DOES MUON NEED FINE-GRAINED SPECTRAL SHAPING?

Anonymous authors
Paper under double-blind review

ABSTRACT

MUON combines current and past gradients into matrix momentum. For $M = U\Sigma V^\top$, the idealized polar update $Q = UV^\top$ gives every singular direction the same weight. We refer to this as the flat profile. Several recent optimizers replace this flat profile with fine-grained spectral maps that give each direction its own gain. *We ask how much of this spectral detail a* MUON *update needs*. Our spectral diagnostics show that approximately 94–97% of measured singular modes lie below an estimated noise edge, yet collectively align positively with a reference gradient.

We introduce BULKBOOST, a two-band spectral reweighting framework with fixed-rank and noise-calibrated variants. The latter uses split-minibatch gradient differences to calibrate a Marchenko–Pastur reference edge for MUON's Nesterov input, separating the *bulk* below the edge from the *spikes* above it. Both variants increase the bulk's relative weight through *one shared gain* while preserving the Frobenius norm of each matrix's unreweighted direction. For a fixed partition, our theory gives the first-order condition under which moving weight toward the bulk lowers the loss. It also quantifies the fraction of the maximal first-order improvement rate, over all per-mode reallocations, that two bands can capture. A polar-plus-low-rank representation enables an approximation without a full SVD, tracking only the spike subspace. Across 30 continued-pretraining settings spanning Pythia-14M to 410M and six corpora, two-band reweighting is competitive with the fine-grained power-law profile of FREON (Shumaylov et al., 2026) and outperforms SPECTRA (Huang et al., 2026). Measured against MUON's flat profile, FREON reduces final loss by 0.022% of the pre-adaptation loss on average, whereas the two-band variants achieve reductions of 0.073–0.147%. These observations suggest that useful departures from the flat profile are surprisingly low-dimensional: a single bulk-to-spike gain captures at least as much benefit as the fine-grained spectral profiles.

1 INTRODUCTION

An optimizer for neural-network weight matrices decides two things: how large an update is, and how the update is spread across directions. MUON, an optimizer for language-model training, combines current and past gradients into a matrix momentum M (Jordan et al., 2024; Liu et al., 2025). Its SVD, $M = U\Sigma V^\top$, writes M as a sum of orthogonal directions weighted by their singular values. The idealized polar update $Q = UV^\top$ keeps these directions but sets every singular value to *one*. Each direction therefore receives the same gain. We refer to this as the *flat profile*. Recent optimizers depart from this flat profile through spectral maps $Ug(\Sigma)V^\top$, where the gain varies across singular directions (Shumaylov et al., 2026; Dong & Sawin, 2026; Wu et al., 2026; Pang et al., 2026; Feoktistov et al., 2026). At the same time, Shumaylov et al. (2026) show that substantially different spectra, including random and inverted ones, can remain effective. These results raise a more basic question than which detailed profile is best: *does* MUON *need fine-grained spectral shaping, or is a coarse reallocation between two groups of directions enough?*

Two observations motivate this question. First, a direction's strength in the sampled momentum, its separation from sampling noise, and the benefit of increasing its update weight are distinct quantities. A small singular value can still carry signal, and a large one need not deserve more weight.

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.

Second, our diagnostics reveal a broad spectral bulk and a small leading band. Under a Gaussian white-noise reference model, random matrix theory provides an asymptotic upper spectral edge, the *noise edge* (Marchenko & Pastur, 1967; Baik et al., 2005; Benaych-Georges & Nadakuditi, 2012). We call directions above this reference edge *spikes* and the remaining directions the *bulk*. On Pythia-70M MUON trajectories, approximately 93.7–97.1% of pooled singular modes lie in the bulk (Appendix D). These below-edge singular modes, the *bulk*, are weak individually, yet their aggregate signed alignment with an independently sampled reference gradient is positive in every analyzed matrix. Thus, the broad spectral bulk makes a non-negligible collective contribution even though its individual directions are weak. These measurements motivate a two-band partition.

We introduce BULKBOOST, whose exact target reweights MUON's polar direction in two bands. The spike and bulk gains are 1 and ρ before normalization, and the update is then rescaled to keep the Frobenius norm of the unreweighted direction. At the exact-target level, $\rho = 1$ recovers Q, while $\rho > 1$ increases the bulk's relative weight. Once the partition is chosen, the same ρ controls reallocation across all spectrally updated matrices. We consider two rules for the split. The *fixed-rank rule*, BB-STATIC, takes a prescribed fraction p of the leading directions, shared across matrices and held fixed within each run. Our shared configuration uses $p = 5\%$, within the pooled spike fraction of 2.9–6.3% measured in our diagnostics (Appendix D). We also test smaller and larger fractions. The *noise-calibrated rule*, BB-ADAPTIVE, estimates the sampling noise from two minibatch halves, propagates it through the momentum, and places the split at the resulting noise edge for each matrix and step. The edge assumes isotropic Gaussian noise. Since the real gradient noise is not isotropic, this split is an approximation. Our contributions are as follows.

1. **A two-band reweighting rule.** BULKBOOST reweights MUON's flat update in two bands with one shared gain. The split between the bands comes either from a fixed fraction of directions or from a noise-calibrated edge. An exact polar-plus-low-rank identity lets us avoid a full SVD (Section 3).
2. **What the two bands capture.** For a fixed partition, we characterize to first order when shifting weight toward the bulk lowers the loss, and how much of the best per-direction improvement two bands can reach (Section 4).
3. **Evidence from the spectrum.** Our spectral diagnostics establish that most singular directions of MUON's momentum lie below the noise edge. This bulk is weak direction by direction, yet it carries useful signal in aggregate. Giving it more relative weight lowers the loss after a single update in most tested states (Appendices D and E.2).
4. **Competitive gains without fine-grained shaping.** In continued pretraining settings spanning Pythia-14M to 410M and six corpora, two-band reweighting gives, on average, a slightly larger improvement over MUON's flat profile than the fine-grained direction-specific spectral profiles we compare against. Thus, a single shared gain between spikes and bulk is enough in these settings (Section 5).

Due to space limitations, we provide the Related work in Appendix A.

2 SETUP AND SPECTRAL REWEIGHTING

MUON and notation. Consider a matrix parameter $W_t \in \mathbb{R}^{m\times n}$ with minibatch gradient G_t after any global gradient clipping. Let $d = \min\{m, n\}$ and $[d] = \{1, \dots, d\}$. We use $\langle A, B\rangle_F = \mathrm{tr}(A^\top B)$, and $\|A\|_F$ and $\|A\|_2$ denote the Frobenius and spectral norms. For $\beta \in [0, 1)$ and $B_0 = 0$, the exponential moving-average (EMA) buffer and the Nesterov spectral input are, for $t \geq 1$,

$$B_t = \beta B_{t-1} + (1-\beta)G_t, \qquad M_t = \beta B_t + (1-\beta)G_t. \tag{1}$$

Let $M_t = U_t \Sigma_t V_t^\top$ be its thin singular value decomposition (SVD), with singular values $\sigma_{t,1} \geq \cdots \geq \sigma_{t,d}$ and paired singular vectors $u_{t,i}, v_{t,i}$. MUON's idealized direction is $Q_t = \mathrm{polar}(M_t) = U_t V_t^\top$, which gives unit gain to every singular direction (Jordan et al., 2024; Liu et al., 2025). In practice, MUON approximates Q_t with a few Newton–Schulz iterations. All spectral quantities refer to M_t. The exact formulas below assume full-rank inputs. Appendix B.3 extends them to rank-deficient inputs by restricting to nonzero singular modes.

Two-band partitions. A partition, which we also call a split, divides the singular directions of M_t into a spike band $\mathcal{I}_{\mathrm{sp},t} \subseteq [d]$ and a bulk band $\mathcal{I}_{\mathrm{bu},t} = [d] \setminus \mathcal{I}_{\mathrm{sp},t}$. Both rules we consider put

the leading directions in the spike band, so $\mathcal{I}_{\mathrm{sp},t} = \{1, \dots, r_t\}$. The rules differ only in how they choose r_t. The fixed-rank rule, BB-STATIC, sets $r_t = \lceil pd \rceil$ for a fixed fraction $p \in (0, 1)$. The noise-calibrated rule, BB-ADAPTIVE, compares the singular values with a threshold τ_t:

$$\mathcal{I}_{\mathrm{sp},t} = \{i \in [d] : \sigma_{t,i} > \tau_t\}, \qquad \mathcal{I}_{\mathrm{bu},t} = \{i \in [d] : \sigma_{t,i} \le \tau_t\}. \tag{2}$$

The threshold is based on the *noise edge*. For a large Gaussian white-noise matrix with entrywise standard deviation s, the top singular value approaches $s(\sqrt{m} + \sqrt{n})$ (Marchenko & Pastur, 1967; Geman, 1980). Section 3 calibrates this edge for the Nesterov input using two minibatch halves. Real gradient noise is not isotropic Gaussian, so the edge is an approximate reference for the split.

Norm-matched spectral reweighting. A nonzero profile $w_t \in [0, \infty)^d$ specifies unnormalized gains along the singular directions of M_t. Its exact norm-matched direction is

$$D_t(w_t) = c_t(w_t) U_t \operatorname{diag}(w_t) V_t^\top, \qquad c_t(w_t) = \sqrt{\frac{d}{\sum_{i=1}^d w_{t,i}^2}}. \tag{3}$$

Thus $\|D_t(w_t)\|_F = \|Q_t\|_F = \sqrt{d}$, which fixes the per-matrix direction norm before learning-rate scaling. A scalar map $g : (0, \infty) \to [0, \infty)$ induces the profile $w_{t,i} = g(\sigma_{t,i})$, and we write $D_t(g) := D_t(w_t)$ when this profile is nonzero. Fine-grained profiles, such as the power law of FREON (Shumaylov et al., 2026), have this form: each direction's gain depends on its own singular value. BULKBOOST instead uses the singular values only to assign bands. Its two-level profile has one bulk-to-spike gain $\rho \ge 0$, shared by all matrices:

$$w_{t,i}^{(\rho)} = \begin{cases} 1, & i \in \mathcal{I}_{\mathrm{sp},t}, \\ \rho, & i \in \mathcal{I}_{\mathrm{bu},t}, \end{cases} \qquad c_{\rho,r_t} := c_t\Big(w_t^{(\rho)}\Big) = \sqrt{\frac{d}{r_t + \rho^2(d - r_t)}}. \tag{4}$$

For a nonzero profile, the applied spike and bulk gains are c_{ρ,r_t} and $\rho c_{\rho,r_t}$. At $\rho = 1$, the profile is flat and the direction is Q_t. With both bands nonempty, $\rho > 1$ gives the bulk more weight relative to the spikes, and $0 \le \rho < 1$ gives it less. Because norm matching rescales both bands by c_{ρ,r_t}, ρ sets only the ratio of the two gains. Multiplying the bulk gains by ρ and dividing the spike gains by ρ therefore give the same update. For small spike fractions and $\rho > 1$, the bulk gain stays close to one, and most of the change comes from a smaller spike gain. Appendix B gives further details on normalization identities, relative and effective gains, and conventions for rank-deficient inputs and empty bands. Section 3 realizes this direction with a numerical polar approximation $\widehat{Q}_t$.

3 NOISE-EDGE REWEIGHTED MUON: BULKBOOST

This section describes how BULKBOOST computes its update.

3.1 NOISE-CALIBRATED PARTITIONING

Here we describe how BB-ADAPTIVE places its threshold. At fixed parameters, let $G_t^{(1)}, G_t^{(2)}$ be the raw mean gradients of two independent, equal-sized minibatch halves from the same sampling distribution. Define

$$G_t^{\mathrm{raw}} = \frac{G_t^{(1)} + G_t^{(2)}}{2}, \qquad Z_t^{\mathrm{split}} = \frac{G_t^{(1)} - G_t^{(2)}}{2}. \tag{5}$$

The half-batch difference Z_t^{split} has zero mean. After vectorization, it has the same covariance as the sampling noise in the raw full-minibatch mean.

Let $\chi_t \in (0, 1]$ be the full-minibatch clipping scalar computed across all model parameters, with $\chi_t = 1$ without clipping. We use $G_t = \chi_t G_t^{\mathrm{raw}}$ in equation 1 and do not clip the halves separately. We apply χ_t^2 to the raw variance estimate before smoothing:

$$\widehat{s}_t^2 = \frac{\|Z_t^{\mathrm{split}}\|_F^2}{mn}, \qquad \overline{s}_t^2 = \gamma \overline{s}_{t-1}^2 + (1 - \gamma)\chi_t^2 \widehat{s}_t^2. \tag{6}$$

For $\gamma \in [0, 1)$, each matrix's EMA starts at $\overline{s}_1^2 = \chi_1^2 \widehat{s}_1^2$, and the recursion applies for $t \ge 2$. The threshold must match the noise in the Nesterov input M_t, which differs from the noise in the gradient

or in the EMA buffer. From equation 1, $M_t = (1-\beta^2)G_t + \beta^2 B_{t-1}$, so the current gradient enters both directly and through the updated buffer. Under a reference model with centered, temporally independent, isotropic input noises of constant entrywise variance s^2 and $B_0 = 0$, the propagated noise variance in M_t is $\nu_t^2(\beta)s^2$, where

$$\nu_t^2(\beta) = (1-\beta)^2\left[(1+\beta)^2 + \frac{\beta^4(1-\beta^{2(t-1)})}{1-\beta^2}\right], \qquad t \geq 1. \tag{7}$$

Using its nonnegative square root $\nu_t(\beta)$, we set the threshold to the estimated noise edge of M_t:

$$\tau_t = \widehat{\tau}_t = \kappa_{\mathrm{edge}}\nu_t(\beta)\sqrt{\overline{s}_t^2}(\sqrt{m}+\sqrt{n}). \tag{8}$$

with a fixed multiplier $\kappa_{\mathrm{edge}} > 0$ ($\kappa_{\mathrm{edge}} = 1$ in all experiments) and the square root taken after variance smoothing. Proposition 4 proves the half-batch and Nesterov noise formulas, and Appendix C.1 states the assumptions behind the threshold.

3.2 Structure and SVD-Free Realization

Exact target. For either rule, the oracle evaluates $D_t^{\mathrm{svd}} = D_t(w_t^{(\rho)})$ using equation 3 and equation 4. For the spike singular-vector blocks $U_{t,\mathrm{sp}}, V_{t,\mathrm{sp}}$, define $P_{t,\mathrm{sp}} = U_{t,\mathrm{sp}}U_{t,\mathrm{sp}}^\top$ and $R_{t,\mathrm{sp}} = V_{t,\mathrm{sp}}V_{t,\mathrm{sp}}^\top$. Since $P_{t,\mathrm{sp}}Q_t = Q_tR_{t,\mathrm{sp}} = U_{t,\mathrm{sp}}V_{t,\mathrm{sp}}^\top$,

$$D_t^{\mathrm{svd}} = \begin{cases} c_{\rho,r_t}[\rho Q_t - (\rho-1)P_{t,\mathrm{sp}}Q_t], & m \leq n, \\ c_{\rho,r_t}[\rho Q_t - (\rho-1)Q_tR_{t,\mathrm{sp}}], & n < m. \end{cases} \tag{9}$$

The flat update Q_t carries the bulk, and a correction of rank at most r_t adjusts the spikes. Theorem 2 extends this identity to direction-dependent spike gains. Both projector expressions are exact for either orientation of the matrix. We keep both cases as the tracking below works on the smaller side.

Spike-subspace tracking. Both variants track the spike subspace on the smaller Gram side,

$$K_t = \begin{cases} M_tM_t^\top, & m \leq n, \\ M_t^\top M_t, & n < m. \end{cases} \tag{10}$$

We evaluate K_t through products with M_t and $M_t^\top$, without forming the Gram matrix. BB-Static keeps a block of fixed width $\lceil pd \rceil$ and refines it with q block-power iterations at each step. BB-Adaptive must also find how many directions lie above the threshold. It augments the previously retained basis X_{t-1} with fresh standard Gaussian probes Ω_t and sets $Y_0 = \mathrm{orth}([X_{t-1}, \Omega_t])$, where orth returns an orthonormal basis of the column space. After q updates $Y_{j+1} = \mathrm{orth}(K_tY_j)$, the eigenvalues of $Y_q^\top K_tY_q$ estimate squared singular values, and multiplying its eigenvectors by Y_q gives the Ritz directions. We keep the Ritz directions whose eigenvalues exceed $\widehat{\tau}_t^2$, in decreasing order, up to an absolute rank cap $r_{\max}$. Reorthonormalizing them gives $X_t \in \mathbb{R}^{d\times\widehat{r}_t}$. Finite iterations, the size of the trial space, and a binding rank cap can make the retained rank $\widehat{r}_t$ differ from the exact rank r_t. Appendix C.3 gives initialization, probe generation, and selection details.

Numerically norm-matched update. We compute a numerical polar factor $\widehat{Q}_t = \mathrm{PolarExpress}(M_t)$ (Amsel et al., 2026a) and form

$$\widetilde{D}_t = \begin{cases} \rho\widehat{Q}_t - (\rho-1)X_t(X_t^\top\widehat{Q}_t), & m \leq n, \\ \rho\widehat{Q}_t - (\rho-1)(\widehat{Q}_tX_t)X_t^\top, & n < m. \end{cases} \tag{11}$$

For nonzero $\widehat{Q}_t$ and $\widetilde{D}_t$, we set

$$\widehat{D}_t = \widehat{c}_{\rho,t}\widetilde{D}_t, \qquad \widehat{c}_{\rho,t} = \frac{\|\widehat{Q}_t\|_F}{\|\widetilde{D}_t\|_F}. \tag{12}$$

In exact arithmetic, $\|\widehat{D}_t\|_F = \|\widehat{Q}_t\|_F$, and $\rho = 1$ returns the unreweighted direction $\widehat{Q}_t$. A zero $\widehat{Q}_t$ gives a zero update. Otherwise, an orthonormal X_t ensures a nonzero raw direction for $\rho > 0$, while $\rho = 0$ requires a nonzero projected spike component. Since $\|\widehat{Q}_t\|_F$ need not equal $\sqrt{d}$, $\widehat{c}_{\rho,t}$ can differ from $c_{\rho,\widehat{r}_t}$. Norm matching fixes the magnitude only, it does not correct polar or subspace approximation errors (Appendix B.4). Algorithm 1 in Appendix C summarizes the realization. Section 4 characterizes when this reallocation lowers the loss to first order.

4 THEORY OF TWO-BAND REWEIGHTING

We first give a first-order test for when two-band reweighting lowers the loss, and quantify how much of the best per-direction improvement two bands can capture. We then give the exact structure behind the SVD-free realization.

4.1 WHAT TWO BANDS CAN CAPTURE

A controlled update comparison. Starting from the same training state, we vary only the applied gains of one matrix's current update. Fix step t, the singular frame U_t, V_t of M_t, and a continuation policy π, which specifies how training continues. Let $\mathcal{L}^{\pi}_{t,K}(w)$ be the expected evaluation loss, given the current state, after using $U_t \operatorname{diag}(w) V_t^{\top}$ as the current direction and then following π for $K \geq 0$ more steps. The learning rate, the current updates of all other parameters, and the rest of the current step are fixed. Later gradients and optimizer states evolve under π. At $K = 0$, the loss is evaluated right after the current update. Here w denotes the applied gains, that is, the normalized gains $c_t(w_t)w_t$ from equation 3. Norm matching keeps w on $\mathcal{S}_d = \{w \in [0,\infty)^d : \|w\|_2^2 = d\}$, and the flat profile $w = \mathbf{1}_d$ gives MUON's idealized update Q_t.

Continuation utility. We take the loss to be defined on an open neighborhood of $\mathbf{1}_d$ in the positive orthant. Set

$$u_i = -\left.\frac{\partial \mathcal{L}^{\pi}_{t,K}(w)}{\partial w_i}\right|_{w=\mathbf{1}_d}, \qquad \overline{u} = \frac{1}{d}\sum_{i=1}^{d} u_i, \qquad u_{\perp} = u - \overline{u}\mathbf{1}_d. \tag{13}$$

The *continuation utility* u_i is the rate at which the loss falls when the applied gain of direction i increases and the other gains stay fixed. Norm matching forces compensating changes across directions, so first-order improvements depend only on differences between utilities, which $u_{\perp}$ captures.

Fix a partition with $r = |\mathcal{I}_{\mathrm{sp}}|$, $k = |\mathcal{I}_{\mathrm{bu}}| = d - r$, and $1 \leq r \leq d-1$, dropping the time index. Let $\overline{u}_{\mathrm{sp}} = r^{-1}\sum_{i\in\mathcal{I}_{\mathrm{sp}}} u_i$ and $\overline{u}_{\mathrm{bu}} = k^{-1}\sum_{i\in\mathcal{I}_{\mathrm{bu}}} u_i$ be the band means, with gap $\Delta = \overline{u}_{\mathrm{bu}} - \overline{u}_{\mathrm{sp}}$. For this partition, varying ρ gives the norm-matched two-level gains $w(\rho) = c_{\rho,r}(\mathbf{1}_{\mathrm{sp}} + \rho\mathbf{1}_{\mathrm{bu}})$, where $\mathbf{1}_{\mathrm{sp}}$ and $\mathbf{1}_{\mathrm{bu}}$ indicate band membership and $c_{\rho,r}$ is defined in equation 4.

Theorem 1 (Local spectral improvement)**.** *Assume $\mathcal{L}^{\pi}_{t,K}$ is differentiable at $\mathbf{1}_d$.*

(i) Which allocation improves loss? *Along the two-level family, $-\frac{d}{d\rho}\mathcal{L}^{\pi}_{t,K}(w(\rho))\Big|_{\rho=1} = \frac{rk}{d}\Delta$. Increasing ρ from 1 gives a strictly positive first-order improvement if and only if $\Delta > 0$. If $\Delta = 0$, the first-order test is inconclusive.*

(ii) How much improvement can two bands capture? *At the flat profile, norm matching requires small gain changes to sum to zero, so they lie in $\mathcal{T} = \{\delta \in \mathbb{R}^d : \mathbf{1}_d^{\top}\delta = 0\}$. If every gain may change separately, the best improvement rate per unit change is $\max_{\delta\in\mathcal{T},\,\|\delta\|_2=1} u^{\top}\delta = \|u_{\perp}\|_2$. Thus the flat profile is first-order stationary in this fixed-frame, norm-matched class if and only if $u_1 = \cdots = u_d$. If the change must be constant within each band, the best rate is $\sqrt{rk/d}\,|\Delta|$. For $u_{\perp} \neq 0$, two bands therefore reach the fraction $R_{\mathrm{bt}} = \frac{\sqrt{rk/d}\,|\Delta|}{\|u_{\perp}\|_2} \in [0,1]$ of the best rate. Both rates are per unit Euclidean change in w, and R_{bt} allows ρ to move in either direction.*

Two bands can use only the difference between the band means. Near the flat profile, any norm-matched reweighting of the same singular directions, however fine-grained, moves w along some $\delta \in \mathcal{T}$. Its first-order rate is therefore at most $\|u_{\perp}\|_2$. Finer shaping can thus add at most the share $1 - R_{\mathrm{bt}}$, which comes from utility differences within the bands. The split rule, whether the noise edge or a fixed fraction, only decides which directions fall in each band. The sign of Δ and the size of R_{bt} depend on the utilities.

When $\Delta > 0$, small enough increases in ρ lower the loss. The theorem does not say how large ρ may be, so it does not justify a specific finite gain such as $\rho = 4$. Here, first order refers to small changes in the gains; the parameter update itself is a finite step. With one ρ shared by all matrices, each norm-matched separately, the rates from Theorem 1(i) add up across matrices under joint differentiability (Appendix E.2). Appendices E.2 and G.2 test finite values of ρ. Appendix F.1 gives the proof.

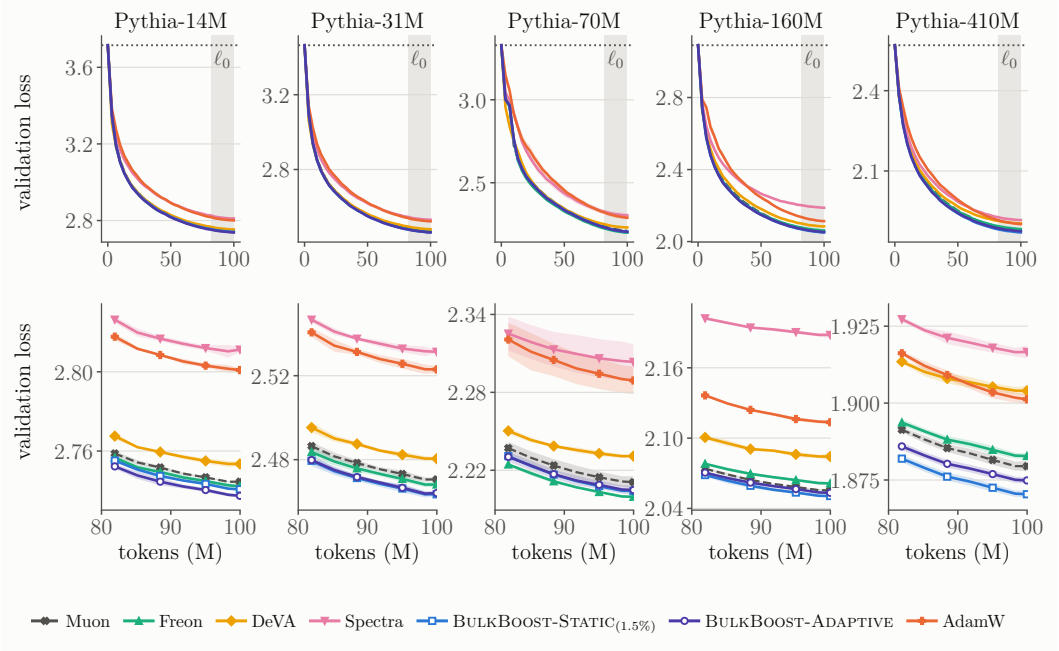


Figure 1: Held-out loss versus training tokens on Wiki-zh, one column per model scale. Top: the full run, with the pre-adaptation loss ℓ_0 dotted and the zoomed interval shaded. Bottom: the shaded interval at a finer vertical scale. The BB-STATIC curve uses $p = 1.5\%$, the leading fraction of SPECTRA, so the two can be compared directly. Appendix G.3 shows the other corpora.

4.2 EXACT POLAR-PLUS-LOW-RANK STRUCTURE

A shared bulk gain also gives the update a simple form. The polar factor carries the bulk, and only a correction in the spike subspace remains.

Theorem 2 (Exact polar-plus-low-rank decomposition)**.** *Let $M \in \mathbb{R}^{m\times n}$ have full rank $d = \min\{m, n\}$, thin SVD $M = U\Sigma V^\top$, and polar factor $Q = UV^\top$. For r selected singular directions, let $U_{\mathrm{sp}}, V_{\mathrm{sp}}$ contain their paired vectors. Assign gains $a \in [0, \infty)^r$ to these directions and a common gain $\rho \geq 0$ to the others. The resulting unnormalized spectral update satisfies*

$$\widetilde{D}(a, \rho) = \rho Q + U_{\mathrm{sp}} \operatorname{diag}(a - \rho \mathbf{1}_r) V_{\mathrm{sp}}^\top, \tag{14}$$

and $\operatorname{rank}(\widetilde{D}(a, \rho) - \rho Q) \leq r$.

With $a = \mathbf{1}_r$, write $\widetilde{D}_\rho = \widetilde{D}(\mathbf{1}_r, \rho)$. Normalizing it gives the projector form equation 9, which needs no bulk singular vectors. The squared Frobenius norm of $\widetilde{D}(a, \rho)$ is $\|a\|_2^2 + \rho^2(d - r)$, so for $a = \mathbf{1}_r$ the normalizer $c_{\rho,r}$ restores $\|Q\|_F = \sqrt{d}$ (Corollary 3). The correction is low rank. Appendices F.2 and F.3 give the proofs.

5 EXPERIMENTS

This section tests whether two bands are enough in practice. The diagnostics in Appendix D motivate this test. On Pythia-70M MUON trajectories, 94–97% of singular directions lie below the noise edge, and this bulk is weak per direction but positively aligned with a reference gradient in aggregate. We compare both BULKBOOST variants, BB-STATIC and BB-ADAPTIVE, with MUON's flat profile and with fine-grained spectral profiles in continued pretraining. We also compare the two variants with each other. Extended results appear in Appendix G, and our code is available online[1].

[1] https://anonymous.4open.science/r/bulkboost_code-7736/README.md

BULKBOOST configurations. Unless varied in an ablation, both variants use $\rho = 4$, $\widehat{Q}_t = \mathrm{PolarExpress}(M_t)$ (Amsel et al., 2026a), $q = 6$ warm-started block iterations for spike tracking, and the norm matching in equation 12. Like Newton–Schulz, Polar Express uses only matrix products. It approximates the polar factor more closely, which the correction in equation 11 needs. MUON and both variants use Nesterov momentum $\beta = 0.95$. BB-ADAPTIVE splits at $\widehat{\tau}_t$ (Section 3.1) with $\gamma = 0.95$, $\kappa_{\mathrm{edge}} = 1$, and the rank cap of Appendix C.3. BB-STATIC keeps $r_t = \lceil pd \rceil$ spike directions, with one fraction p for all matrices and the whole run. We test $p \in \{0.015, 0.02, 0.05, 0.15, 0.20\}$ (Table 9). Our shared configuration, BB-STATIC$_{(5\%)}$, uses $p = 5\%$, which lies within the 2.9–6.3% spike fraction measured in Appendix D. BB-STATIC† instead reports, for each setting, the fraction with the lowest mean final held-out loss. Because this choice uses the reported loss, BB-STATIC† shows what per-setting tuning can reach.

Compared methods. We compare both BULKBOOST variants with five baselines. MUON (Jordan et al., 2024) approximates the flat profile with Newton–Schulz iterations and is the reference for all relative comparisons. Two baselines use fine-grained spectral profiles. FREON (Shumaylov et al., 2026) uses a power-law profile with $c = 0.667$; $c = \frac{1}{2}$ recovers the flat profile. SPECTRA (Huang et al., 2026) rescales the leading singular values to the root-mean-square level of the tail. It keeps the tail's unequal singular values, whereas BULKBOOST gives the whole bulk one gain. We use its recommended leading fraction of 1.5%. We also run BB-STATIC at $p = 1.5\%$, so the two methods can be compared with the same number of spike directions (Figure 1). The other two baselines are ADAMW (Loshchilov & Hutter, 2017) and DEVA (Song et al., 2026), which separates variance adaptation from scale-invariant matrix updates.

Models and datasets. We continue pretraining *converged* Pythia checkpoints (Biderman et al., 2023) at five scales, 14M, 31M, 70M, 160M, and 410M, on six corpora: math (FineMath-4, FineMath-3), Code (CodeParrot), and multilingual text (Wiki-de, Wiki-zh, Wiki-ar). See Appendix G.1.

Token sampling and evaluation. Each corpus is tokenized once with Pythia's GPT-NeoX tokenizer, so all scales and methods reuse the same tokenized data. We reserve the first 1M tokens for held-out evaluation and train on the remainder. Training samples uniformly random contiguous 1024-token windows with a run-specific seed. A run takes $3,051$ steps of 32 sequences ($32,768$ tokens per step), or about 100M tokens, with a cosine learning-rate schedule. Every 100 steps, we evaluate on 128 deterministic, evenly spaced held-out windows, shared across steps and methods. The main comparisons use three seeds.

Pre-adaptation loss. For each model–corpus pair, let ℓ_0 and ℓ_{final} denote the held-out losses before and after continued pretraining, and define $\Delta = \ell_{\mathrm{final}} - \ell_0$. All methods start from the same pretrained checkpoint and are evaluated on the same held-out windows. We report loss differences between methods as a percentage of ℓ_0. Table 8 in Appendix G.1 reports ℓ_0 and Δ for MUON.

Learning-rate calibration. We tune one learning rate per model scale on FineMath-4 and reuse it for every method and all six corpora. To choose it, we run a short learning-rate sweep over multiples of a reference rate, which is 2×10^{-3} for Pythia-14M, 31M, and 70M and 10^{-3} for 160M and 410M. The multipliers are $\{0.01, 0.05, 0.1, 0.25, 0.5, 1, 2, 4\}$. Each sweep run trains one method for 20M tokens with one seed, and we record its final validation loss. The sweep covers every method except ADAMW [2]. In it, BB-STATIC uses $p = 20\%$, and both BULKBOOST variants use $\rho = 4$.

At each scale, we then fix one shared multiplier: the one that is best for most swept methods. It is $0.1\times$ for 14M, 31M, and 160M, $0.25\times$ for 70M, and $0.05\times$ for 410M. The resulting peak learning rates are 2×10^{-4} for 14M and 31M, 5×10^{-4} for 70M, 10^{-4} for 160M, and 5×10^{-5} for 410M. For every swept method, the shared multiplier is also its own best, with two exceptions. At 70M, MUON does best at $0.1\times$ (2×10^{-4}), with a sweep loss of 2.779 against 2.783 at the shared $0.25\times$. Its loss then jumps to 2.919 at the next multiplier, $0.5\times$, so the shared rate stays in MUON's near-optimal range. At 160M, SPECTRA does best at $0.05\times$ (5×10^{-5}), with a sweep loss of 2.552 against 2.553 at the shared $0.1\times$. Figure 2 and Table 11 in Appendix G.2 report the full sweep.

Shared rate in the 100M-token runs. Every 100M-token run uses the shared rate of its scale, with no further tuning. This includes ADAMW, which the sweep does not cover. It also includes all five

[2] Because Pythia models are trained with ADAMW and their learning rate decays to $0.1\times$ the base rate at convergence, we omit ADAMW from the sweep.

Model	Method	FineMath-4	FineMath-3	CodeParrot	Wiki-de	Wiki-zh	Wiki-ar
14M	ADAMW	3.102 ± 0.003	2.650 ± 0.002	1.929 ± 0.003	3.324 ± 0.001	2.801 ± 0.001	2.055 ± 0.001
	MUON	3.086 ± 0.002	2.640 ± 0.002	1.902 ± 0.004	3.283 ± 0.001	2.745 ± 0.001	2.006 ± 0.000
	FREON	$\mathbf{3.083} \pm 0.002$	$\mathbf{2.638} \pm 0.002$	$\underline{1.899} \pm 0.004$	3.279 ± 0.001	2.742 ± 0.001	2.008 ± 0.001
	DEVA	3.110 ± 0.004	2.661 ± 0.002	1.917 ± 0.005	3.286 ± 0.001	2.753 ± 0.001	2.013 ± 0.001
	SPECTRA	3.138 ± 0.004	2.682 ± 0.002	1.955 ± 0.004	3.328 ± 0.003	2.811 ± 0.001	2.058 ± 0.002
	BB-STATIC†	$\mathbf{3.083} \pm 0.002$	$\mathbf{2.638} \pm 0.002$	$\mathbf{1.898} \pm 0.003$	$\underline{3.278} \pm 0.001$	$\underline{2.738} \pm 0.001$	$\mathbf{1.998} \pm 0.001$
	BB-ADAPTIVE	$\underline{3.085} \pm 0.002$	$\underline{2.639} \pm 0.001$	$\underline{1.899} \pm 0.003$	$\mathbf{3.277} \pm 0.002$	$\mathbf{2.737} \pm 0.001$	$\underline{2.000} \pm 0.001$
31M	ADAMW	2.794 ± 0.001	2.443 ± 0.002	1.674 ± 0.002	2.998 ± 0.001	2.523 ± 0.002	1.897 ± 0.000
	MUON	2.776 ± 0.001	2.431 ± 0.002	1.649 ± 0.001	2.959 ± 0.002	2.471 ± 0.001	1.855 ± 0.001
	FREON	$\mathbf{2.771} \pm 0.001$	$\underline{2.429} \pm 0.001$	$\underline{1.648} \pm 0.002$	$\underline{2.956} \pm 0.001$	2.468 ± 0.001	1.861 ± 0.001
	DEVA	2.800 ± 0.001	2.462 ± 0.003	1.668 ± 0.003	2.966 ± 0.003	2.480 ± 0.001	1.862 ± 0.002
	SPECTRA	2.830 ± 0.002	2.482 ± 0.004	1.697 ± 0.003	3.009 ± 0.005	2.531 ± 0.001	1.899 ± 0.002
	BB-STATIC†	$\underline{2.772} \pm 0.001$	$\mathbf{2.428} \pm 0.002$	$\underline{1.648} \pm 0.002$	$\mathbf{2.953} \pm 0.001$	$\mathbf{2.463} \pm 0.001$	$\mathbf{1.848} \pm 0.001$
	BB-ADAPTIVE	2.773 ± 0.001	2.430 ± 0.003	$\mathbf{1.647} \pm 0.003$	$\mathbf{2.953} \pm 0.001$	$\underline{2.464} \pm 0.001$	$\underline{1.849} \pm 0.000$
70M	ADAMW	2.617 ± 0.002	2.343 ± 0.004	1.547 ± 0.005	2.738 ± 0.007	2.289 ± 0.011	1.838 ± 0.014
	MUON	2.551 ± 0.002	2.297 ± 0.004	1.484 ± 0.003	2.680 ± 0.003	2.211 ± 0.005	1.702 ± 0.000
	FREON	$\mathbf{2.536} \pm 0.001$	$\underline{2.288} \pm 0.006$	$\underline{1.471} \pm 0.003$	$\mathbf{2.668} \pm 0.002$	$\mathbf{2.200} \pm 0.000$	$\underline{1.694} \pm 0.001$
	DEVA	2.565 ± 0.001	2.310 ± 0.002	1.493 ± 0.004	2.707 ± 0.001	2.231 ± 0.001	1.723 ± 0.003
	SPECTRA	2.617 ± 0.003	2.339 ± 0.001	1.545 ± 0.001	2.768 ± 0.012	2.303 ± 0.014	1.813 ± 0.021
	BB-STATIC†	$\underline{2.537} \pm 0.004$	$\mathbf{2.287} \pm 0.007$	$\mathbf{1.469} \pm 0.004$	$\underline{2.670} \pm 0.002$	$\underline{2.203} \pm 0.003$	$\mathbf{1.690} \pm 0.001$
	BB-ADAPTIVE	2.545 ± 0.001	2.291 ± 0.003	1.473 ± 0.003	2.674 ± 0.002	2.205 ± 0.004	$\mathbf{1.690} \pm 0.000$
160M	ADAMW	2.292 ± 0.002	$\underline{2.072} \pm 0.001$	1.285 ± 0.003	2.481 ± 0.002	2.114 ± 0.002	1.664 ± 0.001
	MUON	$\mathbf{2.272} \pm 0.003$	$\mathbf{2.066} \pm 0.002$	$\mathbf{1.270} \pm 0.003$	$\mathbf{2.452} \pm 0.002$	2.055 ± 0.001	1.608 ± 0.000
	FREON	2.282 ± 0.001	2.077 ± 0.002	1.282 ± 0.003	2.461 ± 0.002	2.061 ± 0.001	1.613 ± 0.001
	DEVA	2.297 ± 0.003	2.092 ± 0.002	1.287 ± 0.004	2.472 ± 0.002	2.084 ± 0.003	1.639 ± 0.001
	SPECTRA	2.353 ± 0.003	2.125 ± 0.006	1.325 ± 0.004	2.590 ± 0.003	2.188 ± 0.002	1.710 ± 0.001
	BB-STATIC†	$\underline{2.281} \pm 0.003$	2.077 ± 0.003	$\underline{1.280} \pm 0.004$	$\underline{2.457} \pm 0.003$	$\mathbf{2.051} \pm 0.001$	$\mathbf{1.597} \pm 0.002$
	BB-ADAPTIVE	$\underline{2.281} \pm 0.004$	2.077 ± 0.002	1.281 ± 0.004	2.460 ± 0.002	$\underline{2.053} \pm 0.001$	$\underline{1.604} \pm 0.001$
410M	ADAMW	2.058 ± 0.005	1.895 ± 0.002	$\underline{1.119} \pm 0.001$	2.198 ± 0.001	1.901 ± 0.002	1.469 ± 0.000
	MUON	2.046 ± 0.005	1.891 ± 0.002	$\mathbf{1.111} \pm 0.002$	2.181 ± 0.001	1.879 ± 0.002	$\underline{1.452} \pm 0.001$
	FREON	$\underline{2.043} \pm 0.005$	$\mathbf{1.889} \pm 0.001$	$\mathbf{1.111} \pm 0.002$	2.179 ± 0.001	1.883 ± 0.001	1.460 ± 0.001
	DEVA	2.087 ± 0.005	1.929 ± 0.004	1.143 ± 0.002	2.209 ± 0.003	1.904 ± 0.002	1.467 ± 0.001
	SPECTRA	2.099 ± 0.004	1.933 ± 0.006	1.151 ± 0.001	2.224 ± 0.002	1.917 ± 0.002	1.472 ± 0.001
	BB-STATIC†	$\underline{2.043} \pm 0.005$	$\underline{1.890} \pm 0.002$	$\mathbf{1.111} \pm 0.002$	$\mathbf{2.174} \pm 0.001$	$\mathbf{1.870} \pm 0.001$	$\mathbf{1.446} \pm 0.001$
	BB-ADAPTIVE	$\mathbf{2.042} \pm 0.005$	$\underline{1.890} \pm 0.002$	$\mathbf{1.111} \pm 0.002$	$\underline{2.176} \pm 0.001$	$\underline{1.875} \pm 0.001$	1.453 ± 0.002

Table 1: Final validation loss after about 100M tokens, reported as mean $\pm$ standard deviation over three seeds. All runs use the shared learning rate of their scale. For each model–corpus setting, BB-STATIC† reports the fraction with the lowest mean loss among the five in Table 9. **Bold** and Underline mark the lowest and second-lowest means, respectively in each setting, including ties. Appendix G.1 reports the pre-adaptation losses ℓ_0.

BB-STATIC fractions in Table 9, so BB-STATIC$_{(5\%)}$ and BB-STATIC† use it too. Tables 1 and 2 and Figure 1 report these runs.

5.1 COMPARISON OF ALL OPTIMIZATION METHODS

Table 1 reports final validation losses in all 30 model–corpus settings. Table 2 compares each method with MUON and also includes the shared configuration BB-STATIC$_{(5\%)}$. Final loss decreases with model size for every method and corpus. Pre-adaptation losses ℓ_0 follow the same trend (Table 8).

Two bands match or beat fine-grained profiles. Measured against MUON's flat profile, BB-STATIC$_{(5\%)}$, BB-ADAPTIVE, and BB-STATIC† lower the final loss in 23, 24, and 25 of the 30 settings, by 0.073%, 0.077%, and 0.147% of ℓ_0 on average (Table 2). The fine-grained FREON does so in 19 settings, by 0.022%. BB-STATIC$_{(5\%)}$ beats FREON in 13 settings and loses in 13, and BB-ADAPTIVE beats it in 17 and loses in 10. SPECTRA and DEVA trail MUON in all 30 settings, and every BULKBOOST configuration beats both. This includes BB-STATIC at SPECTRA's leading fraction of 1.5% (Table 9). The original DEVA and SPECTRA studies pretrain from scratch (Song et al., 2026; Huang et al., 2026), which differs from our continued pretraining setup. These results suggest that a single bulk-to-spike gain captures at least as much benefit as the fine-grained spectral profiles.

Method	vs. MUON wins	vs. MUON tied	best of all wins	best of all tied	Δ vs. MUON (pp of ℓ_0)	tokens saved (%) mean	tokens saved (%) median
MUON (reference)	–	–	4	1	–	–	–
ADAMW	0	0	0	0	$+1.299 \pm 0.010$	–	–
FREON	19	1	5	3	-0.022 ± 0.010	7.78	7.71
DEVA	0	0	0	0	$+0.811 \pm 0.015$	–	–
SPECTRA	0	0	0	0	$+2.321 \pm 0.047$	–	–
BB-STATIC$_{(5\%)}$	23	2	0	3	-0.073 ± 0.010	7.81	7.80
BB-ADAPTIVE	24	1	4	3	-0.077 ± 0.010	7.99	8.01
BB-STATIC†	**25**	1	**12**	5	$\mathbf{-0.147} \pm 0.021$	**10.00**	**10.28**

Table 2: Summary over the 30 model–corpus settings after about 100M tokens. BB-STATIC† uses the fraction selection of Table 1. Counts compare three-seed mean final losses. *vs.* MUON: settings with lower (*wins*) or equal (*tied*) loss than MUON. *Best of all*: settings with the lowest loss of all rows, alone (*wins*) or shared (*tied*). BB-STATIC$_{(5\%)}$ can at best tie BB-STATIC†, which includes its runs. Δ *vs.* MUON: the loss difference from MUON as a percentage of ℓ_0, averaged over settings within each seed. We report the mean ± std dev of the three seed averages. Negative values favor the method. *Tokens saved*: the mean and median share of the 100M-token budget saved in reaching MUON's final loss, over the winning settings and the seeds that end below MUON (Table 10).

Static and adaptive splits. BB-STATIC† beats BB-ADAPTIVE in 20 settings and loses in 4, but it selects its fraction on the reported loss. Without any fraction search, BB-ADAPTIVE matches BB-STATIC$_{(5\%)}$, with 10 wins, 6 losses, and 14 ties.

Token efficiency and runtime trade-offs. Among settings where it outperforms MUON, BB-ADAPTIVE reaches MUON's final loss using 7.99% fewer tokens on average, with a median saving of 8.01% (Table 2). The extensively tuned BB-STATIC† increases these savings to 10% and 10.28%, respectively. These token savings come with additional per-step computation. At 70M–410M, measured per-step runtimes are 1.36–1.50× MUON's for BB-ADAPTIVE, compared with 1.19–1.27× for the measured static configurations (Appendix G). Adaptive selection avoids fraction search, while static selection offers cheaper execution once a suitable fraction is identified. Neither DEVA nor SPECTRA achieves lower mean final validation loss than MUON in any of the 30 settings (Table 1); their token-saving averages over wins are therefore undefined ("–" in Table 2). DEVA outperforms ADAMW in 18 settings and trails it in 12. Table 10 in Appendix G.2 identifies the winning model–corpus settings and reports their token savings and variation across seeds.

Ablations. Appendix G.2 compares the static and adaptive splits, sweeps the gain ρ, and shows that a third band does not improve on two. In exact-SVD single-update tests on Pythia-70M, $\rho = 1.25$ and 1.5 lower the loss relative to the flat update ($\rho = 1$) in five of six settings (Appendix E.2).

Conclusion. Our diagnostics reveal that MUON's momentum has a broad spectral bulk whose individual directions are weak but collectively aligned with a reference gradient. BULKBOOST exploits this structure with only two bands and a single shared bulk-to-spike gain. Our theory shows when this coarse reallocation is locally beneficial and what finer shaping can add through within-band variation. An exact polar-plus-low-rank identity enables an SVD-free realization. Across 30 continued-pretraining settings, the two-band variants match or exceed the finer spectral baselines in aggregate. Single-update experiments show that bulk emphasis lowers the loss relative to the exact flat update in five of six states. Together, these results suggest that useful departures from MUON's flat spectrum can be surprisingly low-dimensional: *a single bulk-to-spike gain can capture at least as much benefit as the fine-grained spectral profiles*.

Limitations. The strongest BB-STATIC results select the fraction separately for each setting. A shared fraction avoids this search, but the best fraction varies across settings (Table 9). BB-ADAPTIVE instead sets the split online from the noise edge. It needs no fraction search, but it costs more per step. The noise edge comes from a white Gaussian noise reference model, so it separates signal from noise only approximately. Our evaluation is restricted to Pythia continued pretraining under the reported tuning protocol. Extending the evaluation to other architectures and training regimes remains future work.

Reproducibility Statement

The training and evaluation code, configuration files, and data-preprocessing scripts are provided at an anonymous GitHub repository: https://anonymous.4open.science/r/bulkboost_code-7736/README.md. Complete proofs of the theoretical results, including all assumptions, are provided in the appendix. The appendix additionally documents the diagnostic experiments and relevant implementation details needed to reproduce the spectral measurements and ablation studies.

AI Use Statement

Generative AI tools were used during the development of this manuscript. Their use included drafting and debugging portions of experimental code, assisting with the interpretation and presentation of experimental results, and drafting and editing portions of the manuscript. All implementations, experimental protocols, and numerical results included in the final paper were reviewed and verified by the authors. Experiments reported in the paper were executed by the authors on the stated models and datasets. The authors take full responsibility for the correctness and originality of the paper and for parts of the code produced with the assistance of generative AI.

References

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

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, Joshua Lochner, Caleb Fahlgren, Xuan-Son Nguyen, Clémentine Fourrier, Ben Burtenshaw, Hugo Larcher, Haojun Zhao, Cyril Zakka, Mathieu Morlon, Colin Raffel, Leandro von Werra, and Thomas Wolf. Smollm2: When smol goes big – data-centric training of a small language model, 2025. URL https://arxiv.org/abs/2502.02737.

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the Muon algorithm. In *International Conference on Learning Representations (ICLR)*, 2026a.

Noah Amsel, Jack Zhang, Kwangjun Ahn, Ali Naeimi, Austin Feng, Berlin Chen, Tri Dao, and John Langford. Dion3: Full-stack orthogonal updates, 2026b. URL https://arxiv.org/abs/2608.11612.

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.

Florent Benaych-Georges and Raj Rao Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

Pierfrancesco Beneventano, Mahmoud Abdelmoneum, and Tomaso Poggio. The spectral dynamics and noise geometry of muon. *arXiv preprint arXiv:2606.08388*, 2026.

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

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usvsn Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2397–2430. PMLR, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

David Carlson, Volkan Cevher, and Lawrence Carin. Stochastic spectral descent for restricted boltzmann machines. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pp. 111–119. PMLR, 2015. URL https://proceedings.mlr.press/v38/carlson15.html.

CodeParrot. CodeParrot-clean: Cleaned python code dataset, 2022. URL https://huggingface.co/datasets/codeparrot/codeparrot-clean.

Yihe Dong and Will Sawin. Muon^p: Muon with fractional spectral powers. *arXiv preprint arXiv:2606.13867*, 2026.

Katie Everett and Shikai Qiu. Optimizer memory schedules for outscaling the overtraining axis, 2026. URL https://arxiv.org/abs/2609.04577.

Dmitrii Feoktistov, Timofey Belinsky, Andrey Veprikov, Amir Zainullin, and Aleksandr Beznosikov. Softsign: Smooth sign in your optimizer for better parameter heterogeneity handling. *arXiv preprint arXiv:2605.31371*, 2026.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1568–1577. PMLR, 2018. URL https://proceedings.mlr.press/v80/franceschi18a.html.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *CoRR*, abs/2101.00027, 2021. URL https://arxiv.org/abs/2101.00027.

Matan Gavish and David L. Donoho. Optimal shrinkage of singular values. *IEEE Transactions on Information Theory*, 63(4):2137–2152, 2017.

Stuart Geman. A limit theorem for the norm of random matrices. *The Annals of Probability*, pp. 252–261, 1980.

Vineet Gupta, Tomer Koren, and Yoram Singer. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 1842–1850, 2018.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.740. URL https://aclanthology.org/2020.acl-main.740/.

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. URL https://arxiv.org/abs/0909.4061.

Nicholas J. Higham. Computing the polar decomposition—with applications. *SIAM Journal on Scientific and Statistical Computing*, 7(4):1160–1174, 1986. doi: 10.1137/0907079. URL https://doi.org/10.1137/0907079.

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, Fan Yang, and Li Shang. Spectra: Rethinking optimizers for LLMs under spectral anisotropy. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Xiaowen Jiang, Andrei Semenov, and Sebastian U. Stich. Enhancing LLM training via spectral clipping. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.14315.

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

Tim Tsz-Kit Lau, Qi Long, and Weijie Su. Polargrad: A class of matrix-gradient optimizers from a unifying preconditioning perspective. *arXiv preprint arXiv:2505.21799*, 2025.

Liming Liu, Zhenghao Xu, Zixuan Zhang, Hao Kang, Zichong Li, Chen Liang, Weizhu Chen, and Tuo Zhao. COSMOS: A Hybrid Adaptive Optimizer for Efficient Training of Large Language Models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=j2QTOOtM8R.

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.

Zhuanghua Liu, Menglian Wang, and Luo Luo. Musec: MomentUm SpEctral Clipping for stable Muon-type training, 2026. URL https://arxiv.org/abs/2609.11655.

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

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2113–2122. PMLR, 2015. URL https://proceedings.mlr.press/v37/maclaurin15.html.

Gagik Magakyan, Pablo Parrilo, and Asuman Ozdaglar. Spectral scaling laws of Muon, 2026. URL https://arxiv.org/abs/2606.04058.

V. A. Marchenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training, 2018. URL https://arxiv.org/abs/1812.06162.

Tianyu Pang, Yujie Fang, Zihang Liu, Shenyang Deng, Lei Hsiung, Shuhua Yu, and Yaoqing Yang. HTMuon: Improving Muon via Heavy-Tailed Spectral Correction. In *Findings of the Association for Computational Linguistics: ACL*, 2026.

Thomas Pethick, Wanyun Xie, Kimon Antonakopoulos, Zhenyu Zhu, Antonio Silveti-Falls, and Volkan Cevher. Training deep learning models with norm-constrained LMOs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 49069–49104. PMLR, 2025. URL https://proceedings.mlr.press/v267/pethick25a.html.

Zakhar Shumaylov, Nathaël Da Costa, Peter Zaika, Bálint Mucsányi, Alex Massucco, Yoav Gelberg, Carola-Bibiane Schönlieb, Yarin Gal, and Philipp Hennig. Muon is not that special: Random or inverted spectra work just as well. *arXiv preprint arXiv:2605.11181*, 2026.

Zitao Song, Cedar Site Bai, Zhe Zhang, Brian Bullins, and David F. Gleich. Decoupling variance and scale-invariant updates in adaptive gradient descent for unified vector and matrix optimization. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

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

Shuche Wang, Fengzhuo Zhang, Jiaxiang Li, Dirk Bergemann, and Zhuoran Yang. Why muon outperforms adam: A curvature perspective. *arXiv preprint arXiv:2606.04662*, 2026.

Fangzhou Wu, Rikhav Shah, Sandeep Silwal, and Qiuyi Zhang. Dynmuon: A dynamic spectral shaping view of muon. *arXiv preprint arXiv:2605.17109*, 2026.

Qiaozhe Zhang, Jun Sun, and Yingzhuang Liu. Beyond the matrix sign: Quadratic spectral descent, 2026. URL https://arxiv.org/abs/2609.07597.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61121–61143. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhao24s.html.

A Related Work

We position BulkBoost within matrix-aware optimization, spectral reweighting, and subspace selection. Our focus is the allocation of a fixed per-matrix direction norm between two spectral bands, separating partition selection –by fixed rank or a noise-calibrated edge–from the relative gain assigned to the bulk.

A.1 Matrix-Aware Optimization and Spectral Geometry

Stochastic spectral descent exploits matrix geometry rather than treating a parameter matrix as an unstructured vector (Carlson et al., 2015). Bernstein & Newhouse (2024) organize several optimizer constructions through norm-constrained steepest descent. Muon applies an approximate polar map to matrix momentum (Jordan et al., 2024); its large-scale realization addresses update scaling, weight decay, and distributed execution for language-model training (Liu et al., 2025). Scion develops a broader family of norm-constrained linear minimization oracles (Pethick et al., 2025), while PolarGrad places polar-based updates within a matrix-aware preconditioning framework (Lau et al., 2025). These perspectives distinguish the geometry defining an update from its scalar normalization. Within this setting, BulkBoost's exact target retains the singular frame of M_t and reallocates gains between two bands while preserving the Frobenius norm of its polar factor Q_t.

A related line uses accumulated gradient statistics for matrix preconditioning. Shampoo constructs tensor-structured preconditioners (Gupta et al., 2018); SOAP performs Adam-style adaptation in Shampoo's eigenbasis (Vyas et al., 2024). DeVA separates variance adaptation from scale-invariant vector and matrix update rules (Song et al., 2026). COSMOS combines SOAP-style adaptation in a leading eigensubspace with Muon on its complement (Liming Liu et al., 2026), illustrating the use of distinct optimization rules in different spectral subspaces. BulkBoost instead applies one norm-matched reweighting rule with a common gain within each band. Its noise-calibrated variant uses split-minibatch statistics to select band membership, not to assign direction-specific second-moment gains; its fixed-rank variant requires no noise estimate.

A.2 Spectral Shaping Beyond Muon's Flat Profile

Several recent studies replace Muon's equal-gain profile with a nonconstant spectral map. Shumaylov et al. (2026) show that random and inverted spectra can remain effective and introduce Freon through a family of Schatten-based update rules. Muon^p uses fractional spectral powers (Dong & Sawin, 2026), while DynMuon varies the shaping profile over training (Wu et al., 2026). HTMuon uses heavy-tailed spectral correction (Pang et al., 2026), and Softsign introduces smooth alternatives to sign updates, including the matrix optimizer SoftMuon (Feoktistov et al., 2026). These studies show that exact spectral flattening is one effective allocation rather than a necessary choice.

Spectral clipping provides another alternative. The Spectra framework of Jiang et al. (2026) studies clipping before and after preconditioning, with efficient soft spectral clipping via Newton–Schulz iterations. Musec replaces full flattening with momentum spectral clipping and implements smooth saturation using coupled iterations (Liu et al., 2026). These approaches retain or modify variation along the spectrum. BulkBoost instead asks whether useful reallocation can be captured by a norm-matched two-band profile, *without* fine-grained spectral shaping. Its exact target assigns equal gains within each band, with fixed-rank and noise-calibrated rules determining band membership separately from the relative bulk-to-spike gain.

A.3 Head–Bulk Allocation and the Closest Related Methods

The Spectra optimizer of Huang et al. (2026) is distinct from the similarly named Spectra framework of Jiang et al. (2026). It tracks a leading momentum subspace, replaces its singular values with a tail-scale estimate, and retains the residual tail before RMS-normalizing the update. Its leading-subspace intervention and warm-started iteration are closely related to our construction. The central difference is the reference spectrum: Spectra retains the raw tail's unequal singular magnitudes, whereas BulkBoost's exact target reweights the polar factor Q_t with equal gains within each band. Both fixed-rank and noise-calibrated BulkBoost use this two-level rule, differing in band selection. Under norm matching, relative head suppression and relative bulk emphasis are not oppo-

site reallocations. The meaningful distinctions are the underlying spectral profile, partition rule, and normalization, rather than the terminology of suppression versus amplification.

A.4 Random-Matrix Noise Edges and Spectral Estimation

The Marchenko–Pastur law describes the limiting spectrum of appropriate random covariance matrices (Marchenko & Pastur, 1967). Spiked-model theory characterizes transitions in leading empirical eigenvalues (Baik et al., 2005), while rectangular perturbation theory analyzes singular values and singular-vector alignment (Benaych-Georges & Nadakuditi, 2012). Optimal singular-value shrinkage studies reconstruction of an underlying matrix under specified noise and loss models (Gavish & Donoho, 2017). These results provide statistical reference scales and estimation criteria, but do not determine which spectral allocation best reduces evaluation loss after an optimizer update.

Gradient-noise measurements have also been used to characterize useful batch sizes and training efficiency (McCandlish et al., 2018). The noise-calibrated variant of BulkBoost uses split-minibatch differences to estimate the raw full-minibatch noise variance. It rescales this estimate by the squared global clipping factor before smoothing and uses the reference-model Nesterov correction to calibrate an edge for M_t (Section 3.1). The resulting white-noise reference selects band membership; the relative gain ρ remains a separate optimization choice. The edge is neither an exact signal–noise separator nor a prescription to discard below-edge modes. The fixed-rank variant selects its partition without estimating a noise edge.

A.5 Polar Computation, Subspace Tracking, and Low-Rank Updates

Iterative polar decomposition has a classical numerical foundation (Higham, 1986). Polar Express develops optimized matrix-sign iterations and their use in Muon (Amsel et al., 2026a). Randomized range finding and subspace iteration provide standard tools for estimating leading singular spaces without a full decomposition (Halko et al., 2011). Spectral scaling measurements further highlight the role of finite-iteration accuracy: momentum singular-value quantiles vary with layer and model scale, affecting which tail directions Newton–Schulz iterations resolve (Magakyan et al., 2026). Our realization combines Muon's numerical polar approximation $\widehat{Q}_t$ with warm-started block iteration to approximate the selected spike subspace. The noise-calibrated tracker augments the retained basis with fresh probes, whereas fixed-rank tracking uses a block of prescribed width. Neither construction explicitly computes individual bulk singular vectors. The shared bulk gain determines how these numerical ingredients combine into the prescribed reweighted target.

Low-rank optimization methods address related computational objectives. GaLore projects gradients to reduce optimizer-state memory (Zhao et al., 2024), while Dion uses distributed, amortized low-rank orthonormalized updates with error feedback (Ahn et al., 2025). Dion3 combines algorithmic and systems techniques for efficient orthogonal updates (Amsel et al., 2026b). In BulkBoost, the low-rank object is instead a correction to a scaled polar update (Theorem 2). For full-rank M_t and $\rho > 0$, the exact two-level target remains full rank; only its correction is confined to the selected spike subspace. The numerical realization matches $\|\widehat{Q}_t\|_F$, rather than imposing the exact-polar norm $\sqrt{d}$. Its tracking cost depends on the trial-space width and iteration count as well as the retained rank, so avoiding an explicit SVD does not by itself establish a runtime advantage.

A.6 Curvature, Noise Geometry, and Continuation Utility

Pre-update gradient alignment captures only part of update quality. Wang et al. (2026) study Muon's advantage through directional curvature, distinguishing first-order gain from the second-order penalty. Quadratic Spectral Descent incorporates a quadratic surrogate under a spectral-norm constraint (Zhang et al., 2026), while Beneventano et al. (2026) analyze spectral dynamics and noise geometry under Muon. These studies complement spectral-allocation comparisons by examining how curvature and transformed noise affect optimization, beyond singular magnitudes alone.

Differentiating evaluation loss through training has a history in hyperparameter optimization and bilevel learning (Maclaurin et al., 2015; Franceschi et al., 2018). Our continuation utility applies this sensitivity viewpoint to a fixed-frame spectral intervention: it is the negative derivative of post-update or later evaluation loss with respect to a current applied singular-direction gain, evaluated at

MUON's equal-gain allocation. Subsequent training follows a fixed policy, while gradients and optimizer states can respond to the intervention. For either fixed-rank or noise-calibrated band selection, Theorem 1 characterizes local allocation preferences and the fraction of the maximal first-order improvement rate accessible to two bands under norm matching. This is an analysis of the exact spectral target, not an online hypergradient procedure for selecting ρ.

Recent work on optimizer-memory schedules further emphasizes the roles of training horizon and memory timescale in optimizer comparisons (Everett & Qiu, 2026). Our pre-update alignment diagnostics, single-update loss measurements, and continued-pretraining experiments examine distinct aspects of spectral allocation: immediate alignment, the loss response to a current intervention, and performance over a training run.

A.7 CONTROLLED CONTINUED-PRETRAINING EVALUATION

Pythia provides a public model suite designed for analysis across training and scale (Biderman et al., 2023). Continued pretraining on domains and tasks is an established adaptation setting (Gururangan et al., 2020). We use this setting to compare fixed-rank and noise-calibrated spectral reweighting from common pretrained checkpoints, under matched training-token budgets and a common validation protocol. This complements the from-scratch language-model training settings used by several related optimizers and evaluates both allocation and partition-selection rules during adaptation. Our comparisons distinguish validation loss and token efficiency from per-step computation and fraction-selection effort (Section 5).

B SUPPLEMENTARY SETUP AND NORMALIZATION

This appendix supplements Section 2 with normalization identities, gain diagnostics, and rank conventions, and derives the numerical norm matching used in Section 3.

B.1 EXTENDED NOTATION AND SPECTRAL PROFILES

Extended notation. For $A \in \mathbb{R}^{m\times n}$, let $d = \min\{m, n\}$ and write $A = U_A \Sigma_A V_A^\top$ for a thin SVD, where $U_A \in \mathbb{R}^{m\times d}$ and $V_A \in \mathbb{R}^{n\times d}$ have orthonormal columns, $\Sigma_A = \mathrm{diag}(\sigma_{A,1}, \ldots, \sigma_{A,d})$, and $\sigma_{A,1} \geq \cdots \geq \sigma_{A,d} \geq 0$. For $\mathcal{I} \subseteq [d]$, let $U_{A,\mathcal{I}}$ and $V_{A,\mathcal{I}}$ collect the selected singular vectors. Their left and right orthogonal projectors are

$$P_{A,\mathcal{I}} = U_{A,\mathcal{I}} U_{A,\mathcal{I}}^\top, \qquad R_{A,\mathcal{I}} = V_{A,\mathcal{I}} V_{A,\mathcal{I}}^\top.$$

Noise-scale convention. At fixed parameters, write the raw full-minibatch gradient as $G^{\mathrm{raw}} = \overline{G} + Z$, where $\overline{G} = \mathbb{E}[G^{\mathrm{raw}}]$ and $\mathbb{E}[Z] = 0$. In the white-noise reference, s denotes the common entrywise standard deviation of Z. The gradient entering momentum is $G_t = \chi_t G_t^{\mathrm{raw}}$, where χ_t is the global clipping scalar. BB-ADAPTIVE estimates the raw variance from two minibatch halves and applies χ_t^2 before smoothing. Section 3.1 combines this running estimate with the reference-model Nesterov correction to calibrate the noise edge for M_t.

Normalization identity. For the full-rank spectral input $M_t = U_t \Sigma_t V_t^\top$, the rank-one singular modes are Frobenius-orthonormal:

$$\langle u_{t,i} v_{t,i}^\top,\, u_{t,j} v_{t,j}^\top \rangle_F = (u_{t,i}^\top u_{t,j})(v_{t,i}^\top v_{t,j}) = \mathbf{1}\{i = j\}.$$

Consequently, for a nonzero unnormalized profile w_t as in equation 3,

$$\left\| U_t \,\mathrm{diag}(w_t) V_t^\top \right\|_F^2 = \sum_{i=1}^{d} w_{t,i}^2.$$

The normalizer $c_t(w_t)$ therefore gives $\|D_t(w_t)\|_F = \sqrt{d} = \|Q_t\|_F$. The applied gain vector is $c_t(w_t) w_t$, denoted by w in Section 4.1. For either partition, the two-level profile satisfies

$$\sum_{i=1}^{d} \big(w_{t,i}^{(\rho)}\big)^2 = r_t + \rho^2 (d - r_t),$$

yielding $c_t(w_t^{(\rho)}) = c_{\rho,r_t}$ whenever the profile is nonzero. For the noise-calibrated partition, the notation $w_t^{(\rho,\tau_t)}$ makes the threshold dependence explicit.

For a scalar-induced profile $w_{t,i} = g(\sigma_{t,i})$, the equivalent matrix-function notation is

$$g(\Sigma_t) = \mathrm{diag}(g(\sigma_{t,1}), \ldots, g(\sigma_{t,d})), \qquad D_t(g) = c_t(g) U_t g(\Sigma_t) V_t^\top,$$

where $c_t(g) = c_t(w_t)$ and the induced profile is nonzero.

B.2 RELATIVE AND EFFECTIVE GAINS

Absolute gains under a relative reallocation. For a full-rank input with both bands nonempty, let $q_t = r_t/d \in (0, 1)$ denote the spike fraction. For BB-STATIC, $q_t = \lceil pd \rceil / d$, which can differ slightly from p because of rounding. The applied gains relative to the flat profile are

$$g_{\mathrm{sp}} = c_{\rho,r_t} = \frac{1}{\sqrt{q_t + \rho^2(1 - q_t)}}, \qquad g_{\mathrm{bu}} = \rho g_{\mathrm{sp}}. \tag{15}$$

At fixed q_t, increasing ρ above 1 raises g_{bu} toward $1/\sqrt{1 - q_t}$ and lowers g_{sp} toward zero. For a small spike fraction, the bulk gain therefore stays close to one, while the spike gain can fall substantially. With $q_t = 0.05$ and $\rho = 4$, $g_{\mathrm{bu}} \approx 1.02$ and $g_{\mathrm{sp}} \approx 0.26$. At the largest tested fraction, $q_t = 0.2$, the same ρ gives $g_{\mathrm{bu}} \approx 1.11$ and $g_{\mathrm{sp}} \approx 0.28$. Thus ρ sets only the ratio of the two gains, and the absolute bulk gain changes much less than ρ.

More generally, multiplying every unnormalized gain by the same positive scalar leaves the norm-matched direction unchanged. For $\rho > 0$, profiles with spike and bulk gains $(1, \rho)$ and $(1/\rho, 1)$ therefore give the same update. Relative bulk emphasis and relative spike suppression describe the same norm-matched reallocation.

Effective bulk-to-spike gain. For a general spectral profile, define the ratio of root-mean-square gains across the selected bands by

$$\rho_{\text{eff}}(w_t; \mathcal{I}_{\text{sp},t}) = \sqrt{\frac{\frac{1}{d-r_t}\sum_{i\in\mathcal{I}_{\text{bu},t}} w_{t,i}^2}{\frac{1}{r_t}\sum_{i\in\mathcal{I}_{\text{sp},t}} w_{t,i}^2}}. \tag{16}$$

This statistic is defined when both bands are nonempty and the denominator is positive. The common normalizer cancels, so the statistic is the same for w_t and its applied gains $c_t(w_t)w_t$. In particular, the exact two-level target satisfies $\rho_{\text{eff}} = \rho$. For BB-ADAPTIVE, we abbreviate it as $\rho_{\text{eff}}(w_t; \tau_t)$. The partition specifies band membership, whereas ρ_{eff} summarizes the relative allocation; it does not describe gain variation within a band or measure subspace-recovery accuracy.

B.3 Rank Conventions and Degenerate Partitions

Empty bands. For a full-rank input, reallocation between bands requires $0 < r_t < d$. If $r_t = d$, all unnormalized gains equal one and the target is Q_t. If $r_t = 0$ and $\rho > 0$, all unnormalized gains equal ρ, and normalization again gives Q_t. When $r_t = 0$ and $\rho = 0$, the profile is zero and cannot be normalized to $\sqrt{d}$. This case is excluded by the nonzero-profile condition in Section 2.

Rank-deficient inputs. For a rank-deficient spectral input, restrict the exact construction to the active indices $\mathcal{I}_{+,t} = \{i \in [d] : \sigma_{t,i} > 0\}$, with active rank $d_t^+ = |\mathcal{I}_{+,t}|$. The partial polar factor is

$$Q_t = \sum_{i\in\mathcal{I}_{+,t}} u_{t,i}v_{t,i}^\top, \qquad \|Q_t\|_F = \sqrt{d_t^+}.$$

Intersect each partition from Section 2 with $\mathcal{I}_{+,t}$, and let r_t count the selected active spike modes. In particular, the fixed-rank rule gives $r_t = \min\{\lceil pd \rceil, d_t^+\}$: the prescribed fraction still refers to $d = \min\{m, n\}$, while zero singular modes receive no gains.

For $d_t^+ > 0$ and a nonzero active gain vector, the exact normalizers are

$$c_t(w_t) = \sqrt{\frac{d_t^+}{\sum_{i\in\mathcal{I}_{+,t}} w_{t,i}^2}}, \qquad c_{\rho,r_t} = \sqrt{\frac{d_t^+}{r_t + \rho^2(d_t^+ - r_t)}}.$$

The gain diagnostics in Appendix B.2 likewise use the active band sizes. The matrix dimensions remain m, n; replacing d by d_t^+ in these identities does not change the dimensions used to calibrate the noise edge. For a zero spectral input, the partial polar factor and exact update are zero, without evaluating normalization ratios.

B.4 Derivation of Numerical Norm Matching

The numerical update in equation 11 and equation 12 uses an approximate polar factor $\widehat{Q}_t$ and a tracked basis $X_t \in \mathbb{R}^{d\times\widehat{r}_t}$. We first derive the identities in exact arithmetic, assuming that X_t has orthonormal columns; neither an exact polar factor nor exact spike recovery is required for the projection calculation.

Define

$$\widehat{Q}_{\text{sp},t} = \begin{cases} X_t(X_t^\top \widehat{Q}_t), & m \le n, \\ (\widehat{Q}_t X_t)X_t^\top, & n < m, \end{cases} \qquad \widehat{Q}_{\text{bu},t} = \widehat{Q}_t - \widehat{Q}_{\text{sp},t}. \tag{17}$$

These components use the tracked subspace and its orthogonal complement, which can differ from the exact spike and bulk bands. The raw direction is

$$\widetilde{D}_t = \widehat{Q}_{\text{sp},t} + \rho\widehat{Q}_{\text{bu},t} = \rho\widehat{Q}_t - (\rho - 1)\widehat{Q}_{\text{sp},t}.$$

Since $X_tX_t^\top$ is an orthogonal projector,

$$\left\langle \widehat{Q}_{\text{sp},t}, \widehat{Q}_{\text{bu},t} \right\rangle_F = 0.$$

Writing

$$a_t = \|\widehat{Q}_t\|_F^2, \qquad b_t = \|\widehat{Q}_{\mathrm{sp},t}\|_F^2$$

therefore gives $\|\widehat{Q}_{\mathrm{bu},t}\|_F^2 = a_t - b_t$. For nonzero $\widehat{Q}_t$ and $\widetilde{D}_t$,

$$\|\widetilde{D}_t\|_F^2 = b_t + \rho^2(a_t - b_t), \qquad \widehat{c}_{\rho,t} = \sqrt{\frac{a_t}{b_t + \rho^2(a_t - b_t)}} = \frac{\|\widehat{Q}_t\|_F}{\|\widetilde{D}_t\|_F}. \tag{18}$$

Thus $\widehat{D}_t = \widehat{c}_{\rho,t}\widetilde{D}_t$ satisfies $\|\widehat{D}_t\|_F = \|\widehat{Q}_t\|_F$. The projected norm can be evaluated on the smaller side:

$$b_t = \begin{cases} \|X_t^\top \widehat{Q}_t\|_F^2, & m \le n, \\ \|\widehat{Q}_t X_t\|_F^2, & n < m. \end{cases}$$

For $a_t > 0$ and $\rho > 0$, the raw norm is positive because $0 \le b_t \le a_t$. At $\rho = 0$, a nonzero projected spike component is required. A zero $\widehat{Q}_t$ gives a zero numerical update, without evaluating the norm ratio. At $\rho = 1$, $\widetilde{D}_t = \widehat{Q}_t$, so the update reduces to the unreweighted numerical polar direction.

Relation to the exact target. If $\widehat{Q}_t = Q_t$ and M_t has full rank, then

$$Q_t Q_t^\top = I_m \quad \text{when } m \le n, \qquad Q_t^\top Q_t = I_n \quad \text{when } n < m.$$

Consequently, $a_t = d$ and $b_t = \widehat{r}_t$ for any orthonormal tracked basis, giving $\widehat{c}_{\rho,t} = c_{\rho,\widehat{r}_t}$ whenever the raw direction is nonzero. This equality of normalizers does not require an exact spike subspace. If X_t also spans the selected spike subspace, then $\widehat{r}_t = r_t$ and the numerical construction recovers the exact target $D_t(w_t^{(\rho)})$. For an approximate polar factor, a_t need not equal d and b_t need not equal $\widehat{r}_t$, so the normalizers can differ. For rank-deficient inputs, the direct norm ratio remains applicable, but the full-rank simplification above does not apply in general.

Finite-precision computation. The direct norm ratio in equation 12 requires only Frobenius norms and rescaling. Its norm-matching property does not rely on exact orthogonality of the computed components, unlike the expression involving $a_t - b_t$. For a nonzero computed raw direction, exact evaluation of the ratio matches its norm to $\|\widehat{Q}_t\|_F$; finite-precision evaluation is subject to rounding error. Norm matching controls magnitude without correcting polar or subspace approximation errors.

C Implementation Details and SVD-Free Algorithm

This appendix gives the implementation details behind Section 3. It covers noise calibration, the exact-SVD target used in oracle comparisons, spike tracking for both variants, and the full BB-ADAPTIVE algorithm. BB-STATIC and BB-ADAPTIVE share the reweighting and normalization of Section 2. They differ only in how they choose and track the spike band. Appendix B.4 derives numerical norm matching, and Proposition 4 derives the half-batch noise estimate and the Nesterov noise factor.

C.1 Calibration and Estimator State

Half-batch noise estimate. The two halves in equation 5 are sampled independently at the same parameters, so they have the same mean. This mean cancels in the half-batch difference Z_t^{split}. After vectorization, Z_t^{split} has the same covariance as the sampling noise in G_t^{raw}. Hence $\widehat{s}_t^2 = \|Z_t^{\mathrm{split}}\|_F^2/(mn)$ estimates the average entrywise noise variance of the full-minibatch gradient. For isotropic noise, this is the common entrywise variance. For anisotropic noise, one scalar cannot describe the full covariance.

Clipping and variance smoothing. Let $\chi_t \in (0, 1]$ be the global clipping scalar. It is computed from the raw full-minibatch gradient of all model parameters, and all matrices share it. Each matrix uses $G_t = \chi_t G_t^{\mathrm{raw}}$ in the momentum recursion. The halves are not clipped separately. Instead, each matrix's raw variance estimate is scaled by χ_t^2 before smoothing:

$$\overline{s}_t^2 = \begin{cases} \chi_t^2 \widehat{s}_t^2, & \text{if the variance state is uninitialized,} \\ \gamma \overline{s}_{t-1}^2 + (1-\gamma)\chi_t^2 \widehat{s}_t^2, & \text{otherwise.} \end{cases}$$

The EMA averages variances. The square root $\sqrt{\overline{s}_t^2}$ in equation 8 is taken afterwards. For a matrix first observed at step 1, this gives $\overline{s}_1^2 = \chi_1^2 \widehat{s}_1^2$. Without clipping, $\chi_t = 1$.

Nesterov factor. Proposition 4 derives the factor $\nu_t(\beta)$ in equation 7 under a reference model: centered, isotropic noise that is independent over time and has constant variance, with $B_0 = 0$. Writing $M_t = (1-\beta^2)G_t + \beta^2 B_{t-1}$ gives an equivalent two-term form:

$$\nu_t^2(\beta) = (1-\beta^2)^2 + \beta^4 \frac{1-\beta}{1+\beta}\left(1 - \beta^{2(t-1)}\right).$$

The first term comes from the current gradient, which enters M_t both directly and through the updated buffer. The second term comes from the past gradients stored in B_{t-1}.

Approximations. The implementation replaces the reference variance with the clipping-scaled running estimate $\overline{s}_t^2$. This puts the estimate on the scale of the applied gradient. Because χ_t depends on the data, the scaled estimate is in general biased for the noise in the clipped gradient. The running estimate also assumes that the noise level changes slowly. Finally, the noise edge assumes Gaussian white noise (Section 2). The multiplier κ_{edge} can shift the threshold, but a single scalar cannot correct for anisotropic or correlated noise in general. The experiments use $\gamma = 0.95$ and $\kappa_{\mathrm{edge}} = 1$.

C.2 Exact Target and Oracle Comparison

For a full-rank input M_t and a nonzero two-level profile, the exact-SVD oracle evaluates, for either rule,

$$D_t^{\mathrm{svd}} = D_t\left(w_t^{(\rho)}\right) = c_{\rho,r_t} U_t \operatorname{diag}\left(w_t^{(\rho)}\right) V_t^\top. \tag{19}$$

For BB-ADAPTIVE, r_t counts all singular values above the threshold $\tau_t = \widehat{\tau}_t$. For BB-STATIC, $r_t = \lceil pd \rceil$. In both cases, c_{ρ,r_t} is the normalizer in equation 4. Rank-deficient and zero inputs follow Appendix B.3.

Here, *oracle* only means that the rule is evaluated exactly with an SVD. It does not choose the partition or the gain. An oracle comparison fixes M_t, the rule, and ρ. Both realizations therefore use the same threshold for BB-ADAPTIVE and the same rank for BB-STATIC. The exact oracle matches

$\|Q_t\|_F$, and the numerical realization matches $\|\widehat{Q}_t\|_F$. If $\widehat{Q}_t = Q_t$ and the orthonormal tracked basis spans the selected spike subspace, the numerical update equals the oracle (Appendix B.4).

For BB-ADAPTIVE, the rank cap applies only to the numerical tracker. The exact target keeps every direction above the threshold. Both sides use the same spectral rule, so a comparison at a common input measures only numerical error: polar approximation, imperfect subspace recovery, and, for BB-ADAPTIVE, truncation by the cap.

C.3 WARM-STARTED BLOCK ITERATION

This subsection first gives the tracker for BB-ADAPTIVE and then the simpler tracker for BB-STATIC. Both use the smaller-side Gram operator $K_t \in \mathbb{R}^{d\times d}$ from equation 10. Its action on a block Y is

$$K_t Y = \begin{cases} M_t(M_t^\top Y), & m \le n, \\ M_t^\top(M_t Y), & n < m, \end{cases}$$

so the Gram matrix is never formed. As in Section 3.2, orth returns an orthonormal basis of the column space of its argument. An empty column space gives an empty basis.

Initialization and trial-space construction. Let X_{t-1} be the spike basis that BB-ADAPTIVE retained at the previous step, with $\widehat{r}_{t-1}$ columns. Before the first update, X_0 is empty and $\widehat{r}_0 = 0$. For positive initial and refresh probe widths n_{init} and n_{probe}, set

$$n_{\text{fresh},t} = \begin{cases} \min\{d, n_{\text{init}}\}, & \widehat{r}_{t-1} = 0, \\ \min\{d, n_{\text{probe}}\}, & \widehat{r}_{t-1} > 0. \end{cases}$$

The initial width is also used whenever the previous step selected no directions.

Draw $\Omega_t \in \mathbb{R}^{d\times n_{\text{fresh},t}}$ with independent standard Gaussian entries and set

$$Y_0 = \text{orth}([X_{t-1}, \Omega_t]).$$

The concatenated block may have more than d columns, but its orthonormal basis has at most d. The previous basis gives a warm start, and the fresh probes let new directions enter the trial space. At the first step, Y_0 comes from the probes alone. The first retained basis X_1 is obtained only after iteration and threshold selection.

Block iteration and Ritz extraction. Within training step t, perform q block-power iterations:

$$Y_{j+1} = \text{orth}(K_t Y_j), \qquad j = 0, \dots, q-1. \tag{20}$$

Let ℓ_t be the number of columns of Y_q. For a nonempty trial basis, form

$$T_t = Y_q^\top K_t Y_q = O_t \Lambda_t O_t^\top, \qquad \lambda_{t,1} \ge \cdots \ge \lambda_{t,\ell_t} \ge 0,$$

where $\Lambda_t = \text{diag}(\lambda_{t,1}, \dots, \lambda_{t,\ell_t})$. The Ritz directions are the columns of $Y_q O_t$, with singular-value estimates $\sqrt{\lambda_{t,i}}$. The eigenvalues are nonnegative in exact arithmetic because K_t is positive semidefinite.

For a nonnegative integer rank cap $r_{\max}$, let $n_{\text{cand},t}$ count the Ritz values above the threshold:

$$n_{\text{cand},t} = \left|\{i \in [\ell_t] : \lambda_{t,i} > \widehat{\tau}_t^2\}\right|, \qquad \widehat{r}_t = \min\{n_{\text{cand},t}, r_{\max}\}, \qquad X_t = \text{orth}\big(Y_q O_{t,[\widehat{r}_t]}\big). \tag{21}$$

Here $O_{t,[\widehat{r}_t]}$ holds the first $\widehat{r}_t$ eigenvectors in decreasing eigenvalue order. If the cap binds, the largest estimates above the threshold are kept. An empty selection gives an empty spike basis. If $\ell_t = 0$, set $\widehat{r}_t = 0$ and leave X_t empty; no eigendecomposition is needed. The experiments use this hard threshold without hysteresis.

The selected Ritz block is reorthonormalized before it is stored. The projector identities in Appendix B.4 assume orthonormal columns, so in finite precision they hold only up to orthonormalization error.

Experimental settings and random-number state. All BB-ADAPTIVE training runs use

$$q = 6, \qquad n_{\text{init}} = 128, \qquad n_{\text{probe}} = 128, \qquad r_{\max} = 256.$$

Probe widths are limited by d. The cap $r_{\max}$ is an absolute number of directions and does not scale with d. The retained rank satisfies $\widehat{r}_t \le \min\{\ell_t, r_{\max}\}$, with $\ell_t \le d$. The first retained basis has at most $\min\{d, 128\}$ columns, below the cap. The cap applies only to the retained basis. In these runs, the trial space has up to $\min\{d, \widehat{r}_{t-1} + 128\}$ columns, which can exceed the cap.

Gaussian probes are generated in float32 with a dedicated pseudorandom-number generator for each parameter matrix, separate from the training random-number stream. Its state carries over from step to step and is never reseeded.

Fixed-rank tracking (BB-STATIC). BB-STATIC tracks a block of fixed width $r = \lceil pd \rceil$. The fraction p is shared across matrices and fixed for the whole run. If a stored basis $X_{t-1} \in \mathbb{R}^{d \times r}$ exists, it is the warm start, $Y_0 = X_{t-1}$. Otherwise, as at the first step, we draw a Gaussian block $\Omega_t \in \mathbb{R}^{d \times r}$ and set $Y_0 = \operatorname{orth}(\Omega_t)$. At each training step, we run

$$Y_{j+1} = \operatorname{orth}(K_t Y_j), \qquad j = 0, \ldots, q-1, \qquad X_t = Y_q,$$

with $q = 6$ in the experiments. The whole final basis enters equation 11 and is stored for the next step. Unlike the BB-ADAPTIVE tracker, this tracker adds no fresh probes, does no Ritz extraction, and applies no threshold or rank cap. The parameters n_{init}, n_{probe}, and $r_{\max}$ therefore do not apply. For example, $p = 0.20$ and $d = 1024$ give $r = 205$, so initialization and all six iterations use a 1024×205 block.

C.4 FULL ALGORITHM

Algorithm 1 gives one BB-ADAPTIVE step for one matrix. It combines noise calibration, spike tracking, and numerical norm matching. It returns the update direction before learning-rate scaling. The clipping scalar χ_t is computed once from the full-model gradient and passed in. For BB-STATIC, the noise-estimate and threshold steps are skipped, and the fixed-rank tracker of Appendix C.3 replaces the adaptive one.

Before the first update, $B_0 = 0$, X_0 is empty, and the variance state is uninitialized. The algorithm returns B_t, X_t, and $\overline{s}_t^2$ along with $\widehat{D}_t$. The probe generator's state is kept separately for each matrix. Since $\rho > 0$, a nonzero $\widehat{Q}_t$ and an orthonormal X_t give a nonzero raw direction in exact arithmetic, so the norm ratio is defined. Appendix B.4 covers $\rho = 0$ and finite-precision effects.

Algorithm 1 SVD-free BB-ADAPTIVE update for one matrix

Inputs: Mean gradients $G_t^{(1)}, G_t^{(2)} \in \mathbb{R}^{m\times n}$ of the two minibatch halves, before clipping; global clipping scalar $\chi_t \in (0,1]$; previous states B_{t-1}, X_{t-1}; previous variance state $\overline{s}_{t-1}^2$ (possibly uninitialized); the matrix's probe generator; step $t \geq 1$. Write $d = \min\{m,n\}$.
Parameters: Bulk-to-spike gain $\rho > 0$; momentum $\beta \in [0,1)$; smoothing $\gamma \in [0,1)$; edge multiplier $\kappa_{\text{edge}} > 0$; power iterations q; initial probe width n_{init}; refresh probe width n_{probe}; absolute rank cap $r_{\max}$.

$Z_t^{\text{split}} \leftarrow (G_t^{(1)} - G_t^{(2)})/2$
$\widehat{s}_t^2 \leftarrow \|Z_t^{\text{split}}\|_F^2/(mn)$
$G_t^{\text{raw}} \leftarrow (G_t^{(1)} + G_t^{(2)})/2$
$G_t \leftarrow \chi_t G_t^{\text{raw}}$
if the variance state is uninitialized **then**
  $\overline{s}_t^2 \leftarrow \chi_t^2 \widehat{s}_t^2$
else
  $\overline{s}_t^2 \leftarrow \gamma \overline{s}_{t-1}^2 + (1-\gamma)\chi_t^2 \widehat{s}_t^2$
end if
$B_t \leftarrow \beta B_{t-1} + (1-\beta)G_t$
$M_t \leftarrow (1-\beta)G_t + \beta B_t$
Compute $\nu_t(\beta)$ from equation 7
$\widehat{\tau}_t \leftarrow \kappa_{\text{edge}}\nu_t(\beta)\sqrt{\overline{s}_t^2}(\sqrt{m}+\sqrt{n})$
$\widehat{Q}_t \leftarrow \text{PolarExpress}(M_t)$
Define the Gram action using equation 10
if X_{t-1} is empty **then**
  $f \leftarrow \min\{d, n_{\text{init}}\}$
else
  $f \leftarrow \min\{d, n_{\text{probe}}\}$
end if
Draw $\Omega_t \in \mathbb{R}^{d\times f}$ with independent $\mathcal{N}(0,1)$ entries using this matrix's probe generator
$Y_0 \leftarrow \text{orth}([X_{t-1}, \Omega_t])$
for $j = 0, \ldots, q-1$ **do**
  $Y_{j+1} \leftarrow \text{orth}(K_t Y_j)$
end for
$\ell_t \leftarrow$ number of columns of Y_q
if $\ell_t = 0$ **then**
  $\widehat{r}_t \leftarrow 0$; set X_t empty
else
  Compute $Y_q^\top K_t Y_q = O_t \Lambda_t O_t^\top$
  Order eigenpairs by decreasing eigenvalue
  $n_{\text{cand},t} \leftarrow |\{i \in [\ell_t] : (\Lambda_t)_{ii} > \widehat{\tau}_t^2\}|$
  $\widehat{r}_t \leftarrow \min\{n_{\text{cand},t}, r_{\max}\}$
  $X_t \leftarrow \text{orth}(Y_q O_{t,[\widehat{r}_t]})$
end if
if $\|\widehat{Q}_t\|_F = 0$ **then**
  $\widehat{D}_t \leftarrow 0$
else
  Form $\widetilde{D}_t$ using equation 11
  $\widehat{D}_t \leftarrow \dfrac{\|\widehat{Q}_t\|_F}{\|\widetilde{D}_t\|_F}\widetilde{D}_t$
end if
return $\widehat{D}_t, B_t, X_t, \overline{s}_t^2$

Noise calibration. The half-batch difference estimates the average entrywise noise variance of the full-minibatch mean gradient. The clipping scalar rescales the gradient by χ_t and the variance estimate by χ_t^2. The EMA averages variances; the square root and the Nesterov factor are applied afterwards.

Spike tracking. Fresh Gaussian probes are added at every step: n_{init} when the previous basis is empty and n_{probe} otherwise. A nonempty previous basis also gives a warm start. The trial basis has at most d columns. Hard-threshold selection keeps at most $r_{\max}$ Ritz directions, which are then reorthonormalized.

Persistent state. The momentum buffer, retained basis, and variance estimate carry over between steps. The probe generator advances when probes are drawn and is never reseeded.

Norm matching. In exact arithmetic, $\|\widehat{D}_t\|_F = \|\widehat{Q}_t\|_F$. At $\rho = 1$, the update is the unreweighted direction $\widehat{Q}_t$. Norm matching fixes the magnitude only; it does not correct polar or subspace approximation errors.

D BAND CONTENT ALONG MUON TRAJECTORIES

Does the bulk, whose singular values lie at or below the noise edge, contribute meaningfully to reference-gradient alignment under the flat profile? We examine this question along MUON trajectories of continued pretraining, measuring band sizes, per-direction alignment, and signed contributions across matrices. The measured bulk is *weak per direction but substantial in aggregate*: it contains 93.7–97.1% of pooled singular directions and has positive signed aggregate alignment in every analyzed matrix. These observations motivate a broad-bulk/small-spike partition and the fixed fractions tested in Section 5. Whether increasing the bulk's relative gain improves an update is a separate question, which Appendix E examines.

D.1 PROTOCOL AND FRAME-CONSISTENT CALIBRATION

Training trajectories and checkpoints. For each of FineMath-4, CodeParrot, and Wiki-ar, we analyze one Pythia-70M MUON trajectory of continued pretraining. Each run uses approximately 100M tokens, with batches of 32 sequences of length 1024, momentum $\beta = 0.95$, global gradient clipping at norm 1, and Newton–Schulz iterations for the polar map, as in the MUON baseline of Section 5. The matrix optimizer and the auxiliary AdamW optimizer both use peak learning rate 5×10^{-4}, the shared Pythia-70M rate of Section 5. These diagnostics observe the MUON trajectories without changing their updates.

We report four checkpoints as 0.0M, 25.0M, 50.0M, and 99.9M consumed training tokens. The initial measurement is a transient probe before continued pretraining; subsequent measurements precede updates after 24,969,216, 49,971,200, and 99,942,400 tokens. At each checkpoint, we analyze all 24 hidden weight matrices, excluding embeddings and the language-model head. For matrix ℓ with dimensions $m_\ell \times n_\ell$, write $d_\ell = \min\{m_\ell, n_\ell\}$. Here each matrix supplies 512 thin-SVD directions, giving $N_{\mathrm{dir}} = \sum_\ell d_\ell = 12{,}288$ directions per corpus–checkpoint–frame combination.

An independent reference gradient. At fixed model parameters, construct the large-batch reference

$$H_t = \frac{1}{16} \sum_{j=1}^{16} G_t^{\mathrm{ref},j}, \tag{22}$$

where each $G_t^{\mathrm{ref},j}$ is an unclipped mean gradient from an independently sampled full minibatch of the same corpus. The reference contains 512 sequences, or 524,288 input tokens; $H_{t,\ell}$ denotes its block for matrix ℓ. It is neither clipped nor used for parameter updates. Reference sampling uses a separate data random-number stream, and the PyTorch random-number state is restored afterward. Full thin SVDs are computed only at diagnostic checkpoints.

Raw-gradient and Nesterov frames. As in Section 3.1, distinguish the raw gradient $G_{t,\ell}^{\mathrm{raw}}$ from the clipped gradient $G_{t,\ell} = \chi_t G_{t,\ell}^{\mathrm{raw}}$, where χ_t is the full-model clipping scalar. Using the actual previous momentum buffer, reconstruct

$$\begin{aligned} B_{t,\ell} &= \beta B_{t-1,\ell} + (1-\beta) G_{t,\ell}, \\ M_{t,\ell} &= (1-\beta) G_{t,\ell} + \beta B_{t,\ell}. \end{aligned} \tag{23}$$

We analyze $A_{t,\ell}^G = G_{t,\ell}^{\mathrm{raw}}$ in the *raw-gradient frame* and $A_{t,\ell}^M = M_{t,\ell}$ in the *Nesterov frame*, against the same $H_{t,\ell}$. The first is the current minibatch gradient. The second is the input to MUON's polar map, which combines the current gradient with the momentum buffer.

A noise edge matched to each frame. For each training minibatch, compute the raw mean gradients of the two halves and form

$$G_{t,\ell}^{\mathrm{raw}} = \frac{G_{t,\ell}^{(1)} + G_{t,\ell}^{(2)}}{2}, \qquad Z_{t,\ell}^{\mathrm{split}} = \frac{G_{t,\ell}^{(1)} - G_{t,\ell}^{(2)}}{2}. \tag{24}$$

Let $\widehat{v}_{t,\ell} = \|Z_{t,\ell}^{\mathrm{split}}\|_F^2 / (m_\ell n_\ell)$ be the raw half-batch variance estimate, which corresponds to $\widehat{s}_t^2$ in Section 3.1. The diagnostics keep both a raw and a clipping-scaled variance EMA:

$$\begin{aligned} v_{t,\ell}^{\mathrm{raw}} &= \gamma v_{t-1,\ell}^{\mathrm{raw}} + (1-\gamma) \widehat{v}_{t,\ell}, \\ v_{t,\ell}^{\mathrm{clip}} &= \gamma v_{t-1,\ell}^{\mathrm{clip}} + (1-\gamma) \chi_t^2 \widehat{v}_{t,\ell}. \end{aligned} \tag{25}$$

Initialize each EMA directly from its first observation, then apply these recursions every training step with $\gamma = 0.95$. The clipping-scaled state is $\overline{s}_t^2$ of Section 3.1 for matrix ℓ; the raw state serves the raw-gradient frame. The squared clipping factor is applied before smoothing.

Using the Nesterov variance multiplier

$$\nu_t^2(\beta) = (1-\beta^2)^2 + \beta^4 \frac{1-\beta}{1+\beta}\big(1-\beta^{2(t-1)}\big), \qquad t \geq 1, \tag{26}$$

from Proposition 4, set

$$\begin{aligned} \tau_{t,\ell}^G &= \kappa_{\text{edge}} \sqrt{v_{t,\ell}^{\text{raw}}}(\sqrt{m_\ell} + \sqrt{n_\ell}), \\ \tau_{t,\ell}^M &= \kappa_{\text{edge}} \nu_t(\beta) \sqrt{v_{t,\ell}^{\text{clip}}}(\sqrt{m_\ell} + \sqrt{n_\ell}), \end{aligned} \tag{27}$$

with $\kappa_{\text{edge}} = 1$ and $\nu_t(\beta) \geq 0$. Thus the raw-gradient frame uses the raw noise scale, and the Nesterov frame uses the clipping-scaled noise scale adjusted for momentum. In each frame, singular values strictly above the threshold are spikes; the remaining directions form the bulk. Selection uses the full computed spectrum, without a tracker rank cap. As in Section 2, these thresholds are approximate references based on white noise.

The initial Nesterov measurement uses a transient first-step probe with a zero buffer. Direct initialization of the variance states gives

$$M_{1,\ell} = (1-\beta^2)\chi_1 G_{1,\ell}^{\text{raw}}, \qquad \tau_{1,\ell}^M = (1-\beta^2)\chi_1 \tau_{1,\ell}^G.$$

The matched positive rescaling preserves the initial partition and polar direction, which explains the agreement between frames at 0.0M tokens. The probe leaves the training buffer and variance states unchanged.

What per-direction alignment measures. For $F \in \{G, M\}$, write $A_{t,\ell}^F = U_{t,\ell}^F \Sigma_{t,\ell}^F (V_{t,\ell}^F)^\top$ and define

$$\alpha_{t,\ell,i}^F = (u_{t,\ell,i}^F)^\top H_{t,\ell} v_{t,\ell,i}^F. \tag{28}$$

This is the signed projection of the reference gradient onto the unit-Frobenius matrix $u_{t,\ell,i}^F (v_{t,\ell,i}^F)^\top$, without normalization by $\|H_{t,\ell}\|_F$, so it is not a cosine similarity. A positive value means that a step along $-u_{t,\ell,i}^F (v_{t,\ell,i}^F)^\top$ lowers the reference objective to first order. Under the flat profile, the scores sum to the polar alignment:

$$\sum_i \alpha_{t,\ell,i}^F = \big\langle H_{t,\ell}, \operatorname{polar}(A_{t,\ell}^F) \big\rangle_F . \tag{29}$$

These identities concern the exact polar targets of Section 2, with the rank conventions of Appendix B.3. Alignment with the raw input matrix instead weights each $\alpha_{t,\ell,i}^F$ by its singular value. The raw-gradient-frame polar direction is only a diagnostic comparison; MUON uses the Nesterov frame.

Signed and absolute contributions within a matrix. Suppress the frame, time, and matrix indices temporarily. For a nonempty band $\mathcal{I}$, its mean absolute alignment is $\overline{a}(\mathcal{I}) = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} |\alpha_i|$. Its signed and absolute shares are

$$S(\mathcal{I}) = \frac{\sum_{i \in \mathcal{I}} \alpha_i}{\sum_{j=1}^{d_\ell} \alpha_j}, \qquad S_{\text{abs}}(\mathcal{I}) = \frac{\sum_{i \in \mathcal{I}} |\alpha_i|}{\sum_{j=1}^{d_\ell} |\alpha_j|}, \tag{30}$$

when the respective denominators are nonzero. The signed share measures the net contribution after positive and negative projections cancel; the absolute share measures projection magnitude without cancellation. A signed share can be negative or exceed one, and it can be sensitive to a small total signed denominator. For applied gains $w = (w_i)$, the contribution of direction i becomes $w_i \alpha_i$. A common positive normalizer cancels from these within-matrix shares. In general, it does not cancel from sums across matrices with different normalizers.

Corpus	Tokens (M)	Bulk (%)	Γ ($\times$)	S_{bu} (%)	$S_{\mathrm{bu}}^{\mathrm{abs}}$ (%)
FineMath-4	0.0	95.3	66.4	22.6	23.5
	25.0	97.1	85.1	25.8	28.0
	50.0	96.6	92.3	19.7	23.6
	99.9	96.2	33.2	18.6	43.5
CodeParrot	0.0	95.5	74.8	22.6	22.3
	25.0	96.5	72.0	26.0	27.8
	50.0	96.6	64.9	25.6	30.2
	99.9	96.6	58.8	16.6	32.9
Wiki-ar	0.0	93.7	86.7	14.6	14.6
	25.0	94.8	64.8	21.4	22.0
	50.0	94.5	47.0	25.5	26.9
	99.9	94.6	40.7	18.1	30.3

Table 3: **Raw-gradient-frame band content.** Each row pools 24 matrices and 12,288 singular directions from a Pythia-70M MUON trajectory. Bulk (%) is the pooled fraction of directions whose singular values lie at or below the raw-gradient noise edge $\tau_{t,\ell}^{G}$; Γ is the ratio of mean absolute per-direction alignment in spikes to bulk. S_{bu} and $S_{\mathrm{bu}}^{\mathrm{abs}}$ are the signed and absolute shares in equation 32, computed from pooled sums. Every analyzed matrix has positive signed bulk alignment. Token labels are rounded.

How the tables pool matrices. The tables pool counts and alignment sums first and then form ratios; they do not average matrix-wise shares. Suppressing checkpoint and frame indices, define for $b \in \{\mathrm{sp}, \mathrm{bu}\}$

$$N_b = \sum_{\ell} |\mathcal{I}_{b,\ell}|,$$
$$A_b = \sum_{\ell} \sum_{i \in \mathcal{I}_{b,\ell}} \alpha_{\ell,i}, \qquad B_b = \sum_{\ell} \sum_{i \in \mathcal{I}_{b,\ell}} |\alpha_{\ell,i}|. \tag{31}$$

We report the bulk fraction $N_{\mathrm{bu}}/N_{\mathrm{dir}}$, the mean absolute per-direction alignments $a_b = B_b/N_b$, and their spike-to-bulk ratio $\Gamma = a_{\mathrm{sp}}/a_{\mathrm{bu}}$. The pooled bulk shares are

$$S_{\mathrm{bu}} = \frac{A_{\mathrm{bu}}}{A_{\mathrm{sp}} + A_{\mathrm{bu}}}, \qquad S_{\mathrm{bu}}^{\mathrm{abs}} = \frac{B_{\mathrm{bu}}}{B_{\mathrm{sp}} + B_{\mathrm{bu}}}. \tag{32}$$

To quantify cancellation, we also report

$$C = \frac{|A_{\mathrm{all}}|}{B_{\mathrm{all}}}, \qquad A_{\mathrm{all}} = A_{\mathrm{sp}} + A_{\mathrm{bu}}, \qquad B_{\mathrm{all}} = B_{\mathrm{sp}} + B_{\mathrm{bu}}. \tag{33}$$

Small C means that opposing projections largely cancel. All pooled denominators in the reported measurements are positive. No layer-specific learning-rate or shape multipliers enter these sums, so they describe the alignment of polar directions only. Neither BULKBOOST variant uses H_t online.

D.2 RESULTS: A BROAD BULK WITH POSITIVE AGGREGATE ALIGNMENT

Many weak directions contribute collectively. Tables 3 and 4 report the raw-gradient and Nesterov frames, respectively. Across the three corpora and four checkpoints, the bulk contains 93.7–97.1% of pooled directions. On average, spike directions have much larger absolute reference-gradient projections: Γ ranges from 33.2 to 92.3 in the raw-gradient frame and from 18.7 to 86.7 in the Nesterov frame. Despite these per-direction differences, the bulk contributes 14.6–26.0% of pooled signed polar alignment in the raw-gradient frame and 10.0–63.5% in the Nesterov frame. Thus, *weak average per-direction alignment does not imply a negligible collective contribution*.

Positive alignment is present in every measured matrix. All 24 matrices have positive signed bulk sums at every checkpoint in both frames and all three corpora: $3 \times 4 \times 2 \times 24 = 576$ recorded combinations. Positive bulk alignment is therefore spread across all layers.

Corpus	Tokens (M)	Bulk (%)	Γ ($\times$)	S_{bu} (%)	$S_{\mathrm{bu}}^{\mathrm{abs}}$ (%)
FineMath-4	0.0	95.3	66.4	22.6	23.5
	25.0	97.1	56.8	24.7	36.8
	50.0	96.8	78.1	10.0	28.0
	99.9	96.6	18.9	63.5	59.8
CodeParrot	0.0	95.5	74.8	22.6	22.3
	25.0	96.5	60.7	19.4	31.3
	50.0	96.5	37.4	20.5	42.4
	99.9	96.3	18.7	45.8	58.5
Wiki-ar	0.0	93.7	86.7	14.6	14.6
	25.0	95.1	72.6	18.9	21.3
	50.0	94.7	41.5	23.5	30.3
	99.9	94.8	25.3	25.3	41.8

Table 4: **Nesterov-frame band content.** The trajectories, checkpoints, and reference gradients are paired with Table 3, but the singular frame and noise edge $\tau_{t,\ell}^{M}$ are those of MUON's Nesterov input $M_{t,\ell}$. Each row pools 24 matrices and 12,288 directions, using the same statistics as Table 3. Matched scalar rescaling explains the agreement at 0.0M. Every analyzed matrix has positive signed bulk alignment.

Corpus	Frame	A_{sp}	A_{bu}	A_{all}	B_{all}	C
FineMath-4	Gradient	0.5946	0.1363	0.7308	1.2160	0.601
	Nesterov	0.0749	0.1300	0.2049	0.8594	0.238
CodeParrot	Gradient	1.0155	0.2028	1.2183	1.6182	0.753
	Nesterov	0.1132	0.0957	0.2088	0.8246	0.253
Wiki-ar	Gradient	0.9131	0.2013	1.1144	1.3890	0.802
	Nesterov	0.4201	0.1422	0.5623	0.9577	0.587

Table 5: **Signed contributions and cancellation at 99.9M tokens.** A_{sp} and A_{bu} are pooled signed alignments; A_{all} is their sum and B_{all} sums absolute alignments. $C = |A_{\mathrm{all}}|/B_{\mathrm{all}}$ measures the fraction remaining after signed cancellation. Every row pools 24 matrices, and every matrix has positive signed bulk alignment. Displayed sums are rounded independently.

The optimizer's frame changes the relative contributions. At 99.9M tokens, the Nesterov bulk supplies 63.5% of pooled signed alignment on FineMath-4, 45.8% on CodeParrot, and 25.3% on Wiki-ar. The paired raw-gradient shares are 18.6%, 16.6%, and 18.1%. A larger share does not necessarily mean a larger absolute bulk contribution: in Table 5, the final Nesterov bulk sums are smaller than their raw-gradient counterparts, but the spike sums decrease by a greater proportion. The share is also not monotone over training: on FineMath-4, the Nesterov bulk share is 24.7% at 25M, 10.0% at 50M, and 63.5% at 99.9M. What holds in every case is positive net bulk alignment; the share itself varies across corpora and checkpoints.

The contribution remains after signed cancellation. Table 5 reports raw signed and absolute sums at the final checkpoint. For example, the Nesterov bulk on CodeParrot contributes approximately 0.0957 to total signed alignment 0.2088, while total absolute alignment is 0.8246, giving $C \approx 0.253$. The Nesterov cancellation ratios are 0.238 on FineMath-4 and 0.587 on Wiki-ar. The positive bulk sums therefore survive the cancellation between opposing directions.

D.3 Implications for BulkBoost

Motivation for the fixed-fraction sweep. The complementary spike band contains only 2.9–6.3% of the pooled singular directions in these measurements. This broad-bulk/small-spike structure motivates testing small fixed spike fractions, alongside larger ones, in Table 9 (Appendix G.2). The tested grid is $p \in \{0.015, 0.02, 0.05, 0.15, 0.20\}$: 5% lies within the observed pooled range and is the default for BB-Static, 1.5% and 2% examine smaller fractions, and 15% and 20% examine

larger spike bands. The diagnostics inform the grid; we did not convert the noise edge into five calibrated fractions. The loss comparisons in Table 9 decide which fractions perform best.

The 2.9–6.3% range is computed after pooling matrices within each corpus–checkpoint–frame combination. Individual matrices can fall outside it, and it is not an estimate of the best fixed rank. BB-STATIC uses $r_{t,\ell} = \lceil p d_\ell \rceil$ with a fixed fraction for the whole run, whereas BB-ADAPTIVE selects the spike band separately for each matrix and step. The two rules give different partitions even when their pooled band sizes are similar.

Link to the two-band design. Theorem 2 shows that a shared bulk gain needs only the polar factor plus a correction in the spike subspace. The band-content measurements show why the bulk matters in this allocation: it carries positive reference alignment in every matrix. Appendix E.2 evaluates the post-update effect of reweighting, and Section 5 evaluates training performance and computational cost.

E PRE-UPDATE ALIGNMENT AND POST-UPDATE LOSS

This appendix connects the band-content measurements in Appendix D to the effect of spectral reweighting on validation loss. We first compute the pre-update reference-alignment preference and then evaluate the model produced by a controlled update. The single-update measurements show that finite bulk-emphasizing reweighting can improve on the exact flat update Q_t, even when paired pre-update predictions favor less bulk weight.

E.1 PRE-UPDATE REFERENCE ALIGNMENT

A norm-matched alignment score. Use the same singular frames, reference gradients, and partitions as Appendix D, holding them fixed while varying ρ. Suppress the checkpoint and frame indices. For matrix ℓ, let $r_\ell = |\mathcal{I}_{\mathrm{sp},\ell}|$, $k_\ell = d_\ell - r_\ell$, and $q_\ell = r_\ell/d_\ell$, and set $A_{b,\ell} = \sum_{i \in \mathcal{I}_{b,\ell}} \alpha_{\ell,i}$. The exact norm-matched two-level direction has alignment

$$J_\ell(\rho) = c_{\rho,r_\ell}\big(A_{\mathrm{sp},\ell} + \rho A_{\mathrm{bu},\ell}\big), \qquad c_{\rho,r_\ell} = \big[q_\ell + \rho^2(1-q_\ell)\big]^{-1/2}, \tag{34}$$

whenever the profile is nonzero. A larger score predicts a larger first-order decrease in the reference objective under a common step size. This gives the pre-update comparison for the loss measurements below.

For nonempty bands, differentiating at $\rho = 1$ gives

$$\left.\frac{dJ_\ell(\rho)}{d\rho}\right|_{\rho=1} = \frac{r_\ell k_\ell}{d_\ell}\left(\overline{\alpha}_{\mathrm{bu},\ell} - \overline{\alpha}_{\mathrm{sp},\ell}\right), \tag{35}$$

where $\overline{\alpha}_{\mathrm{bu},\ell} = A_{\mathrm{bu},\ell}/k_\ell$ and $\overline{\alpha}_{\mathrm{sp},\ell} = A_{\mathrm{sp},\ell}/r_\ell$. This uses the same norm-matched differentiation as Theorem 1, with pre-update reference projections in place of continuation utilities. The local preference therefore depends on the difference in *mean per-direction* alignment between bands. Appendix D instead measures their *aggregate* contributions under the flat profile.

Normalize each matrix separately before pooling:

$$J(\rho) = \sum_\ell J_\ell(\rho), \qquad R_{\mathrm{mat}}(\rho) = \frac{J(\rho)}{J(1)}. \tag{36}$$

As in Appendix D, no learning-rate or shape multipliers are included. The reference $\rho = 1$ is the flat profile in the same frame; in the Nesterov frame, it gives MUON's idealized update Q_t. The measured $J(1)$ is positive in every pooled comparison.

The pre-update score favors less bulk weight. The aggregate derivative $J'(1)$ is negative in all 24 corpus–checkpoint–frame combinations. At the matrix level, the derivative is negative in 564 of 576 recorded combinations and positive in the remaining 12. The pooled signed per-direction ratio $(A_{\mathrm{bu}}/N_{\mathrm{bu}})/(A_{\mathrm{sp}}/N_{\mathrm{sp}})$ ranges from approximately 0.0037 to 0.0617. Unlike Γ in Appendix D, this ratio keeps the signs of the projections. The derivatives and reweighted scores are computed matrix by matrix; the pooled ratio only summarizes them. Across the recorded comparisons, $R_{\mathrm{mat}}(2)$ ranges from 0.556 to 0.828, and $R_{\mathrm{mat}}(4)$ from 0.330 to 0.738. Table 6 reports the final checkpoint.

Corpus	Frame	$\overline{\alpha}_{\mathrm{bu}}/\overline{\alpha}_{\mathrm{sp}}$	$R_{\mathrm{mat}}(2)$	$R_{\mathrm{mat}}(4)$
FineMath-4	Gradient	0.009	0.601	0.397
	Nesterov	0.062	0.828	0.738
CodeParrot	Gradient	0.007	0.590	0.381
	Nesterov	0.032	0.739	0.604
Wiki-ar	Gradient	0.012	0.602	0.395
	Nesterov	0.019	0.638	0.450

Table 6: **Pre-update reference-alignment trade-off at 99.9M tokens.** The per-direction ratio uses pooled signed means $\overline{\alpha}_b = A_b/N_b$; the ratio Γ in Appendix D uses absolute alignments. $R_{\mathrm{mat}}(\rho)$ applies each matrix's own norm-matching factor before summing, as in equation 36. Values below one indicate lower alignment than the flat profile in the same frame. Each row uses the same 24 matrices and reference gradient as the preceding diagnostics, without learning-rate or shape multipliers.

Interpreting the norm-matched reweighting. Increasing ρ moves the fixed direction norm from the spikes toward the bulk. For a matrix with 95% bulk directions, the applied bulk gains at $\rho = 2$ and 4 are approximately 1.019 and 1.024, whereas the spike gains are 0.510 and 0.256. In the pre-update score, the loss of spike alignment outweighs the additional bulk alignment, which explains the preference for smaller ρ. Relative bulk emphasis and relative spike suppression therefore describe the same norm-matched intervention. The next subsection evaluates its effect directly on post-update validation loss.

E.2 Single-Update Loss Measurements

Question and controlled comparison. *Can a finite increase in the bulk-to-spike gain lower validation loss after one update, even when a pre-update linearized prediction favors less bulk weight?* We use the Pythia-70M Muon trajectories of Appendix D on FineMath-4, CodeParrot, and Wiki-ar, immediately before updates 763 and 1526, after 24,969,216 and 49,971,200 training tokens, respectively. The protocol uses batches of 32 sequences of length 1024, $\beta = 0.95$, $\gamma = 0.95$, $\kappa_{\mathrm{edge}} = 1$, global clipping at norm 1, and peak learning rate 5×10^{-4} for both the matrix and auxiliary optimizers. Validation uses 128 fixed held-out windows.

At each checkpoint, the branches share the current Nesterov singular frames and the noise-edge partitions of Appendix D. We vary one common gain across the 24 hidden weight matrices, using the exact-SVD target and separate Frobenius norm matching for each matrix. The tested gains are $\rho \in \{0.5, 0.75, 1, 1.25, 1.5\}$, with two duplicate $\rho = 1$ branches. The comparator is the exact flat update Q_t at the same input, computed from the same SVD. All branches compute their current update from the same exact SVD, so they differ only in ρ. We report $K = 0$: validation immediately after the current update, without additional training steps. The duplicate $\rho = 1$ branches have identical recorded losses.

Direct improvement and allocation preference. Let $F_{t,K}(\rho)$ denote the expected evaluation loss, given the current state, under a current common gain ρ and K additional steps following a fixed continuation policy. Write $\widehat{F}_{t,0}(\rho)$ for the measured post-update validation loss on the fixed windows. For $\epsilon \in \{0.25, 0.5\}$, define

$$\widehat{B}_\epsilon = \widehat{F}_{t,0}(1) - \widehat{F}_{t,0}(1+\epsilon), \qquad \widehat{S}_\epsilon = \frac{\widehat{F}_{t,0}(1-\epsilon) - \widehat{F}_{t,0}(1+\epsilon)}{2\epsilon}. \tag{37}$$

The loss reduction $\widehat{B}_\epsilon$ directly measures improvement over the flat update $\rho = 1$: positive values mean that the bulk-emphasizing update $\rho = 1 + \epsilon$ gives lower validation loss. The raw central finite difference $\widehat{S}_\epsilon$ instead compares the bulk-emphasizing and bulk-suppressing branches, $\rho = 1 + \epsilon$ and $\rho = 1 - \epsilon$. A positive value indicates lower loss for the bulk-emphasizing branch. Together, these statistics give the benefit over the flat update and the preference between opposing reweightings. Table 7 reports both quantities for all six corpus–checkpoint settings.

Bulk emphasis improves the post-update model. Both $\rho = 1.25$ and $\rho = 1.5$ lower validation loss relative to $\rho = 1$ in five of six corpus–checkpoint settings: all three corpora at 25M tokens, and

		Loss reduction $\widehat{B}_\epsilon$ (10^{-4})		Slope $\widehat{S}_\epsilon$ (10^{-3})	
Corpus	Tokens (M)	$\epsilon = 0.25$	$\epsilon = 0.5$	$\epsilon = 0.25$	$\epsilon = 0.5$
FineMath-4	25.0	+2.524	+3.725	+1.709	+2.774
CodeParrot	25.0	+1.842	+2.574	+1.350	+2.320
Wiki-ar	25.0	+6.593	+10.226	+4.082	+5.944
FineMath-4	50.0	-0.604	-1.191	-0.177	+0.032
CodeParrot	50.0	+0.844	+1.262	+0.554	+0.866
Wiki-ar	50.0	+1.530	+2.318	+0.978	+1.471

Table 7: **Single-update validation-loss improvement ($K = 0$).** Both $\rho = 1.25$ and $\rho = 1.5$ lower post-update validation loss relative to the exact flat update $\rho = 1$ in five of six tested states. Each row is a Pythia-70M corpus–checkpoint state with a common gain across 24 hidden matrices and separate per-matrix Frobenius norm matching. $\widehat{B}_\epsilon$ measures the loss reduction relative to $\rho = 1$, whereas $\widehat{S}_\epsilon$ compares the bulk-emphasizing and bulk-suppressing branches through equation 37. Loss reductions and raw central finite-difference slopes are reported in units of 10^{-4} and 10^{-3}, respectively. All six states are shown; token labels are rounded, and the duplicate $\rho = 1$ branches have identical recorded losses.

CodeParrot and Wiki-ar at 50M tokens. The positive loss reductions $\widehat{B}_\epsilon$ measure these improvements directly. FineMath-4 at 50M is the exception, where the flat update has lower loss than both bulk-emphasizing updates.

The improvements occur despite paired pre-update linearized predictions favoring less bulk weight at both perturbation widths in every tested state. These predictions are recorded for the intervention branches and differ from the pooled score $J(\rho)$ in Appendix E.1. Thus, finite, norm-matched spectral reweighting can improve the post-update model even when the paired pre-update approximation favors the opposite allocation. The benefit is already visible at $K = 0$, without additional continuation steps.

Connection to Theorem 1. The theorem varies the gains of one matrix, whereas this experiment changes a common ρ across matrices. For matrix ℓ, let $r_{t,\ell}$ be its spike rank, $k_{t,\ell} = d_\ell - r_{t,\ell}$, and $\Delta_{\ell,t,K}$ its bulk-minus-spike mean continuation utility when every matrix uses the flat profile. Each matrix's utilities are partial derivatives with the other matrices' current gains held at one. Define $\omega_{t,\ell} = r_{t,\ell}k_{t,\ell}/d_\ell$, the matrix-specific version of the prefactor rk/d in the theorem. Under the corresponding joint differentiability assumption, the chain rule gives

$$S_{t,K} := -F'_{t,K}(1) = \sum_{\ell:\, 0<r_{t,\ell}<d_\ell} \omega_{t,\ell}\Delta_{\ell,t,K}. \tag{38}$$

Indeed, the derivative along the common-gain curve is the sum of its matrix-wise partial derivatives, and Theorem 1 gives $\omega_{t,\ell}\Delta_{\ell,t,K}$ for each nonempty partition. Empty-band matrices have constant normalized targets and contribute zero. The identity applies to the joint loss, including interactions across matrices.

When $\sum_\ell \omega_{t,\ell} > 0$, the exact ratio $S_{t,K}/\sum_\ell \omega_{t,\ell}$ is a weighted mean of the matrix-wise continuation-utility gaps. Its finite-difference counterpart is $\widehat{S}_\epsilon/\sum_\ell \omega_{t,\ell}$. Table 7 reports the raw central finite differences $\widehat{S}_\epsilon$ in units of 10^{-3}. The theorem characterizes the derivative at $\rho = 1$; $\widehat{S}_\epsilon$ is its finite-difference counterpart, and $\widehat{B}_\epsilon$ directly tests improvement at the chosen gains. These diagnostics measure the joint response to a shared gain. They do not measure individual utilities or R_{bt}.

Continuation utility evaluates the updated model. Even at $K = 0$, the evaluation point has changed. For an additive intervention with fixed effective step multiplier $\eta_{t,\ell}$, let θ_1^+ be the complete parameter state after the flat update, and let $\mathcal{E}$ be the evaluation objective defining the loss. Differentiation with respect to an applied singular-direction gain gives

$$u_{\ell,i,t,0} = \eta_{t,\ell}\left\langle \nabla_{W_\ell}\mathcal{E}(\theta_1^+), u_{t,\ell,i}v_{t,\ell,i}^\top \right\rangle_F. \tag{39}$$

The singular vectors remain those of the fixed pre-intervention Nesterov input, but the evaluation gradient is taken *after* the flat update. The chain rule uses

$$\frac{\partial W_\ell^+}{\partial w_{\ell,i}} = -\eta_{t,\ell} u_{t,\ell,i} v_{t,\ell,i}^\top,$$

and continuation utility is the negative loss derivative. The multiplier $\eta_{t,\ell}$ includes the learning rate and fixed shape scaling. Thus, "first order" describes changes in the applied gains around the flat update, with the actual parameter-update step size held fixed. This identifies the evaluation point relevant to local allocation preferences and explains the difference from pre-update reference alignment.

What the experiment establishes. *The experiment provides direct evidence that finite, norm-matched two-band reweighting can improve the model produced by a single update, already at* $K = 0$. The measurements cover $\rho = 1.25$ and 1.5 at checkpoints near 25M and 50M tokens; Appendix D also measures band content at initialization and 99.9M. Section 5 separately evaluates the full training rule with the shared default $\rho = 4$. Because norm matching couples relative bulk emphasis with spike suppression, the measured benefit concerns their combined reweighting.

F PROOFS FOR SECTIONS 3 AND 4

This appendix proves Theorems 1 and 2, and states and proves Corollary 3. It also proves the half-batch and Nesterov noise identities used in Section 3.1 (Proposition 4), and treats the Gaussian model behind the noise edge separately. Appendix B.4 derives numerical norm matching.

F.1 PROOF OF THEOREM 1

Theorem 1 (restated). Fix the current training state, singular frame, nonempty two-band partition, and continuation policy as in Section 4.1. The intervention varies the applied gains of one matrix's current update; the current updates of all other parameters and the rest of the current step stay fixed. Assume $\mathcal{L}^{\pi}_{t,K}$ is differentiable at $\mathbf{1}_d$. Along the two-level family,

$$-\frac{d}{d\rho}\mathcal{L}^{\pi}_{t,K}(w(\rho))\Big|_{\rho=1} = \frac{rk}{d}\Delta. \tag{40}$$

Increasing ρ from 1 gives a strictly positive first-order improvement if and only if $\Delta > 0$. If $\Delta = 0$, the first-order test is inconclusive.

Small gain changes at the flat profile lie in the tangent space $\mathcal{T} = \{\delta \in \mathbb{R}^d : \mathbf{1}_d^{\top}\delta = 0\}$. If every gain may change separately, the best first-order improvement rate is

$$\max_{\substack{\delta\in\mathcal{T}\\ \|\delta\|_2=1}} u^{\top}\delta = \|u_{\perp}\|_2. \tag{41}$$

Thus the flat profile is first-order stationary in this fixed-frame, norm-matched class if and only if $u_1 = \cdots = u_d$.

If the change must be constant within each band, the best rate is $\sqrt{rk/d}\,|\Delta|$. For $u_{\perp} \neq 0$, two bands therefore reach the fraction

$$R_{\mathrm{bt}} = \frac{\sqrt{rk/d}\,|\Delta|}{\|u_{\perp}\|_2} \in [0,1] \tag{42}$$

of the best rate. Both rates are per unit Euclidean change in w, and R_{bt} allows ρ to move in either direction.

Between-band and within-band components. Use the fixed state, frame, partition, and utilities from Section 4.1. Define

$$\begin{aligned}(u_{\mathrm{bt}})_i &= \begin{cases}\overline{u}_{\mathrm{sp}} - \overline{u}, & i \in \mathcal{I}_{\mathrm{sp}},\\ \overline{u}_{\mathrm{bu}} - \overline{u}, & i \in \mathcal{I}_{\mathrm{bu}},\end{cases}\\ (u_{\mathrm{wi}})_i &= \begin{cases}u_i - \overline{u}_{\mathrm{sp}}, & i \in \mathcal{I}_{\mathrm{sp}},\\ u_i - \overline{u}_{\mathrm{bu}}, & i \in \mathcal{I}_{\mathrm{bu}}.\end{cases}\end{aligned} \tag{43}$$

The decomposition $u_{\perp} = u_{\mathrm{bt}} + u_{\mathrm{wi}}$ is orthogonal and satisfies

$$\|u_{\perp}\|_2^2 = \frac{rk}{d}\Delta^2 + \sum_{i\in\mathcal{I}_{\mathrm{sp}}}(u_i - \overline{u}_{\mathrm{sp}})^2 + \sum_{i\in\mathcal{I}_{\mathrm{bu}}}(u_i - \overline{u}_{\mathrm{bu}})^2. \tag{44}$$

In particular, $\|u_{\mathrm{bt}}\|_2 = \sqrt{rk/d}\,|\Delta|$, so for $u_{\perp} \neq 0$ the capture ratio can also be written as $R_{\mathrm{bt}} = \|u_{\mathrm{bt}}\|_2/\|u_{\perp}\|_2$. Its square is the share of squared tangent utility that lies between the bands; the rest lies within the bands. This share says nothing directly about the size of a finite loss reduction. The proof below establishes these identities together with the theorem.

Proof. Write $\mathcal{L} = \mathcal{L}^{\pi}_{t,K}$ for brevity. As specified in Section 4.1, $\mathcal{L}$ is defined on an open neighborhood of $\mathbf{1}_d$ in the positive orthant and is differentiable at $\mathbf{1}_d$, with $u = -\nabla\mathcal{L}(\mathbf{1}_d)$. All gain comparisons below lie on the norm-matched sphere. The singular frame, current training state, partition, and continuation policy are fixed throughout the argument. The policy is fixed as a rule: subsequent gradients and optimizer states may respond to the current intervention. We proceed in four steps.

Step 1: first-order rates on the norm-matched sphere. The feasible set is

$$\mathcal{S}_d = \{w \in [0,\infty)^d : \|w\|_2^2 = d\}.$$

Differentiating the norm constraint along a differentiable feasible curve through $\mathbf{1}_d$ shows that its velocity δ satisfies $\mathbf{1}_d^\top \delta = 0$. Conversely, for any unit vector δ with $\mathbf{1}_d^\top \delta = 0$, define

$$w_\delta(\varepsilon) = \sqrt{d}\,\frac{\mathbf{1}_d + \varepsilon\delta}{\|\mathbf{1}_d + \varepsilon\delta\|_2}.$$

Since

$$\|\mathbf{1}_d + \varepsilon\delta\|_2^2 = d + \varepsilon^2,$$

this curve lies in $\mathcal{S}_d$ for sufficiently small $|\varepsilon|$, has positive coordinates there, and satisfies

$$w_\delta(0) = \mathbf{1}_d, \qquad w_\delta'(0) = \delta.$$

Thus the tangent space is $\mathcal{T} = \{\delta : \mathbf{1}_d^\top \delta = 0\}$, and every unit tangent direction is realized by such a curve.

Differentiability gives

$$\mathcal{L}(w_\delta(\varepsilon)) = \mathcal{L}(\mathbf{1}_d) - \varepsilon u^\top \delta + o(|\varepsilon|).$$

Hence $u^\top \delta$ is the first-order improvement rate per unit gain-vector displacement. For tangent δ,

$$u^\top \delta = (u - \overline{u}\mathbf{1}_d)^\top \delta = u_\perp^\top \delta.$$

Step 2: orthogonal between-band and within-band decomposition. For $i \in \mathcal{I}_{\mathrm{sp}}$,

$$(u_{\mathrm{bt}})_i + (u_{\mathrm{wi}})_i = (\overline{u}_{\mathrm{sp}} - \overline{u}) + (u_i - \overline{u}_{\mathrm{sp}}) = u_i - \overline{u}.$$

The same identity holds on $\mathcal{I}_{\mathrm{bu}}$, giving $u_\perp = u_{\mathrm{bt}} + u_{\mathrm{wi}}$. Moreover,

$$\sum_{i\in\mathcal{I}_{\mathrm{sp}}} (u_i - \overline{u}_{\mathrm{sp}}) = 0, \qquad \sum_{i\in\mathcal{I}_{\mathrm{bu}}} (u_i - \overline{u}_{\mathrm{bu}}) = 0.$$

Therefore u_{wi} is orthogonal to both band indicator vectors. Since u_{bt} is in their span,

$$u_{\mathrm{bt}} \perp u_{\mathrm{wi}}, \qquad \|u_\perp\|_2^2 = \|u_{\mathrm{bt}}\|_2^2 + \|u_{\mathrm{wi}}\|_2^2.$$

By definition,

$$\|u_{\mathrm{wi}}\|_2^2 = \sum_{i\in\mathcal{I}_{\mathrm{sp}}} (u_i - \overline{u}_{\mathrm{sp}})^2 + \sum_{i\in\mathcal{I}_{\mathrm{bu}}} (u_i - \overline{u}_{\mathrm{bu}})^2.$$

Let $\Delta = \overline{u}_{\mathrm{bu}} - \overline{u}_{\mathrm{sp}}$. Using $d = r + k$ and $\overline{u} = (r\overline{u}_{\mathrm{sp}} + k\overline{u}_{\mathrm{bu}})/d$, we obtain

$$\overline{u}_{\mathrm{sp}} - \overline{u} = -\frac{k}{d}\Delta, \qquad \overline{u}_{\mathrm{bu}} - \overline{u} = \frac{r}{d}\Delta.$$

Consequently,

$$\begin{aligned}\|u_{\mathrm{bt}}\|_2^2 &= r\frac{k^2}{d^2}\Delta^2 + k\frac{r^2}{d^2}\Delta^2 \\ &= \frac{rk(r+k)}{d^2}\Delta^2 = \frac{rk}{d}\Delta^2.\end{aligned}$$

Combining these identities proves equation 44.

Step 3: maximal first-order rate and stationarity. For every unit tangent direction, Cauchy–Schwarz gives

$$u^\top \delta = u_\perp^\top \delta \le \|u_\perp\|_2.$$

If $u_\perp \neq 0$, equality holds at

$$\delta^\star = \frac{u_\perp}{\|u_\perp\|_2},$$

which is a feasible unit tangent direction because $\mathbf{1}_d^\top u_\perp = 0$. If $u_\perp = 0$, every tangent direction has zero rate. It follows that

$$\max_{\substack{\delta\in\mathcal{T}\\ \|\delta\|_2=1}} u^\top \delta = \|u_\perp\|_2.$$

This proves equation 41. In particular, the flat profile is first-order stationary with respect to the fixed-frame, norm-matched interventions if and only if

$$u_\perp = 0 \quad\Longleftrightarrow\quad u \in \mathrm{span}\{\mathbf{1}_d\} \quad\Longleftrightarrow\quad u_1 = \cdots = u_d.$$

Step 4: the two-level direction and capture ratio. A band-constant tangent direction has the form $a\mathbf{1}_{\mathrm{sp}} + b\mathbf{1}_{\mathrm{bu}}$ with $ra + kb = 0$. Since $r, k > 0$, these directions form the line spanned by

$$v = \frac{r}{d}\mathbf{1}_{\mathrm{bu}} - \frac{k}{d}\mathbf{1}_{\mathrm{sp}}, \qquad \|v\|_2^2 = \frac{rk}{d} > 0.$$

For the two-level curve $w(\rho) = c_{\rho,r}(\mathbf{1}_{\mathrm{sp}} + \rho\mathbf{1}_{\mathrm{bu}})$, the exact normalizer satisfies

$$c_{\rho,r} = \sqrt{\frac{d}{r + k\rho^2}}, \qquad c_{1,r} = 1, \qquad \frac{d}{d\rho}c_{\rho,r}\Big|_{\rho=1} = -\frac{k}{d}.$$

Therefore

$$\begin{aligned} w'(1) &= -\frac{k}{d}(\mathbf{1}_{\mathrm{sp}} + \mathbf{1}_{\mathrm{bu}}) + \mathbf{1}_{\mathrm{bu}} \\ &= -\frac{k}{d}\mathbf{1}_{\mathrm{sp}} + \frac{r}{d}\mathbf{1}_{\mathrm{bu}} = v. \end{aligned}$$

The chain rule now yields

$$\begin{aligned} -\frac{d}{d\rho}\mathcal{L}(w(\rho))\Big|_{\rho=1} &= u^\top v \\ &= \frac{r}{d}\sum_{i\in\mathcal{I}_{\mathrm{bu}}} u_i - \frac{k}{d}\sum_{i\in\mathcal{I}_{\mathrm{sp}}} u_i \\ &= \frac{rk}{d}(\overline{u}_{\mathrm{bu}} - \overline{u}_{\mathrm{sp}}) = \frac{rk}{d}\Delta. \end{aligned}$$

This proves equation 40. More explicitly, as $h \to 0$,

$$\mathcal{L}(w(1+h)) = \mathcal{L}(\mathbf{1}_d) - h\frac{rk}{d}\Delta + o(|h|).$$

If $\Delta > 0$, every sufficiently small positive h reduces the loss. If $\Delta < 0$, every sufficiently small positive h increases it, and a sufficiently small negative h reduces it. When $\Delta = 0$, the first derivative vanishes and does not determine whether a higher-order improvement occurs. Thus the strict sign criterion is an if-and-only-if statement about first-order improvement.

For the capture ratio, observe that

$$\mathrm{proj}_{\mathrm{span}\{v\}}\, u = \frac{u^\top v}{\|v\|_2^2}v = \Delta v = u_{\mathrm{bt}}.$$

The two unit directions in this line are $\pm v/\|v\|_2$, and hence

$$\max_{\substack{\delta\in\mathrm{span}\{v\}\\ \|\delta\|_2=1}} u^\top\delta = \frac{|u^\top v|}{\|v\|_2} = \sqrt{\frac{rk}{d}}\,|\Delta| = \|u_{\mathrm{bt}}\|_2.$$

Both signs can be realized along the two-level family: the curves $w(1\pm\varepsilon/\|v\|_2)$ have unit velocities $\pm v/\|v\|_2$ at $\varepsilon = 0$. Thus, when $u_\perp \neq 0$, the ratio of the restricted and unrestricted unit-tangent optima is

$$R_{\mathrm{bt}} = \frac{\sqrt{rk/d}\,|\Delta|}{\|u_\perp\|_2} = \frac{\|u_{\mathrm{bt}}\|_2}{\|u_\perp\|_2}, \qquad R_{\mathrm{bt}}^2 = \frac{\|u_{\mathrm{bt}}\|_2^2}{\|u_{\mathrm{bt}}\|_2^2 + \|u_{\mathrm{wi}}\|_2^2} \in [0,1].$$

This proves equation 42. The ratio is not defined when $u_\perp = 0$, since both first-order optima then vanish. □

What finer shaping can add. Two bands can use u_{bt}, the differences between band means. They cannot use u_{wi}, the variation within bands. By Step 1, every differentiable curve of norm-matched gains through $\mathbf{1}_d$, however fine-grained, has its velocity in $\mathcal{T}$. By Step 3, its first-order rate per unit velocity is therefore at most $\|u_\perp\|_2$. The best two-band change reaches $\|u_{\mathrm{bt}}\|_2$. Finer shaping can thus add at most

$$\|u_\perp\|_2 - \|u_{\mathrm{bt}}\|_2 = (1 - R_{\mathrm{bt}})\|u_\perp\|_2 \le \|u_{\mathrm{wi}}\|_2$$

to the first-order rate. In particular, if utilities are constant within each band, finer shaping cannot improve on two bands to first order. The discussion after Theorem 1 covers the remaining caveats.

F.2 Proof of Theorem 2

Proof. After a common reordering of the singular directions, write

$$U = [U_{\mathrm{sp}}\ U_{\mathrm{bu}}], \qquad V = [V_{\mathrm{sp}}\ V_{\mathrm{bu}}].$$

Then

$$Q = UV^\top = U_{\mathrm{sp}}V_{\mathrm{sp}}^\top + U_{\mathrm{bu}}V_{\mathrm{bu}}^\top.$$

The prescribed unnormalized update is

$$\widetilde{D}(a,\rho) = U_{\mathrm{sp}}\,\mathrm{diag}(a)V_{\mathrm{sp}}^\top + \rho U_{\mathrm{bu}}V_{\mathrm{bu}}^\top.$$

Substituting $U_{\mathrm{bu}}V_{\mathrm{bu}}^\top = Q - U_{\mathrm{sp}}V_{\mathrm{sp}}^\top$ gives

$$\begin{aligned}\widetilde{D}(a,\rho) &= U_{\mathrm{sp}}\,\mathrm{diag}(a)V_{\mathrm{sp}}^\top + \rho\big(Q - U_{\mathrm{sp}}V_{\mathrm{sp}}^\top\big)\\ &= \rho Q + U_{\mathrm{sp}}\big(\mathrm{diag}(a) - \rho I_r\big)V_{\mathrm{sp}}^\top\\ &= \rho Q + U_{\mathrm{sp}}\,\mathrm{diag}(a - \rho\mathbf{1}_r)V_{\mathrm{sp}}^\top.\end{aligned}$$

This is equation 14. The correction factors through an r-dimensional space, so

$$\mathrm{rank}\big(\widetilde{D}(a,\rho) - \rho Q\big) \le r.$$

For $a = \mathbf{1}_r$, it follows that

$$\widetilde{D}_\rho = \rho Q - (\rho-1)U_{\mathrm{sp}}V_{\mathrm{sp}}^\top.$$

Define

$$P_{\mathrm{sp}} = U_{\mathrm{sp}}U_{\mathrm{sp}}^\top, \qquad R_{\mathrm{sp}} = V_{\mathrm{sp}}V_{\mathrm{sp}}^\top.$$

Orthonormality within each singular-vector matrix gives

$$\begin{aligned}P_{\mathrm{sp}}Q &= U_{\mathrm{sp}}U_{\mathrm{sp}}^\top\big(U_{\mathrm{sp}}V_{\mathrm{sp}}^\top + U_{\mathrm{bu}}V_{\mathrm{bu}}^\top\big)\\ &= U_{\mathrm{sp}}V_{\mathrm{sp}}^\top,\end{aligned}$$

and

$$\begin{aligned}QR_{\mathrm{sp}} &= \big(U_{\mathrm{sp}}V_{\mathrm{sp}}^\top + U_{\mathrm{bu}}V_{\mathrm{bu}}^\top\big)V_{\mathrm{sp}}V_{\mathrm{sp}}^\top\\ &= U_{\mathrm{sp}}V_{\mathrm{sp}}^\top.\end{aligned}$$

Both identities hold for either rectangular orientation, giving

$$\widetilde{D}_\rho = \begin{cases}\rho Q - (\rho-1)P_{\mathrm{sp}}Q, & m \le n,\\ \rho Q - (\rho-1)QR_{\mathrm{sp}}, & n < m.\end{cases} \tag{45}$$

With time indices restored, multiplication by c_{ρ,r_t} recovers the main-method identity equation 9. No approximation or random-matrix assumption is used. □

F.3 Exact Frobenius Normalization

Section 4.2 uses the following consequence of Theorem 2.

Corollary 3 (Exact Frobenius normalization)**.** *Under Theorem 2,*

$$\|\widetilde{D}(a,\rho)\|_F^2 = \|a\|_2^2 + \rho^2(d-r). \tag{46}$$

For $r + \rho^2(d-r) > 0$, the normalizer $c_{\rho,r}$ from equation 4 gives $\|c_{\rho,r}\widetilde{D}_\rho\|_F = \|Q\|_F = \sqrt{d}$.

Proof. The rank-one singular modes are Frobenius-orthonormal:

$$\begin{aligned}\big\langle u_iv_i^\top, u_jv_j^\top\big\rangle_F &= \mathrm{tr}\big(v_iu_i^\top u_jv_j^\top\big)\\ &= (u_i^\top u_j)(v_i^\top v_j) = \mathbf{1}\{i=j\}.\end{aligned}$$

Consequently, the squared Frobenius norm of a spectral profile is the sum of its squared gains. In particular,

$$\|\widetilde{D}(a,\rho)\|_F^2 = \sum_{i=1}^{r} a_i^2 + \rho^2(d-r) = \|a\|_2^2 + \rho^2(d-r),$$

which proves equation 46. For $a = \mathbf{1}_r$,

$$\|\widetilde{D}_\rho\|_F^2 = r + \rho^2(d-r), \qquad \|Q\|_F^2 = \left\|\sum_{i=1}^{d} u_i v_i^\top\right\|_F^2 = d.$$

Whenever $r + \rho^2(d-r) > 0$, the stated normalizer is well-defined and satisfies

$$\|c_{\rho,r}\widetilde{D}_\rho\|_F^2 = \frac{d}{r+\rho^2(d-r)}\big[r + \rho^2(d-r)\big] = d = \|Q\|_F^2.$$

Taking nonnegative square roots proves the corollary. If $r + \rho^2(d-r) = 0$, the raw two-level direction is zero and cannot be rescaled to norm $\sqrt{d}$; this case is excluded by the corollary's hypothesis. □

F.4 Online Calibration of the Noise Edge

BB-ADAPTIVE estimates the noise scale online from two minibatch halves, without a large-batch reference gradient. The proposition below proves the exact half-batch and Nesterov covariance identities. The Gaussian model behind the noise edge is treated separately after the proof.

Proposition 4 (Half-batch calibration of MUON's spectral input)**.** *Let $X_1, \dots, X_b \in \mathbb{R}^{m\times n}$ be independent per-example gradients with common mean $\overline{G}$ and*

$$\mathrm{Cov}(\mathrm{vec}\, X_i) = s_{\mathrm{ex}}^2 I_{mn},$$

where b is a positive even integer. Let $G^{(1)}, G^{(2)}$ be the two half-minibatch means, and define

$$G = \frac{G^{(1)} + G^{(2)}}{2}, \qquad Z^{\mathrm{split}} = \frac{G^{(1)} - G^{(2)}}{2}.$$

Then

$$\mathrm{Cov}(\mathrm{vec}(G - \overline{G})) = \mathrm{Cov}(\mathrm{vec}\, Z^{\mathrm{split}}) = s_b^2 I_{mn}, \qquad s_b^2 = \frac{s_{\mathrm{ex}}^2}{b}, \tag{47}$$

and the variance estimator satisfies

$$\widehat{s}_b^2 = \frac{\|Z^{\mathrm{split}}\|_F^2}{mn}, \qquad \mathbb{E}[\widehat{s}_b^2] = s_b^2. \tag{48}$$

Across training steps, suppose $G_t = \overline{G}_t + \varepsilon_t$, where the noises ε_t are centered, independent across time, and have the same covariance $s_b^2 I_{mn}$ after vectorization. For $\beta \in [0,1)$ and $B_0 = 0$, define

$$B_t = \beta B_{t-1} + (1-\beta)G_t, \qquad M_t = (1-\beta)G_t + \beta B_t. \tag{49}$$

The component obtained by propagating the sampling noises through these recursions is

$$N_t^M = (1-\beta^2)\varepsilon_t + \beta^2(1-\beta)\sum_{j=1}^{t-1}\beta^{t-1-j}\varepsilon_j,$$

where the sum is empty for $t = 1$. For every $t \geq 1$,

$$\mathrm{Cov}(\mathrm{vec}\, N_t^M) = \nu_t^2(\beta) s_b^2 I_{mn}, \tag{50}$$

with

$$\nu_t^2(\beta) = (1-\beta^2)^2 + \beta^4 \frac{1-\beta}{1+\beta}\big(1 - \beta^{2(t-1)}\big). \tag{51}$$

Here $\nu_t(\beta)$ denotes the nonnegative square root of this variance multiplier.

Interpretation. The half-batch difference removes the unknown mean gradient. Its covariance equals that of the full-minibatch mean; each half alone has twice this variance. The factor $\nu_t^2(\beta)$ accounts for the current gradient entering the Nesterov input both directly and through the updated buffer. For $\beta > 0$, the EMA-only multiplier $\frac{1-\beta}{1+\beta}\left(1-\beta^{2t}\right)$ would therefore understate the noise in M_t. The implementation in Section 3.1 uses $G_t = \chi_t G_t^{\mathrm{raw}}$ in the momentum recursion, keeps $\overline{s}_t^2$ as an EMA of $\chi_t^2 \widehat{s}_t^2$, substitutes $\sqrt{\overline{s}_t^2}$ for s_b, and applies κ_{edge}, which gives equation 8. Appendix C.1 lists the approximations this involves. Neither the covariance identities nor the noise edge show that the resulting split is the best one for optimization.

Proof. We first establish the half-batch identities and then propagate the noise through the momentum recursion.

Step 1: half-batch covariance and variance estimation. The half-minibatch means are

$$G^{(1)} = \frac{2}{b}\sum_{i=1}^{b/2} X_i, \qquad G^{(2)} = \frac{2}{b}\sum_{i=b/2+1}^{b} X_i.$$

By the proposition's assumptions, they are independent, have common mean $\overline{G}$, and satisfy

$$\mathrm{Cov}(\mathrm{vec}\, G^{(1)}) = \mathrm{Cov}(\mathrm{vec}\, G^{(2)}) = \frac{2s_{\mathrm{ex}}^2}{b} I_{mn}.$$

Let

$$g_1 = \mathrm{vec}(G^{(1)} - \overline{G}), \qquad g_2 = \mathrm{vec}(G^{(2)} - \overline{G}).$$

Then

$$\mathrm{vec}(G - \overline{G}) = \frac{g_1 + g_2}{2}, \qquad \mathrm{vec}\, Z^{\mathrm{split}} = \frac{g_1 - g_2}{2}.$$

The cross-covariances vanish by independence, so

$$\begin{aligned}\mathrm{Cov}(\mathrm{vec}(G - \overline{G})) &= \frac{1}{4}\big(\mathrm{Cov}(g_1) + \mathrm{Cov}(g_2)\big) \\ &= \frac{s_{\mathrm{ex}}^2}{b} I_{mn} = s_b^2 I_{mn},\end{aligned}$$

and the identical calculation for the difference gives

$$\mathrm{Cov}(\mathrm{vec}\, Z^{\mathrm{split}}) = s_b^2 I_{mn}.$$

This proves equation 47. Since Z^{split} has zero mean,

$$\begin{aligned}\mathbb{E}[\widehat{s}_b^2] &= \frac{1}{mn}\mathbb{E}\|Z^{\mathrm{split}}\|_F^2 \\ &= \frac{1}{mn}\,\mathrm{tr}\big(\mathrm{Cov}(\mathrm{vec}\, Z^{\mathrm{split}})\big) \\ &= \frac{1}{mn}\,\mathrm{tr}(s_b^2 I_{mn}) = s_b^2.\end{aligned}$$

This proves equation 48.

For completeness, the average and difference also satisfy

$$\mathrm{Cov}\big(\mathrm{vec}(G - \overline{G}), \mathrm{vec}\, Z^{\mathrm{split}}\big) = \frac{1}{4}\big(\mathrm{Cov}(g_1) - \mathrm{Cov}(g_2)\big) = 0.$$

If (g_1, g_2) is jointly Gaussian, the average and difference are therefore independent. Without Gaussianity, zero cross-covariance does not by itself imply independence; that independence is not needed for the variance identity.

Step 2: propagation through the Nesterov spectral input. Write $G_t = \overline{G}_t + \varepsilon_t$ and consider the component obtained by linearly propagating the centered sampling noises through the recursions in equation 49. The EMA noise component satisfies

$$N_t^B = \beta N_{t-1}^B + (1-\beta)\varepsilon_t, \qquad N_0^B = 0.$$

Unrolling the recursion gives, for $t \geq 1$,

$$N_{t-1}^B = (1-\beta)\sum_{j=1}^{t-1} \beta^{t-1-j}\varepsilon_j,$$

with an empty sum when $t = 1$. Because the noises are independent across time and have the same covariance $s_b^2 I_{mn}$,

$$\begin{aligned}\operatorname{Cov}(\operatorname{vec} N_{t-1}^B) &= (1-\beta)^2 \sum_{j=1}^{t-1} \beta^{2(t-1-j)} s_b^2 I_{mn} \\ &= \frac{1-\beta}{1+\beta}\big(1-\beta^{2(t-1)}\big) s_b^2 I_{mn}.\end{aligned}$$

The spectral-input noise is

$$\begin{aligned}N_t^M &= (1-\beta)\varepsilon_t + \beta N_t^B \\ &= (1-\beta)\varepsilon_t + \beta\big(\beta N_{t-1}^B + (1-\beta)\varepsilon_t\big) \\ &= (1-\beta^2)\varepsilon_t + \beta^2 N_{t-1}^B \\ &= (1-\beta^2)\varepsilon_t + \beta^2(1-\beta)\sum_{j=1}^{t-1}\beta^{t-1-j}\varepsilon_j.\end{aligned}$$

The pair $(\varepsilon_t, N_{t-1}^B)$ is independent, so

$$\begin{aligned}\operatorname{Cov}(\operatorname{vec} N_t^M) &= (1-\beta^2)^2 \operatorname{Cov}(\operatorname{vec}\varepsilon_t) + \beta^4 \operatorname{Cov}(\operatorname{vec} N_{t-1}^B) \\ &= \left[(1-\beta^2)^2 + \beta^4 \frac{1-\beta}{1+\beta}\big(1-\beta^{2(t-1)}\big)\right] s_b^2 I_{mn} \\ &= \nu_t^2(\beta) s_b^2 I_{mn}.\end{aligned}$$

This proves equation 50 and equation 51. The current gradient appears twice, directly and through the buffer, so its two coefficients are combined before taking covariances.

Finally,

$$\begin{aligned}\nu_t^2(\beta) &= (1-\beta^2)^2 + \beta^4 \frac{1-\beta}{1+\beta}\big(1-\beta^{2(t-1)}\big) \\ &= (1-\beta)^2\left[(1+\beta)^2 + \frac{\beta^4\big(1-\beta^{2(t-1)}\big)}{1-\beta^2}\right],\end{aligned}$$

which is the method-section factor in equation 7. The entrywise standard deviation of N_t^M is $s_b\nu_t(\beta)$, where $\nu_t(\beta)$ is the nonnegative square root of the variance multiplier. □

The noise edge under the Gaussian model. The covariance calculation above does not need Gaussian entries, and it does not determine a spectral edge by itself. Assume, in addition, that the entries of each ε_t are independent centered Gaussian variables with variance s_b^2, and that the noise matrices are independent across time. Then the linear combination for N_t^M has independent centered Gaussian entries with variance $s_b^2\nu_t^2(\beta)$.

For fixed t and $\beta \in [0, 1)$, assume first that $s_b > 0$ and set $E_{m,n} = N_t^M/(s_b\nu_t(\beta))$. This is a standard Gaussian matrix. Its largest singular value converges to the upper edge of the Marchenko–Pastur law (Marchenko & Pastur, 1967; Geman, 1980):

$$\frac{\|E_{m,n}\|_2}{\sqrt{m}+\sqrt{n}} \xrightarrow{\text{a.s.}} 1 \qquad \text{as } m, n \to \infty, \quad m/n \to \zeta \in (0, \infty).$$

Rescaling gives $\|N_t^M\|_2/\big(s_b\nu_t(\beta)(\sqrt{m}+\sqrt{n})\big) \to 1$ almost surely, so the noise edge of N_t^M is

$$\tau_t^M = s_b\nu_t(\beta)(\sqrt{m}+\sqrt{n}). \tag{52}$$

If $s_b = 0$, the centered noise vanishes almost surely and the edge is zero. This step needs the Gaussian model and an asymptotic result. Isotropic covariance alone does not give it, and it is not a finite-sample bound.

	14M		31M		70M		160M		410M	
Corpus	ℓ_0	Δ	ℓ_0	Δ	ℓ_0	Δ	ℓ_0	Δ	ℓ_0	Δ
Wiki-de	4.530	-1.247 ± 0.001	4.009	-1.050 ± 0.002	3.699	-1.019 ± 0.003	3.171	-0.719 ± 0.002	2.626	-0.445 ± 0.001
Wiki-zh	3.715	-0.970 ± 0.001	3.465	-0.994 ± 0.001	3.329	-1.118 ± 0.005	3.090	-1.035 ± 0.001	2.570	-0.691 ± 0.002
FineMath-4	3.572	-0.486 ± 0.002	3.278	-0.502 ± 0.001	3.088	-0.537 ± 0.002	2.753	-0.481 ± 0.003	2.421	-0.375 ± 0.005
Wiki-ar	2.936	-0.930 ± 0.000	2.761	-0.906 ± 0.001	2.747	-1.045 ± 0.000	2.620	-1.012 ± 0.000	2.150	-0.698 ± 0.001
FineMath-3	2.930	-0.290 ± 0.002	2.692	-0.261 ± 0.002	2.566	-0.269 ± 0.004	2.320	-0.254 ± 0.002	2.050	-0.159 ± 0.002
CodeParrot	2.381	-0.479 ± 0.004	2.040	-0.391 ± 0.001	1.845	-0.361 ± 0.003	1.508	-0.238 ± 0.003	1.229	-0.118 ± 0.002

Table 8: Corpus adaptation across five model scales. ℓ_0 is the held-out loss before continued pretraining. $\Delta = \ell_{\text{final}} - \ell_0$ is the change after about 100M tokens with MUON, reported as mean $\pm$ standard deviation over three seeds. Negative values indicate improvement. ℓ_0 has no spread, because all seeds start from the same checkpoint and use the same held-out windows. Rows are ordered by decreasing ℓ_0, which gives the same order at every scale. Corpora are grouped into mathematics, code, and multilingual Wikipedia.

G OMITTED EXPERIMENT RESULTS

In this section, we give the detailed experimental results.

G.1 CORPORA ADAPTATION.

Models and datasets. We conduct continued pretraining by adapting *converged*, publicly available Pythia checkpoints on new corpora with the cross-entropy loss. We use five scales, Pythia-14M, 31M, 70M, 160M, and 410M, with hidden widths $d_{\text{model}} = 128$, 256, 512, 768, and 1024, respectively. For the matrix methods, ADAMW updates the embeddings and the language-model head. Table 15 in Appendix G lists the widths, batch settings, and training budgets. Pythia was pretrained on the Pile (Gao et al., 2021), a mainly English mixture that includes GitHub code, English Wikipedia, and arXiv. We adapt it to six corpora in three groups. For mathematics, FineMath-4 and FineMath-3 (Allal et al., 2025) come from Common Crawl with different math-quality thresholds. For code, CodeParrot is CodeParrot-clean (CodeParrot, 2022), a filtered and deduplicated Python corpus from GitHub. For multilingual text, Wiki-de, Wiki-zh, and Wiki-ar are German, Chinese, and Arabic Wikipedia.

Initial loss reflects corpus shift. Table 8 reports the pre-adaptation held-out loss ℓ_0 and the change $\Delta = \ell_{\text{final}} - \ell_0$ after 100M tokens of continued pretraining with MUON. We use ℓ_0 as a proxy for how far each corpus is from the Pythia pretraining distribution. A lower ℓ_0 means that the pretrained model already assigns higher probability to the corpus. A higher ℓ_0 suggests a larger distribution shift. The corpora keep the same order at every scale. CodeParrot has the lowest ℓ_0, as GitHub code is already part of the Pile. Wiki-de and Wiki-zh have the highest ℓ_0, as they are far from the mainly English Pile. The two math corpora, which overlap with mathematical text in the Pile, lie in between. Wiki-ar also lies in between, below FineMath-4 and above FineMath-3 at every scale.

Continued pretraining with MUON lowers the validation loss in every setting, but a larger ℓ_0 does not always give a larger reduction. For example, Wiki-ar starts below FineMath-4 yet gains about twice as much at every scale. FineMath-4 has higher ℓ_0 than FineMath-3 and gains more at every scale. CodeParrot starts with the lowest ℓ_0 and has the smallest reductions at 160M and 410M, with $\Delta = -0.479$ at 14M and -0.118 at 410M. Among the multilingual corpora, Wiki-de starts with the highest ℓ_0. Wiki-zh and Wiki-ar keep large gains through 160M and still improve by about 0.7 at 410M. Thus, ℓ_0 describes how well the pretrained model already fits a corpus, while Δ records the change achieved within the adaptation budget.

G.2 ABLATION STUDIES

Fixed fraction versus noise calibration. Both variants use the same norm-matched two-level rule with $\rho = 4$. They differ in how they choose the split and in how they track the spike subspace (Appendix C.3). Table 9 shows that the best fixed fraction varies across models and corpora. BB-ADAPTIVE instead sets the split for each matrix and step from the noise edge. On average, it keeps 4.6% of d_{model} directions as spikes (5th–95th percentile: 1.3–7.7%; per-corpus means: 2.8–5.8%).

Without any fraction search, it performs on par with the shared BB-STATIC$_{(5\%)}$. Their aggregate losses are close (Table 2), and BB-ADAPTIVE has lower loss in 10 settings, higher loss in 6, and equal loss in 14. BB-STATIC$_{(5\%)}$ has the lower per-step cost (Table 14). Selecting the fraction for each setting gives further gains: BB-STATIC† has lower loss than BB-ADAPTIVE in 20 settings, higher loss in 4, and equal loss in 6 (Table 1). Noise calibration thus removes the fraction search, a shared fraction lowers the per-step cost, and per-setting selection lowers the loss further at an extra tuning cost.

Adding a third band. We also test whether a finer split improves on two bands. The three-band variant of BB-STATIC splits the singular directions of each matrix into the leading 5%, the next 15%, and the remaining 80%. Its leading band is the same as in BB-STATIC$_{(5\%)}$, so the comparison isolates the effect of splitting the bulk into two parts. Averaged over all 30 model–corpus settings, the three-band variant ends with a final loss 0.002 ± 0.001 higher than BB-STATIC$_{(5\%)}$. The extra band therefore does not help, and two bands remain the better choice. This is consistent with Theorem 1, under which a finer split can improve on two bands only through utility differences within the bands.

Single-update allocation experiment. On Pythia-70M MUON trajectories for FineMath-4, CodeParrot, and Wiki-ar, we compare exact-SVD updates with $\rho = 1.25$ and 1.5 against the exact flat update $\rho = 1$ near 25M and 50M tokens. Each comparison fixes the current singular frames and noise-edge partitions, applies a common ρ across 24 hidden weight matrices, and matches each matrix's Frobenius direction norm separately. Both bulk-emphasizing gains lower the validation loss immediately after the update in five of six corpus–checkpoint settings, despite paired pre-update linearized predictions that favor less bulk weight. These $K = 0$ measurements illustrate the difference between pre-update alignment and the post-update loss response studied in Theorem 1. Appendix E.2 gives the protocol, and Table 7 reports all six settings.

Gain sensitivity and numerical accuracy. For BB-ADAPTIVE, we sweep $\rho \in \{2, 4, 6, 8, 10\}$ across 30 model–corpus settings using 20M tokens, one seed, and the shared learning rates (Table 12). Within a setting, the loss changes by a median of 0.009 across the tested gains, with larger changes at 70M. The best tested gain improves on $\rho = 4$ by about 0.0045 on average. The smallest gain, $\rho = 2$, is best in 20 settings. We keep $\rho = 4$ as the shared default in all 100M-token runs. The sweep covers bulk-emphasizing gains, and Appendix E.2 studies gains near $\rho = 1$. At $\rho = 4$, the per-scale median effective gains ρ_{eff} lie in $[3.963, 4.005]$, and the per-scale median Rayleigh–Ritz residuals stay below 10^{-5} (Table 13). These diagnostics check the realized gain and the accuracy of the retained spike subspaces.

Effect of the spike fraction in BB-STATIC. Table 9 presents the results of tuning the fixed fraction p used by BB-STATIC. The optimal value varies substantially across model scales and corpora. Thus, no single value of p is optimal across all experiments, and selecting one requires an expensive tuning of parameter p. In contrast, BB-ADAPTIVE determines the rank directly from the spectrum of each matrix using the Marchenko–Pastur noise edge. It selects, on average, 4.6% of d_{model} (5th–95th percentile: 1.3%–7.7%; per-corpus means: 2.8%–5.8%) and, without tuning, closely matches the best fixed fraction across scales. The shared $p = 5\%$ configuration achieves aggregate validation loss comparable to BB-ADAPTIVE (Table 2). Table 1 instead reports BB-STATIC†, selected separately for each model–corpus setting from the five fractions in Table 9. *The benefit of the adaptive rule is therefore not improved accuracy over a tuned fixed fraction, but achieving comparable accuracy without the tuning*.

Frequency of improvement and token savings. Table 10 shows which settings each method wins and its token savings in each, complementing the aggregate comparison in Table 2. BB-STATIC$_{(5\%)}$, BB-ADAPTIVE, and BB-STATIC† have lower three-seed mean final loss than MUON in 23, 24, and 25 of the 30 settings, respectively, compared with 19 for FREON. These counts match the filled cells and exclude ties. Within each winning setting, the savings are averaged over the seeds that also finish below MUON's same-seed final loss. A superscript marks a setting won in fewer than three seeds, and an entry without one is won in all three seeds. For example, on Pythia-70M CodeParrot, BB-STATIC$_{(5\%)}$ and BB-ADAPTIVE save $11.2 \pm 1.6\%$ and $10.7 \pm 1.4\%$ of the budget, and BB-STATIC† saves $13.1 \pm 1.9\%$. Table 14 reports the per-step cost.

Model	Method	FineMath-4	FineMath-3	CodeParrot	Wiki-de	Wiki-zh	Wiki-ar
14M	BB-STATIC$_{(1.5\%)}$	3.088 ± 0.003	2.642 ± 0.002	1.903 ± 0.003	$\mathbf{3.278} \pm 0.001$	2.741 ± 0.000	1.999 ± 0.001
	BB-STATIC$_{(2\%)}$	3.086 ± 0.002	2.639 ± 0.001	1.899 ± 0.003	3.279 ± 0.000	2.739 ± 0.001	$\mathbf{1.998} \pm 0.001$
	BB-STATIC$_{(5\%)}$	3.085 ± 0.003	2.639 ± 0.002	1.900 ± 0.003	$\mathbf{3.278} \pm 0.001$	$\mathbf{2.738} \pm 0.001$	2.000 ± 0.000
	BB-STATIC$_{(15\%)}$	$\mathbf{3.083} \pm 0.002$	2.639 ± 0.002	$\mathbf{1.898} \pm 0.003$	3.279 ± 0.001	2.742 ± 0.001	2.006 ± 0.001
	BB-STATIC$_{(20\%)}$	$\mathbf{3.083} \pm 0.002$	$\mathbf{2.638} \pm 0.002$	1.899 ± 0.004	3.282 ± 0.002	2.745 ± 0.001	2.009 ± 0.001
31M	BB-STATIC$_{(1.5\%)}$	2.776 ± 0.001	2.433 ± 0.002	1.650 ± 0.002	2.956 ± 0.001	$\mathbf{2.463} \pm 0.001$	$\mathbf{1.848} \pm 0.001$
	BB-STATIC$_{(2\%)}$	2.776 ± 0.002	2.433 ± 0.002	1.649 ± 0.001	2.956 ± 0.002	2.464 ± 0.001	$\mathbf{1.848} \pm 0.001$
	BB-STATIC$_{(5\%)}$	2.773 ± 0.002	2.431 ± 0.003	$\mathbf{1.648} \pm 0.002$	$\mathbf{2.953} \pm 0.001$	2.464 ± 0.001	1.849 ± 0.001
	BB-STATIC$_{(15\%)}$	2.773 ± 0.002	$\mathbf{2.428} \pm 0.002$	$\mathbf{1.648} \pm 0.002$	2.956 ± 0.001	2.469 ± 0.001	1.858 ± 0.000
	BB-STATIC$_{(20\%)}$	$\mathbf{2.772} \pm 0.001$	2.429 ± 0.002	1.650 ± 0.003	2.960 ± 0.001	2.472 ± 0.001	1.862 ± 0.001
70M	BB-STATIC$_{(1.5\%)}$	2.549 ± 0.004	2.295 ± 0.006	1.478 ± 0.003	2.677 ± 0.002	2.204 ± 0.000	$\mathbf{1.690} \pm 0.001$
	BB-STATIC$_{(2\%)}$	2.547 ± 0.002	2.294 ± 0.004	1.478 ± 0.002	2.678 ± 0.002	2.205 ± 0.002	$\mathbf{1.690} \pm 0.000$
	BB-STATIC$_{(5\%)}$	2.541 ± 0.004	2.290 ± 0.003	1.472 ± 0.003	2.673 ± 0.002	$\mathbf{2.203} \pm 0.003$	$\mathbf{1.690} \pm 0.001$
	BB-STATIC$_{(15\%)}$	$\mathbf{2.537} \pm 0.002$	$\mathbf{2.287} \pm 0.007$	$\mathbf{1.469} \pm 0.003$	$\mathbf{2.670} \pm 0.002$	2.204 ± 0.002	1.695 ± 0.001
	BB-STATIC$_{(20\%)}$	$\mathbf{2.537} \pm 0.004$	$\mathbf{2.287} \pm 0.003$	$\mathbf{1.469} \pm 0.004$	2.674 ± 0.001	2.205 ± 0.002	1.700 ± 0.001
160M	BB-STATIC$_{(1.5\%)}$	$\mathbf{2.281} \pm 0.003$	$\mathbf{2.077} \pm 0.002$	$\mathbf{1.280} \pm 0.004$	2.458 ± 0.002	$\mathbf{2.051} \pm 0.001$	$\mathbf{1.597} \pm 0.002$
	BB-STATIC$_{(2\%)}$	$\mathbf{2.281} \pm 0.002$	$\mathbf{2.077} \pm 0.003$	1.281 ± 0.003	$\mathbf{2.457} \pm 0.003$	$\mathbf{2.051} \pm 0.000$	1.598 ± 0.002
	BB-STATIC$_{(5\%)}$	2.283 ± 0.003	$\mathbf{2.077} \pm 0.003$	1.283 ± 0.004	2.459 ± 0.002	2.054 ± 0.001	1.606 ± 0.001
	BB-STATIC$_{(15\%)}$	2.285 ± 0.002	2.079 ± 0.002	1.283 ± 0.004	2.466 ± 0.002	2.066 ± 0.000	1.625 ± 0.001
	BB-STATIC$_{(20\%)}$	2.284 ± 0.003	2.080 ± 0.002	1.284 ± 0.003	2.470 ± 0.003	2.072 ± 0.001	1.632 ± 0.001
410M	BB-STATIC$_{(1.5\%)}$	$\mathbf{2.043} \pm 0.005$	$\mathbf{1.890} \pm 0.002$	$\mathbf{1.111} \pm 0.002$	$\mathbf{2.174} \pm 0.001$	$\mathbf{1.870} \pm 0.001$	$\mathbf{1.446} \pm 0.001$
	BB-STATIC$_{(2\%)}$	$\mathbf{2.043} \pm 0.005$	1.891 ± 0.001	$\mathbf{1.111} \pm 0.002$	$\mathbf{2.174} \pm 0.001$	1.871 ± 0.001	1.447 ± 0.001
	BB-STATIC$_{(5\%)}$	$\mathbf{2.043} \pm 0.005$	$\mathbf{1.890} \pm 0.001$	$\mathbf{1.111} \pm 0.002$	2.176 ± 0.000	1.875 ± 0.001	1.453 ± 0.001
	BB-STATIC$_{(15\%)}$	2.044 ± 0.005	$\mathbf{1.890} \pm 0.002$	1.112 ± 0.002	2.182 ± 0.001	1.888 ± 0.001	1.468 ± 0.001
	BB-STATIC$_{(20\%)}$	2.044 ± 0.005	$\mathbf{1.890} \pm 0.002$	1.112 ± 0.001	2.185 ± 0.001	1.894 ± 0.001	1.473 ± 0.001

Table 9: Fixed-fraction sweep for BB-STATIC: final validation loss after about 100M training tokens, reported as mean $\pm$ standard deviation over three seeds. For $p \in \{0.015, 0.02, 0.05, 0.15, 0.20\}$, the leading band contains $r_t = \lceil pd \rceil$ singular directions, where $d = \min\{m, n\}$ and p is held fixed within each run. All fractions use $\rho = 4$ and the shared learning rate of their scale, with the same training budget, schedule, and seeds as Table 1. For each model–corpus setting, the fraction with the lowest three-seed mean gives the BB-STATIC† entry in Table 1. Table 2 summarizes the shared configuration $p = 5\%$, and Figure 1 shows $p = 1.5\%$. **Bold** marks the lowest mean among the fractions in each setting, including ties at the printed precision.

Arm	Model	FineMath-4	FineMath-3	CodeParrot	Wiki-de	Wiki-zh	Wiki-ar
FREON	14M	8.3 ± 0.7	3.5 ± 2.9	7.1 ± 1.6	4.9 ± 2.6	4.4 ± 2.6	–
	31M	7.8 ± 1.8	8.7 ± 3.5	3.3 ± 2.9	4.4 ± 1.7	4.7 ± 0.7	–
	70M	11.1 ± 1.4	10.6 ± 3.2	12.5 ± 0.7	9.8 ± 2.6	10.9 ± 3.3	10.2 ± 1.5
	160M	–	–	–	–	–	–
	410M	9.9 ± 3.1	9.6 ± 3.5	–	6.0 ± 5.1	–	–
BB-STATIC$_{(5\%)}$	14M	4.6 ± 0.4	3.4 ± 2.9	6.1 ± 2.1	6.3 ± 2.1	10.5 ± 2.4	12.3 ± 2.2
	31M	5.2 ± 0.6	–	3.1 ± 2.8	8.2 ± 2.2	9.5 ± 0.6	9.8 ± 2.1
	70M	8.4 ± 2.9	9.2 ± 3.3	11.2 ± 1.6	7.0 ± 1.7	7.7 ± 1.7	13.2 ± 0.6
	160M	–	–	–	–	2.6 ± 0.8	4.6 ± 2.1
	410M	10.0 ± 2.9	6.9 ± 5.5	–	10.6 ± 3.1	9.3 ± 1.6	–
BB-ADAPTIVE	14M	5.2 ± 1.8	$4.4 \pm 1.6^{\,2/3}$	6.5 ± 2.6	7.3 ± 2.8	11.3 ± 2.6	11.5 ± 2.0
	31M	4.5 ± 2.2	$7.5 \pm 1.3^{\,2/3}$	4.6 ± 3.9	8.7 ± 1.2	10.0 ± 0.2	9.6 ± 2.4
	70M	5.6 ± 2.9	8.0 ± 4.8	10.7 ± 1.4	5.8 ± 1.2	6.7 ± 1.6	13.1 ± 0.3
	160M	–	–	–	–	3.6 ± 0.8	6.8 ± 1.2
	410M	10.7 ± 3.0	7.2 ± 4.6	–	11.3 ± 1.8	9.7 ± 1.9	–
BB-STATIC†	14M	8.1 ± 0.8	4.4 ± 3.9	7.8 ± 2.7	6.3 ± 2.1	10.5 ± 2.4	13.7 ± 1.6
	31M	6.3 ± 0.7	10.0 ± 1.7	3.1 ± 2.8	8.2 ± 2.2	10.7 ± 0.7	12.6 ± 2.8
	70M	11.1 ± 3.5	11.7 ± 6.2	13.1 ± 1.9	8.0 ± 4.5	7.7 ± 1.7	13.2 ± 0.6
	160M	–	–	–	–	7.5 ± 1.1	15.3 ± 2.4
	410M	10.1 ± 2.9	8.5 ± 4.5	–	13.9 ± 1.6	15.3 ± 1.2	12.8 ± 1.0

Table 10: Token savings against MUON, as a percentage of the 100M-token budget, for each method and model–corpus setting. MUON is the reference, so it has no block of its own. A cell is filled only where the method's three-seed mean final loss is below MUON's in Table 1, and "–" marks settings where MUON is at least as good. A filled cell reports $1 - D^{\star}/100\mathrm{M}$, where $D^{\star}$ is the number of tokens at which the method's curve first reaches MUON's 100M-token loss in the same seed. Values are mean $\pm$ standard deviation over the seeds in which the method ends below MUON. A superscript $k/3$ marks a setting won in only k of the three seeds. DEVA and SPECTRA never beat MUON, so they are omitted. The last block, BB-STATIC†, uses the best of the five fractions in each setting (Table 9). Because this fraction is chosen on the same runs, its savings are optimistic. Separately, intermediate points of a cosine schedule are evaluated before the learning rate has fully decayed.

Learning-rate tuning and the shared-rate policy. We distinguish each method's best multiplier in the 20M-token sweep from the shared multiplier used in the 100M-token runs. Figure 2 and Table 11 report the sweep on FineMath-4, run with one seed over multiples of the reference rate of Section 5. The swept methods are MUON, FREON, DEVA, SPECTRA, BB-STATIC, and BB-ADAPTIVE. In the sweep, BB-STATIC uses $p = 20\%$, and both BULKBOOST variants use $\rho = 4$. Each method's best multiplier at a scale is the one with the lowest validation loss after 20M tokens.

The best multipliers agree across all six methods at 14M, 31M, and 410M, and across five of the six at 70M and 160M. At 70M, MUON does best at $0.1\times$, while the other methods do best at $0.25\times$. At 160M, SPECTRA does best at $0.05\times$, while the other methods do best at $0.1\times$. The shared multipliers are therefore 0.1, 0.1, 0.25, 0.1, and 0.05 for Pythia-14M, 31M, 70M, 160M, and 410M, and each is the best multiplier for most of the six methods. All 100M-token runs use the resulting peak learning rates for every method, including ADAMW, on all six corpora. In particular, the 70M MUON runs use 5×10^{-4} and the 160M SPECTRA runs use 10^{-4}, instead of their own best sweep rates. In the sweep, this raises the loss from 2.779 to 2.783 for 70M MUON, and from 2.552 to 2.553 for 160M SPECTRA. Table 11 marks each method's best multiplier, and its caption lists the shared multipliers.

Computation-time overhead. Table 14 reports per-step runtimes relative to MUON at each model scale, averaged over the six corpora. The measured fixed-rank configurations with $p = 1.5\%$ and 5% require 1.19–$1.59\times$ MUON's step time, compared with 1.36–$2.13\times$ for BB-ADAPTIVE. At 70M–410M, these ranges narrow to 1.19–$1.27\times$ and 1.36–$1.50\times$, respectively. The two measured fixed fractions have nearly identical runtimes and are faster per step than the noise-calibrated variant at every scale. For the other methods, ADAMW and SPECTRA take 0.92–$0.98\times$ and 0.98–$1.00\times$ MUON's step time, respectively, while FREON and DEVA incur overheads of up to 20% and 34%.

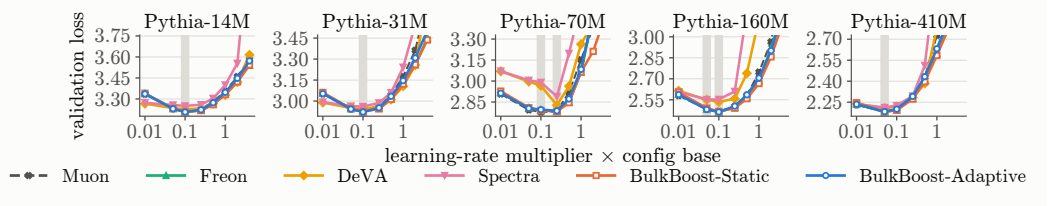


Figure 2: Learning-rate sweeps on FineMath-4 with a 20M-token budget and one seed. Curves show final validation loss against multiples of each scale's reference rate. Grey bands mark each method's best multiplier. BB-STATIC uses $p = 20\%$ in the sweep. Table 11 lists the full results.

Model	Method	0.01×	0.05×	0.1×	0.25×	0.5×	1×	2×	4×	best
14M	MUON	3.335	3.232	**3.209**	3.227	3.281	3.354	3.459	3.581	0.1×
	FREON	3.338	3.231	**3.207**	3.219	3.265	3.335	3.427	3.540	0.1×
	DEVA	3.266	3.235	**3.226**	3.232	3.267	3.328	3.421	3.615	0.1×
	SPECTRA	3.272	3.257	**3.250**	3.259	3.304	3.400	3.555	4.458	0.1×
	BB-STATIC$_{(20\%)}$	3.340	3.232	**3.207**	3.217	3.259	3.332	3.422	3.543	0.1×
	BB-ADAPTIVE	3.336	3.230	**3.207**	3.223	3.273	3.347	3.448	3.572	0.1×
31M	MUON	3.051	2.945	**2.922**	2.958	3.042	3.176	3.363	3.733	0.1×
	FREON	3.055	2.945	**2.924**	2.948	3.016	3.125	3.259	3.450	0.1×
	DEVA	2.989	2.950	**2.942**	2.963	3.011	3.106	3.257	3.747	0.1×
	SPECTRA	2.993	2.965	**2.960**	2.987	3.060	3.242	3.547	4.848	0.1×
	BB-STATIC$_{(20\%)}$	3.061	2.947	**2.925**	2.944	3.012	3.122	3.261	3.436	0.1×
	BB-ADAPTIVE	3.053	2.943	**2.922**	2.952	3.034	3.146	3.308	3.504	0.1×
70M	MUON	2.908	2.794	**2.779**	2.783	2.919	3.152	3.386	3.986	0.1×
	FREON	2.918	2.808	2.792	**2.785**	2.848	3.064	3.473	3.674	0.25×
	DEVA	3.070	2.996	2.966	**2.827**	2.963	3.263	3.415	4.610	0.25×
	SPECTRA	3.071	3.005	2.991	**2.890**	3.198	3.547	5.466	5.936	0.25×
	BB-STATIC$_{(20\%)}$	2.928	2.809	2.789	**2.782**	2.845	3.065	3.212	3.467	0.25×
	BB-ADAPTIVE	2.913	2.805	2.796	**2.787**	2.872	3.083	3.484	3.634	0.25×
160M	MUON	2.580	2.483	**2.466**	2.503	2.579	2.747	2.964	3.434	0.1×
	FREON	2.590	2.484	**2.463**	2.498	2.569	2.687	2.918	3.196	0.1×
	DEVA	2.615	2.550	**2.535**	2.558	2.739	3.130	3.545	4.937	0.1×
	SPECTRA	2.607	**2.552**	2.553	2.610	3.182	3.940	5.445	6.425	0.05×
	BB-STATIC$_{(20\%)}$	2.603	2.490	**2.466**	2.490	2.558	2.666	2.857	3.107	0.1×
	BB-ADAPTIVE	2.587	2.481	**2.463**	2.505	2.585	2.706	2.901	3.180	0.1×
410M	MUON	2.232	**2.182**	2.199	2.290	2.429	2.728	3.047	3.204	0.05×
	FREON	2.236	**2.180**	2.196	2.278	2.412	2.598	2.799	2.989	0.05×
	DEVA	2.239	**2.208**	2.218	2.282	2.384	2.779	3.263	3.884	0.05×
	SPECTRA	2.235	**2.212**	2.226	2.294	2.512	3.228	3.529	3.963	0.05×
	BB-STATIC$_{(20\%)}$	2.246	**2.182**	2.193	2.273	2.397	2.584	2.783	2.990	0.05×
	BB-ADAPTIVE	2.235	**2.183**	2.200	2.291	2.433	2.632	2.849	3.060	0.05×

Table 11: Final held-out loss on FineMath-4 after 20M tokens with one seed, at multiples of each scale's reference rate. **Bold** marks each method's lowest loss, and underline marks its second lowest. BB-STATIC uses $p = 20\%$. The 100M-token runs use one shared multiplier per scale, the best for most methods: 0.1, 0.1, 0.25, 0.1, and 0.05 from 14M to 410M.

These timings compare the implemented variants, which differ in both noise calibration and subspace tracking, so they do not isolate the cost of rank selection. The overhead of the adaptive variant is largest at 14M (113%) and lies between 36% and 66% at the larger scales.

Model	Corpus	ρ=2	ρ=4	ρ=6	ρ=8	ρ=10	best	range
14M	FineMath-4	3.209	3.209	**3.209**	3.209	3.210	6	0.000
	FineMath-3	2.737	2.737	2.736	2.737	**2.735**	10	0.002
	CodeParrot	2.046	**2.046**	2.048	2.048	2.048	4	0.002
	Wiki-de	**3.518**	3.522	3.523	3.526	3.528	2	0.010
	Wiki-zh	**2.960**	2.962	2.964	2.965	2.967	2	0.007
	Wiki-ar	**2.184**	2.188	2.190	2.193	2.194	2	0.011
31M	FineMath-4	**2.923**	2.925	2.927	2.927	2.928	2	0.005
	FineMath-3	**2.530**	2.534	2.535	2.536	–	2	0.006
	CodeParrot	**1.796**	1.797	1.798	1.800	1.802	2	0.006
	Wiki-de	**3.203**	3.205	3.208	3.210	3.212	2	0.009
	Wiki-zh	**2.702**	2.705	2.709	2.711	2.712	2	0.011
	Wiki-ar	**2.046**	2.051	2.053	2.056	2.058	2	0.012
70M	FineMath-4	**2.780**	2.795	2.806	2.818	2.837	2	0.057
	FineMath-3	2.458	**2.457**	2.481	2.482	2.492	4	0.035
	CodeParrot	**1.677**	1.691	1.714	1.712	1.731	2	0.054
	Wiki-de	**2.985**	3.020	3.030	3.034	3.043	2	0.057
	Wiki-zh	**2.481**	2.493	2.503	2.516	2.528	2	0.047
	Wiki-ar	**1.915**	1.919	1.923	1.926	1.931	2	0.016
160M	FineMath-4	2.466	**2.466**	2.467	2.466	2.468	4	0.002
	FineMath-3	2.205	**2.201**	2.206	2.203	2.210	4	0.009
	CodeParrot	1.408	**1.406**	1.407	1.407	1.409	4	0.002
	Wiki-de	**2.698**	2.702	2.704	2.706	2.707	2	0.010
	Wiki-zh	**2.328**	2.332	2.337	2.339	2.340	2	0.012
	Wiki-ar	**1.859**	1.863	1.864	1.866	1.870	2	0.011
410M	FineMath-4	2.181	2.181	**2.180**	2.181	2.181	6	0.001
	FineMath-3	1.979	1.978	1.977	**1.976**	1.976	8	0.002
	CodeParrot	1.194	1.192	1.191	1.191	**1.190**	10	0.003
	Wiki-de	**2.348**	2.348	2.348	2.349	2.349	2	0.001
	Wiki-zh	**2.093**	2.097	2.099	2.100	2.101	2	0.008
	Wiki-ar	**1.668**	1.674	1.676	1.678	1.679	2	0.012

Table 12: ρ tuning across all 30 model–corpus pairs, measured by final held-out validation loss after 20M tokens. The best loss in each row is **bold**, and the second-best is underlined. The *range* column is the difference between the largest and smallest loss, computed before rounding; its median is 0.009. At each model scale, the sweep uses the same peak learning rate as the main 100M-token BB-ADAPTIVE runs (Table 11), held fixed across ρ values and corpora. The sweep uses a different seed from the learning-rate sweep, so its $\rho = 4$ entries on FineMath-4 differ from the BB-ADAPTIVE entries of Table 11 by at most 0.008.

Model	matrices	median ρ_{eff}	p5–p95	$\rho_{\text{eff}} - \rho$	Ritz residual
Pythia-14M	144	3.980	3.88–4.07	-0.020	4.1×10^{-7}
Pythia-31M	144	3.980	3.91–4.04	-0.020	4.7×10^{-7}
Pythia-70M	144	4.005	3.95–4.06	$+0.005$	6.2×10^{-7}
Pythia-160M	288	3.963	3.92–4.05	-0.037	1.5×10^{-6}
Pythia-410M	576	3.965	3.93–4.02	-0.035	8.2×10^{-6}

Table 13: Numerical diagnostics for BB-ADAPTIVE at the target gain $\rho = 4$, summarized by model scale. "matrices" counts the instrumented two-dimensional weight matrices across the six corpora. We report the median effective gain ρ_{eff}, its 5th–95th percentile interval, and the difference between the median and the target gain. The final column reports the median Rayleigh–Ritz residual of the retained spike subspaces.

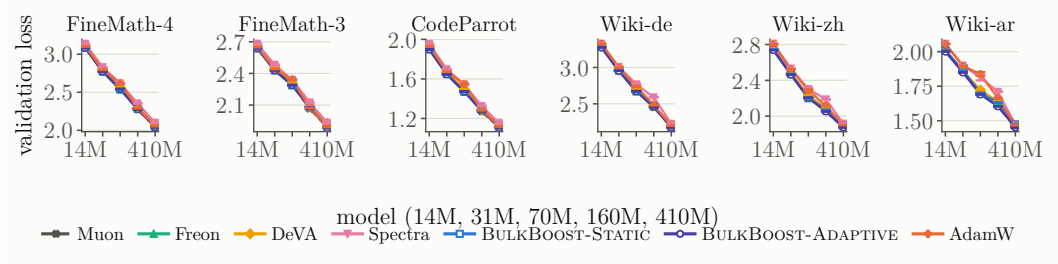


Figure 3: Final held-out validation loss as a function of model size, with one panel per corpus and all seven optimization methods shown. Across all corpora, final loss decreases consistently as model size increases, matching the expected scaling trend. The rate of decrease differs by corpus, reflecting differences in adaptation difficulty, but the overall monotonic improvement with scale remains stable. The BB-STATIC curve uses $p = 20\%$ (Table 9); the other curves match Table 1.

Arm	14M	31M	70M	160M	410M
MUON (ms/step)	173 ± 5	308 ± 1	617 ± 0	635 ± 1	1748 ± 10
ADAMW	$0.93 \pm 0.05\times$	$0.92 \pm 0.01\times$	$0.98 \pm 0.01\times$	$0.98 \pm 0.00\times$	$0.98 \pm 0.01\times$
FREON	$1.17 \pm 0.01\times$	$1.10 \pm 0.00\times$	$1.06 \pm 0.00\times$	$1.17 \pm 0.00\times$	$1.20 \pm 0.00\times$
DEVA	$1.13 \pm 0.00\times$	$1.10 \pm 0.00\times$	$1.11 \pm 0.00\times$	$1.27 \pm 0.00\times$	$1.34 \pm 0.00\times$
SPECTRA	$0.98 \pm 0.01\times$	$1.00 \pm 0.00\times$	$1.00 \pm 0.00\times$	$1.00 \pm 0.00\times$	$1.00 \pm 0.00\times$
BB-STATIC$_{(1.5\%)}$	$1.59 \pm 0.05\times$	$1.32 \pm 0.01\times$	$1.19 \pm 0.00\times$	$1.27 \pm 0.00\times$	$1.27 \pm 0.01\times$
BB-STATIC$_{(5\%)}$	$1.59 \pm 0.04\times$	$1.33 \pm 0.01\times$	$1.19 \pm 0.00\times$	$1.27 \pm 0.00\times$	$1.27 \pm 0.01\times$
BB-ADAPTIVE	$2.13 \pm 0.05\times$	$1.66 \pm 0.00\times$	$1.36 \pm 0.00\times$	$1.50 \pm 0.00\times$	$1.48 \pm 0.01\times$

Table 14: Per-step wall-clock time at each model scale. For each run, the per-step time is its total wall-clock time divided by its number of steps. Entries are mean $\pm$ standard deviation over the runs at that scale, across six corpora and three seeds. The MUON row reports milliseconds per step; all other rows report runtime relative to MUON at the same scale. Fixed-rank timings cover $p = 1.5\%$ and 5%. Measurements were collected on shared cluster nodes and are subject to scheduling variability. The ratios within a scale are stable.

Model	d_{model}	micro-batch $\times$ accum	steps	tokens	MUON ms/step
Pythia-14M	128	32×1	3051	100M	173 ± 5
Pythia-31M	256	32×1	3051	100M	308 ± 1
Pythia-70M	512	32×1	3051	100M	617 ± 0
Pythia-160M	768	32×1	3051	100M	635 ± 1
Pythia-410M	1024	32×1	3051	100M	1748 ± 10

Table 15: Pythia model configurations and training budgets. d_{model} denotes the hidden width, and "micro-batch $\times$ accum" gives the number of sequences per microbatch and the number of gradient accumulation steps. All scales use an effective batch of 32 sequences of length 1024, giving 32,768 tokens per step and approximately 100M tokens over 3,051 steps. The MUON timing entries reproduce those in Table 14: mean $\pm$ standard deviation over six corpora and three seeds at each scale.

G.3 ADDITIONAL LEARNING CURVES

Final loss against model size. Figure 3 plots final loss against model size, with one panel per corpus and all seven methods shown on an absolute loss axis. The curves match Table 1, except that BB-STATIC is shown at $p = 20\%$ (Table 9). This view makes the scaling trend clear: final loss decreases steadily with model size on every corpus.

Figures 4–8 show learning curves for the five remaining corpora—FineMath-4, FineMath-3, CodeParrot, Wiki-de, and Wiki-ar. Those figures use the same layout as Figure 1 (Wiki-zh). All fixed-rank curves use $p = 1.5\%$.

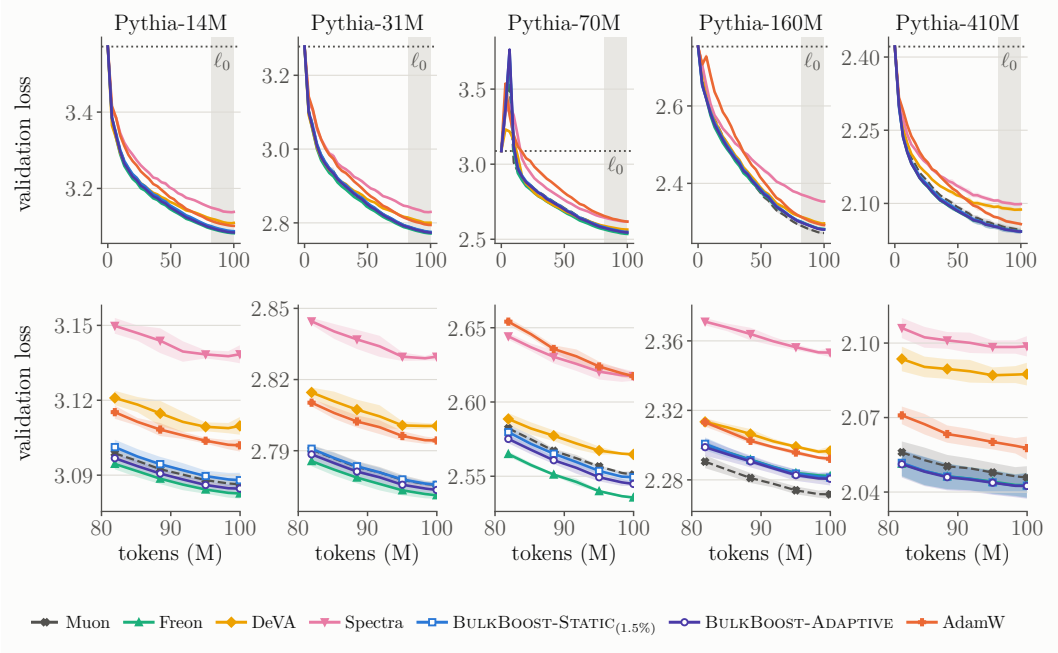


Figure 4: Held-out loss vs. tokens on FineMath-4, one column per scale. Top: the full run, with the pre-adaptation loss ℓ_0 dotted and the zoom window shaded. Bottom: the shaded window, rescaled. Note the y-ranges differ per panel.

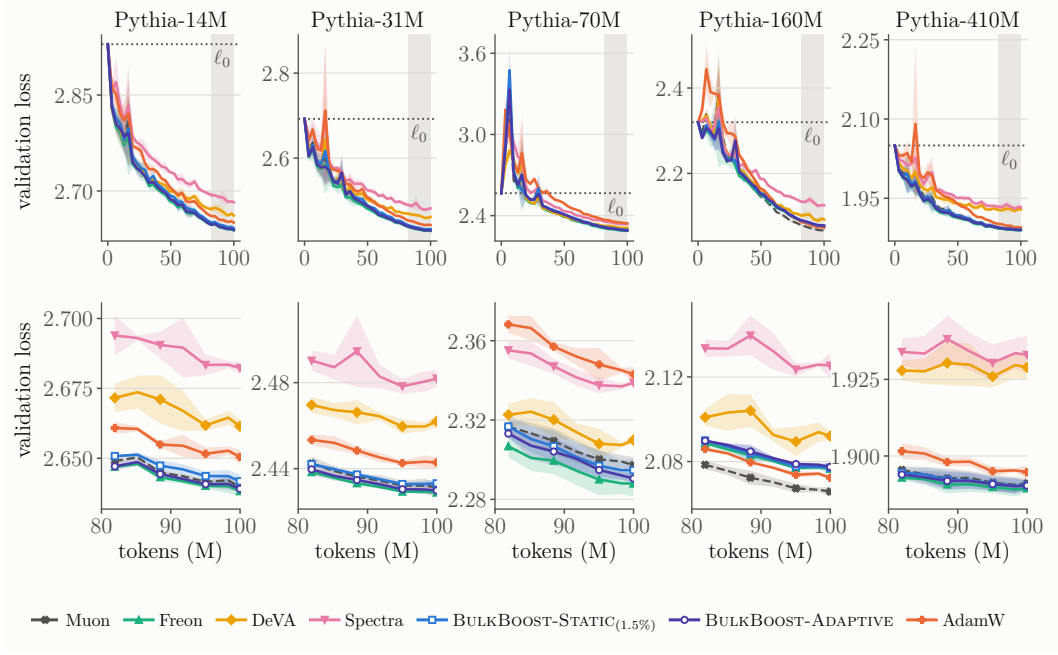


Figure 5: Held-out loss vs. tokens on FineMath-3, one column per scale. Top: the full run, with the pre-adaptation loss ℓ_0 dotted and the zoom window shaded. Bottom: the shaded window, rescaled. Note the y-ranges differ per panel.

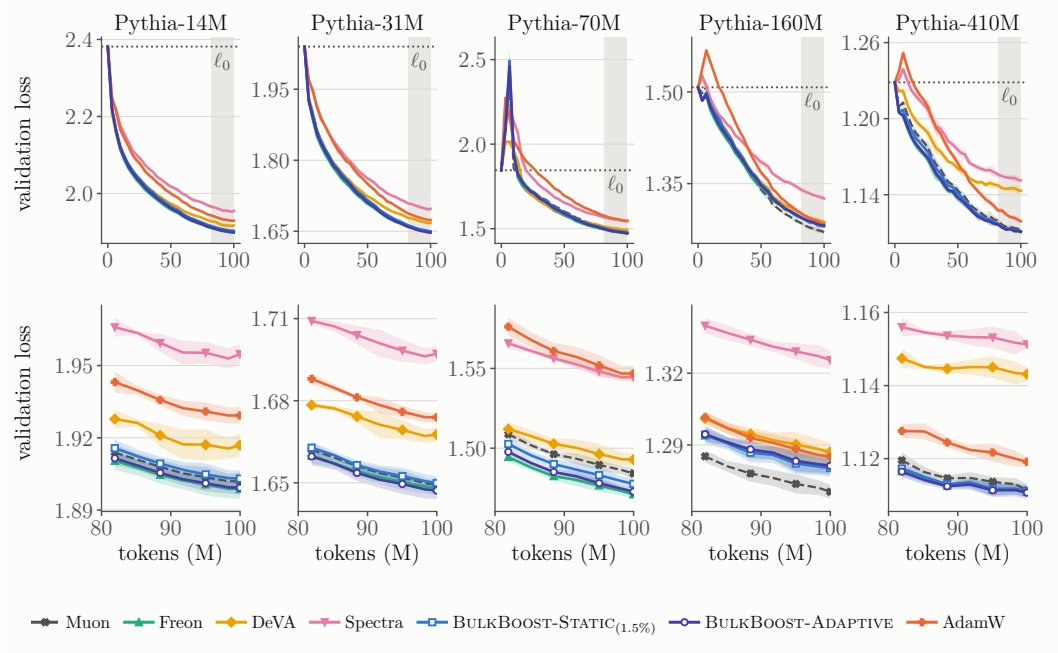


Figure 6: Held-out loss vs. tokens on CodeParrot, one column per scale. Top: the full run, with the pre-adaptation loss ℓ_0 dotted and the zoom window shaded. Bottom: the shaded window, rescaled. Note the y-ranges differ per panel.

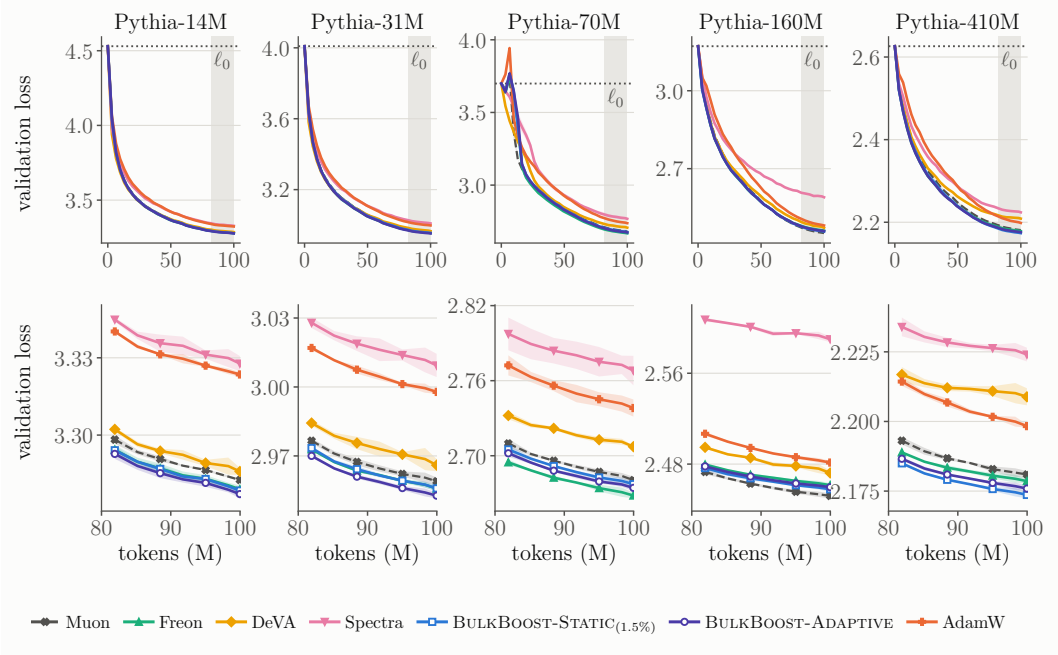


Figure 7: Held-out loss vs. tokens on Wiki-de, one column per scale. Top: the full run, with the pre-adaptation loss ℓ_0 dotted and the zoom window shaded. Bottom: the shaded window, rescaled. Note the y-ranges differ per panel.

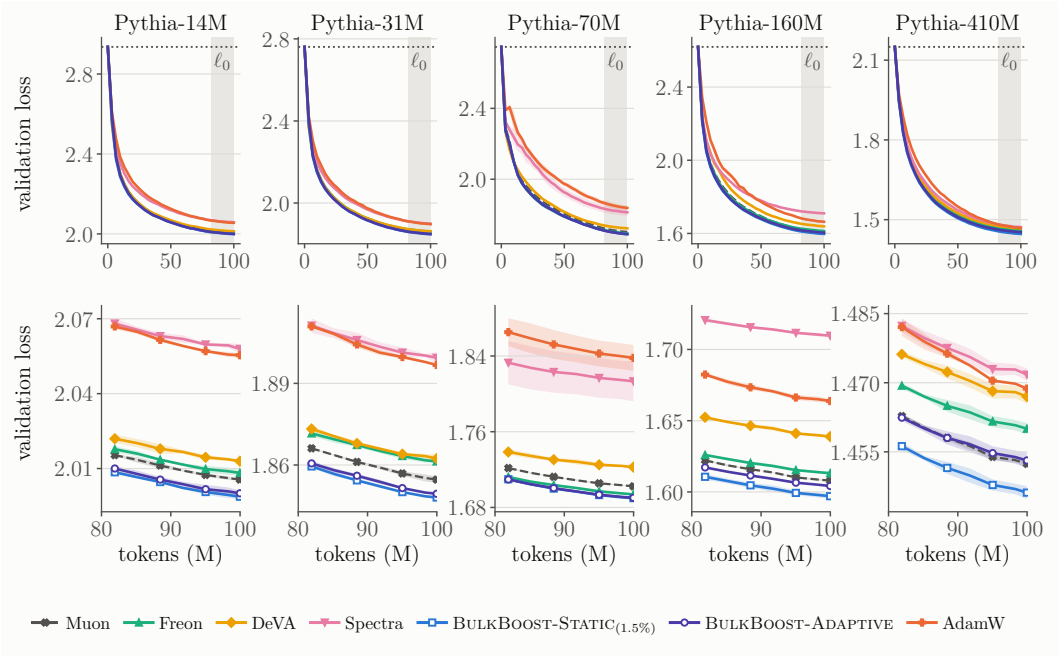


Figure 8: Held-out loss vs. tokens on Wiki-ar, one column per scale. Top: the full run, with the pre-adaptation loss ℓ_0 dotted and the zoom window shaded. Bottom: the shaded window, rescaled. Note the y-ranges differ per panel.