SPECTRAL DEFLATION FOR FACTORIZATION-FREE MATRIX FILTERING IN MUON AND SEMIDEFINITE PROGRAMMING

Anonymous authors
Paper under double-blind review

ABSTRACT

GPU implementations of the Muon optimizer and of first-order semidefinite programming (SDP) solvers share one computational pattern: a matrix factorization is replaced by a fixed-depth polynomial filter applied after normalization. When a few dominant spectral components carry most of the input's scale, normalization pushes the remaining spectrum toward zero, where the filter is least accurate. We propose *spectral deflation*: estimate the dominant components, remove them, filter the normalized residual with the unchanged filter, and restore them. Deflation preserves the target matrix function, and we prove that it strictly reduces the finite-step error of the classical Newton–Schulz family. Implemented with batched randomized SVD for Muon and warm-started subspace tracking for ADMM, deflation consistently improves GPT-2 pretraining over the corresponding Muon baselines with both the Newton–Schulz and Polar Express mappings, also in wall-clock time, and lowers the KKT residuals of factorization-free ADMM on large-scale SDPs within a similar projection time.

1 INTRODUCTION

Large-scale numerical computing on GPUs increasingly replaces matrix factorizations with matrix–matrix products. A prominent instance of this trend approximates a spectral matrix function $f(A)$ by a *fixed-depth polynomial filter*: the input is normalized once, and a short fixed sequence of polynomials is applied,

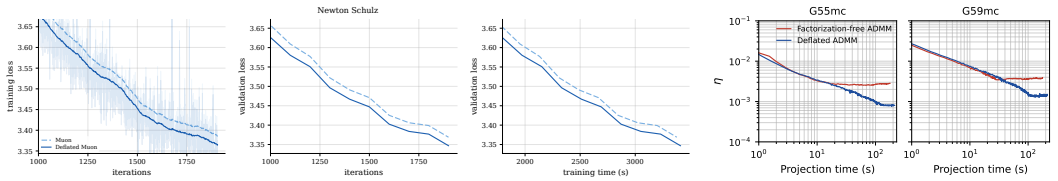
$$X_0 = \frac{A}{\theta(A)}, \qquad X_t = p_t(X_{t-1}), \quad t = 1, \dots, L, \qquad X_L \approx f(A), \tag{1}$$

where $\theta(A) > 0$ is a normalization scale and each p_t is evaluated by a few matrix–matrix multiplications. The filter avoids any factorization and is therefore fast and low-precision friendly; however, because the depth L is fixed, its accuracy depends strongly on the normalized spectrum of X_0.

Two applications. In deep learning, the Muon optimizer updates every matrix-valued parameter W in the direction of the polar factor of its momentum matrix $M \in \mathbb{R}^{n\times m}$ (Jordan et al., 2024; Liu et al., 2025): $W \leftarrow W - \lambda\,\mathrm{polar}(M)$, where $\mathrm{polar}(M) := UV^\top$ for a compact singular value decomposition $M = U\Sigma V^\top$. In words, Muon keeps the singular directions of M and maps every nonzero singular value to one. An SVD at every optimizer step is too expensive, so Muon computes (1) with $\theta = \|M\|_F$ and five fixed Newton–Schulz-type polynomials. In numerical optimization, first-order semidefinite programming (SDP) solvers repeatedly project a symmetric matrix Z onto the positive semidefinite (PSD) cone (Wen et al., 2010; O'Donoghue et al., 2016; Boyd et al., 2011). The projection applies the ReLU function $\max(x, 0)$ to the eigenvalues of Z. Since $\max(x, 0) = x\big(1+\mathrm{sign}(x)\big)/2$, computing the projection reduces to computing the matrix sign function $\mathrm{sign}(Z)$. Recent factorization-free solvers again approximate $\mathrm{sign}(Z)$ by (1), where θ is an estimated upper bound on $\|Z\|_2$ and the fixed polynomials p_t form a composite sign filter (Kang et al., 2025). Muon and factorization-free ADMM therefore share one computational core, and we treat them together.

The spectral floor. In both applications, the matrices entering (1) have spectra of a specific shape. Momentum matrices in GPT-2 training carry a few large singular values and many near zero (Figure 7; see also Magakyan et al., 2026), and matrices inside SDP solvers often approach a head-dominated regime (Kang et al., 2025). Normalizing by the whole matrix then compresses the tail of

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.



(a) DEFLATED MUON (b) ADMM with deflated projection

Figure 1: (a) Training loss against optimizer iterations (left), validation loss against optimizer iterations (middle) and validation loss against training time (right) of GPT-2 Large trained on one billion tokens from FineWeb; we compare Muon with the Newton–Schulz mapping against DEFLATED MUON. (b) KKT residual η of ADMM on the max-cut SDPs `G55mc` and `G59mc` ($n = 5{,}000$) against cumulative projection time, with every PSD projection performed in FP16: the factorization-free filter of Kang et al. (2025) (red) and the same filter behind deflation (blue).

the spectrum. We call the smallest normalized nonzero spectral magnitude of X_0 its *spectral floor*. To see the problem, assume that Muon's polynomial has slope $a > 1$ at the origin, so five iterations amplify a small singular value by at most a^5; a singular value below $1/a^5$ of the Frobenius norm therefore ends far from one, and its direction is strongly attenuated in the computed polar factor. This raises the question at the center of this paper:

Can we lift the small spectral components that a fixed-depth polynomial filter fails to resolve, without redesigning the filter and without any matrix factorization?

We give an affirmative answer through a simple principle: *deflate* the dominant components, *filter* the normalized residual, and *restore* what was removed.

Spectral deflation. Let $M = U_k \Sigma_k V_k^\top + R_k$ split off the k largest singular triplets of M, and let $Z = U_k \Lambda_k U_k^\top + R_k$ analogously split off the k eigenvalues of largest magnitude of a symmetric Z. Both targets decompose exactly:

$$\operatorname{polar}(M) = U_k V_k^\top + \operatorname{polar}(R_k), \qquad \Pi_{\mathbb{S}_+^n}(Z) = U_k (\Lambda_k)_+ U_k^\top + \Pi_{\mathbb{S}_+^n}(R_k), \tag{2}$$

where $(\Lambda_k)_+$ zeroes the negative eigenvalues of the head. In words, deflation never changes the target; it only changes the input that the filter sees. Applying (1) to R_k instead of the full matrix raises every remaining spectral magnitude by the factor $\alpha_k = \theta(A)/\theta(R_k) > 1$, which is large exactly when the head dominates, and the deflated head is handled explicitly through (2). The filter itself is untouched. This distinguishes our approach from work that redesigns the polynomials (Grishina et al., 2025; Amsel et al., 2025), from Muon variants that change the update's scaling or geometry (Appendix A), and from randomized low-rank polar approximations that replace the full-rank target (Choudhury et al., 2026). Classical deflation treats obstructive eigencomponents separately in Krylov methods (Saad et al., 2000; Eiermann et al., 2011), and we adapt the same idea to dense fixed-depth filters, where the obstacle is the normalization scale. Figure 1 previews the end-to-end effect on both applications.

Contributions. Our contributions are threefold. (i) We identify the spectral floor as a shared accuracy bottleneck of fixed-depth polynomial filters in Muon and in factorization-free ADMM, and we formalize *spectral deflation* as a target-preserving wrapper around an unchanged filter, based on the exact decompositions (2). (ii) We instantiate the wrapper as two GPU primitives: a *deflated polar factor* for Muon, driven by a batched randomized SVD that processes all momentum matrices of one shape at once (§2.2), and a *deflated PSD projection* for ADMM, driven by a warm-started subspace iteration that tracks the dominant eigenspace across consecutive projections (§2.3). On the theory side, we prove that exact deflation raises the spectral floor from ℓ to $\min(\alpha_k \ell, \gamma^{-1})$, where γ is the padding parameter, and contracts the finite-step Newton–Schulz error accordingly (Theorem 1), and we prove the analogous statement for the Newton–Schulz PSD projection (Proposition 2); Appendix E.3 gives a matching bound for the fixed projection filter used in practice. (iii) Empirically, deflation improves both applications under the same filter budget: pretraining GPT-2 Large on FineWeb, DEFLATED MUON attains lower validation loss than Muon across learning rates, momentum coefficients, iteration counts, and training horizons, for both mappings and also against

wall-clock time; and on four large SDPs with FP16 projections, ADMM with the deflated projection reaches lower KKT residuals within the same projection-time budget.

Organization. §2 develops spectral deflation: the mechanism (§2.1), the two primitives (§2.2–2.3), and the error analysis (§2.4). §3 evaluates matrix-function accuracy, GPT-2 pretraining, and SDP solving; §4 concludes. Related work is reviewed in Appendix A.

2 Spectral Deflation

Notation. $\sigma_1(A) \geq \sigma_2(A) \geq \cdots$ are singular values and $|\lambda_1(Z)| \geq |\lambda_2(Z)| \geq \cdots$ eigenvalues of a symmetric Z ordered by magnitude; $\|\cdot\|_F$ and $\|\cdot\|_2$ are the Frobenius and spectral norms; $\mathbb{S}^n$ and $\mathbb{S}^n_+$ are the symmetric matrices and the PSD cone, with $\Pi_{\mathbb{S}^n_+}$ the Euclidean projection onto $\mathbb{S}^n_+$; $(x)_+ := \max\{x, 0\}$; and $\sigma(X)$ denotes the set of nonzero singular values of X.

2.1 The Spectral Floor

We first make the mechanism precise for Muon. For the polar factor, the filter (1) with $\theta = \|M\|_F$ is a Newton–Schulz iteration; we write it using the classical degree-five map (Björck & Bowie, 1971),

$$X_0 = \frac{M}{\|M\|_F},\ X_{t+1} = \mathcal{N}(X_t) := aX_t + b(X_tX_t^\top)X_t + c(X_tX_t^\top)^2X_t,\ t = 0, \ldots, L-1, \quad (3)$$

with $a = 15/8$, $b = -5/4$, and $c = 3/8$ (Muon's implementation retunes these coefficients for a larger slope at the origin; §3.1 gives the values). At the spectral level, each step applies the scalar polynomial $p(x) = ax + bx^3 + cx^5$ to every singular value; under this map, every nonzero singular value of X_0 increases monotonically to one, so $X_t \to \mathrm{polar}(M)$ as $t \to \infty$. The difficulty lies in the fixed budget L: since $p(x) \leq ax$ on $[0, 1]$, a singular value starting at x reaches at most $a^L x$ after L steps, so any $x \ll a^{-L}$ remains far from one within the budget.

What deflation does. For a momentum matrix M, the singular values of X_0 in (3) are the Frobenius-normalized values $\hat{\sigma}_i(M) = \sigma_i(M)/\|M\|_F$. Across the attention and MLP momentum matrices recorded during GPT-2 training (setup in §3.1), these spectra share a common structure: a few large singular values—the *head*—dominate, while the remaining ones—the *tail*—concentrate near zero (Figure 7 in Appendix C). The spectral floor of X_0 is then tiny, and the fixed-depth filter fails to resolve the tail, as discussed in §1. Deflation changes the normalization, not the polynomial: consider the exact rank-k decomposition $M = U_k\Sigma_kV_k^\top + R_k$ with $1 \leq k < q = \mathrm{rank}(M)$, and normalize the residual instead of M. Every remaining singular value is then rescaled by

$$\frac{\sigma_i(M)}{\|M\|_F} \longrightarrow \frac{\sigma_i(M)}{\|R_k\|_F} = \alpha_k \frac{\sigma_i(M)}{\|M\|_F},\ \text{where } \alpha_k := \frac{\|M\|_F}{\|R_k\|_F} > 1, \quad (4)$$

while the target is unchanged by (2): the deflated head is restored explicitly, and only the residual passes through the filter. When the head carries most of the Frobenius mass, α_k is large, and the whole tail is lifted into a region where the fixed-depth approximation is substantially more accurate. §2.4 makes this quantitative: the spectral floor of the initialization rises by exactly the factor α_k, and this factor controls both error bounds of Theorem 1.

Example 1 (A head-dominated toy spectrum)**.** *Let M have singular values $(10, 1, 0.5, 0.2, 0.1)$, so the single head σ_1 dominates the Frobenius norm, $\|M\|_F \approx 10.06$. Under the standard normalization, the smallest singular value starts at the spectral floor $\sigma_5/\|M\|_F \approx 10^{-2}$; after five steps of* (3) *it has reached only* 0.23*. Deflating σ_1 gives $\|R_1\|_F \approx 1.14$, hence $\alpha_1 \approx 8.8$: the same singular value now starts at 8.8×10^{-2} and reaches 0.988 after five steps, and the remaining tail values, lifted by the same factor, converge earlier still (Figure 8 in Appendix C).*

One computational question remains: deflation needs the leading spectral components of every input matrix at every step, and estimating them must cost far less than the filter itself. The next two subsections give one such estimator for each application.

2.2 Deflated Polar Factor and Deflated Muon

Suppose estimated leading triplets (U, s, V) of M are available, with $s_1 \geq \cdots \geq s_r$ (from Algorithm 3 below); Algorithm 1 states the resulting primitive. Deflation activates only when the gate

Algorithm 1 DEFLATED POLAR FACTOR (one matrix)

Require: $M \in \mathbb{R}^{n\times m}$, $n \geq m$; its leading triplets (U, s, V) from Algorithm 3; threshold $\tau \in (0, 1)$; padding $\gamma > 1$; polynomial map $\mathcal{N}$; iteration count $L \geq 0$

if $s_r/s_1 < \tau$ **then** ▷ deflation gate
  $k \leftarrow \max\{i : s_i > \tau s_1\}$; $U_k, V_k \leftarrow$ leading k columns of U, V ▷ the gate implies $k < r$
  $R \leftarrow M - U_k \operatorname{diag}(s_1, \ldots, s_k)\, V_k^\top$
  $X_0 \leftarrow R/\|R\|_F + \gamma^{-1} U_k V_k^\top$
else
  $X_0 \leftarrow M/\|M\|_F$
end if
return $\mathcal{N}^{\circ L}(X_0)$

$s_r < \tau s_1$ fires, which implies $k < r$: the computed window always contains the entire deflated head, and the same τ serves as gate and rank threshold.

Padding. Because the estimated triplets are inexact, the subtraction leaves small components along the head directions, which the polynomial's large slope near the origin would amplify; we therefore reinsert the estimated head into the filter initialization at magnitude γ^{-1}. The margin $\gamma > 1$ is a practical safeguard against overshoot from approximate deflation and low-precision arithmetic (with exact triplets, $\|X_0\|_2 \leq 1$ holds automatically).

Estimating the head. Deflation needs only the leading $r \ll m$ singular triplets, which randomized SVD estimates from a sketch of $r' = r + \lceil \delta m \rceil$ columns, where δ is an oversampling ratio (Halko et al., 2011). Applying it to each momentum matrix separately, however, is inefficient on a GPU for two reasons. First, momentum matrices are small relative to the device, so a per-matrix loop is dominated by kernel-launch latency rather than arithmetic. Second, all matrices of one shape can share every stage of the computation. We therefore stack same-shape matrices into a tensor $A \in \mathbb{R}^{B\times n\times m}$ and run one BATCHED RANDOMIZED SVD per group (Algorithm 3, Appendix D): a Gaussian sketch, CholeskyQR orthonormalization, J batched subspace iterations, a thin SVD of the reduced $r' \times m$ matrices $Q^\top A$, and truncation to the leading $r = \lceil wm \rceil$ triplets, where the window ratio $w \in (0, 1)$ sets what fraction of the spectrum is estimated; wide matrices are handled through $\operatorname{polar}(M^\top) = \operatorname{polar}(M)^\top$. Every large operation is a batched GEMM or a batched CholeskyQR, and an SVD touches only the reduced matrices, so the front-end is asymptotically cheaper than the polynomial iterations.

Muon with deflation. DEFLATED MUON replaces the polar-factor routine inside Muon by Algorithm 1: at every optimizer step, Algorithm 3 runs once per shape group, and each momentum matrix is then processed for $L = 5$ steps with the practical mapping in use—Muon's retuned map or Polar Express (§3.1). The primitive is agnostic to $\mathcal{N}$; the classical map (3) is the reference for the analysis in §2.4. When the gate does not fire, $X_0 = M/\|M\|_F$ and the method reduces exactly to Muon. The values of w, τ, and γ are reported in §3.

2.3 DEFLATED PSD PROJECTION AND ADMM

We now instantiate the same wrapper for PSD projection. For a symmetric matrix $Z = V\Lambda V^\top$, the PSD-cone projection is a matrix ReLU (Higham, 1988), and it can be rewritten through the matrix sign function:

$$\Pi_{\mathbb{S}^n_+}(Z) = V \max(\Lambda, 0) V^\top = \frac{1}{2} Z\big(I + \operatorname{sign}(Z)\big). \tag{5}$$

Factorization-free solvers approximate $\operatorname{sign}(Z)$ by a fixed composite odd filter $g = p_T \circ \cdots \circ p_1$ built offline (Appendix E.2; Kang et al., 2025; Lee et al., 2022), which yields the baseline projection

$$\widehat{\Pi}_+(Z) = \frac{1}{2} Z\big(I + g(Z/\theta)\big), \tag{6}$$

where θ is a Lanczos-based upper estimate of $\|Z\|_2$ that places the spectrum of Z/θ in $[-1, 1]$ (Appendix E.1). The deflated projection, stated as Algorithm 2, mirrors Algorithm 1. Its gate acts on the magnitudes $|\lambda_i|$ of the estimated eigenpairs (λ_i, u_i), because the sign filter must resolve both large positive and large negative eigenvalues; the deflated directions are padded at $\pm\gamma^{-1}$; and the

Algorithm 2 DEFLATED PSD PROJECTION (one projection)

Require: $Z \in \mathbb{S}^n$; basis $V \in \mathbb{R}^{n\times K}$, $K = \lceil w'n \rceil$, from the previous iteration; window ratios $w \le w'$; threshold τ; padding γ; composite filter $g = p_T \circ \cdots \circ p_1$

1: $\widetilde{Z} \leftarrow ZV$; eigendecompose $V^\top \widetilde{Z} = E\,\mathrm{diag}(\lambda)E^\top$; $U \leftarrow VE$ ▷ Ritz pairs (λ_i, u_i)
2: $s \leftarrow$ the values $|\lambda_i|$ sorted decreasingly; $r \leftarrow \lceil wn \rceil$
3: **if** $s_r < \tau s_1$ **then** ▷ the gate of Algorithm 1, on magnitudes
4: $\mathcal{I} \leftarrow \{i : |\lambda_i| > \tau s_1\}$
5: $R \leftarrow Z - \sum_{i\in\mathcal{I}} \lambda_i u_i u_i^\top$
6: $\hat{\theta} \leftarrow$ Lanczos-based upper estimate of $\|R\|_2$
7: $X_0 \leftarrow R/\hat{\theta} + \gamma^{-1} \sum_{i\in\mathcal{I}} \mathrm{sign}(\lambda_i) u_i u_i^\top$ ▷ padded initialization, as in Algorithm 1
8: $P \leftarrow \frac{1}{2} R\big(I + g(X_0)\big) + \sum_{i\in\mathcal{I},\,\lambda_i>0} \lambda_i u_i u_i^\top$ ▷ deflated projection
9: **else**
10: $\theta \leftarrow$ Lanczos-based upper estimate of $\|Z\|_2$; $X_0 \leftarrow Z/\theta$
11: $P \leftarrow \frac{1}{2} Z\big(I + g(X_0)\big)$ ▷ eq. (6)
12: **end if**
13: $V \leftarrow \mathrm{CholeskyQR}(\widetilde{Z})$
14: **return** P and the updated basis V

residual $R = Z - \sum_{i\in\mathcal{I}} \lambda_i u_i u_i^\top$, normalized by its own Lanczos-based upper estimate $\hat{\theta} \ge \|R\|_2$, passes through the same filter g. When reconstructing the projection, only the positive deflated eigencomponents are restored, since $(\lambda)_+ = 0$ for $\lambda < 0$. To ensure that P is symmetric despite rounding errors in the reconstruction, we symmetrize it explicitly, $P \leftarrow \frac{1}{2}(P + P^\top)$.

Warm-started subspace tracking. The estimator differs from Muon's for a structural reason. Momentum matrices are numerous and grouped by shape, so the polar routine sketches each group afresh at every optimizer step. ADMM instead produces one large, slowly varying symmetric matrix per iteration, so its dominant eigenspace can be carried across iterations. We maintain an orthonormal basis $V \in \mathbb{R}^{n\times K}$ with $K = \lceil w'n \rceil$: each projection performs one block step $\widetilde{Z} = ZV$ and one Rayleigh–Ritz step, uses the leading $r = \lceil wn \rceil \le K$ Ritz magnitudes for the gate, and orthonormalizes $\widetilde{Z}$ as the next basis, in the spirit of warm-started block eigensolvers (Knyazev, 2001). Because consecutive iterates are close, one block subspace step per iteration is sufficient in practice; V starts as a Gaussian random block, and the gate stays off during an initial warm-up phase while the basis stabilizes. Algorithm 2 summarizes one projection; the constants are reported in §3.3.

Use inside ADMM. We evaluate the deflated projection inside a first-order solver for the standard-form semidefinite program and its dual,

$$\begin{aligned} \text{Primal:}\quad & \min_X \langle C, X\rangle && \text{Dual:}\quad & \max_{y,S}\ b^\top y \\ & \text{s.t. } \mathcal{A}X = b,\ X \in \mathbb{S}^n_+, && & \text{s.t. } \mathcal{A}^* y + S = C,\ S \in \mathbb{S}^n_+, \end{aligned} \tag{7}$$

where $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ with $\mathcal{A}X = (\langle A_1, X\rangle, \ldots, \langle A_m, X\rangle)$ and $\mathcal{A}^* y = \sum_i y_i A_i$. A classical three-block ADMM (Wen et al., 2010) alternates a linear solve for y, the projection $S^{(k+1)} = \Pi_{\mathbb{S}^n_+}(Z^{(k+1)})$ with $Z^{(k+1)} := C - \mathcal{A}^* y^{(k+1)} - \sigma^{-1} X^{(k)}$, and a matrix addition for X (the full iteration is Appendix E.1, eq. (25)). For large instances the projection dominates the per-iteration cost (Kang et al., 2025; Groudiev et al., 2025); we leave the y- and X-updates untouched and replace only the projection by Algorithm 2.

2.4 ERROR ANALYSIS

To isolate the spectral mechanism of deflation, we analyze an idealized setting: the deflated components are exact, and both filters are the classical Newton–Schulz family, of which (3) is the degree-five member—the polar iteration composes it L times, and the PSD projection uses the same composition as the sign filter g. In practice the deflated components are estimated, but the estimators of §2.2 and §2.3 are accurate enough to recover the head singular triplets and eigenpairs: Appendix D measures their errors against exact decompositions on the matrices of §3 and finds them small. The degree-$(2d+1)$ classical Newton–Schulz map (Björck & Bowie, 1971) is

$$\mathcal{N}_d(X) := a_0 X + a_1 (XX^\top)X + a_2 (XX^\top)^2 X + \cdots + a_d (XX^\top)^d X, \qquad d \ge 1, \tag{8}$$

with the Björck–Bowie coefficients $a_0, \dots, a_d$ given explicitly in Appendix B. The corresponding iteration is $X_{t+1} = \mathcal{N}_d(X_t)$; the degree-five map (3) is the case $d = 2$.

Deflated polar-factor analysis. Let A denote the input, $M \in \mathbb{R}^{n\times m}$ for the polar factor or $Z \in \mathbb{S}^n$ for the projection, with the normalization scale of (1): $\theta(M) := \|M\|_F$ as in (3) and $\theta(Z) := \|Z\|_2$ as in (6). Let $\sigma_1 \geq \cdots \geq \sigma_q > 0$ be the nonzero singular values of A; for Z, $\sigma_i = |\lambda_i|$ with the eigenvalues ordered by magnitude. Let $X^{\mathrm{base}} := A/\theta(A)$ be the standard initialization and $\ell := \sigma_q/\theta(A)$ its spectral floor. For $1 \leq k < q$, let $A = H_k + R_k$ be the exact head split of (2), with $H_k = U_k\Sigma_k V_k^\top$ for M and $H_k = U_k\Lambda_k U_k^\top$ for Z, and for $\gamma \geq 1$ let

$$X_\gamma^{\mathrm{def}} := \gamma^{-1}\,\mathrm{polar}(H_k) + R_k/\theta(R_k), \qquad \alpha_k := \frac{\theta(A)}{\theta(R_k)}, \qquad \varphi_k := \min\big(\alpha_k\ell,\ \gamma^{-1}\big), \tag{9}$$

be the padded deflated initialization of Algorithms 1 and 2, the tail amplification factor of (4), and the spectral floor of X_γ^{def}; here $\mathrm{polar}(H_k)$ is $U_kV_k^\top$ for M and $U_k\,\mathrm{sign}(\Lambda_k)U_k^\top$ for Z, and $\alpha_k\ell = \sigma_q/\theta(R_k)$ is the floor of the normalized residual. For M, $\alpha_k > 1$ always; for Z, $\alpha_k = \sigma_1/\sigma_{k+1}$. The following theorem quantifies the effect of deflation.

Theorem 1 (Exact deflation raises the spectral floor)**.** *Let $M \in \mathbb{R}^{n\times m}$ be a nonzero matrix, and let X^{base}, X_γ^{def}, ℓ, α_k, and φ_k be defined as above with $A = M$. Then $\sigma(X^{\mathrm{base}}) \subset [\ell, 1]$ and $\sigma(X_\gamma^{\mathrm{def}}) \subset [\varphi_k, 1]$. Moreover, if $\gamma^{-1} > \ell$, then for every $d \geq 1$ and $L \geq 0$ the deflated-to-baseline error ratio satisfies*

$$\frac{\big\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}\big(X_\gamma^{\mathrm{def}}\big)\big\|_2}{\big\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}\big(X^{\mathrm{base}}\big)\big\|_2} \leq \left(\frac{1-\varphi_k^2}{1-\ell^2}\right)^{(d+1)^L} < 1. \tag{10}$$

The proof is deferred to Appendix B, which also bounds the deflated error itself by $(1-\varphi_k^2)^{(d+1)^L}$, eq. (12). In words, deflation multiplies the spectral floor by α_k (capped by the padding level γ^{-1}), and the error ratio decays doubly exponentially in L; with $\gamma = 1$, $\varphi_k = \alpha_k\ell$ and the unpadded case is recovered.

Deflated PSD-projection analysis. We now take $A = Z$, so that $\theta(Z) = \|Z\|_2 = \sigma_1$, $\theta(R_k) = \|R_k\|_2 = \sigma_{k+1}$, and $\alpha_k = \sigma_1/\sigma_{k+1}$; we assume $\sigma_{k+1} < \sigma_1$, so that $\alpha_k > 1$. For a symmetric X, the map (8) applies the odd polynomial $p_d(x) = \sum_{i=0}^d a_i x^{2i+1}$ to every eigenvalue, and $p_d^{\circ L}(x) \to 1$ on $(0, 1]$, so $g = \mathcal{N}_d^{\circ L}$ is a sign filter for (6) and Algorithm 2. Write $\widehat{\Pi}^{\mathrm{base}}(Z) := \frac{1}{2}Z\big(I + \mathcal{N}_d^{\circ L}(X^{\mathrm{base}})\big)$ for (6) with this filter and $\theta = \theta(Z)$, and $\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) := U_k(\Lambda_k)_+U_k^\top + \frac{1}{2}R_k\big(I + \mathcal{N}_d^{\circ L}(X_\gamma^{\mathrm{def}})\big)$ for the output of Algorithm 2 with this filter and exact inputs.

Proposition 2 (Deflated PSD projection)**.** *Let $Z \in \mathbb{S}^n$ be a nonzero matrix with $\sigma_{k+1} < \sigma_1$, and let X^{base}, X_γ^{def}, ℓ, α_k, $\widehat{\Pi}^{\mathrm{base}}$, and $\widehat{\Pi}_\gamma^{\mathrm{def}}$ be defined as above with $A = Z$. Then $\sigma(X^{\mathrm{base}}) \subset [\ell, 1]$ and $\sigma(X_\gamma^{\mathrm{def}}) \subset \{\gamma^{-1}\} \cup [\alpha_k\ell, 1]$, $\widehat{\Pi}_\gamma^{\mathrm{def}}(Z)$ does not depend on γ, and for every $d \geq 1$ and $L \geq 0$ the deflated-to-baseline error ratio satisfies*

$$\frac{\big\|\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2}{\big\|\widehat{\Pi}^{\mathrm{base}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2} \leq \left(\frac{1-\alpha_k^2\ell^2}{1-\ell^2}\right)^{(d+1)^L} < 1. \tag{11}$$

The proof (Appendix E.3) follows the three steps of the proof of Theorem 1: the error is diagonal in the eigenbasis of Z, with entry $-\frac{\sigma_i}{2}\big(1 - p_d^{\circ L}(\sigma_i/\theta)\big)$ in the direction of every eigenvalue that passes through the filter (θ the scale in use), and Lemma 4 bounds this residual at the two floors. In words, the projection obeys the ratio bound of Theorem 1 at $\gamma = 1$: deflation replaces the condition number of Z by that of the residual. Two differences from Theorem 1 are worth noting. First, the padded head is annihilated by R_k and never reaches the output, so the bounds involve the residual floor $\alpha_k\ell$ rather than φ_k, and no side condition is needed beyond $\alpha_k > 1$, which the gate guarantees under exact spectral estimates. Second, the error in each eigendirection is weighted by σ_i: small eigenvalues that the filter cannot resolve are cheap for the projection, whereas they dominate the error of the polar factor. Appendix E.3 also bounds the deflated error itself, eq. (28), and gives complementary bounds for the implemented fixed filter (Proposition 5) and for an idealized minimax filter regenerated on the deflated interval (Theorem 6).

3 NUMERICAL EXPERIMENTS

General setup. All language-model experiments pretrain GPT-2 Large (Radford et al., 2019) ($n_{\text{embd}} = 1280$, $n_{\text{layer}} = 36$, $n_{\text{head}} = 20$, vocabulary size 50,257, context length 1024) on subsets of FineWeb (Penedo et al., 2024), in `bfloat16` mixed precision on four NVIDIA H200 GPUs; the SDP experiments run on a single H200 with CUDA 12.9. Muon is applied to all eligible parameters, excluding token embeddings, positional embeddings, and the output embedding layer; the remaining parameters are optimized with AdamW (Loshchilov & Hutter, 2019). The learning rate is held constant for the first 40% of training and then decays linearly to zero, following Jordan et al. (2024); the ten-billion-token runs shorten the constant phase to the first 5% for stability. §2 analyzes the classical Newton–Schulz coefficients; the experiments use the two practical mappings: Muon's retuned degree-five coefficients $a = 3.4445$, $b = -4.7750$, $c = 2.0315$ with $L = 5$ (Jordan et al., 2024; Cesista et al., 2025), whose larger slope at the origin amplifies small singular values faster, and Polar Express, which selects iteration-dependent degree-five polynomials by a minimax criterion under the same multiplication budget (Amsel et al., 2025). We refer to them as Newton–Schulz and Polar Express below; deflation changes neither mapping's coefficients. Unless stated otherwise, deflation uses the window ratio $w = 0.025$, threshold $\tau = 0.1$, and padding $\gamma = 1.01$ ($\gamma = 1.1$ for the Polar Express mapping).

3.1 MATRIX-FUNCTION ACCURACY

We train GPT-2 Large with Muon on 1B tokens from FineWeb and record the momentum matrices of the four attention projections Q, K, V, O and the two MLP projections in the middle transformer block (layer $\lfloor N/2 \rfloor$ of N layers), at training steps 1, 500, 1000, and 1500. For each momentum matrix, we compare four methods with the entries the optimizer uses: Muon's Newton–Schulz iteration, Polar Express, and their deflated counterparts, whose head is estimated by the batched randomized SVD of Algorithm 3 with the gate and constants of the training runs, not read off an exact SVD. Using $UV^\top$ from a double-precision SVD as the reference polar factor, we measure the relative Frobenius error $\|X_L - UV^\top\|_F / \|UV^\top\|_F$ after every iteration; the polynomial iterations are evaluated in single precision, so that the effects of estimation and of the mapping are not confounded with the `bfloat16` rounding that all four arms share (Appendix F quantifies the latter). Figure 2 shows that when the momentum matrix has a few dominant singular values, the deflated variants attain the lower error at every iteration. Appendix F reports all six recorded matrices (Figures 10 and 11).

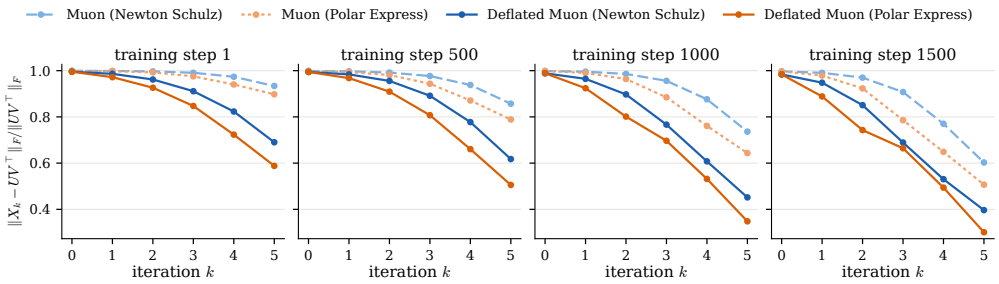
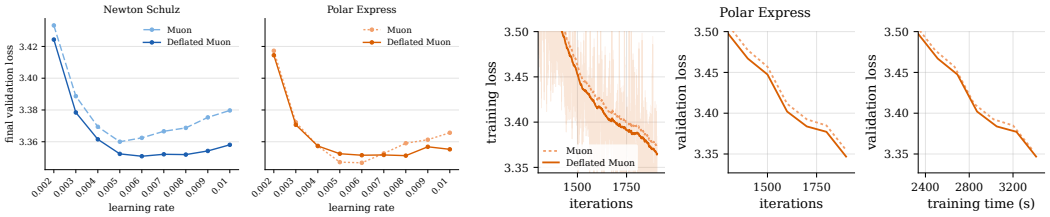


Figure 2: Matrix-level accuracy on the attention V momentum matrix of GPT-2 Large, recorded at training steps 1, 500, 1000, and 1500, as a function of the number of iterations. Each panel reports the relative error $\|X_L - UV^\top\|_F / \|UV^\top\|_F$.

3.2 LANGUAGE-MODEL PRETRAINING

We integrate DEFLATED MUON into full pretraining: one epoch over a one-billion-token subset of FineWeb. Figures 1 and 3 report the results. In both cases, DEFLATED MUON achieves better validation loss than Muon, and the improvement is sustained over the whole run. Although the batched randomized SVD and deflation add per-step computation, the advantage persists against wall-clock time. DEFLATED MUON attains lower final validation loss than Muon at most learning rates for both mappings, and at every learning rate for Newton–Schulz.

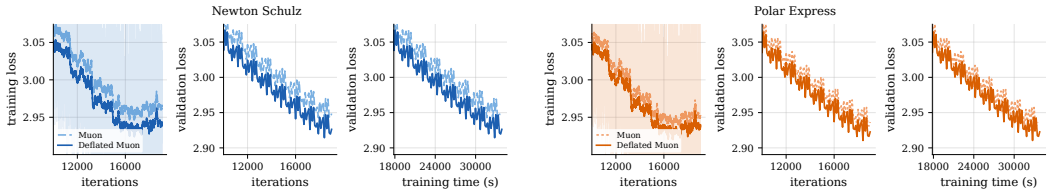


(a) Final validation loss across learning rates

(b) Polar Express mapping, learning rate 0.01

Figure 3: GPT-2 Large trained on one billion tokens from FineWeb. (a) Final validation loss across learning rates, for both mappings. (b) Polar Express loss curves at learning rate 0.01; layout as in Figure 1a.

Longer training horizon. To test whether the benefit persists over a longer horizon, we train GPT-2 Large on ten billion tokens, roughly the Chinchilla-optimal budget for this model size (Hoffmann et al., 2022). As shown in Figure 4, DEFLATED MUON continues to outperform the original Muon, reaching a lower final validation loss.



(a) Newton–Schulz mapping.

(b) Polar Express mapping.

Figure 4: GPT-2 Large trained on ten billion tokens from FineWeb, for the Newton–Schulz (a) and Polar Express (b) mappings; layout as in Figure 1a.

3.3 SEMIDEFINITE PROGRAMMING

We next evaluate the deflated PSD projection inside ADMM on four large instances from Mittelmann's sparse SDP collection (Mittelmann, 2006): the max-cut relaxations (Goemans & Williamson, 1995) `G55mc` and `G59mc` ($n = 5{,}000$) and `G60mc` ($n = 7{,}000$), and the min-bisection relaxation `G60_mb` ($n = 7{,}000$). We run the ADMM iteration (25) with $\sigma = 1$ for a fixed budget of 10^4 iterations, with every PSD projection performed in FP16, and compare the factorization-free projection (6) against its deflated version (Algorithm 2), everything else being identical. The deflated projection uses $\tau = 0.1$, $w = 0.025$, $w' = 0.05$, and $\gamma = 1.1$, with the gate off for the first 100 iterations while the tracked basis stabilizes. We monitor the eigenvalue-free part of the maximum KKT residual η (definition and end-of-run PSD-feasibility checks in Appendix E.4).

Figure 5 reports η against cumulative projection time: across the four instances, ADMM with the deflated projection converges to a lower KKT residual within a similar projection-time budget.

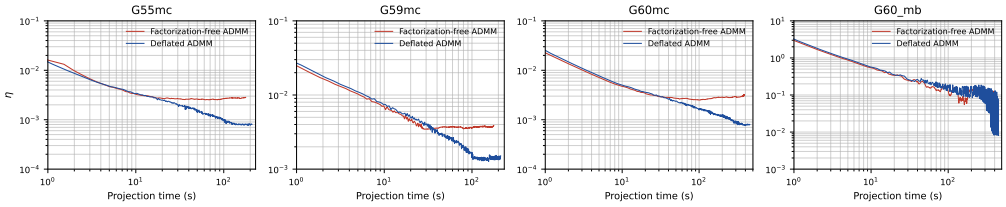


Figure 5: KKT residual η (the three eigenvalue-free terms of (36)) against cumulative projection time on four instances from Mittelmann's collection, with the setup of §3.3. We compare the factorization-free ADMM (red) with ADMM using the deflated projection (blue).

3.4 Ablations

Number of polynomial iterations. We vary the number of iterations over $L \in \{0, 1, 2, 3, 4, 5\}$ (Table 1): Deflated Muon is less sensitive to L and maintains low validation loss even with fewer iterations; notably, with Polar Express, $L = 3$ deflated iterations (3.3493) already outperform the $L = 5$ baseline (3.3671).

		Number of iterations L					
Mapping	Method	0	1	2	3	4	5
Newton–Schulz	Muon	4.1879	3.8746	3.6646	3.5201	3.4084	3.3794
	Deflated Muon	**4.0898**	**3.7895**	**3.5870**	**3.4400**	**3.3681**	**3.3568**
Polar Express	Muon	4.1916	3.7698	3.5122	3.4244	3.3770	3.3671
	Deflated Muon	**4.0805**	**3.6717**	**3.4814**	**3.3493**	**3.3545**	**3.3604**

Table 1: Final validation loss of GPT-2 Large as a function of the number of polynomial iterations L. For Polar Express, with deflation, $L = 3$ is already sufficient to reach the best validation loss.

Momentum coefficient. We evaluate the momentum coefficient $\beta \in \{0.90, 0.95, 0.98\}$ on the one-billion-token setup of §3.2. As shown in Table 2, the improvement from deflation is robust across all three values, and Figure 6 shows the results for $\beta = 0.90$ with the Newton–Schulz mapping.

	Momentum coefficient β		
Method	0.90	0.95	0.98
Newton–Schulz			
Muon	3.4205	3.3803	3.3875
Deflated Muon	**3.3845**	**3.3599**	**3.3675**
Polar Express			
Muon	3.3967	3.3672	3.3786
Deflated Muon	**3.3776**	**3.3576**	**3.3785**

Table 2: Final validation loss of GPT-2 Large for momentum coefficients β; Deflated Muon outperforms Muon at every β.

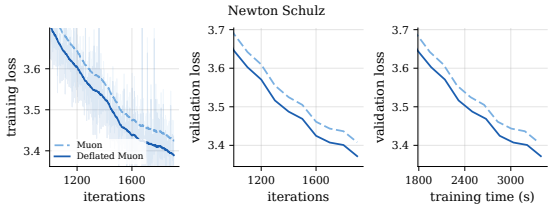


Figure 6: GPT-2 Large with momentum coefficient $\beta = 0.90$, Newton–Schulz mapping; layout as in Figure 1a.

Deflation ratio. Deflated Muon has two constants of its own: the window ratio w, which determines how deep into the spectrum the gate looks, and the threshold τ, which is shared by the gate and the rank cutoff (§2.2). Starting from the operating point $(w, \tau) = (0.025, 0.1)$ used in the main experiments, we test $w \in \{0.0025, 0.005, 0.01, 0.025\}$ with $\tau = 0.1$, and $\tau \in \{0.05, 0.1, 0.2, 0.3\}$ with $w = 0.025$. Tables 5 and 6 (Appendix F) show that these settings affect the size of the deflated head: the number of deflated singular pairs decreases as we narrow the window or raise the threshold, while the improvement in validation loss remains robust across different settings.

Deflation cutoff. Momentum spectra generally become less head-dominated as training progresses (Magakyan et al., 2026), so deflation may not be needed throughout training; we therefore vary the cutoff over $\{10\%, 30\%, 50\%, 100\%\}$ of the optimizer steps (the main experiments use 100%). Table 4 (Appendix F) shows that Deflated Muon improves on the baseline at every tested cutoff; a cutoff of 50% or even 30% already substantially improves performance.

4 Conclusion

We introduced spectral deflation, a general framework that preserves the target exactly while raising the spectral floor of the input and therefore improves the accuracy of polynomial approximations. Integrated into Muon, Deflated Muon consistently improves GPT-2 pretraining over the corresponding Muon baselines; integrated into factorization-free ADMM for SDP, the deflated PSD projection substantially reduces the KKT residual the solver converges to within a similar projection-time budget. The framework is orthogonal to the design of the polynomial, and natural next instances are inverse matrix roots in preconditioned optimizers (Gupta et al., 2018) and other spectral functions computed by fixed-depth filters.

AI USE STATEMENT

In this work, we used generative AI tools for language polishing and code assistance. We did not use generative AI for research ideation, experimental design, or experimental data generation. All LLM-generated code was verified and tested for correctness by the authors. We take full responsibility for the final content of this work, including all text, claims, code, and other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The two primitives are fully specified in §2 (Algorithms 1–2), with implementation details in Appendices D and E. Details of the experiments, including the model architecture, training data, hyperparameters, and evaluation metrics, are provided in §3 and Appendix E.4. Complete proofs of the theoretical results are provided in Appendices B and E.3. Source code to reproduce all experiments is included in the supplementary materials and will be open-sourced upon acceptance.

REFERENCES

Noah Amsel, David Persson, Christopher Musco, and Robert M. Gower. The polar express: Optimal matrix sign methods and their application to the Muon algorithm. *arXiv preprint arXiv:2505.16932*, 2025.

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

Åke Björck and Clazett Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.

Jacques C. R. Bloch, Tobias Breu, Andreas Frommer, Simon Heybrock, Katrin Schäfer, and Tilo Wettig. Krylov subspace methods and the sign function: Multishifts and deflation in the non-hermitian case. *Proceedings of Science*, LAT2009:043, 2009.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Franz Louis Cesista, Jiacheng You, and Keller Jordan. Squeezing 1-2% efficiency gains out of Muon by optimizing the Newton–Schulz coefficients. `https://leloykun.github.io/ponder/muon-opt-coeffs/`, 2025.

Da Chang, Qiankun Shi, Lvgang Zhang, Yu Li, Ruijie Zhang, Yao Lu, Yongxiang Liu, and Ganzhao Yuan. MuonEq: Balancing before orthogonalization with lightweight equilibration. *arXiv preprint arXiv:2603.28254*, 2026.

Lizhang Chen, Jonathan Li, and Qiang Liu. Muon optimizes under spectral norm constraints. *arXiv preprint arXiv:2506.15054*, 2025.

Sayantan Choudhury, Xiaoran Cheng, Martin Takáč, Sen Na, and Mladen Kolar. Muon with Nesterov momentum: Heavy-tailed noise and (randomized) inexact polar decomposition. *arXiv preprint arXiv:2605.06884*, 2026.

Bishnu Dev, Sushil Bohara, Martin Takáč, and Samuel Horváth. CacheMuon: Using temporal preconditioning to approximate polar factor. *arXiv preprint arXiv:2606.16371*, 2026.

Michael Eiermann, Oliver G. Ernst, and Stefan Güttel. Deflated restarting for matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 32(2):621–641, 2011.

Li Fan, David I. Shuman, Shashanka Ubaru, and Yousef Saad. Spectrum-adapted polynomial approximation for matrix functions. *arXiv preprint arXiv:1808.09506*, 2018.

Juliano B. Francisco and Douglas S. Gonçalves. A fixed-point method for approximate projection onto the positive semidefinite cone. *Linear Algebra and its Applications*, 523:59–78, 2017.

Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

Ekaterina Grishina, Matvey Smirnov, and Maxim Rakhuba. Accelerating Newton–Schulz iteration for orthogonalization via chebyshev-type polynomials. *arXiv preprint arXiv:2506.10935*, 2025.

Antoine Groudiev, Shucheng Kang, and Heng Yang. cuADMM: GPU-accelerated first-order optimization for large-scale multi-block semidefinite programs. In *RSS Workshop on Fast Motion Planning and Control in the Era of Parallelism*, 2025.

Ming Gu. Subspace iteration randomization and singular value problems. *SIAM Journal on Scientific Computing*, 37(3):A1139–A1173, 2015.

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.

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.

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

Nicholas J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103–118, 1988.

Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2008.

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

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

Shucheng Kang, Haoyu Han, Antoine Groudiev, and Heng Yang. Factorization-free orthogonal projection onto the positive semidefinite cone with composite polynomial filtering. *arXiv preprint arXiv:2507.09165*, 2025.

Andrew V. Knyazev. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing*, 23(2):517–541, 2001.

Eunsang Lee, Joon Woo Lee, Jong Seon No, and Young Sik Kim. Minimax approximation of sign function by composite polynomial for homomorphic comparison. *IEEE Transactions on Dependable and Secure Computing*, 19(6):3711–3727, 2022.

Ren-Cang Li. New perturbation bounds for the unitary polar factor. *SIAM Journal on Matrix Analysis and Applications*, 16(1):327–332, 1995.

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

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

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

Gagik Magakyan, Pablo Parrilo, and Asuman Ozdaglar. Spectral scaling laws of Muon. *arXiv preprint arXiv:2606.04058*, 2026.

Hans D. Mittelmann. Several SDP-codes on sparse and other SDP problems. `https://plato.asu.edu/ftp/sparse_sdp.html`, 2006.

Yuji Nakatsukasa, Zhaojun Bai, and François Gygi. Optimizing halley's iteration for computing the matrix polar decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2700–2720, 2010.

Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

Beresford N. Parlett. *The Symmetric Eigenvalue Problem*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, 2024.

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

Nikitas Rontsis, Paul Goulart, and Yuji Nakatsukasa. Efficient semidefinite programming with approximate ADMM. *Journal of Optimization Theory and Applications*, 192(1):292–320, 2022.

Yousef Saad, Man Yeung, Jocelyne Erhel, and Frédéric Guyomarc'h. A deflated version of the conjugate gradient algorithm. *SIAM Journal on Scientific Computing*, 21(5):1909–1926, 2000.

Arvind K. Saibaba. Randomized subspace iteration: Analysis of canonical angles and unitarily invariant norms. *SIAM Journal on Matrix Analysis and Applications*, 40(1):23–48, 2019.

Chongjie Si, Debing Zhang, and Wei Shen. AdaMuon: Adaptive Muon optimizer. *arXiv preprint arXiv:2507.11005*, 2025.

Mario Souto, Joaquim D. Garcia, and Álvaro Veiga. Exploiting low-rank structure in semidefinite programming by approximate operator splitting. *Optimization*, 71(1):117–144, 2022.

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.

Zaiwen Wen, Donald Goldfarb, and Wotao Yin. Alternating direction augmented lagrangian methods for semidefinite programming. *Mathematical Programming Computation*, 2(3–4):203–230, 2010.

Chenrui Xu, Wenjing Yan, and Ying-Jun Angela Zhang. FISMO: Fisher-structured momentum-orthogonalized optimizer. *arXiv preprint arXiv:2601.21750*, 2026.

Ruijie Zhang, Yequan Zhao, Ziyue Liu, Zhengyang Wang, Yupeng Su, Liyan Tan, and Zheng Zhang. Muon+: Towards more effective Muon via one additional normalization step for LLM pre-training. *arXiv preprint arXiv:2602.21545*, 2026a.

Yechen Zhang, Shuhao Xing, Junhao Huang, Kai Lv, Yunhua Zhou, Xipeng Qiu, Qipeng Guo, and Kai Chen. Mousse: Rectifying the geometry of Muon with curvature-aware preconditioning. *arXiv preprint arXiv:2603.09697*, 2026b.

Yang Zheng, Giovanni Fantuzzi, Antonis Papachristodoulou, Paul Goulart, and Andrew Wynn. Fast ADMM for homogeneous self-dual embedding of sparse SDPs. *IFAC-PapersOnLine*, 50 (1):8411–8416, 2017.

A Related Work

Muon and orthogonalized-update optimizers. Muon constructs updates for matrix-valued parameters by approximating the polar factors of their momentum matrices with a small, fixed number of Newton–Schulz iterations (Jordan et al., 2024; Liu et al., 2025). Bernstein & Newhouse (2024) interpret the update as steepest descent under spectral-norm geometry, and Chen et al. (2025) analyze Muon through implicit spectral-norm constraints. Subsequent work modifies either the output of orthogonalization or the coordinates in which it is performed. Among post-orthogonalization methods, Muon+ adds row- or column-wise normalization to correct norm imbalance in the polar update (Zhang et al., 2026a), and NorMuon combines orthogonalization with neuron-wise second-moment normalization (Li et al., 2025); AdaMuon is a hybrid, applying a sign-stabilized momentum transformation before orthogonalization and element-wise adaptive scaling after it (Si et al., 2025). More broadly, geometry-aware preconditioners motivate working in transformed coordinates: SOAP performs Adam-style adaptation in an eigenbasis obtained from Shampoo preconditioners (Gupta et al., 2018; Vyas et al., 2024). Pre-orthogonalization corrections modify the matrix seen by the orthogonalization polynomial: FISMO and Mousse compute spectral updates in curvature-whitened coordinates (Xu et al., 2026; Zhang et al., 2026b), and MuonEq applies lightweight row and column equilibration before Newton–Schulz (Chang et al., 2026). These methods deliberately change the update's scaling or geometry, and hence its exact target. In contrast, exact spectral deflation preserves the original polar target and aims specifically at the finite-step approximation error.

Polar-factor computation. The polar factor can be computed accurately through an SVD, but a full factorization for every matrix at every optimizer step is expensive (Higham, 1986). Classical iterative alternatives include Newton, inverse Newton, Newton–Schulz, and Halley-type methods for the polar decomposition or the matrix sign function (Higham, 2008; Nakatsukasa et al., 2010; Björck & Bowie, 1971). Prioritizing speed over asymptotic accuracy, Muon uses a retuned degree-five Newton–Schulz map for only a few iterations; the retuned coefficients accelerate tail singular values although one is no longer a fixed point of the scalar map (Jordan et al., 2024; Cesista et al., 2025). Recent work improves this trade-off by redesigning the polynomials: CANS derives Chebyshev-type coefficients for controlled approximate orthogonalization (Grishina et al., 2025), and Polar Express selects iteration-dependent minimax polynomials under a fixed multiplication budget (Amsel et al., 2025). A complementary line reduces the cost of the polar computation itself: Choudhury et al. (2026) analyze Muon with a randomized low-rank polar approximation, and CacheMuon reuses temporal information across optimizer steps (Dev et al., 2026). Deflation is orthogonal to both lines: it changes the normalized spectrum presented to an existing filter, keeps the full-rank target through the exact identity (2), and we evaluate it with both Newton–Schulz and Polar Express; the temporal reuse of CacheMuon parallels our warm-started tracking on the ADMM side.

ADMM and semidefinite programming. ADMM is a widely used operator-splitting framework whose low per-iteration cost and decomposable structure make it suitable for large-scale constrained optimization (Boyd et al., 2011). For semidefinite programming, Wen et al. (2010) developed an alternating-direction augmented-Lagrangian method in which each iteration consists of linear-system operations and a projection onto the PSD cone; related first-order solvers include the homogeneous self-dual splitting of SCS (O'Donoghue et al., 2016) and ADMM for sparse SDPs (Zheng et al., 2017), with recent GPU implementations for large multi-block problems (Groudiev et al., 2025). Because the PSD projection (Higham, 1988) is conventionally computed by a full eigendecomposition, it often dominates the cost for large dense instances. Approximate ADMM reduces this cost with partial eigendecompositions (Rontsis et al., 2022), low-rank operator-splitting methods exploit low-rank solution structure (Souto et al., 2022), and fixed-point methods approximate the projection through the matrix sign function (Francisco & Gonçalves, 2017). Most recently, Kang et al. (2025) designed a factorization-free composite polynomial filter that approximates the matrix ReLU using dense GEMMs and low-precision arithmetic; we retain that filter and improve the normalized spectrum it receives.

Spectral deflation and spectrum-aware filtering. Deflation is a classical acceleration technique for linear systems, eigenvalue problems, and matrix-function approximation: selected invariant components are handled explicitly, and an iterative approximation is applied only to their complement. Deflated conjugate gradient removes the outlying eigenvalues that slow convergence (Saad et al.,

2000); deflated restarting accelerates Krylov approximations of matrix functions (Eiermann et al., 2011); and deflated rational Krylov methods for the matrix sign function explicitly treat eigenvalues near the function's discontinuity (Bloch et al., 2009). Other approaches adapt the approximation to the spectrum instead: spectrum-adapted polynomial methods estimate the spectral density and allocate accuracy to densely populated regions (Fan et al., 2018). We adapt the deflation principle to short, dense-GEMM polynomial filters, where the dominant components are removed not because they impede a Krylov expansion but because they inflate the normalization scale of the fixed-depth filter.

B Proof of Theorem 1

We restate the theorem for self-containment, adding the bound on the deflated error itself that the proof yields alongside the ratio bound.

Theorem 1 (Exact deflation raises the spectral floor; restated with the deflated-error bound). *Let M, X^{base}, X_γ^{def}, ℓ, α_k, and $\varphi_k = \min(\alpha_k \ell, \gamma^{-1})$ be as defined in §2.4, with $1 \le k < q$ and $\gamma \ge 1$. Then $\sigma(X^{\mathrm{base}}) \subset [\ell, 1]$ and $\sigma(X_\gamma^{\mathrm{def}}) \subset [\varphi_k, 1]$. Moreover, for every $d \ge 1$ and $L \ge 0$,*

$$\left\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X_\gamma^{\mathrm{def}})\right\|_2 \le \left(1 - \varphi_k^2\right)^{(d+1)^L}. \tag{12}$$

If in addition $\gamma^{-1} > \ell$, the deflated-to-baseline error ratio satisfies

$$\frac{\left\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X_\gamma^{\mathrm{def}})\right\|_2}{\left\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X^{\mathrm{base}})\right\|_2} \le \left(\frac{1 - \varphi_k^2}{1 - \ell^2}\right)^{(d+1)^L} < 1.$$

We prove the theorem in three steps. First, two scalar lemmas describe how one Newton–Schulz step, and then L steps, act on a squared singular value. Second, we reduce the matrix error to a scalar quantity at the spectral floor. Third, we apply the lemmas at the floors ℓ and φ_k. The first lemma gives an exact formula for the one-step squared-singular-value residual, and the second iterates it. Throughout, the Björck–Bowie coefficients of (8) are $a_i := (-1)^i \sum_{j=i}^d \frac{1}{4^j}\binom{2j}{j}\binom{j}{i}$ for $i = 0, \dots, d$.

Lemma 3 (One-step residual identity)**.** *For $d \ge 1$, let $c_j := 4^{-j}\binom{2j}{j}$ and $h_d(z) := \sum_{j=0}^d c_j z^j$, and let $p_d(x) := \sum_{i=0}^d a_i x^{2i+1} = x\, h_d(1 - x^2)$ be the scalar polynomial induced by $\mathcal{N}_d$. Then p_d is nondecreasing and maps $[0, 1]$ into itself. Moreover,*

$$1 - p_d(x)^2 = (1 - x^2)^{d+1} \omega_d(1 - x^2), \tag{13}$$

where

$$\omega_d(z) := (2d + 1) c_d \sum_{j=0}^d \frac{c_j}{d + j + 1} z^j. \tag{14}$$

The function ω_d is positive and nondecreasing on $[0, 1]$ and satisfies $0 < \omega_d(z) \le 1$ for every $z \in [0, 1]$. Consequently,

$$0 \le 1 - p_d(x)^2 \le (1 - x^2)^{d+1} \qquad \text{for every } x \in [0, 1]. \tag{15}$$

Proof. We first differentiate p_d. The coefficients c_j satisfy $c_{j+1}/c_j = \frac{1}{4}\binom{2j+2}{j+1}/\binom{2j}{j} = \frac{2j+1}{2j+2}$, and hence

$$2(j + 1)c_{j+1} = (2j + 1)c_j. \tag{16}$$

Set $z = 1 - x^2$. By the chain rule, $p_d'(x) = h_d(z) - 2(1 - z)h_d'(z)$. Expanding the right-hand side gives

$$\begin{aligned} p_d'(x) &= \sum_{j=0}^d c_j z^j - 2(1 - z) \sum_{j=1}^d j c_j z^{j-1} \\ &= (c_0 - 2c_1) + \sum_{j=1}^{d-1} \big((2j + 1)c_j - 2(j + 1)c_{j+1}\big) z^j + (2d + 1)c_d z^d. \end{aligned}$$

Because $c_0 = 1$, $c_1 = 1/2$, and (16) holds, every term except the last one vanishes. Thus $p_d'(x) = (2d+1)c_d(1-x^2)^d$. In particular, $p_d'(x) \geq 0$ on $[0,1]$. Since $p_d(0) = 0$ and $p_d(1) = 1$, it follows that $p_d([0,1]) \subset [0,1]$.

It remains to prove the residual identity. For $z \in [0,1]$, direct differentiation and the formula for p_d' give

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}z}\big[1-(1-z)h_d(z)^2\big] &= h_d(z)^2 - 2(1-z)h_d(z)h_d'(z)\\ &= h_d(z)\big[h_d(z) - 2(1-z)h_d'(z)\big]\\ &= (2d+1)c_d z^d h_d(z).\end{aligned}$$

Because $h_d(0) = c_0 = 1$, the expression inside the derivative is zero at $z = 0$. Integrating from 0 to z therefore yields

$$\begin{aligned}1-(1-z)h_d(z)^2 &= (2d+1)c_d\int_0^z u^d h_d(u)\,\mathrm{d}u\\ &= (2d+1)c_d\sum_{j=0}^{d} c_j\int_0^z u^{d+j}\,\mathrm{d}u\\ &= (2d+1)c_d\sum_{j=0}^{d}\frac{c_j}{d+j+1}z^{d+j+1}\\ &= z^{d+1}\omega_d(z).\end{aligned}$$

Substituting $z = 1-x^2$ and using $p_d(x)^2 = x^2 h_d(1-x^2)^2$ proves (13).

Every coefficient of ω_d is positive. Hence $\omega_d(z) > 0$ and ω_d is nondecreasing on $[0,1]$. Taking $x = 0$ in (13) gives $\omega_d(1) = 1$. It follows that $0 < \omega_d(z) \leq 1$ for every $z \in [0,1]$. Combining this inequality with (13) proves (15). □

Lemma 4 (Iterated squared-residual bounds)**.** *For an integer $t \geq 0$, let $p_d^{\circ t}$ denote the t-fold composition of p_d, with $p_d^{\circ 0}$ the identity. Then, for every $x \in [0,1]$ and every $L \geq 0$,*

$$1 - p_d^{\circ L}(x) \leq 1 - \big(p_d^{\circ L}(x)\big)^2 \leq (1-x^2)^{(d+1)^L}. \tag{17}$$

Moreover, if $0 < x \leq y \leq 1$ and $x < 1$, then

$$\frac{1-p_d^{\circ L}(y)}{1-p_d^{\circ L}(x)} \leq \left(\frac{1-y^2}{1-x^2}\right)^{(d+1)^L}. \tag{18}$$

Proof. Lemma 3 shows that p_d maps $[0,1]$ into itself, so $p_d^{\circ t}(x) \in [0,1]$ for every $t \geq 0$. Applying (13) to $p_d^{\circ t}(x)$ and writing $p_d^{\circ(t+1)}(x) = p_d\big(p_d^{\circ t}(x)\big)$ gives

$$1 - \big(p_d^{\circ(t+1)}(x)\big)^2 = \Big(1 - \big(p_d^{\circ t}(x)\big)^2\Big)^{d+1}\omega_d\Big(1 - \big(p_d^{\circ t}(x)\big)^2\Big). \tag{19}$$

Because $0 < \omega_d(z) \leq 1$ on $[0,1]$,

$$1 - \big(p_d^{\circ(t+1)}(x)\big)^2 \leq \Big(1 - \big(p_d^{\circ t}(x)\big)^2\Big)^{d+1}. \tag{20}$$

At $t = 0$, we have $1 - \big(p_d^{\circ 0}(x)\big)^2 = 1 - x^2$. Induction using (20) therefore yields

$$1 - \big(p_d^{\circ L}(x)\big)^2 \leq (1-x^2)^{(d+1)^L} \qquad \text{for every } L \geq 0. \tag{21}$$

Furthermore, since $p_d^{\circ L}(x) \in [0,1]$, $1 - p_d^{\circ L}(x) = \big(1-(p_d^{\circ L}(x))^2\big)/\big(1+p_d^{\circ L}(x)\big) \leq 1 - \big(p_d^{\circ L}(x)\big)^2$. Together with (21), this proves (17).

We next prove the ratio estimate. By Lemma 3, p_d is nondecreasing on $[0,1]$, and hence so is $p_d^{\circ t}$. Since $x \leq y$ and both iterates lie in $[0,1]$,

$$p_d^{\circ t}(x) \leq p_d^{\circ t}(y) \qquad \text{and} \qquad 0 \leq 1 - \big(p_d^{\circ t}(y)\big)^2 \leq 1 - \big(p_d^{\circ t}(x)\big)^2 \qquad \text{for every } t \geq 0. \tag{22}$$

Because $x < 1$, we have $1 - x^2 > 0$, and since ω_d is positive, (19) shows inductively that $1 - \big(p_d^{\circ t}(x)\big)^2 > 0$ for every finite t. Hence the following ratios are well-defined. Applying (19) to x and to y, and using the monotonicity of ω_d together with (22), we obtain

$$\frac{1-\big(p_d^{\circ(t+1)}(y)\big)^2}{1-\big(p_d^{\circ(t+1)}(x)\big)^2} = \left(\frac{1-\big(p_d^{\circ t}(y)\big)^2}{1-\big(p_d^{\circ t}(x)\big)^2}\right)^{d+1} \frac{\omega_d\Big(1-\big(p_d^{\circ t}(y)\big)^2\Big)}{\omega_d\Big(1-\big(p_d^{\circ t}(x)\big)^2\Big)} \leq \left(\frac{1-\big(p_d^{\circ t}(y)\big)^2}{1-\big(p_d^{\circ t}(x)\big)^2}\right)^{d+1}.$$

Induction on t now gives

$$\frac{1-\big(p_d^{\circ L}(y)\big)^2}{1-\big(p_d^{\circ L}(x)\big)^2} \leq \left(\frac{1-y^2}{1-x^2}\right)^{(d+1)^L}. \tag{23}$$

Finally, (22) gives $p_d^{\circ L}(x) \leq p_d^{\circ L}(y)$. Rationalizing the two differences therefore gives

$$\frac{1-p_d^{\circ L}(y)}{1-p_d^{\circ L}(x)} = \frac{1-\big(p_d^{\circ L}(y)\big)^2}{1-\big(p_d^{\circ L}(x)\big)^2} \cdot \frac{1+p_d^{\circ L}(x)}{1+p_d^{\circ L}(y)} \leq \frac{1-\big(p_d^{\circ L}(y)\big)^2}{1-\big(p_d^{\circ L}(x)\big)^2}.$$

Combining this inequality with (23) proves (18). □

Proof of Theorem 1. Let $M = U\,\mathrm{diag}(\sigma_1, \dots, \sigma_q)V^\top$ with $\sigma_1 \geq \cdots \geq \sigma_q > 0$ be a compact singular value decomposition of M. Then $\mathrm{polar}(M) = UV^\top$. The nonzero singular values of $X^{\mathrm{base}} = M/\|M\|_F$ are $\sigma_i/\|M\|_F$, $i = 1, \dots, q$, and their smallest value is $\sigma_q/\|M\|_F = \ell$. Therefore, $\sigma(X^{\mathrm{base}}) \subset [\ell, 1]$.

By the definition of the deflated initialization $X_\gamma^{\mathrm{def}} = \gamma^{-1}U_kV_k^\top + R_k/\|R_k\|_F$, its two terms have orthogonal row and column spaces, so its nonzero singular values are γ^{-1} (with multiplicity k) and $\sigma_i/\|R_k\|_F$ for $i = k+1, \dots, q$. The smallest of the latter is $\sigma_q/\|R_k\|_F = (\|M\|_F/\|R_k\|_F)(\sigma_q/\|M\|_F) = \alpha_k\ell$, so the smallest nonzero singular value of X_γ^{def} is $\min(\alpha_k\ell, \gamma^{-1}) = \varphi_k$, and consequently $\sigma(X_\gamma^{\mathrm{def}}) \subset [\varphi_k, 1]$: indeed $\gamma^{-1} \leq 1$ since $\gamma \geq 1$, and $\sigma_i/\|R_k\|_F \leq \|R_k\|_2/\|R_k\|_F \leq 1$ for $i > k$. We record two endpoint inequalities for later use. Since $1 \leq k < q$, we have $q \geq 2$, hence $\|M\|_F > \sigma_q$ and $\ell < 1$. Moreover, $\|R_k\|_F \geq \sigma_q$, with equality exactly when $k = q-1$, so $\ell < \alpha_k\ell \leq 1$; together with $\gamma^{-1} \in (0, 1]$ this gives $0 < \varphi_k \leq 1$.

If $X = U\,\mathrm{diag}(x_1, \dots, x_q)V^\top$ with $x_i \in [0, 1]$, then $\mathcal{N}_d(X) = U\,\mathrm{diag}\big(p_d(x_1), \dots, p_d(x_q)\big)V^\top$. Thus, after L iterations, a singular value with initial value x_i becomes $p_d^{\circ L}(x_i)$.

Lemma 3 shows that p_d is nondecreasing and maps $[0, 1]$ into itself; the same is therefore true of $p_d^{\circ L}$. Hence the largest error occurs in a direction corresponding to the smallest initial singular value. If $x_{\min}^\star$ denotes the smallest nonzero singular value of $X^\star$, where $\star \in \{\mathrm{base}, \mathrm{def}\}$, then

$$\big\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X^\star)\big\|_2 = 1 - p_d^{\circ L}(x_{\min}^\star). \tag{24}$$

Indeed, the nonzero singular values of $\mathrm{polar}(M)$ are all one, and $p_d^{\circ L}(x_i) \in [0, 1]$.

For the deflated initialization, $x_{\min}^{\mathrm{def}} = \varphi_k$. Since $0 < \varphi_k \leq 1$, applying (17) with $x = \varphi_k$ gives

$$\big\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X_\gamma^{\mathrm{def}})\big\|_2 = 1 - p_d^{\circ L}(\varphi_k) \leq \big(1-\varphi_k^2\big)^{(d+1)^L}.$$

For the relative bound, note that $x_{\min}^{\mathrm{base}} = \ell$ and $x_{\min}^{\mathrm{def}} = \varphi_k$. The hypothesis $\gamma^{-1} > \ell$, together with $\alpha_k\ell > \ell$, gives $\varphi_k > \ell$. Since $0 < \ell < \varphi_k \leq 1$ and $\ell < 1$, the hypotheses of Lemma 4 hold; applying (18) with $x = \ell$ and $y = \varphi_k$, together with (24), gives

$$\frac{\big\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X_\gamma^{\mathrm{def}})\big\|_2}{\big\|\mathrm{polar}(M) - \mathcal{N}_d^{\circ L}(X^{\mathrm{base}})\big\|_2} = \frac{1-p_d^{\circ L}(\varphi_k)}{1-p_d^{\circ L}(\ell)} \leq \left(\frac{1-\varphi_k^2}{1-\ell^2}\right)^{(d+1)^L}.$$

Finally, $\varphi_k > \ell$ gives $\varphi_k^2 > \ell^2$, and $\varphi_k \leq 1$ gives $1 - \varphi_k^2 \geq 0$. Since $\ell < 1$, the denominator $1 - \ell^2$ is positive, and we conclude $0 \leq (1-\varphi_k^2)/(1-\ell^2) < 1$, which proves the final strict inequality. □

C SUPPORTING FIGURES FOR §2

Figure 7 shows the momentum spectra measured during GPT-2 training that motivate §2.1, and Figure 8 traces every singular value of the toy spectrum of Example 1 through five steps of the classical degree-five Newton–Schulz iteration (3), without and with deflation of the head σ_1.

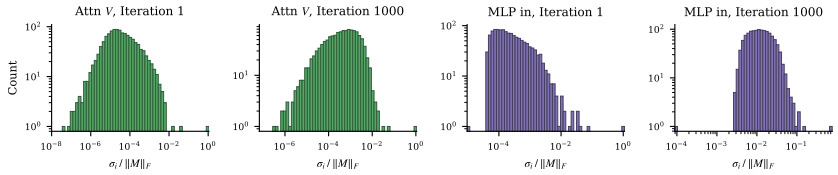


Figure 7: Frobenius-normalized singular value spectra $\hat{\sigma}_i(M)$ of the momentum matrices tracked by Muon in the middle transformer block (layer $\lfloor N/2 \rfloor$ of N layers) of GPT-2 Large: the four attention projections Q, K, V, O and the two MLP projections, at training steps 1 and 1000.

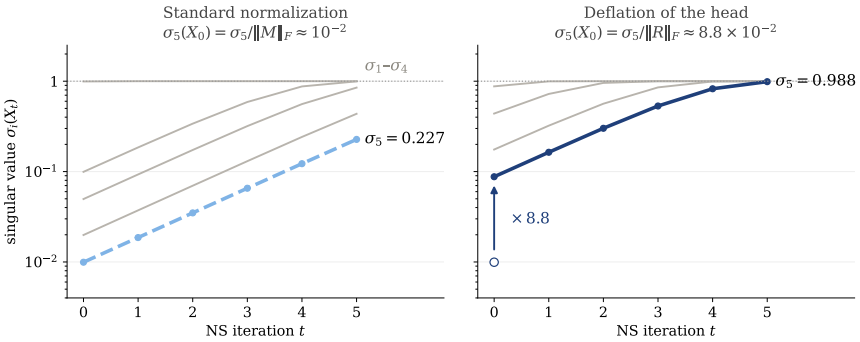


Figure 8: Effect of deflation on the toy spectrum $(\sigma_1, \ldots, \sigma_5) = (10, 1, 0.5, 0.2, 0.1)$ of Example 1 under five steps of the classical degree-five Newton–Schulz iteration (3). Left: standard Frobenius normalization $X_0 = M/\|M\|_F$. Right: deflation of the head followed by normalization of the residual. The hollow marker shows where σ_5 would start without deflation, and the arrow its lift by $\alpha_1 = \|M\|_F/\|R\|_F \approx 8.8$; the remaining singular values (gray) are lifted by the same factor.

D BATCHED RANDOMIZED SVD: IMPLEMENTATION AND ESTIMATION ACCURACY

This appendix states the batched randomized SVD used by §2.2 (Algorithm 3), records the implementation choices behind it, and quantifies how accurately the two estimators of §2—Algorithm 3 for Muon and the warm-started tracking of Algorithm 2 for ADMM—recover the components they deflate.

Grouping. Momentum matrices are grouped by shape: all matrices with the same (n, m) are stacked into one batch, and Algorithm 3 runs once per group per optimizer step. Wide matrices ($n < m$) are transposed before grouping and the resulting polar factor is transposed back, using $\mathrm{polar}(M^\top) = \mathrm{polar}(M)^\top$. For GPT-2 Large this yields two groups per optimizer step: the 1280×1280 group holding the attention Q, K, V, and O projections of all 36 blocks ($B = 144$), and the 5120×1280 group holding the two MLP projections ($B = 72$).

Operating point. The experiments use $w = 0.025$ and $\delta = 0.025$, so the sketch has $r' = 2r = \lceil 0.05\, m \rceil$ columns, with $J = 1$ subspace iteration.

Algorithm 3 BATCHED RANDOMIZED SVD

Require: $A \in \mathbb{R}^{B\times n\times m}$, $n \geq m$; window $w \in (0,1)$; oversampling ratio $\delta \geq 0$; subspace iterations $J \geq 0$
Ensure: $U \in \mathbb{R}^{B\times n\times r}$, $s \in \mathbb{R}^{B\times r}$, $V \in \mathbb{R}^{B\times m\times r}$, with $r = \lceil wm \rceil$

$r \leftarrow \lceil wm \rceil$; $\quad r' \leftarrow \min\{m,\ r + \lceil \delta m \rceil\}$
$\Omega \leftarrow \mathrm{Gaussian}(B, m, r')$
$Q \leftarrow \mathrm{CholeskyQR}(A\Omega)$
for $j = 1, \ldots, J$ **do**
    $Y \leftarrow \mathrm{CholeskyQR}(A^\top Q)$; $\quad Q \leftarrow \mathrm{CholeskyQR}(AY)$
end for
$G \leftarrow Q^\top A$
$(U_G, s, V) \leftarrow \mathrm{BatchedThinSVD}(G)$ ▷ $G = U_G\,\mathrm{diag}(s)V^\top$, $s_1 \geq \cdots \geq s_{r'} \geq 0$
$U \leftarrow QU_G$
$U \leftarrow U_{:,\,:,\,1:r}$; $\quad s \leftarrow s_{:,\,1:r}$; $\quad V \leftarrow V_{:,\,:,\,1:r}$ ▷ drop the oversampling columns
return (U, s, V)

Symmetric warm-started variant. The deflated PSD projection (§2.3) uses a symmetric variant of Algorithm 3 in the spirit of warm-started block eigensolvers (Knyazev, 2001). For a symmetric input $Z \in \mathbb{S}^n$, the routine tracks a single orthonormal basis $V \in \mathbb{R}^{n\times K}$ with $K = \lceil w'n \rceil$, returns Ritz values and vectors from the Rayleigh–Ritz step, and warm-starts the subspace from the previous ADMM iterate; since consecutive iterates are close, a single subspace-iteration pass ($J = 1$) suffices per ADMM iteration.

How inexact components enter the bounds. Let $(\widehat{U}_k, \hat{s}, \widehat{V}_k)$ be the estimated triplets that Algorithm 1 actually uses, $\widehat{H} := \widehat{U}_k\,\mathrm{diag}(\hat{s})\widehat{V}_k^\top$ and $\widehat{R} := M - \widehat{H}$, and let

$$\widehat{X}_0 := \widehat{R}/\|\widehat{R}\|_F + \gamma^{-1}\widehat{U}_k\widehat{V}_k^\top$$

be the resulting entry; X_γ^{def} of (9) is the same entry built from the exact triplets with the same k. The iteration $\mathcal{N}_d^{\circ L}$ acts on the singular values of its argument and leaves its singular vectors unchanged, so the output $\mathcal{N}_d^{\circ L}(\widehat{X}_0)$ has singular values $p_d^{\circ L}\big(\sigma_i(\widehat{X}_0)\big)$ in the singular directions of $\widehat{X}_0$. By Weyl's inequality, $|\sigma_i(\widehat{X}_0) - \sigma_i(X_\gamma^{\mathrm{def}})| \leq \eta := \|\widehat{X}_0 - X_\gamma^{\mathrm{def}}\|_2$, hence $\sigma(\widehat{X}_0) \subset [\varphi_k - \eta,\ 1+\eta]$: the spectral floor of Theorem 1 degrades from φ_k to $\varphi_k - \eta$, and since p_d is nondecreasing with $p_d(1+t) - 1 = O(t^{d+1})$ (Lemma 3), the part of the spectrum above one is contracted at the same order; in particular, the orthogonality defect $\|X_L^\top X_L - I\|$ of the output depends on $\sigma(\widehat{X}_0)$ alone. The limit of the iteration is $\mathrm{polar}(\widehat{X}_0)$ rather than $\mathrm{polar}(M) = \mathrm{polar}(X_\gamma^{\mathrm{def}})$, and the perturbation theory of the polar factor (Li, 1995; Higham, 2008) bounds the difference by $2\|\widehat{X}_0 - X_\gamma^{\mathrm{def}}\|_F/(2\varphi_k - \eta)$. Both effects are governed by η, and expanding

$$\widehat{X}_0 - X_\gamma^{\mathrm{def}} = \frac{H_k - \widehat{H}}{\|\widehat{R}\|_F} + R_k\Big(\frac{1}{\|\widehat{R}\|_F} - \frac{1}{\|R_k\|_F}\Big) + \gamma^{-1}\big(\widehat{U}_k\widehat{V}_k^\top - U_kV_k^\top\big)$$

shows that η is small whenever the head singular values and the two head subspaces are accurate; the singular values are compared with an exact SVD below, and the same holds for Algorithm 2 with eigenpairs in place of singular triplets.

Bounds for randomized subspace iteration. Algorithm 3 is randomized subspace iteration with J power steps and a Gaussian sketch of r' columns (Halko et al., 2011). Write $\Omega_1 = V_k^\top\Omega$ and $\Omega_2 = V_\perp^\top\Omega$ for the components of the sketch along and orthogonal to the exact right head subspace, and $p = r' - k$ for the oversampling. Saibaba (2019) bounds the canonical angles between the estimated and the exact left head subspaces by $\sin\theta_i \leq (\sigma_{k+1}/\sigma_i)^{2J+1}\|\Omega_2\Omega_1^\dagger\|_2$ for $i \leq k$, with an analogous bound for the right subspace, and Gu (2015) shows $\sigma_i \geq \hat{s}_i \geq \sigma_i/\sqrt{1 + (\sigma_{k+1}/\sigma_i)^{2(2J+1)}\|\Omega_2\Omega_1^\dagger\|_2^2}$ for the singular values; for a Gaussian sketch, $\mathbb{E}\|\Omega_2\Omega_1^\dagger\|_2 \leq \sqrt{k/(p-1)} + e\sqrt{(k+p)(m-k)}/p$ (Halko et al., 2011; Saibaba, 2019). The error decays with the gap ratio σ_{k+1}/σ_i raised to the power $2J+1$, so the single subspace iteration of the training runs

($J = 1$) already suppresses it. The bounds depend on the gap at the cutoff rather than on the gate threshold τ: the gate certifies $\sigma_{k+1} < \tau\sigma_1 < \sigma_k$ but not a gap between σ_k and σ_{k+1}, so when the spectrum is flat around $\tau\sigma_1$ the trailing head directions are resolved less accurately; such directions carry singular values close to $\tau\sigma_1$ on both sides of the cutoff, and exchanging them changes $\widehat{X}_0$ little.

Accuracy of the estimated head. On the 24 momentum matrices of §3.1 we run Algorithm 3 with the constants of the training runs ($w = \delta = 0.025$, $J = 1$, single precision) for 20 Gaussian sketches each and compare the k deflated singular values—k being the width chosen by the gate—with the leading k singular values of a double-precision SVD; the gate fires on 21 of the 24 matrices. Table 3 summarizes the relative error $\max_{i \le k} |\hat{s}_i - \sigma_i| / \sigma_i$ for each momentum matrix over the training steps and sketches.

Momentum matrix	median	min	max
Attention Q	8.0e-3	2.3e-8	5.6e-2
Attention K	1.3e-2	2.9e-7	6.9e-2
Attention V	8.8e-8	8.1e-9	9.0e-7
Attention O	5.8e-7	1.1e-8	2.8e-2
MLP (first)	6.7e-3	3.4e-8	4.8e-2
MLP (second)	5.1e-4	1.7e-8	2.4e-3

Table 3: Relative error $\max_{i \le k} |\hat{s}_i - \sigma_i| / \sigma_i$ of the deflated singular values estimated by the batched randomized SVD on the momentum matrices of §3.1, at the operating point of the training runs; statistics over the training steps at which the gate fires and over 20 sketches per matrix.

Tracking accuracy along the ADMM trajectory. We rerun the deflated ADMM of Figure 5 on the four instances of §3.3 and, every 100 iterations, eigendecompose the current Z exactly in double precision. Let λ_i be the Ritz values of Algorithm 2, $\mathcal{I} = \{i : |\lambda_i| > \tau s_1\}$ the deflated set, and $\lambda_i^\star$ the exact eigenvalues, matched by sign and magnitude order. Figure 9 reports $\max_{i \in \mathcal{I}} |\lambda_i - \lambda_i^\star| / |\lambda_i^\star|$ at the checkpoints where the gate is active: the error stays below 3×10^{-7}.

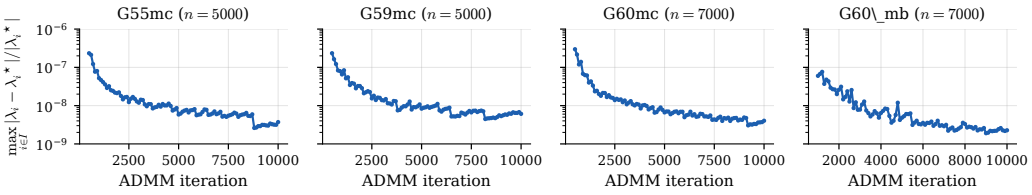


Figure 9: Relative error of the deflated Ritz values inside deflated ADMM on the instances of §3.3, against an exact double-precision eigendecomposition of the current Z every 100 iterations, at the checkpoints where the gate is active.

E Deflated PSD Projection: Details and Analysis

This appendix collects the material behind §2.3: the full ADMM iteration (§E.1), the construction of the fixed composite filter (§E.2), the proofs and the idealized analysis (§E.3), and the experimental details (§E.4).

E.1 ADMM FOR STANDARD-FORM SDP

We solve the standard-form pair (7) with a classical three-block ADMM (Wen et al., 2010); with penalty parameter $\sigma > 0$, one iteration is

$$\begin{aligned} y^{(k+1)} &= (\mathcal{A}\mathcal{A}^*)^{-1}\big(\sigma^{-1}b - \mathcal{A}(\sigma^{-1}X^{(k)} + S^{(k)} - C)\big), \\ S^{(k+1)} &= \Pi_{\mathbb{S}^n_+}\big(Z^{(k+1)}\big), \qquad Z^{(k+1)} := C - \mathcal{A}^* y^{(k+1)} - \sigma^{-1}X^{(k)}, \\ X^{(k+1)} &= X^{(k)} + \sigma\big(S^{(k+1)} + \mathcal{A}^* y^{(k+1)} - C\big). \end{aligned} \tag{25}$$

The y-update is a linear solve with the fixed matrix $\mathcal{A}\mathcal{A}^*$ and the X-update is a matrix addition, so the per-iteration cost is dominated by the projection in the S-update, which we evaluate by Algorithm 2.

Normalization scale. To place the spectrum of X_0 in $[-1, 1]$, the scale θ (and $\hat{\theta}$ for the residual) is obtained from a 20-step Lanczos run (Parlett, 1998, Chapter 13) on the squared matrix: if (ρ, v) is the largest Ritz pair of Z^2, we use $\theta = \sqrt{\rho + \|Z^2 v - \rho v\|_2}$. This value is a certified upper bound on $\|Z\|_2$ when ρ is the Ritz value closest to $\lambda_{\max}(Z^2)$ (Kang et al., 2025); taking the *largest* Ritz value of a random-start run is a heuristic, which Kang et al. (2025) found to yield a valid upper bound in all their experiments.

E.2 COMPOSITE MINIMAX FILTER

The component polynomials of the fixed filter g are constructed offline in two stages, following Kang et al. (2025), whose construction adapts the composite minimax sign approximation of Lee et al. (2022). Let $\mathcal{P}^{\mathrm{odd}}_{d_t}$ denote the space of odd real polynomials of degree at most d_t, and let $\varepsilon > 0$ exclude a small neighborhood of the discontinuity of the sign function. The first stage constructs a minimax approximation to $\mathrm{sign}(x)$ through sequential Remez problems. Starting from $[a_1, b_1] = [\varepsilon, 1]$, the polynomial at stage t is chosen as

$$g_t = \operatorname*{arg\,min}_{g \in \mathcal{P}^{\mathrm{odd}}_{d_t}} \max_{x \in [a_t, b_t]} |g(x) - 1|, \qquad [a_{t+1}, b_{t+1}] = g_t([a_t, b_t]). \tag{26}$$

Because g_t is odd, it simultaneously approximates -1 on the reflected negative interval. Repeating this construction yields a minimax-optimal composite approximation $g_T \circ \cdots \circ g_1$ to the sign function on $[-1, -\varepsilon] \cup [\varepsilon, 1]$.

Since a composition optimized for the sign function is not necessarily optimal for the matrix ReLU, the second stage initializes the component polynomials with $\{g_t\}_{t=1}^T$ and jointly refines their coefficients by approximately minimizing

$$\min_{p_1, \ldots, p_T} \max_{x \in [-1,1]} \left| \frac{x}{2}\left(1 + (p_T \circ \cdots \circ p_1)(x)\right) - \max(x, 0) \right|. \tag{27}$$

The maximum is evaluated over a dense set of points, and the polynomial coefficients are refined by gradient descent while retaining the composite representation. This avoids expanding the composition into a single high-degree polynomial and preserves efficient matrix evaluation through GEMMs.

For all experiments, we use the resulting filter in half precision, with $\varepsilon = 10^{-3}$ and $T = 7$ degree-five odd polynomials of the form $p_t(x) = \alpha_{t,1}x + \alpha_{t,3}x^3 + \alpha_{t,5}x^5$. Their coefficients remain fixed throughout ADMM. Deflation changes only the normalized input to this filter and the subsequent restoration of the deflated eigenspaces.

E.3 ERROR ANALYSIS

We first prove Proposition 2, restated for self-containment together with the bound on the deflated error itself that the proof yields alongside the ratio bound. The proof follows the three steps of Appendix B: it identifies the spectral floors of the two initializations, reduces the matrix error to a scalar residual by a spectral computation, and applies Lemmas 3–4 at the floors ℓ and $\alpha_k \ell$.

Proposition 2 (Deflated PSD projection; restated with the deflated-error bound). *Let $Z \in \mathbb{S}^n$ be nonzero with nonzero eigenvalues $\lambda_1, \ldots, \lambda_q$ ordered so that $\sigma_i = |\lambda_i|$ satisfies $\sigma_1 \geq \cdots \geq \sigma_q > 0$,*

let $1 \le k < q$ *with* $\sigma_{k+1} < \sigma_1$, *let* $\gamma \ge 1$, *and let* $X^{\mathrm{base}} = Z/\|Z\|_2$, $X_\gamma^{\mathrm{def}} = \gamma^{-1} U_k \operatorname{sign}(\Lambda_k) U_k^\top + R_k/\|R_k\|_2$, $\ell = \sigma_q/\|Z\|_2$, *and* $\alpha_k = \|Z\|_2/\|R_k\|_2$ *be as in §2.4, eq.* (9) *with* $A = Z$, *and let* $\widehat{\Pi}^{\mathrm{base}}(Z) = \frac{1}{2} Z\big(I + \mathcal{N}_d^{\circ L}(X^{\mathrm{base}})\big)$ *and* $\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) = U_k(\Lambda_k)_+ U_k^\top + \frac{1}{2} R_k\big(I + \mathcal{N}_d^{\circ L}(X_\gamma^{\mathrm{def}})\big)$. *Then* $\sigma(X^{\mathrm{base}}) \subset [\ell, 1]$ *and* $\sigma(X_\gamma^{\mathrm{def}}) \subset \{\gamma^{-1}\} \cup [\alpha_k \ell, 1]$, *and* $\widehat{\Pi}_\gamma^{\mathrm{def}}(Z)$ *does not depend on* γ. *Moreover, for every* $d \ge 1$ *and* $L \ge 0$,

$$\big\|\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2 \le \frac{\|R_k\|_2}{2} \max_{\alpha_k \ell \le x \le 1} x\big(1 - x^2\big)^{(d+1)^L} \le \frac{\|R_k\|_2}{2}\big(1 - \alpha_k^2 \ell^2\big)^{(d+1)^L}, \quad (28)$$

and the deflated-to-baseline error ratio satisfies

$$\frac{\big\|\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2}{\big\|\widehat{\Pi}^{\mathrm{base}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2} \le \left(\frac{1 - \alpha_k^2 \ell^2}{1 - \ell^2}\right)^{(d+1)^L} < 1.$$

Proof of Proposition 2. Write the spectral decomposition $Z = \sum_{i=1}^q \lambda_i u_i u_i^\top$ with orthonormal $u_1, \dots, u_q$ and $\sigma_i = |\lambda_i|$, so that $U_k \Lambda_k U_k^\top = \sum_{i \le k} \lambda_i u_i u_i^\top$, $R_k = \sum_{i > k} \lambda_i u_i u_i^\top$, and $\Pi_{\mathbb{S}_+^n}(Z) = \sum_{i=1}^q (\lambda_i)_+ u_i u_i^\top$. Set $\theta := \theta(Z) = \|Z\|_2 = |\lambda_1|$ and $\theta_k := \theta(R_k) = \|R_k\|_2 = |\lambda_{k+1}|$, so that $\alpha_k = \theta/\theta_k$.

Step 1: spectral floors. The nonzero eigenvalues of $X^{\mathrm{base}} = Z/\theta$ are λ_i/θ, $i = 1, \dots, q$, with magnitudes in $[|\lambda_q|/\theta, 1] = [\ell, 1]$; hence $\sigma(X^{\mathrm{base}}) \subset [\ell, 1]$. The two terms of X_γ^{def} act on orthogonal eigenspaces, so its nonzero eigenvalues are $\gamma^{-1} \operatorname{sign}(\lambda_i)$ for $i \le k$ and λ_i/θ_k for $i > k$. The magnitudes of the latter lie in $[|\lambda_q|/\theta_k, 1]$, and $|\lambda_q|/\theta_k = (\theta/\theta_k)(|\lambda_q|/\theta) = \alpha_k \ell$; hence $\sigma(X_\gamma^{\mathrm{def}}) \subset \{\gamma^{-1}\} \cup [\alpha_k \ell, 1]$. We record two endpoint inequalities for later use: the hypothesis $|\lambda_{k+1}| < |\lambda_1|$ gives $\alpha_k > 1$ and $\ell \le |\lambda_{k+1}|/|\lambda_1| < 1$, and $|\lambda_q| \le |\lambda_{k+1}|$ gives $\alpha_k \ell \le 1$; thus $0 < \ell < \alpha_k \ell \le 1$.

Step 2: scalarization. For a symmetric X we have $XX^\top = X^2$, so (8) reads $\mathcal{N}_d(X) = \sum_{i=0}^d a_i X^{2i+1} = p_d(X)$, with p_d the odd scalar polynomial of Lemma 3. Consequently, if $X = \sum_i x_i u_i u_i^\top$, then $\mathcal{N}_d^{\circ L}(X) = \sum_i p_d^{\circ L}(x_i) u_i u_i^\top$, and $p_d^{\circ L}(-x) = -p_d^{\circ L}(x)$. Since $R_k u_i = 0$ for $i \le k$, the padded head of X_γ^{def} is annihilated in the product

$$R_k \mathcal{N}_d^{\circ L}\big(X_\gamma^{\mathrm{def}}\big) = \sum_{i > k} \lambda_i \, p_d^{\circ L}(\lambda_i/\theta_k) \, u_i u_i^\top,$$

which does not involve γ; this proves that $\widehat{\Pi}_\gamma^{\mathrm{def}}(Z)$ is independent of γ. Using $(\lambda)_+ = \frac{\lambda}{2}\big(1 + \operatorname{sign}(\lambda)\big)$ and the oddness of $p_d^{\circ L}$, we have $\lambda\big(p_d^{\circ L}(\lambda/\vartheta) - \operatorname{sign}(\lambda)\big) = -|\lambda|\big(1 - p_d^{\circ L}(|\lambda|/\vartheta)\big)$ for every $\lambda \ne 0$ and every scale $\vartheta > 0$. The head terms $U_k(\Lambda_k)_+ U_k^\top$ and $\sum_{i \le k} (\lambda_i)_+ u_i u_i^\top$ cancel, and the kernel of Z contributes nothing to either approximation, so

$$\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) - \Pi_{\mathbb{S}_+^n}(Z) = -\frac{1}{2} \sum_{i > k} |\lambda_i| \big(1 - p_d^{\circ L}(|\lambda_i|/\theta_k)\big) u_i u_i^\top, \quad (29)$$

$$\widehat{\Pi}^{\mathrm{base}}(Z) - \Pi_{\mathbb{S}_+^n}(Z) = -\frac{1}{2} \sum_{i=1}^q |\lambda_i| \big(1 - p_d^{\circ L}(|\lambda_i|/\theta)\big) u_i u_i^\top. \quad (30)$$

Let $x_i := |\lambda_i|/\theta \in [\ell, 1]$ for $i = 1, \dots, q$, so that $|\lambda_i|/\theta_k = \alpha_k x_i \in [\alpha_k \ell, 1]$ for $i > k$. By Lemma 3, $p_d^{\circ L}$ maps $[0, 1]$ into itself, so every coefficient in (29)–(30) is nonnegative; both error matrices are negative semidefinite, and by the unitary invariance of the spectral norm,

$$\big\|\widehat{\Pi}_\gamma^{\mathrm{def}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2 = \frac{\theta_k}{2} \max_{k < i \le q} \alpha_k x_i \big(1 - p_d^{\circ L}(\alpha_k x_i)\big), \quad (31)$$

$$\big\|\widehat{\Pi}^{\mathrm{base}}(Z) - \Pi_{\mathbb{S}_+^n}(Z)\big\|_2 = \frac{\theta}{2} \max_{1 \le i \le q} x_i \big(1 - p_d^{\circ L}(x_i)\big). \quad (32)$$

These are the counterparts of (24); the weight x_i is new, because the projection error in the direction of λ_i is proportional to $|\lambda_i|$.

Step 3: bounds at the floors. For $i > k$, apply (17) with $x = \alpha_k x_i \in [\alpha_k \ell, 1]$:

$$\alpha_k x_i\big(1 - p_d^{\circ L}(\alpha_k x_i)\big) \le \alpha_k x_i\big(1 - \alpha_k^2 x_i^2\big)^{(d+1)^L} \le \max_{\alpha_k \ell \le x \le 1} x(1-x^2)^{(d+1)^L} \le (1 - \alpha_k^2 \ell^2)^{(d+1)^L},$$

where the last step uses $x \le 1$ and the monotonicity of $1 - x^2$. Inserting this into (31) proves (28).

For the ratio bound, fix $i > k$. Then $0 < x_i \le \alpha_k x_i \le 1$ and $x_i \le |\lambda_{k+1}| / |\lambda_1| < 1$, so (18) applies with $x = x_i$ and $y = \alpha_k x_i$:

$$\frac{1 - p_d^{\circ L}(\alpha_k x_i)}{1 - p_d^{\circ L}(x_i)} \le \left(\frac{1 - \alpha_k^2 x_i^2}{1 - x_i^2}\right)^{(d+1)^L} \le \left(\frac{1 - \alpha_k^2 \ell^2}{1 - \ell^2}\right)^{(d+1)^L} =: \rho.$$

The second inequality holds because $\frac{1-\alpha^2 x^2}{1-x^2} = 1 - (\alpha^2 - 1)\frac{x^2}{1-x^2}$ is decreasing in $x \in (0,1)$ for $\alpha > 1$ and $x_i \ge \ell$; both bases are nonnegative because $\alpha_k x_i \le 1$, so the inequality survives raising to the power $(d+1)^L$. Multiplying by $|\lambda_i| = \theta_k \alpha_k x_i = \theta x_i$ gives, for every $i > k$,

$$\theta_k\, \alpha_k x_i\big(1 - p_d^{\circ L}(\alpha_k x_i)\big) \le \rho\, \theta\, x_i\big(1 - p_d^{\circ L}(x_i)\big) \le \rho\, \theta \max_{1 \le j \le q} x_j\big(1 - p_d^{\circ L}(x_j)\big).$$

Taking the maximum over $i > k$ and using (31)–(32) proves the inequality in (11). The denominator is positive: $0 < x_q = \ell < 1$, and the recursion (19) in the proof of Lemma 4 shows inductively that $1 - \big(p_d^{\circ L}(\ell)\big)^2 > 0$, hence $1 - p_d^{\circ L}(\ell) > 0$. Finally, $0 < \ell < \alpha_k \ell \le 1$ gives $0 \le (1-\alpha_k^2\ell^2)/(1-\ell^2) < 1$, which is the strict inequality. □

The proof uses the exact scales $\|Z\|_2$ and $\|R_k\|_2$ only through $\theta \ge \|Z\|_2$, $\hat\theta \ge \|R_k\|_2$, and $\hat\theta < \theta$; hence Proposition 2 remains valid for the Lanczos upper estimates of Algorithm 2, with $\ell = |\lambda_q| / \theta$, $\alpha_k = \theta/\hat\theta$, $\alpha_k \ell = |\lambda_q| / \hat\theta$, and $\|R_k\|_2$ replaced by $\hat\theta$ in the prefactor of (28).

The fixed filter. The implemented filter is fixed offline on $[\varepsilon, 1]$ (§E.2); the following bound takes it as given.

Proposition 5 (Deflated projection with a fixed filter)**.** *Let $Z \in \mathbb{S}^n$ have nonzero eigenvalues $\lambda_1, \dots, \lambda_q$ with $|\lambda_1| \ge \cdots \ge |\lambda_q| > 0$, let $1 \le k < q$, and let $Z = U_k \Lambda_k U_k^\top + R_k$ be the exact head split of* (2)*. Let g be a fixed odd polynomial (possibly composite) with sign error $\delta_\varepsilon = \max_{\varepsilon \le x \le 1} |g(x) - 1|$ on $[\varepsilon, 1]$, and let $\hat\theta \ge \|R_k\|_2$. If $|\lambda_q| \ge \varepsilon\hat\theta$, then*

$$\left\| U_k(\Lambda_k)_+ U_k^\top + \frac{1}{2} R_k\big(I + g(R_k/\hat\theta)\big) - \Pi_{\mathbb{S}^n_+}(Z) \right\|_2 \;\le\; \frac{\hat\theta}{2} \max_{\varepsilon \le x \le 1} x\, |g(x) - 1| \;\le\; \frac{\hat\theta}{2}\, \delta_\varepsilon.$$

Proof of Proposition 5. Write the spectral decomposition $Z = \sum_{i=1}^q \lambda_i u_i u_i^\top$, so that $R_k = \sum_{i=k+1}^q \lambda_i u_i u_i^\top$ and $U_k(\Lambda_k)_+ U_k^\top = \sum_{i=1}^k (\lambda_i)_+ u_i u_i^\top$, the u_i being the columns of U_k. Define $h(x) := \frac{x}{2}\big(1 + g(x)\big)$, so that $\frac{1}{2} R_k\big(I + g(R_k/\hat\theta)\big) = \hat\theta\, h(R_k/\hat\theta)$ and $h(0) = 0$: the head directions and the kernel of Z contribute nothing to the filtered term. Using spectral functional calculus and $(\lambda)_+ = \frac{\lambda}{2}\big(1 + \operatorname{sign}(\lambda)\big)$,

$$U_k(\Lambda_k)_+ U_k^\top + \hat\theta\, h(R_k/\hat\theta) - \Pi_{\mathbb{S}^n_+}(Z) = \frac{1}{2} \sum_{i=k+1}^{q} \lambda_i \Big[g\big(\lambda_i/\hat\theta\big) - \operatorname{sign}(\lambda_i) \Big]\, u_i u_i^\top.$$

By the orthogonality of the eigenvectors and the unitary invariance of the spectral norm, with $x_i := \lambda_i/\hat\theta$,

$$\left\| U_k(\Lambda_k)_+ U_k^\top + \hat\theta\, h(R_k/\hat\theta) - \Pi_{\mathbb{S}^n_+}(Z) \right\|_2 = \frac{\hat\theta}{2} \max_{k < i \le q} |x_i|\; \big|g(x_i) - \operatorname{sign}(x_i)\big|.$$

Since $\hat\theta \ge \|R_k\|_2$ and $|\lambda_q| \ge \varepsilon\hat\theta$, every x_i with $i > k$ satisfies $\varepsilon \le |x_i| \le 1$. Because g is odd, $|x|\; |g(x) - \operatorname{sign}(x)| = |x|\; |g(|x|) - 1|$, so the maximum is at most $\max_{\varepsilon \le x \le 1} x\, |g(x) - 1| \le \delta_\varepsilon$. □

If the floor condition fails, an eigenvalue with $|\lambda_i| < \varepsilon\hat{\theta}$ contributes at most $\frac{1}{2}|\lambda_i|\big(1+|g(x_i)|\big) \le \frac{\hat{\theta}}{2}\varepsilon\big(1+\max_{|x|\le\varepsilon}|g(x)|\big)$ to the maximum above, which yields a two-regime bound.

We next analyze the idealized case in which the composite filter is regenerated on the deflated interval, so that the sequential minimax optimality of the construction (26) can be invoked. This result explains what the fixed filter gives up, and it recovers the doubly exponential rate of Theorem 1.

Theorem 6 (Idealized sequential-minimax filter)**.** *Let $Z = U_k\Lambda_k U_k^\top + R_k$ and k be as in Proposition 5, let $\theta_k := \|R_k\|_2 = |\lambda_{k+1}|$, and let $a_k := |\lambda_q|/\theta_k \in (0,1]$, so that the nonzero eigenvalues of R_k/θ_k lie in $[-1,-a_k]\cup[a_k,1]$. Let $g = g_T \circ \cdots \circ g_1$ be the composite filter generated by the sequential minimax construction* (26) *initialized on $[a_k,1]$, where each g_t is an odd polynomial of degree at most $2d+1$, with integers $d, T \ge 1$. Define $h_g(x) := \frac{x}{2}\big(1+g(x)\big)$. Then the deflated PSD approximation $\widehat{P}_{\mathrm{def}} := U_k(\Lambda_k)_+U_k^\top + \theta_k h_g(R_k/\theta_k)$ satisfies*

$$\big\|\widehat{P}_{\mathrm{def}} - \Pi_{\mathbb{S}^n_+}(Z)\big\|_2 \le \frac{\theta_k}{2}(1-a_k^2)^{(d+1)^T}. \tag{33}$$

Proof. The spectral computation in the proof of Proposition 5, with $\hat{\theta} = \theta_k$, gives

$$\big\|\widehat{P}_{\mathrm{def}} - \Pi_{\mathbb{S}^n_+}(Z)\big\|_2 \le \frac{\theta_k}{2}\max_{a_k\le x\le 1}|1-g(x)|\,. \tag{34}$$

It remains to bound the scalar sign-approximation error. Consider the degree-$(2d+1)$ Newton–Schulz polynomial $p_{\mathrm{NS},d}(x) := x\sum_{j=0}^{d} c_j(1-x^2)^j$, with $c_j := 4^{-j}\binom{2j}{j}$, and its T-fold composition $f_T := p_{\mathrm{NS},d}^{\circ T}$. By Lemma 3, $p_{\mathrm{NS},d}$ maps $[0,1]$ into itself and satisfies $0 \le 1 - p_{\mathrm{NS},d}(x)^2 \le (1-x^2)^{d+1}$ for $x \in [0,1]$. For $x_0 = x \in [0,1]$ and $x_{t+1} = p_{\mathrm{NS},d}(x_t)$, set $r_t := 1 - x_t^2$. The preceding inequality gives $r_{t+1} \le r_t^{d+1}$, and induction yields $0 \le 1 - f_T(x)^2 \le (1-x^2)^{(d+1)^T}$. Moreover, $f_T(x) \in [0,1]$, and hence

$$|1-f_T(x)| = \frac{1-f_T(x)^2}{1+f_T(x)} \le (1-x^2)^{(d+1)^T}.$$

The polynomial f_T satisfies the same degree and composition constraints as g. By Theorem 1 of Kang et al. (2025), the sequential minimax construction makes g optimal over this class for uniform approximation of the sign function on $[-1,-a_k]\cup[a_k,1]$. Therefore,

$$\max_{a_k\le x\le 1}|1-g(x)| \le \max_{a_k\le x\le 1}|1-f_T(x)| \le \max_{a_k\le x\le 1}(1-x^2)^{(d+1)^T} = (1-a_k^2)^{(d+1)^T}. \tag{35}$$

Substituting (35) into (34) proves (33). □

E.4 Experimental Details

For the experiments of §3.3, we monitor the maximum KKT residual

$$\eta = \max\Big\{\tfrac{\|\mathcal{A}X-b\|_2}{1+\|b\|_2},\ \tfrac{\|\mathcal{A}^*y+S-C\|_F}{1+\|C\|_F},\ \tfrac{\big|\langle C,X\rangle - b^\top y\big|}{1+|\langle C,X\rangle|+|b^\top y|},\ \tfrac{\max\{0,-\lambda_{\min}(X)\}}{1+\|b\|_2},\ \tfrac{\max\{0,-\lambda_{\min}(S)\}}{1+\|C\|_F}\Big\}. \tag{36}$$

Because computing the last two terms requires explicit eigenvalue information, they are not evaluated during the ADMM iterations. The convergence curves therefore report the maximum of the first three, eigenvalue-free terms of (36). At the end of each run, the two PSD-feasibility terms are additionally evaluated using a double-precision eigendecomposition of the final iterates. In our experiments, the first three terms dominate the final two terms, so the eigenvalue-free residual accurately reflects the observed convergence behavior.

F Additional Experiments

Deflation cutoff. Table 4 reports the final validation loss when deflation is enabled for different percentages of the optimizer steps (§3.4).

		DEFLATED MUON, deflation cutoff			
Mapping	Muon	10%	30%	50%	100%
Newton–Schulz	3.3813	3.3716	3.3604	3.3599	**3.3575**
Polar Express	3.3675	3.3587	3.3574	**3.3552**	3.3579

Table 4: Final validation loss of GPT-2 Large when deflation is enabled for different percentages of the optimizer steps. The *Muon* column is the baseline without deflation; bold marks the best cutoff for each mapping, and every tested cutoff improves on the baseline.

Deflation ratio. Tables 5 and 6 report the sweeps over the window ratio w and the threshold τ of §3.4: the final validation loss and the number k of singular pairs deflated per matrix, averaged over the optimizer steps at which the gate fires.

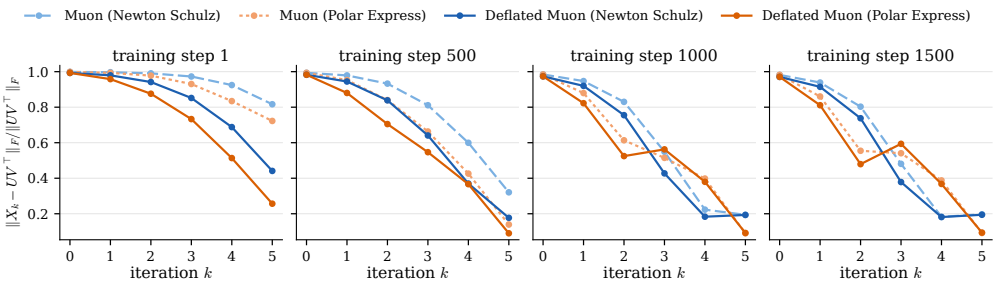
	Window ratio w ($\tau = 0.1$)				
	0.0025	0.005	0.01	0.025	Muon
Newton–Schulz					
Val. loss	**3.3574**	**3.3597**	**3.3568**	**3.3574**	3.3794
Removed k	1.5	2.3	3.8	8.8	–
Polar Express					
Val. loss	**3.3546**	**3.3573**	**3.3542**	**3.3581**	3.3671
Removed k	1.6	2.5	4.0	9.4	–

Table 5: Window ratio w at $\tau = 0.1$: final validation loss of GPT-2 Large under DEFLATED MUON and the average number k of singular pairs deflated per matrix, over the optimizer steps at which the gate fires. The column "Muon" shows the baseline without deflation.

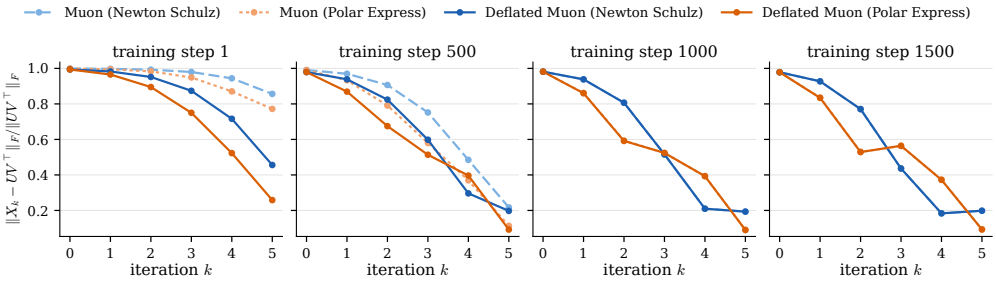
	Threshold τ ($w = 0.025$)				
	0.05	0.1	0.2	0.3	Muon
Newton–Schulz					
Val. loss	**3.3581**	**3.3574**	**3.3575**	**3.3565**	3.3794
Removed k	8.8	8.8	5.2	3.3	–
Polar Express					
Val. loss	**3.3605**	**3.3581**	**3.3596**	**3.3552**	3.3671
Removed k	9.6	9.4	5.8	4.0	–

Table 6: Threshold τ at $w = 0.025$, same quantities as Table 5. Deflation consistently improves the validation loss over the Muon baseline.

Matrix-function accuracy on all recorded matrices. Figures 10 and 11 extend Figure 2 to all six momentum matrices recorded in the middle block of GPT-2 Large: the MLP input and output projections and the attention projections Q, K, V, O, at training steps 1, 500, 1000, and 1500. The deflated arms run the batched randomized SVD of Algorithm 3 (a sketch of $\lceil 0.025m \rceil$ columns plus $\lceil 0.025m \rceil$ oversampling columns, one subspace iteration) and the gate with $w = 0.025$, $\tau = 0.1$, $\gamma = 1.01$ for Newton–Schulz and $\gamma = 1.1$ for Polar Express, exactly as in training; the sketch is seeded per matrix. Wherever the momentum has a pronounced head, deflation lowers the error at every iteration for both mappings; on K at step 1000 and on the MLP output projection at steps 1000 and 1500 the spectrum has flattened, the gate does not fire, and the deflated curves coincide with the baselines. On K at step 1500 the exact spectrum sits just above the gate threshold ($\sigma_{32}/\sigma_1 = 0.105$ for the window $r = 32$), but the estimated σ_{32} lies 5–7% below the exact value for every sketch seed we tried, so the production gate fires and deflates 31 directions; the deflated curves then improve on the baselines by up to 0.053 (Newton–Schulz) and 0.042 (Polar Express) over the five iterations. Replacing the estimated head by the exact one changes no other gate decision and moves every other curve by at most 0.003. With Newton–Schulz the deflated iterate is never less accurate than the baseline at any iteration; with Polar Express, the curves on the MLP projections at later steps cross at intermediate iterations and agree to within 0.003 after $L = 5$ iterations. Evaluating the polynomial iterations in `bfloat16`, the training precision, instead of single precision moves individual curves by at most 0.017; the ordering of the arms is unchanged, except that rounding lets the Newton–Schulz deflated iterate exceed its baseline by up to 0.001 at one point (MLP input projection, step 1000) and widens the Polar Express agreement on the MLP projections to 0.004.

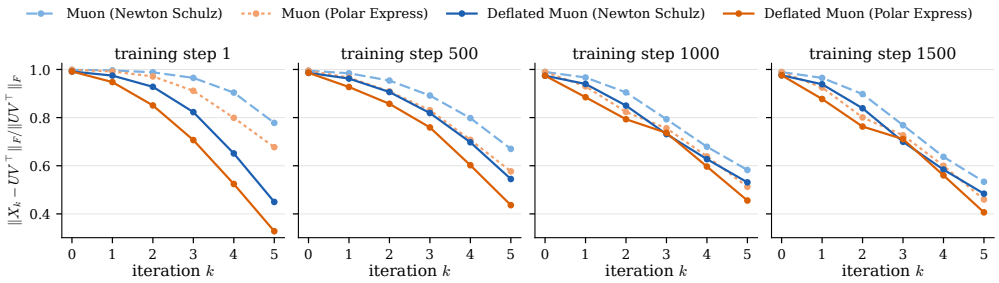


(a) MLP input projection.

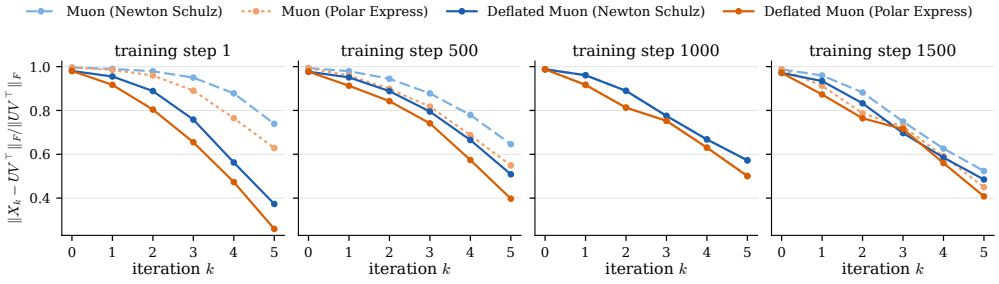


(b) MLP output projection.

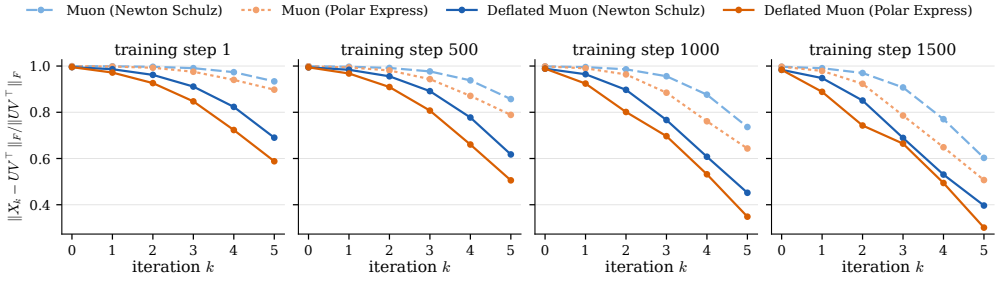
Figure 10: Matrix-level accuracy on the two MLP momentum matrices of the middle block of GPT-2 Large, recorded at training steps 1, 500, 1000, and 1500, as a function of the number of iterations. Each panel reports the relative error $\|X_L - UV^\top\|_F / \|UV^\top\|_F$ of the Newton–Schulz and Polar Express mappings and of their deflated counterparts. Where the gate does not fire (output projection at steps 1000 and 1500), the deflated curve coincides with its baseline.



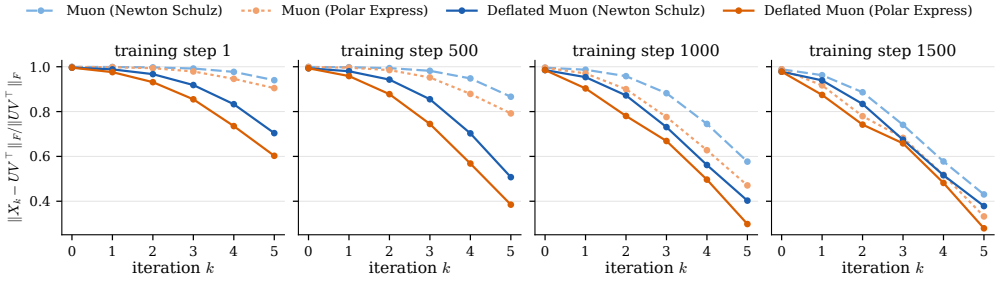
(a) Attention Q projection.



(b) Attention K projection.



(c) Attention V projection (Figure 2 of the main text).



(d) Attention O projection.

Figure 11: Matrix-level accuracy on the four attention momentum matrices of the middle block of GPT-2 Large, in the setting of Figure 10. Where the gate does not fire (K at step 1000), the deflated curve coincides with its baseline; on K at step 1500 the estimated spectrum triggers the gate although the exact one sits just above the threshold (see text).