuum with no identifiable boundary between signal and noise.

**Probe 2: Variance decomposition.** Because stochastic sampling noise scales inversely with batch size (McCandlish et al., 2018), a fourfold reduction in batch size should double the standard deviation of the sampling component. We leverage this scaling law to decompose our estimated noise floor into persistent structure $C$ and sampling noise $S$, writing $\hat{\sigma}^2 = C + S$. Under a fourfold batch reduction, the ratio of estimated noise floors becomes:

$$\left(\frac{\hat{\sigma}_{\text{small batch}}}{\hat{\sigma}_{\text{large batch}}}\right)^2 = \frac{C + 4S}{C + S} = 1 + 3f, \qquad f \coloneqq \frac{S}{C + S}. \tag{9}$$

The measured variance ratio is $1.119$ at 60M and $1.056$ at 350M, which corresponds to noise fractions of $f = 0.084 \pm 0.020$ and $f = 0.038 \pm 0.019$, respectively. Thus, **stochastic sampling noise accounts for under $11\%$ of the estimated noise floor at 60M and under $6\%$ at 350M**. In other words, over $89\%$ of the energy below the threshold is not sampling noise.

**Probe 3: Momentum suppression dynamics.** Exponential moving averages, $\boldsymbol{M}_t = \mu \boldsymbol{M}_{t-1} + \boldsymbol{G}_t$, attenuate temporally uncorrelated noise relative to persistent signal by a factor of $(1-\mu)/(1+\mu)$ in variance, a standard property of exponential averaging; while preserving persistent signals. We validate this physical model by varying $\mu$ across a thirteen-fold range. The measured noise share $f$ scales linearly with $(1-\mu)/(1+\mu)$, with regression residuals remaining below $0.03$ (Section D). This linear agreement confirms that our decomposition accurately separates stochastic fluctuations from persistent momentum components.

**Implications for rank truncation.** Together, these three probes show that the sub-threshold spectrum is not sampling noise but weak, diffuse components that persist across steps. Combined with the monotonic loss curves of Section 5.1, this indicates that rank truncation discards information that matters for optimization: the theoretical noise edge identifies only a lower bound on structured rank, rather than an optimal truncation threshold.

## 5 Training Dynamics and Feedback Collapse

Having shown that momentum spectra lack a gap at the noise edge, we now examine its consequences during actual model training. First, we establish baseline performance under fixed ranks, showing that validation loss improves monotonically as rank increases. Second, we investigate online adaptive rank control, revealing that coupling spectral selection with error feedback triggers a severe instability, which we formalize and resolve through a pool-wide subtraction fix.

### 5.1 Baselines and Strict Monotonicity

We train GPT-style language models (Radford et al., 2019) at 60M, 124M and 350M parameters on FineWeb (Penedo et al., 2024), from scratch for $5000$ steps at $294{,}912$ tokens per step, with AdamW (Loshchilov & Hutter, 2019) for embeddings and one-dimensional parameters (Jordan et al., 2024). We compare full-rank Muon against fixed ranks $r/\min(m,n) \in \{1/16, 1/4, 1/2\}$ in Low-rank Muon (He et al., 2026) and Dion (Ahn et al., 2025); within a host optimizer, only the rank selection rule differs. Every configuration is run with three seeds, with standard deviations between $0.0005$ and $0.0061$ nats; widths, matrix counts and further details are in Section A.1.

**Instrument calibration.** In the baseline host without error feedback, the adaptive criterion selects rank fractions of $0.353$, $0.357$, and $0.361$ across the three scales. As shown in Figure 2, the resulting validation losses fall within $0.005$–$0.008$ nats of the interpolated fixed-rank baseline, confirming that our instrument accurately identifies a viable operating point on the trade-off curve.

**Strict monotonicity.** The primary empirical outcome of this benchmark is that **validation loss improves strictly monotonically with rank across every setting** (Table 1). In none of the six configurations does an interior optimal rank exist. Extending Dion to full rank continues the progression

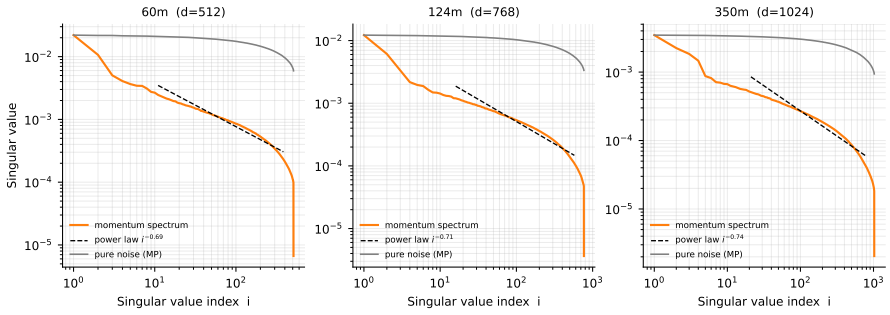



Figure 1: Singular value spectrum of momentum versus an equivalent pure noise matrix (log–log scale). Momentum singular values follow a continuous power law with no spectral gap at the noise edge.

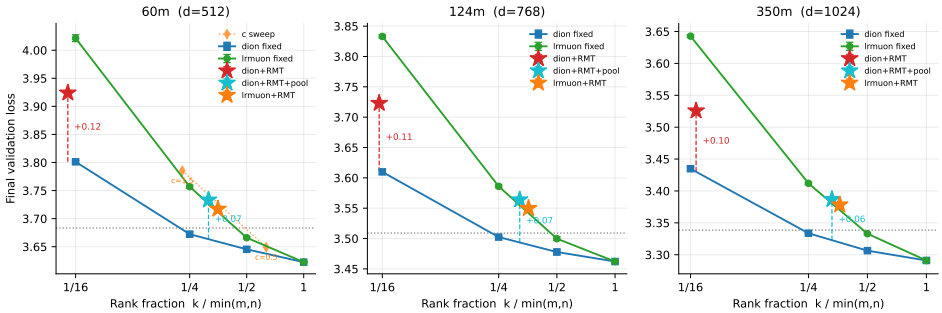



Figure 2: Validation loss versus rank fraction across three model scales. Solid lines interpolate fixed-rank baselines; stars mark the adaptive spectral criterion placed at its selected average rank fraction. Without error feedback, the adaptive measurement matches the baseline curve. Under error feedback, it degrades severely.

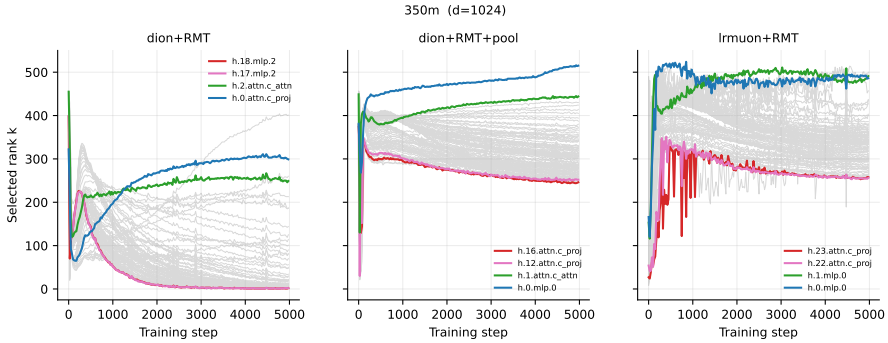



Figure 3: Trajectory of selected rank across training steps for all matrices in Dion under dynamic rank selection. Most layers progressively collapse toward rank one.

and matches full-rank Muon within seed variance. Together with the measurements of Section 4, this indicates that the tail is not disposable: it is not sampling noise, and discarding it costs loss.

Table 1: Final validation loss across rank fractions $r/\min(m,n)$. Every row decreases monotonically.

| Scale | Host | 1/16 | 1/4 | 1/2 | 1 | Muon (Full) |
|---|---|---|---|---|---|---|
| 60M | Dion | 3.8012 | 3.6724 | 3.6453 | 3.6357 | 3.6225 |
| 60M | Low-rank Muon | 4.0214 | 3.7569 | 3.6661 | — | 3.6225 |
| 124M | Dion | 3.6098 | 3.5024 | 3.4778 | 3.4683 | 3.4621 |
| 124M | Low-rank Muon | 3.8330 | 3.5860 | 3.4998 | — | 3.4621 |
| 350M | Dion | 3.4348 | 3.3337 | 3.3064 | — | 3.2910 |
| 350M | Low-rank Muon | 3.6427 | 3.4120 | 3.3330 | — | 3.2910 |

### 5.2 Online Feedback Collapse

While adaptive selection operates stably in Low-rank Muon (where selected rank fraction remains steady near 0.35, drifting by only $-12.8\%$ across a full run), coupling it with error feedback in Dion triggers a severe failure mode. In Dion, adaptive rank selection collapses: the selected rank fraction drops to 0.057–0.067, with 83–88% **of all weight matrices collapsing toward rank one** (Figure 3). This collapse incurs an optimization penalty of 0.096–0.123 nats compared to fixed-rank baselines evaluated at the same average rank.

**The positive feedback mechanism.** This instability is not caused by stochastic fluctuations. Instead, it is driven by a four-stage positive feedback loop between spectral thresholding and residual retention:

1. **Selection:** At step $t$, the spectral criterion identifies a subset of directions $\mathcal{S}$ whose singular values exceed the threshold $\tau$.
2. **Selective Subtraction:** Error feedback (Ahn et al., 2025) subtracts only these applied directions from the buffer: $\boldsymbol{M}_t \leftarrow \boldsymbol{M}_t - \alpha \boldsymbol{Q}_t[:, \mathcal{S}](\boldsymbol{P}_t[:, \mathcal{S}])^\top$.
3. **Accumulation:** Unselected directions outside $\mathcal{S}$ remain in the buffer, where they accumulate energy across subsequent iterations via momentum.
4. **Threshold Inflation:** This accumulated energy inflates the estimated noise floor $\hat{\sigma}$, which raises the threshold $\tau$ and forces the criterion to select an even smaller subset $\mathcal{S}$ at the next step.

Under fixed-rank optimization, the subset $\mathcal{S}$ is predefined and independent of the buffer state, so the feedback loop cannot close. Under adaptive control, however, the rank decision directly alters future buffer states, locking the optimizer into a self-reinforcing cycle.

**Open-loop spectral flattening.** We verify this mechanism in open-loop experiments. Lowering the fixed rank leaves more unapplied momentum in the buffer, which monotonically flattens the singular value spectrum. Specifically, the fitted power-law exponent drops from $0.75$ to $0.30$, while the peak-to-median ratio falls from $35$ to $7$. At a fixed rank, this flattened spectrum remains stable. In a closed loop, however, the adaptive selector responds to this flattening by shrinking $\mathcal{S}$ further, precipitating rank collapse.

### 5.3 Resolution via Pool-Wide Subtraction

If Feedback Collapse is caused by retaining unselected candidate directions, subtracting energy across the entire candidate subspace should break the loop. To verify this hypothesis, we introduce a direct causal intervention: we modify the error-feedback step to subtract across the **entire observed candidate pool**, rather than only the selected columns:

$$\boldsymbol{M}_t \leftarrow \boldsymbol{M}_t - \alpha \boldsymbol{Q}_t \boldsymbol{P}_t^\top. \tag{10}$$

The rank selection criterion, the weight update rule, and all other hyperparameters remain unchanged. This is the step that fixed-rank Dion applies to all of its tracked directions (Ahn et al., 2025).

This single-line modification prevents the collapse: the proportion of degenerate rank-one matrices drops from $83$–$88\%$ to $0\%$ **across all three scales**. The selected rank fraction stabilizes at $0.31$–$0.33$, the estimated noise floor $\hat{\sigma}$ saturates normally, and validation loss recovers by $0.139$–$0.191$ nats. Figure 4 shows that the intervention acts on the mediating quantity itself, not only on the selected rank. Because only one line changes within one host, this provides direct causal evidence that Feedback Collapse is an instability of closed-loop selection rather than an artifact of the spectral estimator itself.

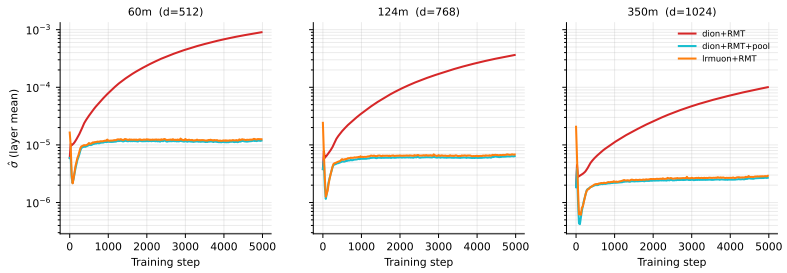



Figure 4: Noise-floor estimate $\hat{\sigma}$ over training, averaged over matrices, one panel per scale. Under selected-EF it grows without settling; under pool-wide subtraction (Equation (10)) it saturates and coincides with the host without error feedback, at all three scales.

## 6 DISCUSSION

**Denoising versus lossy compression.** Momentum and weight updates have been reported to be approximately low-rank in fine-tuning (Hu et al., 2022; Shen et al., 2026). In pretraining, the momentum spectrum is instead gapless, and most of it below the noise edge is not sampling noise (Section 4). Weight matrices of modern networks likewise lack a scale separating signal from noise (Martin & Mahoney, 2021), and our measurements extend this picture to the optimizer state. Unlike in matrix denoising, discarding the tail costs loss (Section 5.1).

**Limits of error feedback.** Convergence guarantees for error feedback assume a compression operator that is independent of the state or has a uniform contraction factor (Richtárik et al., 2021; Makarenko et al., 2023). A rank chosen from the current spectrum satisfies neither, and pool subtraction (Equation (10)) locates the resulting instability in the selectively retained residual. Two further conditions are needed for the loop to form. The threshold must scale with the magnitude of the spectral tail: a criterion that fixes a cumulative energy share selects more directions as the spectrum flattens, and did not collapse in our 60M runs (three seeds; Section H). The structured rank fraction must also be declining, as in language models ($-12.8\%$ over a run) but not in vision transformers ($+1.2\%$), where the loop does not form (Section G).

**Recommendations and limitations.** Adaptive rank selection inside an error-feedback loop is best avoided: besides the risk of collapse, computing the spectrum at every step costs 20 to $65\times$ more than a full-rank update (Section F). Offline, an $800$-step diagnostic gives a stable estimate that can be fixed for the full run, at about $2.2\times$ lower total cost than a three-level grid (Section A.3). Because the measured fraction differs across modalities ($0.35$ against $0.45$; Section G) and across depth ($0.06$ to $0.53$; Section A.13), it should be measured rather than assumed. *Limitations:* We study transformers up to 350M parameters on a single GPU, with spectra taken from attention projections. The noise-floor estimator assumes a compact noise bulk and overestimates rank under strong row correlation, a bias that is conservative for our conclusions (full details in Section A.15).

## 7 CONCLUSION

We studied whether the momentum spectrum provides a principled basis for rank selection in low-rank orthogonal optimization. Across transformer language-model pretraining at three scales, momentum spectra are approximately gapless and power-law, and most of the sub-threshold component cannot be attributed to sampling noise. Although our spectral diagnostic yields a stable estimate of structured rank, validation loss continues to improve beyond that point. The noise edge therefore provides a lower bound on structured rank rather than an optimization-optimal cutoff in the settings we study.

We further show that spectral measurement and adaptive control need not behave equivalently. When online rank selection is coupled with selective error feedback, the decision changes the momentum state on which subsequent decisions are based, creating a self-reinforcing Feedback Collapse in our language-model experiments. Pool-wide subtraction breaks this loop across all three scales, isolating the instability to the interaction between adaptive selection and residual retention. The collapse is not universal: our ViT experiments indicate that its formation also depends on the training dynamics, in particular whether the structured-rank fraction declines over time.

Overall, our results support using spectral rank estimation primarily as an offline diagnostic, while treating low-rank orthogonal optimization as a cost–quality trade-off rather than a denoising problem. This perspective suggests that future adaptive rank methods should account for the marginal optimization value of additional directions, rather than rely on a presumed sharp separation between signal and noise.

### ETHICS STATEMENT

All authors have read and adhere to the ICLR Code of Ethics. This work studies optimization algorithms using publicly available datasets (FineWeb and a subset of ImageNet) under their research licenses, and involves no human subjects or personally identifiable data. The authors take full responsibility for the content of this paper, including any part prepared with the assistance of AI tools.

### AI USE STATEMENT

- **Text Drafting and Editing:** Generative AI tools (e.g., Claude, ChatGPT) were used to draft initial sections of the manuscript and polish the text for readability. All text was critically reviewed, edited, and revised by the authors.
- **Ideation, Methodology, and Code:** AI tools were used for brainstorming, discussing experimental methodology, and assisting with writing and debugging code for experiments and data analysis. All code was verified and tested for correctness by the authors.
- **Literature Discovery:** AI tools assisted in searching and identifying related literature. Every cited paper and reference was manually verified against original sources.
- **Scope Limitations and Responsibility:** We did not use generative AI to formulate or prove mathematical claims, nor to generate synthetic training datasets. All experiments were conducted by the authors, all reported numbers derive directly from our experimental logs, and the authors take full responsibility for the entire content of this work.

### REPRODUCIBILITY STATEMENT

The models, data, and training configuration are described in Section 5.1; the full per-configuration results with seed counts are in Section H. The noise floor estimator is specified with pseudocode in Section C, and the planted-rank test in Section C.2 allows any reimplementation to be checked. The single-GPU form of Dion is given in Section B. The training framework, a single script that reproduces every run by stage, and the analysis scripts that produce all tables and figures are included in the supplementary material as a zip archive.

## REFERENCES

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

Zhidong Bai and Jack W. Silverstein. *Spectral Analysis of Large Dimensional Random Matrices*. Springer, second edition, 2010. doi: 10.1007/978-1-4419-0661-8.

Jinho Baik, Gérard Ben Arous, and Sandrine Péché. Phase transition of the largest eigenvalue for nonnull complex sample covariance matrices. *The Annals of Probability*, 33(5):1643–1697, 2005. doi: 10.1214/009117905000000233.

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

Chuyan Chen, Yutong He, Pengrui Li, Weichen Jia, and Kun Yuan. Greedy low-rank gradient compression for distributed learning with convergence guarantees. *IEEE Transactions on Signal Processing*, 74:340–354, 2026. doi: 10.1109/TSP.2026.3541178.

David L. Donoho, Matan Gavish, and Elad Romanov. ScreeNOT: Exact MSE-optimal singular value thresholding in correlated noise. *The Annals of Statistics*, 51(1):122–148, 2023. doi: 10.1214/22-AOS2232.

Matan Gavish and David L. Donoho. The optimal hard threshold for singular values is $4/\sqrt{3}$. *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014. doi: 10.1109/TIT.2014.2323359.

Guy Gur-Ari, Daniel A. Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Nathan Halko, Per-Gunnar Martinsson, and Joel A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53 (2):217–288, 2011. doi: 10.1137/090771806.

Chuan He, Zhanwang Deng, and Zhaosong Lu. Low-rank orthogonalization for large-scale matrix optimization with applications to foundation model training. *Transactions on Machine Learning Research*, 2026.

Yutong He, Pengrui Li, Yipeng Hu, Chuyan Chen, and Kun Yuan. Subspace optimization for large language models with convergence guarantees. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22468–22522, 2025.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Zhendong Huang, Hengjie Cao, Fang Dong, Ruijun Huang, Mengyi Chen, Yifeng Yang, Xin Zhang, Anrui Chen, Mingzhi Dong, Yujiang Wang, Jinlong Hou, Qin Lv, Robert P. Dick, Yuan Cheng, Tun Lu, Yang Fan, and Li Shang. Spectra: Rethinking optimizers for LLMs under spectral anisotropy. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

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.

Xianliang Li, Zihan Zhang, Weiyang Liu, and Han Bao. Denoise first, orthogonalize later: Understanding momentum in Muon via spectral filtering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2026.

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

Maksim Makarenko, Elnur Gasanov, Abdurakhmon Sadiev, Rustem Islamov, and Peter Richtárik. Adaptive compression for communication-efficient distributed training. *Transactions on Machine Learning Research*, 2023.

Vladimir A. Marčenko and Leonid A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967. doi: 10.1070/SM1967v001n04ABEH001994.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165):1–73, 2021.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 30811–30849, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019.

Hariharan Ramesh and Jyotikrishna Dass. SpecTraL: Spectral transformation for layer-wise global rank discovery in federated LoRA for vision transformers. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, 2026.

Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. AdaRankGrad: Adaptive gradient-rank and moments for memory-efficient LLMs training and fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2025.

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

Thomas Robert, Mher Safaryan, Ionut-Vlad Modoranu, and Dan Alistarh. LDAdam: Adaptive optimization from low-dimensional gradient statistics. In *International Conference on Learning Representations (ICLR)*, 2025.

Wei Shen, Yaxiang Zhang, Minhui Huang, Mengfan Xu, Jiawei Zhang, and Cong Shen. MLorc: Momentum low-rank compression for memory efficient large language model adaptation. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 300 of *Proceedings of Machine Learning Research*, pp. 1891–1899, 2026.

Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? In *International Conference on Learning Representations (ICLR)*, 2025.

Noel Thomas. No subspace to track: Non-identifiability and optimizer state in low-rank training. *arXiv preprint arXiv:2607.05872*, 2026.

Haochen Zhang, Junze Yin, Guanchu Wang, Zirui Liu, Lin F. Yang, Tianyi Zhang, Anshumali Shrivastava, and Vladimir Braverman. Breaking the frozen subspace: Importance sampling for low-rank optimization in LLM pretraining. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, pp. 11276–11328, 2025.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61502–61524, 2024.

# A Additional details and results

## A.1 Experimental setup details

We train GPT-style language models (Radford et al., 2019) at three scales (Table 2) on FineWeb (Penedo et al., 2024), sequence length $1024$ tokens, $294{,}912$ tokens per step, for $5000$ steps. Embeddings and one-dimensional parameters use AdamW (Loshchilov & Hutter, 2019) following the convention of Jordan et al. (2024). All experiments run on a single GPU, so the communication axis can only be modeled (Section F.4).

Table 2: The three model scales. The last column is the number of weight matrices that pass through the orthogonalized optimizer.

| Scale | Parameters | Blocks | Width $d$ | Heads | Matrices |
|---|---|---|---|---|---|
| 60M | 57.8M | 10 | 512 | 8 | 40 |
| 124M | 124.5M | 12 | 768 | 12 | 48 |
| 350M | 354.9M | 24 | 1024 | 16 | 96 |

At each scale we run twelve configurations: a full-rank reference; a fixed-rank grid $r/\min(m,n) \in \{1/16,\ 1/4,\ 1/2\}$ in both host optimizers; the adaptive-rank branch in both, plus one variant introduced in Section 5.3; and AdamW and GaLore (Zhao et al., 2024) as references outside the orthogonalized family. The fixed-rank grid is indispensable: judging whether an adaptive criterion is good requires knowing what the loss–rank curve looks like, and a single point says nothing.

Within each host optimizer, the only variable between the fixed-rank and adaptive branches is the rank selection rule; the normalization coefficient, learning rate schedule, embedding handling and data order are held identical. The learning rate is swept separately for each optimizer before the main grid, and the best value coincides between the two branches within a host. This has a consequence for interpretation that we use again in Section 5.2: comparisons *within a single host* are controlled, whereas comparisons *between hosts* are only contrasts, since the hosts differ in details beyond rank selection.

Beyond loss, we log the selected rank $k$, the noise floor estimate $\hat{\sigma}$ and the threshold $\tau$ separately for each matrix at each step, along with the full singular value spectrum periodically. The reason for logging more than usual is that the map of rank across layers and over time is the primary data of this paper, while loss is a consequence and arrives too late — the phenomenon in Section 5.2 shows up in those quantities about a thousand steps before it shows up in the loss.

Table 3: Distance between the adaptive branch and the interpolated fixed-rank curve, evaluated at exactly the rank fraction the criterion selects. Positive means above the curve, i.e., worse than the corresponding fixed rank.

| Scale | Host | Mode | Rank fraction | Loss | Distance |
|---|---|---|---|---|---|
| 60M | Low-rank Muon | — | 0.353 | 3.7171 | +0.005 |
| 124M | Low-rank Muon | — | 0.357 | 3.5498 | +0.008 |
| 350M | Low-rank Muon | — | 0.361 | 3.3783 | +0.008 |
| 60M | Dion | pool-EF | 0.314 | 3.7334 | +0.070 |
| 124M | Dion | pool-EF | 0.321 | 3.5639 | +0.071 |
| 350M | Dion | pool-EF | 0.330 | 3.3868 | +0.064 |
| 60M | Dion | selected-EF | 0.057 | 3.9240 | +0.123 |
| 124M | Dion | selected-EF | 0.060 | 3.7230 | +0.113 |
| 350M | Dion | selected-EF | 0.067 | 3.5257 | +0.096 |

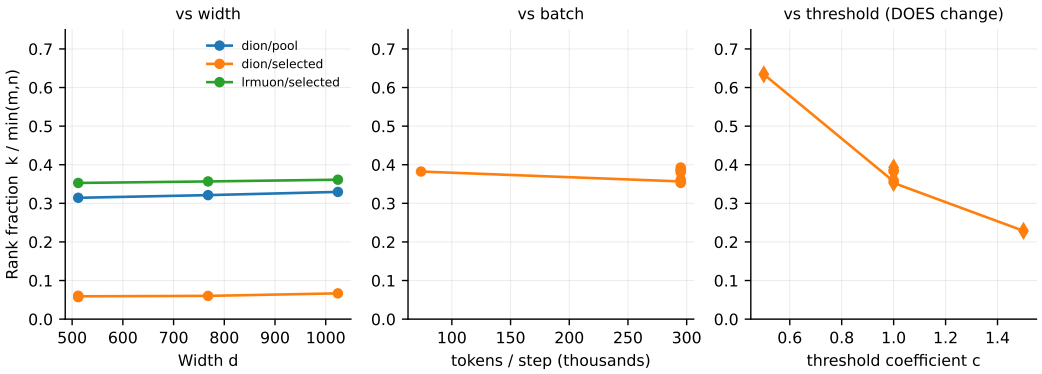



Figure 5: Selected rank fraction measured along three independent axes of variation: model width, batch size, and threshold coefficient $c$. The quantity is nearly invariant along the first two, and changes only when the threshold itself is moved.

Every configuration at all three scales is run three times from different initializations. For most configurations the standard deviation lies between $0.0005$ and $0.0061$ nats with no trend across scale; the Dion branch at full rank is an exception at $0.0125$ nats, twice the rest. We use the standard deviation of the *specific configuration being compared* as the yardstick rather than a single pooled value.

Full results for every configuration at all three scales are in Section H, Table 10.

We record four hypotheses that we registered and subsequently had to reject, listed in Section A.15, since a procedure that reports only its correct predictions is indistinguishable from telling the story after the fact.

### A.2 Stability of the selected rank fraction

A quantity that swings with training conditions measures nothing. We check along three independent axes of variation, summarized in Figure 5.

Across model scale, the selected rank fraction is nearly constant: $0.353$, $0.357$ and $0.361$ as parameter count grows sixfold. Across batch size, when compared at equal tokens seen, the two batch regimes trace the same trajectory. This stability does not make $0.35$ a universal constant: the same measurement on ViT and ImageNet-100 gives $0.43$–$0.49$, and there it varies with width about seven times more (Section G). Over the course of training the fraction declines rather than staying fixed — and this is something a constant cannot follow, so a fixed rank is necessarily wrong at some stage of training.

Table 4: Cost of finding a reasonable rank, in GPU-hours. The middle column is the search step alone; the right column adds the final training run. The grid uses $r/\min(m,n) \in \{1/16, 1/4, 1/2\}$ on the corresponding host.

| Scale | Approach | Find rank | Total | Ratio |
|---|---|---|---|---|
| 60M | Three-level grid sweep | 4.90 | 4.90 | — |
| 60M | Diagnostic, then fix | 0.38 | 2.32 | $2.1\times$ |
| 350M | Three-level grid sweep | 19.41 | 19.41 | — |
| 350M | Diagnostic, then fix | 1.97 | 8.73 | $2.2\times$ |

### A.3 Cost of finding a rank: details

The two subsections above show the measurement is accurate and stable. This one quantifies the resulting saving: the cost of *finding* a reasonable rank.

The comparison is set up as follows. The usual approach sweeps a grid of fixed ranks, three levels in our experiments, with each level requiring a full training run, and then picks the best. The alternative runs one $800$-step diagnostic per Section 3.3, reads $\bar{\rho}$, fixes the rank, and trains once. Table 4 gives the numbers at two scales.

Two numbers must be distinguished. Counting only the *search* step, the diagnostic is about an order of magnitude cheaper: $13\times$ at 60M and $9.8\times$ at 350M. But after fixing the rank one still has to train, so in terms of the total cost of obtaining a trained model at a reasonable rank the ratio comes down to $2.1\times$ at 60M and $2.2\times$ at 350M. We report the second number as the headline, since that is what a user actually pays.

The ratio is stable between the two scales but is not invariant. It depends on how many levels the grid has, since sweeping five instead of three makes the grid more expensive while leaving the diagnostic unchanged. It also depends on the ratio between diagnostic cost and the cost of one full run. Here the diagnostic covers $800/5000 = 16\%$ of the steps but uses the more expensive criterion, so its actual cost exceeds that fraction.

Finally, this analysis concerns the cost of *finding* the rank. The cost of *running* is a separate matter: on a single GPU across the scale range we examine, full orthogonalization is cheaper than every low-rank variant, and the compute advantage of low-rank appears only at considerably larger matrix sizes. Details in Section F.

### A.4 Sensitivity to the threshold coefficient

If the curve has no break, changing $c$ in equation 6 merely moves the operating point along it rather than finding a better one. We test this by sweeping $c$ in the host without error feedback; see Table 8.

The measured elasticity $\mathrm{d}\log k/\mathrm{d}\log c$ is $-0.92$. This is consistent with the spectral shape measured in Section 4: on a power-law spectrum, the number of directions above the threshold responds smoothly to threshold position, with no jump. Two independent measurements give the same conclusion about shape.

The practical consequence: choosing $c$ becomes a hyperparameter to sweep, in place of choosing $r$. The original goal of removing a hyperparameter is not achieved; the hyperparameter is merely renamed.

### A.5 Divergence between denoising and optimization

There is a compact way to state this entire section, using two numbers.

For matrix denoising, the threshold of Gavish & Donoho (2014) and equation 5 place the optimal threshold *above* the noise edge, corresponding to $c \approx 1.155$ in the parameterization of equation 6. For training, Table 8 places the best value in the swept range *below* the noise edge, at $c = 0.5$, and still improving as $c$ decreases further.

The two problems pull in opposite directions from the same reference point. This is a quantitative statement, not a qualitative remark: optimal denoising and optimization are not the same task, and the noise edge is where the two objectives part ways.

### A.6 Separation under the diagnostic indicator

The counts in Section 5 use a per-matrix indicator: the normalized drift of $\hat{\sigma}$ over the $40$–$80\%$ window of training, with a matrix flagged when the drift exceeds $0.5$. Figure 6 shows that it separates the two populations cleanly. Matrices in the selected-EF branch that collapse show sustained growth of $\hat{\sigma}$, while no matrix in the pool-EF branch or in the host without error feedback is flagged. The gap between the two groups narrows with scale, from $0.61$ at 60M to $0.29$ at 124M and $0.07$ at 350M, and the number of matrices falling inside it grows from $0$ to $1$ and $6$; we therefore report counts at two thresholds rather than one. Because $\hat{\sigma}$ is already computed by the criterion, the indicator costs nothing extra, and it moves about a thousand steps before the loss does, which is why we propose it as an early warning in Section A.13.

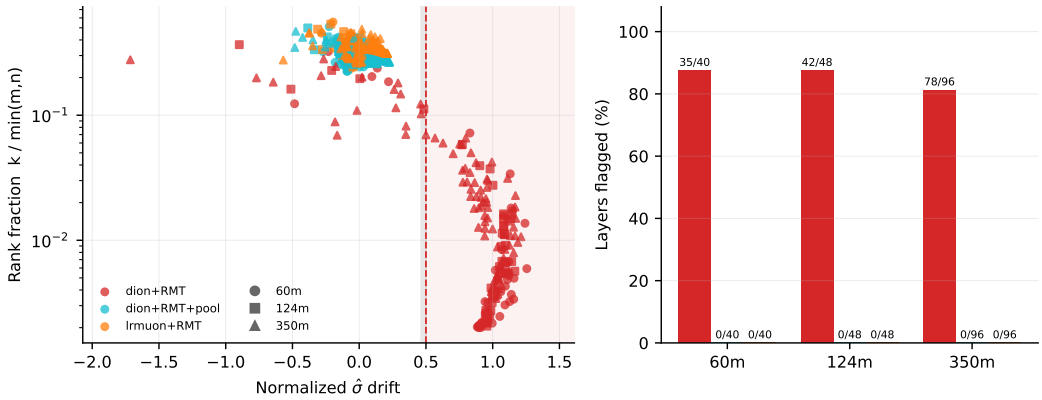



Figure 6: Left: each point is a weight matrix, placed by the drift of $\hat{\sigma}$ during mid-training and by its average selected rank fraction; marker shape distinguishes scale. The grey band is the gap between the two groups, drawn for the scale with the narrowest gap — any threshold inside it yields the same classification. Right: fraction of matrices flagged, grouped by scale. The figure shows one representative initialization; Table 3 and the text use the mean over three.

### A.7 Layer-wise map of the collapse

The collapse is not uniform across the network. Figure 7 plots the selected rank fraction of each matrix in the selected-EF branch against its block index. The rank fraction declines smoothly with depth rather than splitting into two sharply separated groups: matrices in shallow blocks retain the highest rank, attention matrices retain more than MLP matrices, and the MLP output projection is lowest throughout. At 350M the rank fraction rises again in the last few blocks. We do not have an explanation for this depth dependence; it is listed among the open questions in Section A.15.

### A.8 Tracing the mechanism

We present this in the order it happened, because each step ruled out a competing explanation.

The first sign was that the noise floor estimate $\hat{\sigma}$ grew without settling, whereas in the host without error feedback it saturates (Figure 4). Since both branches run on the same data with the same learning rate schedule, this rules out the hypothesis that gradient noise naturally grows over training.

The second step was to separate by matrix. Not all decline: some matrices hold their rank or increase it. This rules out a global effect and shows the system has two stable states.

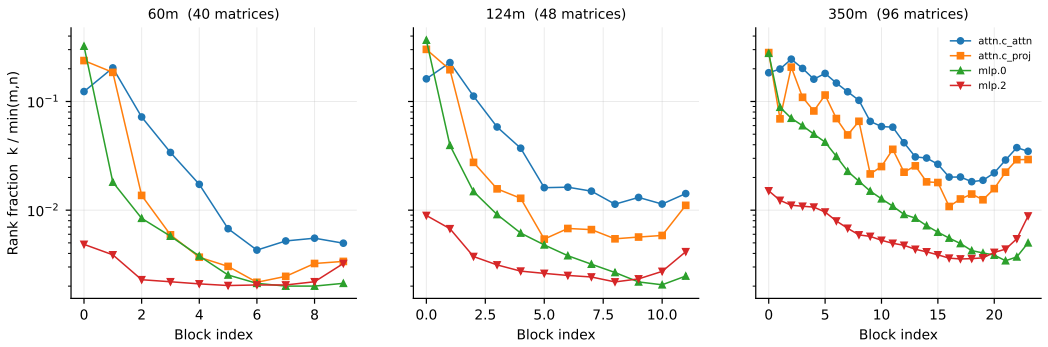



Figure 7: Selected rank fraction by matrix position in the network, at three scales. The horizontal axis is block index; the four markers correspond to the four matrix types in a transformer block. The group that retains high rank sits in the shallow blocks and favors attention matrices.

The third step was to look at the spectrum over time. In the affected matrices, the spectrum gradually loses its original shape and approaches that of pure noise. This shows the problem lies not in the counting but in what is being counted.

### A.9 Residual gap after the correction

After the fix, the adaptive point still sits above the fixed-rank curve by $+0.070$, $+0.071$ and $+0.064$ nats at the three scales. This figure is stable across scales, and nearly an order of magnitude larger than the $0.005$–$0.008$ measured in the host without error feedback (Table 3).

We offer a hypothesis, stated as such and not yet tested: the selected direction set still changes from step to step, so the backlog keeps shifting even though it no longer accumulates in one direction.

The conclusion: adaptive rank selection inside an optimizer with error feedback is never entirely free, even when the right thing is fixed.

### A.10 End state of the loop

Once the loop runs to completion, the spectrum of affected matrices resembles that of pure noise: the ratio between the largest and the median singular value falls from about 37 to $1.7$, close to the pure-noise value of about $1.6$ noted in Section 4. Figure 8 traces this process.

By that point the criterion is correct: there genuinely is nothing left worth keeping in that matrix. It becomes accurate exactly when it has destroyed what it was measuring. One point on scope: this feedback loop does not form in every setting. On ViT and ImageNet-100, the same criterion with the same error feedback produces only a near-constant offset rather than progressive tightening (Section G); the condition separating the two cases is whether the rank fraction is declining over time.

### A.11 Measurement versus control

A quantity that is measured accurately is not automatically a quantity that should be used for control. The two roles have different requirements. For measurement, it suffices to reflect the state correctly; the instrument sits outside the system. For control, the decision changes the system, so one must also ensure the feedback loop does not amplify itself.

The result of this paper is that the criterion meets the first requirement but not the second, and the results give two different reasons. Sections 4 and 5.1 show that it measures correctly, but that the noise edge it measures is not what should be optimized. Section 5.2 shows that when made to control a system with feedback, it destroys its own measurement.

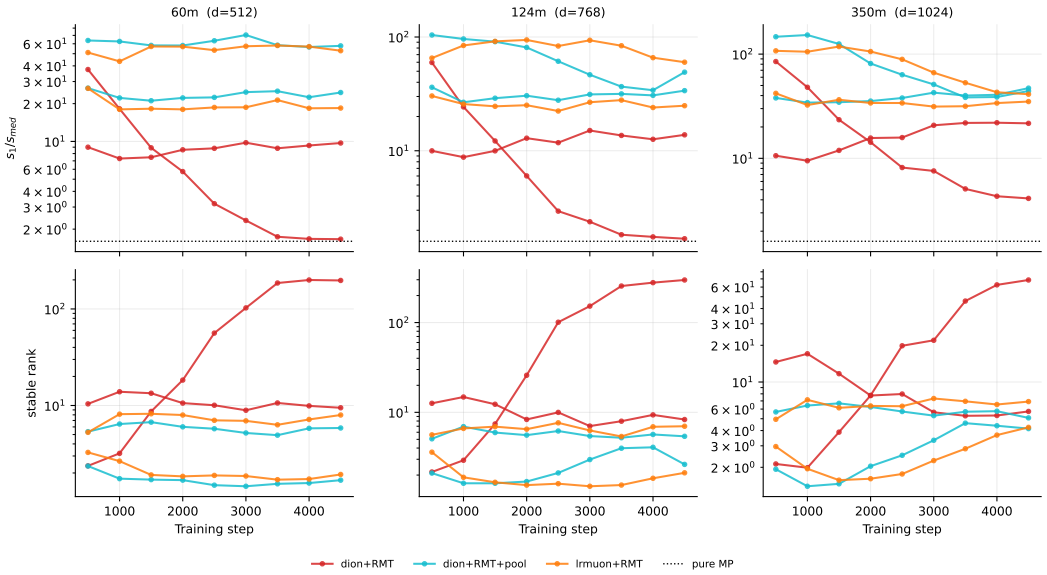



Figure 8: Spectral deformation over training. Top row: ratio between the largest and the median singular value, with the dotted line marking the pure noise value. Bottom row: stable rank. One column per scale. In the selected-EF branch both quantities approach the pure noise values at all three scales; the other two branches retain their spectral shape.

Table 5: Loss difference between Dion and Low-rank Muon in each configuration. Negative means error feedback helps. The sign flips at the adaptive branch, and returns near 0 once equation 10 is applied.

| Scale | $r = 1/16$ | $r = 1/4$ | $r = 1/2$ | adaptive | adaptive + equation 10 |
|---|---|---|---|---|---|
| 60M | $-0.220$ | $-0.085$ | $-0.021$ | $+0.207$ | $+0.016$ |
| 124M | $-0.223$ | $-0.084$ | $-0.022$ | $+0.173$ | $+0.014$ |
| 350M | $-0.208$ | $-0.078$ | $-0.027$ | $+0.147$ | $+0.008$ |

### A.12 Opposite effects of the same mechanism

Table 5 is the most compact statement of this subsection. The first columns give the loss difference between the two hosts at each fixed rank; the last columns give it for the adaptive branch.

Error feedback helps substantially at low fixed ranks, the benefit shrinks as rank grows, and it turns into a liability in the adaptive branch. All three scales show the same shape, with the first three columns nearly constant across scale. The fourth column declines ($+0.207$, $+0.173$, $+0.147$), so the damage from the loop shrinks as models grow; the last column does too. Put briefly: what makes low ranks work is what breaks adaptive rank selection.

### A.13 Appropriate uses

**Diagnosis instead of grid search.** Run a short diagnostic pass to measure the required rank fraction, then fix the rank for the main run. Section A.3 quantifies the gain: total cost drops by about $2.2\times$ relative to a three-level sweep, and the rank-finding step alone is nearly ten times cheaper.

The validity of this procedure rests on two earlier findings: Section 5.1 shows the measurement is accurate, and Section A.2 shows it is stable. Without both, the measured number could not be used to infer what rank to set.

**Allocating a rank budget across layers.** When hardware allows only a fixed total rank, the question is no longer how much but how to divide it. The measured rank fraction ranges from about $0.06$ to

$0.53$ depending on the layer (Figure 7), so dividing evenly means wasting budget in one place while starving another. This is information a fixed rank does not provide.

**An early warning indicator.** The drift of $\hat{\sigma}$ is computed from quantities the optimizer already maintains, at no extra cost. On our data (Figure 6) this indicator classifies more than $1600$ layer–model pairs correctly, and detects the phenomenon of Section 5.2 about a thousand steps before the loss begins to diverge.

One limitation should be stated: at the run level, classification is unambiguous at all three scales; but at the level of individual matrices, the gap between the two groups narrows as scale grows, and at 350M several matrices fall into a transition band. Details in Section A.15.

### A.14 Broader implications

The result of Section 4, that what is treated as noise is mostly weak and diffuse signal, does not apply only to our criterion. Any method that discards the tail of the momentum spectrum is cutting into this material.

That does not make low-rank methods useless. They trade quality for cost, and Section F shows the trade pays off above a certain scale. What must be abandoned is only the expectation that some rank level exists at which cutting costs nothing. No such level exists.

As for future directions, we believe that an automatic rank criterion should be based on the cost–quality trade-off rather than on separating signal from noise. The right question is not *which directions are noise* but *is the* $(k+1)$*-th direction worth what it costs to compute*.

### A.15 Limitations

**Experimental scope.** One architecture, one dataset, one hardware type. Three scales, but the step from 124M to 350M changes both width and depth (Table 2), so we cannot separate their effects; every statement about scaling trends should be read as covering both at once.

**Robustness of the noise-floor estimator.** The estimator of Section 3.2 assumes noise whose spectrum has a compact bulk. On a planted rank-5 problem it recovers $k = 5$ under Gaussian noise and moderately heavy-tailed noise (Student-$t$, $\nu = 5$), but overestimates the rank under very heavy tails ($\nu = 3$: $k = 10$–$13$) and under row correlation ($\rho = 0.3$: $k = 16$–$19$). The bias is always upward, toward retaining more directions. It is therefore conservative for our main conclusion: if momentum noise is correlated, the true structured rank is lower than the measured $0.35$, and loss still improves beyond it. Estimators designed for correlated noise such as ScreeNOT (Donoho et al., 2023) address the correlation but retain the compact-support assumption, and so would not recover a threshold on a power-law spectrum either.

**The noise separation measurement.** The share $f$ is derived from a ratio close to $1$ and so carries a large relative uncertainty: $0.084 \pm 0.020$ at 60M and $0.038 \pm 0.019$ at 350M, two overlapping intervals. The present data suffice to say $f$ is small at both scales, but *not* to establish that $f$ decreases with scale; concluding that would require more runs per pair and at least a third scale. The momentum comparison in Section D is separately confounded with step count, so the effect of $\mu$ may be overestimated. The noise share also depends on the measurement window; we report the standard window and state conclusions as upper bounds so as not to depend on that choice. The two-component model fits the data well, but fitting well is not the same as being correct.

**The diagnostic indicator.** At the run level, classification is unambiguous at all three scales. But the gap between the two groups narrows steadily with scale — $0.61$, $0.29$ and $0.07$ — and the number of matrices falling in the transition band goes from $0$ to $1$ to $6$. Extrapolating further may therefore require recalibrating the threshold, and we report counts at two thresholds rather than one.

**Unmeasured cost axis.** The communication axis is only modeled (Section F.4), and the estimate for the amortized variant is derived from a model rather than implemented.

**Four rejected hypotheses.** We registered and subsequently rejected: that matrix shape determines a layer's fate; that early-training spectral statistics predict which layers will fail; that the selected rank fraction decreases as models grow; and that a small batch size would produce an interior optimal rank. The conclusions that remain are those that survived these tests.

**What we cannot explain.** Why some matrices escape the loop in Section 5.2 while most do not. Architectural position is the best predictor we found but is not sufficient, and no early-training spectral statistic separates the two groups. One open possibility is that the determining factor lies in how much new signal arrives per step, i.e., in the forward and backward pass rather than the optimizer; testing this requires data we did not collect.

# B DION ALGORITHM DETAILS

Algorithm 1 is a reduced single-GPU form of Dion (Ahn et al., 2025), written in the notation of this paper. The only difference between the fixed-rank and adaptive branches is line 6: with a fixed rank, $\mathcal{S}$ is all $r$ columns; with adaptive rank, $\mathcal{S}$ is determined by equation 7.

**Algorithm 1** Dion, single-GPU form. The marked line is where the two branches differ.

**Require:** $\boldsymbol{W}_0$, learning rate $\eta$, feedback coefficient $\alpha$, pool rank $q$

1: $\boldsymbol{M}_0 \leftarrow \mathbf{0}$; $\boldsymbol{V}_0 \leftarrow$ random $n \times q$ matrix, orthonormalized
2: **for** $t = 1, 2, \ldots$ **do**
3: $\quad \boldsymbol{M}_t \leftarrow \mu \boldsymbol{M}_{t-1} + \boldsymbol{G}_t$ ▷ accumulate momentum, equation 1
4: $\quad \boldsymbol{Q}_t \leftarrow \mathrm{QR}\big(\boldsymbol{M}_t \boldsymbol{V}_{t-1}\big)$ ▷ one power iteration
5: $\quad \boldsymbol{P}_t \leftarrow \boldsymbol{M}_t^\top \boldsymbol{Q}_t$
6: $\quad \mathcal{S} \leftarrow$ set of columns to update ▷ fixed: all; adaptive: by equation 7
7: $\quad \boldsymbol{W}_t \leftarrow \boldsymbol{W}_{t-1} - \eta\gamma\, \boldsymbol{Q}_t[:, \mathcal{S}]\ \mathrm{colnorm}\big(\boldsymbol{P}_t[:, \mathcal{S}]\big)^\top$
8: $\quad \boldsymbol{M}_t \leftarrow \boldsymbol{M}_t - \alpha\, \boldsymbol{Q}_t[:, \mathcal{S}]\big(\boldsymbol{P}_t[:, \mathcal{S}]\big)^\top$ ▷ error feedback, equation 4
9: $\quad \boldsymbol{V}_t \leftarrow \mathrm{colnorm}(\boldsymbol{P}_t)$ ▷ basis for the next step, over the whole pool
10: **end for**

Two details are worth noting. First, $\boldsymbol{V}_t$ on the last line is updated over the *entire* pool rather than only over $\mathcal{S}$; otherwise the unselected directions would lose their tracking basis and could never return. Second, columns whose norm approaches 0 are reinitialized with a random orthonormal vector, so that the pool does not degenerate.

The variant in Section 5.3 changes only the error feedback line to equation 10, dropping the $\mathcal{S}$ index from both factors. Every other line is unchanged.

# C NOISE FLOOR ESTIMATION DETAILS

## C.1 THE TWO ESTIMATORS AND HOW THEY COMBINE

The residual-energy estimator — the natural approach described in Section 3.2 — has the form

$$\hat{\sigma}^2_{\text{res}} \;=\; \frac{\|\boldsymbol{M}\|_F^2 - \sum_{i \le q} \sigma_i^2}{\big(\min(m,n) - q\big) \cdot \max(m,n)}, \tag{11}$$

and fails because its denominator assumes a flat noise spectrum. We replace it by the larger of two independent estimates:

$$\hat{\sigma} \;=\; \max\big(\hat{\sigma}_{\text{res-MP}},\ \hat{\sigma}_{\text{tail}}\big). \tag{12}$$

**Corrected residual.** Let $E_{\mathrm{MP}}(\psi)$ denote the fraction of noise energy contained in the leading $\psi$ fraction of the noise spectrum. If $k$ of the $q$ observed directions are signal, the noise outside the leading $q - k$ noise directions accounts for $1 - E_{\mathrm{MP}}\big((q-k)/\min(m,n)\big)$ of total noise energy, so

$$\hat{\sigma}^2_{\text{res-MP}} \;=\; \frac{\|\boldsymbol{M}\|_F^2 - \sum_{i \le q} \sigma_i^2}{\min(m,n)\max(m,n)\big[\,1 - E_{\mathrm{MP}}\big((q-k)/\min(m,n)\big)\,\big]}. \tag{13}$$

**Tail quantile matching.** For each $\sigma_j$ in the lower half of the observed range, if it were noise it would sit at quantile $j/\min(m,n)$ of the Marchenko–Pastur distribution, which implies a value of $\sigma$. Taking the median over $j$:

$$\hat{\sigma}_{\text{tail}} \;=\; \text{median}_j \,\frac{\sigma_j}{\sqrt{\max(m,n)\,t_j}}, \tag{14}$$

with $t_j$ the corresponding Marchenko–Pastur quantile, following the spirit of the median-based estimator of Gavish & Donoho (2014). If signal directions remain in the lower half, equation 14 is only pushed upward, i.e., biased toward retaining fewer directions.

**Why take the larger.** equation 13 breaks down when the observed range covers nearly the whole spectrum, since the residual goes to zero; equation 14 breaks down when the observed range is too shallow to take a median over. The two failure regimes do not overlap, so taking the larger both guarantees one estimator remains valid and biases conservatively.

### C.2 Testing on a problem with a known answer

The constructed matrix is $\boldsymbol{M} = \boldsymbol{U}_5\boldsymbol{\Sigma}_5\boldsymbol{V}_5^\top + \boldsymbol{N}$, of size $512 \times 384$, with five signal directions of singular values $(120, 100, 85, 70, 60)$ and standard normal noise $\sigma = 1$. The noise edge from equation 5 sits near 42.2, so the correct answer is $k = 5$. Table 6 and Figure 9 give the results.

Table 6: Noise floor estimation on a matrix of rank exactly 5. The column $q$ is the number of observed directions. The residual estimator equation 11 drifts with $q$; the corrected estimator equation 12 remains invariant, except when the entire spectrum is observed.

| $q$ | $k$ (residual) | $\hat{\sigma}$ (residual) | $k$ (corrected) | $\hat{\sigma}$ (corrected) |
|---|---|---|---|---|
| 32 | 15 | 0.918 | 5 | 1.014 |
| 64 | 38 | 0.837 | 5 | 1.013 |
| 128 | 89 | 0.691 | 5 | 1.007 |
| 256 | 214 | 0.420 | 5 | 1.007 |
| 340 | 303 | 0.247 | 5 | 1.002 |
| 384 | 358 | 0.141 | 6 | 1.001 |

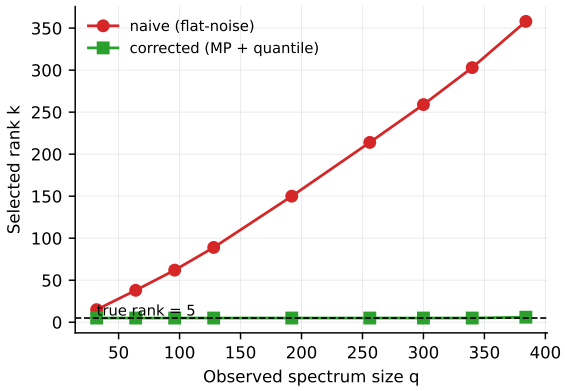



Figure 9: Selected rank on a matrix whose correct answer is 5, against the number of observed directions $q$. The residual estimator drifts roughly linearly with $q$; the corrected estimator is independent of it.

### C.3 The energy distribution function $E_{\text{MP}}$

With aspect ratio $c = \min(m,n)/\max(m,n)$, set $t = \sigma_i^2/\big(\sigma^2 \max(m,n)\big)$. The Marchenko–Pastur density of $t$ is (Marčenko & Pastur, 1967; Bai & Silverstein, 2010)

$$f_c(t) \;=\; \frac{\sqrt{(t_+ - t)(t - t_-)}}{2\pi c\, t}, \qquad t_\pm = \big(1 \pm \sqrt{c}\big)^2, \tag{15}$$

normalized so that $\int f_c = 1$ and $\int t f_c = 1$. Let $\psi$ be the fraction *by count* of noise singular values above some cut level; then $E_{\mathrm{MP}}(\psi)$ is the corresponding fraction *by energy*:

$$\psi(u) = \int_u^{t_+} f_c(t)\,\mathrm{d}t, \qquad E_{\mathrm{MP}}\big(\psi(u)\big) = \int_u^{t_+} t\, f_c(t)\,\mathrm{d}t. \tag{16}$$

The two integrals in equation 16 are taken over the density equation 15. In our implementation we tabulate them once per $c$ by quadrature on a uniform grid of $4000$ points, then interpolate linearly. The tabulation cost is negligible against one optimizer step, and the table is cached across steps.

The quantiles $t_j$ used in equation 14 come from the same table: $t_j$ is the value of $t$ at which $\psi(t) = (j + 1/2)/\min(m, n)$.

### C.4 The self-consistent iteration

The estimator equation 13 depends on $k$, while $k$ from equation 7 depends on $\hat{\sigma}$. We solve by iteration: initialize $k = 1$, compute $\hat{\sigma}$ and $\tau$, recount $k$, and repeat until $k$ is unchanged or three passes have elapsed. Across all our experiments the iteration terminated within three passes.

### C.5 Pseudocode for the criterion

**Algorithm 2** Corrected noise-edge rank selection.

**Require:** singular values $\sigma_1 \geq \cdots \geq \sigma_q$ of the observed range, $\|\boldsymbol{M}\|_F^2$, dimensions $m, n$, coefficient $c$

$\hat{\sigma}_{\text{tail}} \leftarrow$ equation 14 over the lower half of $\{\sigma_j\}$
$k \leftarrow 1$
**for** pass $= 1, 2, 3$ **do**
  $\hat{\sigma}_{\text{res-MP}} \leftarrow$ equation 13 with the current $k$
  $\hat{\sigma} \leftarrow \max(\hat{\sigma}_{\text{res-MP}}, \hat{\sigma}_{\text{tail}})$
  $\tau \leftarrow c\,\hat{\sigma}(\sqrt{m} + \sqrt{n})$
  $k' \leftarrow \max\big(1, |\{i : \sigma_i > \tau\}|\big)$
  **if** $k' = k$ **then break**; $k \leftarrow k'$
**end for**
**return** $k$, $\tau$, $\hat{\sigma}$

## D Noise separation: details

Table 7 gives the noise separation results at three momentum coefficients, measured at the 60M scale using the pairing described in Section 4. The column $f$ is the sampling noise share from equation 9; Figure 10 plots the same data.

Table 7: Separating sampling noise from persistent structure at three momentum coefficients. The fit is a linear regression through the origin, giving $f \approx 2.10 \cdot (1 - \mu)/(1 + \mu)$ with residuals no larger than $0.03$.

| $\mu$ | $(1-\mu)/(1+\mu)$ | $\hat{\sigma}$ ratio | measured $f$ | fitted $f$ | residual |
|---|---|---|---|---|---|
| 0.95 | 0.026 | 1.119 | 0.084 | 0.054 | $+0.030$ |
| 0.80 | 0.111 | 1.330 | 0.256 | 0.233 | $+0.023$ |
| 0.50 | 0.333 | 1.751 | 0.689 | 0.699 | $-0.010$ |

The linear relation is an approximation over the measured range, not an identity: $f$ is bounded above by 1, so linearity cannot hold as $\mu \to 0$. The low-$\mu$ configurations also ran for fewer steps than the standard one, and the rank fraction changes over the course of training, so the effect of $\mu$ may be overestimated.

The replication at the 350M scale uses the same pairing (same step count, same $\mu$, same host) with batch sizes of $294{,}912$ and $73{,}728$ tokens per step; the measured $\hat{\sigma}$ ratio is $1.056$.

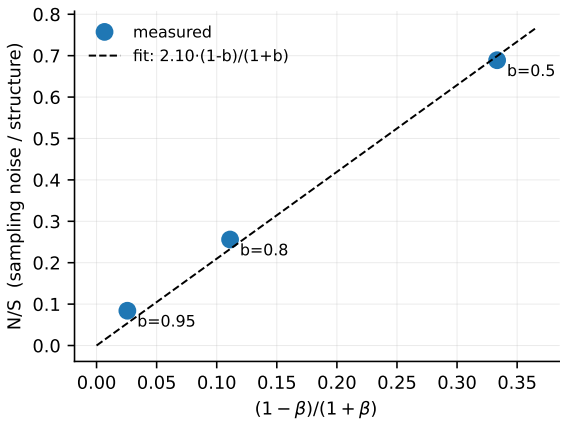



Figure 10: Sampling noise share $f$ against the moving-average compression factor $(1-\mu)/(1+\mu)$. Three measured points spanning a thirteen-fold range; the dashed line is a linear fit through the origin.

## E THRESHOLD COEFFICIENT SWEEP

Table 8 gives the rank fraction and loss at three values of $c$, measured at the 60M scale on the host without error feedback.

Table 8: Sweeping the threshold coefficient $c$. Lowering $c$ raises the selected rank fraction and improves loss; no value gives an interior optimum. The measured elasticity $\mathrm{d}\log k/\mathrm{d}\log c$ is $-0.92$.

| $c$ | Rank fraction | Loss |
|---|---|---|
| 0.5 | 0.639 | 3.6490 |
| 1.0 | 0.353 | 3.7171 |
| 1.5 | 0.211 | 3.7853 |

## F OPTIMIZER STEP COST BY MATRIX SIZE

Table 9 gives the full cost table, measured over the four matrices of one transformer block (of sizes $3d \times d$, $d \times d$, $4d \times d$ and $d \times 4d$), taking the median over ten repetitions after three warmup passes.

Table 9: Optimizer step cost relative to the full-rank branch, by matrix width. Values below 1 mean cheaper. Measured over the four matrices of one transformer block, median of ten repetitions after three warmup passes.

| Method | $d = 512$ | 768 | 1024 | 2048 | 4096 | 8192 |
|---|---|---|---|---|---|---|
| Full rank (Muon) | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 |
| Dion, fixed rank | 1.52 | 2.53 | 2.77 | 1.63 | 0.92 | 0.60 |
| Low-rank Muon (SVD) | 9.01 | 13.91 | 16.39 | 9.31 | 4.71 | 2.85 |
| Low-rank Muon (NS) | 1.51 | 2.52 | 2.82 | 1.70 | 0.99 | 0.68 |
| Dion, adaptive rank | 20.81 | 39.24 | 46.55 | 31.20 | 23.80 | 20.79 |
| Low-rank Muon, adaptive | 30.57 | 53.28 | 65.23 | 46.96 | 36.14 | 31.80 |

### F.1 THREE AXES OF BENEFIT

Low-rank methods save one of three things. *Compute*: fewer directions mean fewer multiplications. *Communication*: in distributed training, only two low-rank factors need be exchanged rather than

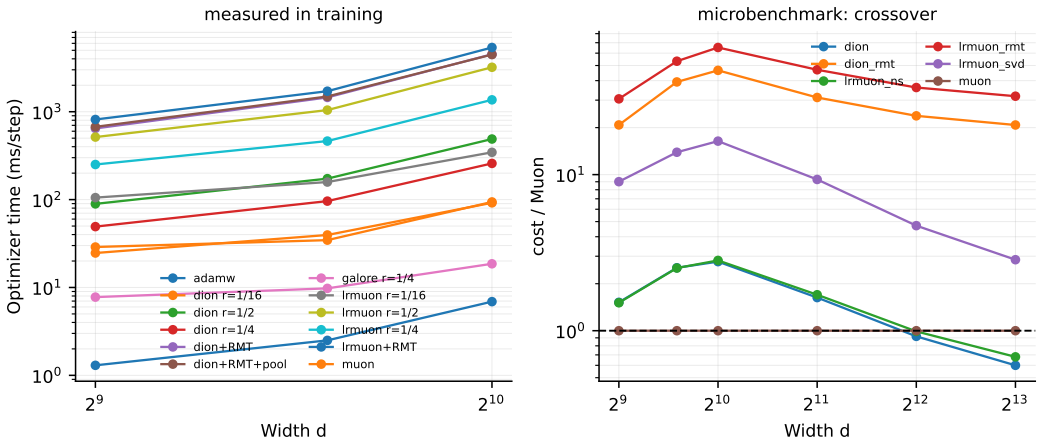



Figure 11: Left: optimizer step cost measured during training, against model width. Right: cost ratio against the full-rank branch, measured separately on matrices of increasing size. The horizontal line at 1 marks break-even.

the full matrix. *Memory*: not an axis for the methods studied here, since as noted in Section 2.1, momentum is still stored in full.

Our experiments run on a single GPU, so the compute axis is measured directly while the communication axis can only be modeled. We distinguish the two clearly below.

### F.2 Why low-rank costs more at small scale

The first observation runs against intuition: at 60M, the full-rank branch runs about $1.7\times$ faster than the branch cut to $r/\min(m,n) = 1/4$, despite performing far fewer operations in theory.

The cause lies not in the operation count but in the kind of operation. The full-rank branch approximates $\boldsymbol{U}\boldsymbol{V}^\top$ with a chain of large matrix multiplications — exactly what the hardware is built to run at high efficiency. The low-rank branch requires a matrix decomposition, which runs at much lower efficiency on the same hardware.

This is a constant-factor effect rather than an asymptotic one: the ratio between the two costs shrinks as matrices grow. A break-even size must therefore exist.

### F.3 The break-even point

To locate that size without training, we measure the optimizer step cost alone on matrices of increasing size at a fixed rank fraction. Results are in Figure 11 and Table 9.

At fixed rank, break-even occurs near $d = 4096$: from that size upward the low-rank branch is cheaper than full rank, and at $d = 8192$ it is about $1.7\times$ cheaper.

This cost model matches the times measured during actual training at all three scales to within $10\%$, so the extrapolation to larger sizes is trustworthy. One directly measured signal points the same way: at 350M, the lowest rank level has already reached $0.98\times$ the cost of the full-rank branch, against $0.85$ at 60M.

The answer to the question opening this subsection is therefore: low-rank *does* have a compute advantage, but it appears only above a certain size. The quality experiments in Section 5.1 and Section 5.2 sit below that point.

### F.4 The communication axis

The volume exchanged per step scales as $mn$ for the full-rank branch and as $r(m+n)$ for the low-rank branch. Substituting the rank fractions measured in Section A.2, at $d = 8192$ this gives roughly $3\times$ savings at fixed rank $1/4$ and roughly $1.9\times$ with adaptive rank.

This is an analytical model, not a measurement. Validating it requires a multi-machine setup, outside the scope of this paper. We include it because this is the axis Dion was designed to optimize (Ahn et al., 2025), and omitting it would leave the cost picture incomplete.

### F.5 Why adaptive rank forfeits that advantage

The cost of automatic rank selection scales not with the number of directions *used* but with the number that must be *observed*. As noted in Section 3.3, the observed range must exceed the rank to be selected, plus a tail for estimating the noise floor via equation 14.

The consequence is a paradox worth recording: in our experiments the average rank fraction the criterion selects is *lower* than the cheapest fixed rank in the grid, yet its cost is more than twenty times higher. The adaptive branches do not break even at any size in the range examined: at $d = 8192$, Dion's is still about $21\times$ and Low-rank Muon's about $32\times$ more expensive than full rank.

There is a remedy, and we state plainly that it is a model-based estimate rather than a measurement. The selected rank fraction changes very slowly between consecutive steps, so the criterion could be run periodically rather than every step, holding the rank fixed in between. At a period of $25$ steps, most of the observation cost is amortized away. We did not implement this variant, since any mid-course change would break comparability across the grid in Section 5.1.

### F.6 Summary

The compute advantage of low-rank is real and measurable, but appears only above the range we examined. The communication advantage is larger but outside our measurement scope. Automatic rank selection forfeits most of the compute advantage unless amortized.

These three statements bound the scope of the paper's conclusions: every claim about wall-clock time applies only to the size range examined, whereas the claims about quality in Section 5.1 and Section 5.2 do not depend on cost and are not bound by this limitation.

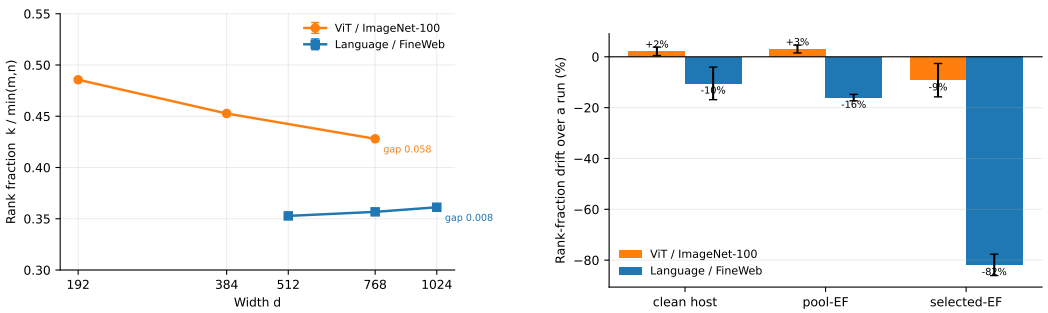
## G The measurement on a different modality: ViT and ImageNet-100

The stability of $0.353$–$0.361$ in Section A.2 raises a practical question: if the rank fraction barely moves, why not hard-code $0.35$ and drop the measurement entirely? This appendix answers by running the same measurement on a different modality.

**Setup.** We train ViTs at three widths ($d = 192, 384, 768$; 12 blocks, 48 matrices each) on ImageNet-100, keeping the matrix shapes identical to the transformer block of Section 2.1 so that the optimizer is reused unchanged. Each run is $800$ steps at a batch size of $256$ images, enough to measure rank but not aimed at best achievable quality. Details are in the supplementary code; Figure 12 summarizes the two main results.

**A higher rank fraction, and not invariant to width.** In the host without error feedback, the measured rank fractions are $0.486$, $0.452$ and $0.431$ across the three widths, averaging $0.456$ against $0.357$ for language models (Figure 12a). The shape matters more than the value: the spread across widths is $0.055$ and *decreases* with $d$, whereas for language models it is $0.008$ and increases slightly.

Figure 12a plots the two curves side by side. A hard-coded constant is therefore wrong twice over: wrong by about $28\%$ if $0.35$ is carried over to vision, and still wrong at one end of the range if a separate vision constant is chosen.




(a) Against model width.

(b) Drift over a run.

Figure 12: The same measurement on two modalities. Left: rank fraction against width in the host without error feedback; ViT gives higher values and varies more with width, in the opposite direction. Right: drift in rank fraction from the start to the end of a run, by mode. The feedback loop of Section 5 appears in the selected-EF column for language models and does not form in ViT.

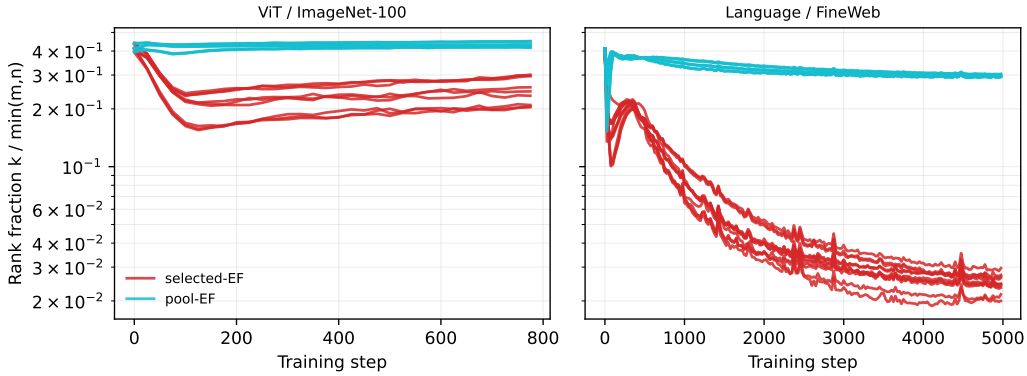



Figure 13: Rank fraction trajectory over training in Dion, under both subtraction modes. Left: ViT, where the two branches are separated by a near-constant offset. Right: language models, where selected-EF tightens by more than an order of magnitude while pool-EF holds steady. Log vertical axis.

**Insensitive to batch size.** Raising the batch from 256 to 1024 moves the rank fraction from 0.457 to 0.458. By contrast, at the 350M language scale, reducing the batch fourfold raises the rank fraction from 0.353 to 0.391. Since the width effect in ViT (0.055) exceeds the batch effect (0.001) by roughly a factor of fifty, the difference between the two modalities cannot be attributed to the batch mismatch.

**The feedback loop of Section 5 does not form.** We run both subtraction modes on Dion at all three widths. The selected-EF branch still selects a rank about $1.8\times$ lower than pool-EF (0.23 against 0.43), but this is a roughly constant offset rather than a progressive tightening: Figure 13 shows the contrast: the drift in rank fraction from the start to the end of a run is $-9.2\%$ in ViT against $-82\%$ for language models in the same configuration. Most of the $-9.2\%$ occurs during the initial transient, after which the trajectory is flat (Figure 12b).

This observation narrows the claim in Section 5. The feedback loop is not an inevitable consequence of combining a spectral criterion with error feedback; it additionally requires that the structured rank fraction be declining over time. In language models that condition holds even in the clean host (the rank fraction drifts $-12.8\%$ over a run); in ViT it does not ($+1.2\%$). Practitioners in the increasing regime are therefore at lower risk, but we still recommend measuring once and fixing the rank, since the condition can only be checked after a run.

**Limitations.** The ViT runs are 800 steps from a single initialization and are not intended to compare quality across rank levels. They suffice to rule out a hard-coded rank and to test whether the feedback loop reproduces, but not to build a loss–rank curve as in Section 5.1.

## H FULL RESULTS TABLE

The table below lists every configuration at all three scales, with the standard deviation across initializations, the average selected rank fraction, and running time. Rows labelled *energy* use the alternative criterion discussed in Section 6: instead of thresholding at the noise edge, it retains the smallest set of directions whose discarded Frobenius share stays below a fixed fraction. Under selected-EF it selects a higher rank than the clean host rather than collapsing, by a factor of 2.4 averaged over three initializations, which is the evidence for the statement made there. Its variance across initializations is markedly larger than that of the noise-edge branches, and we have not investigated why.

Table 10: Full results. Loss is the mean over initializations, with the standard deviation where more than one run is available. Rank fraction is the average fraction of directions the optimizer applies per step; it is 1 for full-rank Muon and empty for AdamW, which uses no decomposition.

| Scale | Method | Loss | Std. dev. | Runs | Rank fraction | GPU-hours |
|---|---|---|---|---|---|---|
| 60M | Muon (full rank) | 3.6225 | 0.0005 | 3 | 1.000 | 1.28 |
| 60M | Dion, $r = 1$ | 3.6356 | 0.0127 | 3 | 1.000 | 1.44 |
| 60M | Dion, $r = 1/16$ | 3.8012 | 0.0026 | 3 | 0.062 | 1.27 |
| 60M | Dion, $r = 1/2$ | 3.6453 | 0.0038 | 3 | 0.500 | 1.35 |
| 60M | Dion, $r = 1/4$ | 3.6724 | 0.0044 | 3 | 0.250 | 1.30 |
| 60M | Low-rank Muon, $r = 1/16$ | 4.0214 | 0.0061 | 3 | 0.062 | 1.37 |
| 60M | Low-rank Muon, $r = 1/2$ | 3.6661 | 0.0021 | 3 | 0.500 | 1.93 |
| 60M | Low-rank Muon, $r = 1/4$ | 3.7569 | 0.0028 | 3 | 0.250 | 1.57 |
| 60M | Dion, noise edge, pool-EF | 3.7334 | 0.0010 | 3 | 0.314 | 2.14 |
| 60M | Dion, noise edge, selected-EF | 3.9240 | 0.0039 | 3 | 0.057 | 2.13 |
| 60M | Low-rank Muon, noise edge | 3.7171 | 0.0013 | 3 | 0.353 | 2.34 |
| 60M | Dion, energy, pool-EF | 4.0547 | 0.2387 | 3 | 0.252 | 2.24 |
| 60M | Dion, energy, selected-EF | 3.6510 | 0.0111 | 3 | 0.513 | 2.35 |
| 60M | Low-rank Muon, energy | 3.9269 | 0.0559 | 3 | 0.203 | 2.53 |
| 60M | GaLore | 3.8826 | 0.0027 | 3 | 0.250 | 1.26 |
| 60M | AdamW | 3.6834 | 0.0036 | 3 | — | 1.24 |
| 124M | Muon (full rank) | 3.4621 | 0.0015 | 3 | 1.000 | 2.31 |
| 124M | Dion, $r = 1$ | 3.4683 | — | 1 | 1.000 | 2.69 |
| 124M | Dion, $r = 1/16$ | 3.6098 | 0.0037 | 3 | 0.062 | 2.31 |
| 124M | Dion, $r = 1/2$ | 3.4778 | 0.0044 | 3 | 0.500 | 2.49 |
| 124M | Dion, $r = 1/4$ | 3.5024 | 0.0009 | 3 | 0.250 | 2.39 |
| 124M | Low-rank Muon, $r = 1/16$ | 3.8330 | 0.0027 | 3 | 0.062 | 2.47 |
| 124M | Low-rank Muon, $r = 1/2$ | 3.4998 | 0.0032 | 3 | 0.500 | 3.68 |
| 124M | Low-rank Muon, $r = 1/4$ | 3.5860 | 0.0025 | 3 | 0.250 | 2.88 |
| 124M | Dion, noise edge, pool-EF | 3.5639 | 0.0019 | 3 | 0.321 | 4.29 |
| 124M | Dion, noise edge, selected-EF | 3.7230 | 0.0012 | 3 | 0.060 | 4.26 |
| 124M | Low-rank Muon, noise edge | 3.5498 | 0.0024 | 3 | 0.357 | 4.59 |
| 124M | GaLore | 3.6949 | 0.0009 | 3 | 0.250 | 2.29 |
| 124M | AdamW | 3.5090 | 0.0021 | 3 | — | 2.27 |
| 350M | Muon (full rank) | 3.2910 | 0.0030 | 3 | 1.000 | 6.41 |
| 350M | Dion, $r = 1/16$ | 3.4348 | 0.0030 | 3 | 0.062 | 6.23 |
| 350M | Dion, $r = 1/2$ | 3.3064 | 0.0013 | 3 | 0.500 | 6.78 |
| 350M | Dion, $r = 1/4$ | 3.3337 | 0.0009 | 3 | 0.250 | 6.45 |
| 350M | Low-rank Muon, $r = 1/16$ | 3.6427 | 0.0022 | 3 | 0.062 | 6.59 |
| 350M | Low-rank Muon, $r = 1/2$ | 3.3330 | 0.0015 | 3 | 0.500 | 10.48 |
| 350M | Low-rank Muon, $r = 1/4$ | 3.4120 | 0.0021 | 3 | 0.250 | 7.96 |
| 350M | Dion, noise edge, pool-EF | 3.3868 | 0.0032 | 3 | 0.330 | 12.24 |
| 350M | Dion, noise edge, selected-EF | 3.5257 | 0.0027 | 3 | 0.067 | 12.31 |
| 350M | Low-rank Muon, noise edge | 3.3783 | 0.0012 | 3 | 0.361 | 13.52 |
| 350M | GaLore | 3.5173 | 0.0036 | 3 | 0.250 | 6.20 |
| 350M | AdamW | 3.3386 | 0.0013 | 3 | — | 6.13 |