Pre-Normalization Momentum and Rank Bias in Adaptive Optimizers

Raghav Kaushik Ravi [*1] **Srivarshinee Sridhar** [*1]

Abstract

Adaptive optimizers induce strikingly different implicit rank biases. Adam reliably recovers low-rank solutions in deep matrix factorization, whereas RMSProp substantially overshoots despite using the same adaptive normalization mechanism. We trace much of this behavior to Adam's pre-normalization momentum, which suppresses fluctuations in trailing directions that would otherwise inflate rank. Controlled ablations are consistent with this explanation. Removing momentum substantially weakens Adam's low-rank recovery, whereas post-normalization momentum does not reproduce Adam-like behavior. Similar behavior appears during TinyLlama fine-tuning, where the measured optimizer differences are broadly consistent with the theoretical prediction. Together, these results suggest that pre-normalization temporal filtering is an important source of optimizer-induced spectral bias.

1. Introduction

Adaptive optimization methods such as Adam (Kingma & Ba, 2017), AdamW (Loshchilov & Hutter, 2019), RMSProp (Tieleman & Hinton, 2012), and their variants dominate modern deep learning practice. Beyond their role in accelerating optimization, these methods are now understood to induce distinct forms of *implicit bias*, shaping the geometry, sparsity, margin structure, and spectral properties of learned solutions even in the absence of explicit regularization (Arora et al., 2019; Cattaneo et al., 2024; Wang et al., 2022). Understanding which architectural components of an optimizer determine these biases remains an important open problem.

A particularly striking example arises in deep matrix factorization. Classical analyses of deep linear networks show that gradient flow preferentially learns dominant singular directions while suppressing trailing components, yielding an implicit bias toward low-rank solutions (Saxe et al., 2014; Arora et al., 2019). Recently, Cao et al. (2023) observed that this behavior depends strongly on the optimizer. Adam preserves low-rank structure far more effectively than RMSProp despite both methods employing the same adaptive normalization mechanism of the form $g/\sqrt{v}$. The source of this discrepancy remains unclear.

In this work, we identify a previously overlooked optimizer component that appears to play an important role in this phenomenon. We show that Adam's β_1 momentum buffer acts as a pre-normalization termporal filter that suppresses fluctuations in trailing-directions, whereas RMSProp lacks such a mechanism. These observations suggest that the placement of temporal filtering relative to adaptive normalization is an important determinant of the resulting spectral bias.

Our analysis proceeds in two stages. First, we show that adaptive normalization removes the depth-dependent suppression mechanism responsible for incremental rank learning in deep linear networks. As a result, adaptive methods become susceptible to rank overshoot. Second, we show that Adam partially restores suppression through its pre-normalization exponential moving average. Under a simple stationary-noise model, this filter reduces update variance to a fraction

$$\frac{1-\beta_1}{1+\beta_1},$$

of the unfiltered value (a standard-deviation reduction factor ≈ 0.23 at $\beta_1 = 0.9$).

We test this explanation through a series of controlled ablation experiments. Removing β_1 from Adam substantially degrades low-rank recovery, while adding post-normalization momentum to RMSProp fails to reproduce Adam-like behavior. Sweeping β_1 further reveals a sharp transition. Weak momentum produces substantial rank overshoot, whereas stable low-rank recovery emerges once $\beta_1 \gtrsim 0.7$ (refer to Fig. 1).

Finally, we observe similar behavior beyond deep linear networks. During TinyLlama fine-tuning, the measured optimizer separation is broadly consistent with the theoretical prediction, and optimizer-dependent update spectra emerge

[*]Equal contribution [1]Vellore Institute of Technology. Correspondence to: Raghav Kaushik Ravi <raghavkaushik.r2022@vitstudent.ac.in>, Srivarshinee Sridhar <srivarshinee.s2022@vitstudent.ac.in>.

Workshop on Weight-Space Symmetries, held in conjunction with the 43rd International Conference on Machine Learning, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

across Transformer layers.

Overall, the paper makes four contributions:

- We identify pre-normalization temporal filtering as an important contributor to implicit low-rank regularization in adaptive methods.
- We analyze how adaptive normalization alters the depth-induced suppression of trailing singular directions, potentially increasing susceptibility to rank overshoot.
- We derive a simple variance characterization explaining how Adam's β_1 buffer suppresses trailing-direction updates relative to RMSProp.
- We provide empirical evidence consistent with the proposed mechanism through ablation studies, β_1 sweeps, and experiments on LLM fine-tuning.

2. Setup

We study a depth-L deep linear network

$$W = W_L W_{L-1} \cdots W_1 \in \mathbb{R}^{d\times d}$$

trained on the regression objective

$$\mathcal{L}(W) = \frac{1}{2}\|XW^\top - Y\|_F^2, \qquad Y = X(M^*)^\top,$$

where the target matrix M^* has rank $r^* \ll d$. Unless otherwise specified, experiments use $d = 64$, $L = 2$, and $r^* = 4$. This classical deep matrix factorization setup (Saxe et al., 2014; Arora et al., 2019) admits analytically tractable singular-value dynamics and isolates optimizer-induced implicit bias. Full experimental details appear in Appendix A.1.

We compare four optimizers spanning the two architectural axes studied in this work: adaptive normalization and pre-normalization momentum filtering. Specifically, we evaluate AdamW, NAdam, RMSProp (without momentum), and SGD with momentum. For a matrix M with singular values $\{\sigma_i\}$, we measure stable rank

$$\mathrm{sr}(M) = \frac{\|M\|_F^2}{\|M\|_2^2},$$

numerical rank at threshold $\tau = 0.01$, and the trailing-energy fraction

$$\rho(M) = \frac{\sum_{i>r^*}\sigma_i^2}{\sum_i \sigma_i^2}.$$

We track ρ on both the per-step update Δ_t and the integrated weights W_t. The ratio

$$\rho(W)/\rho(\Delta)$$

emerges as the central diagnostic of optimizer-induced spectral suppression.

3. A Mechanistic Hypothesis

We hypothesize that the key architectural component governing optimizer-induced rank behavior is the temporal filter applied *before* adaptive normalization, namely Adam's β_1 momentum buffer. Our argument has two parts: (i) adaptive normalization removes the depth-dependent suppression that ordinarily attenuates trailing singular directions, and (ii) Adam partially restores this suppression through pre-normalization temporal filtering.

3.1. Adaptive Normalization Removes Depth Suppression

In deep linear networks, gradient flow decouples along singular directions (Saxe et al., 2014; Arora et al., 2019):

$$\frac{d\sigma_i}{dt} = -L\,\sigma_i^{2-2/L}(\sigma_i - \sigma_i^*).$$

For trailing directions $i > r^*$ with $\sigma_i^* = 0$ and $\sigma_i \approx \varepsilon$, the update scales as

$$\frac{d\sigma_i}{dt} \propto \varepsilon^{3-2/L},$$

yielding exponentially suppressed optimization dynamics with depth. This mechanism underlies incremental rank learning in SGD.

Adaptive methods remove this suppression. Adam-family optimizers apply updates

$$\Delta_t \propto \frac{m_t}{\sqrt{v_t}}, \qquad v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2.$$

At stationarity,

$$\sqrt{v_i} \approx |g_i|_{\mathrm{rms}},$$

so the gradient magnitude cancels:

$$\Delta_i \propto \frac{m_i}{|g_i|_{\mathrm{rms}}}.$$

Consequently, small trailing gradients receive updates comparable in scale to dominant directions. The depth-dependent suppression factor is therefore removed universally across adaptive normalization methods, including Adam, NAdam, and RMSProp.

3.2. Pre-Normalization Momentum Filtering

Adam differs from RMSProp in the placement of its temporal filter:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t, \qquad \Delta_t = \frac{m_t}{\sqrt{v_t}}.$$

The EMA therefore acts on the raw gradient before normalization.

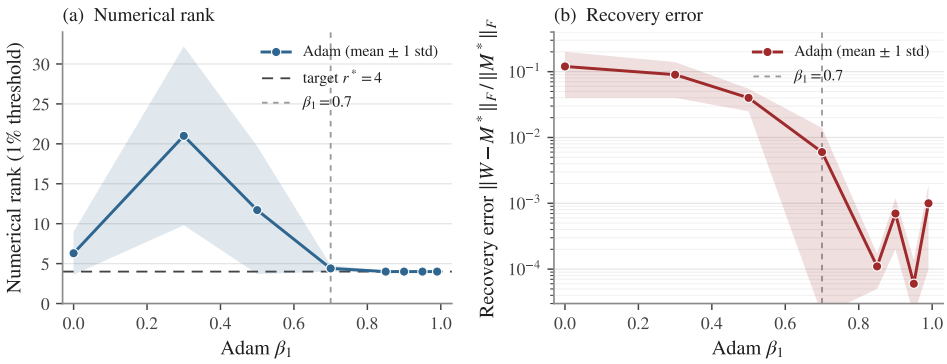

Figure 1. Numerical-rank recovery as a function of β_1. Weak momentum produces severe rank overshoot, while $\beta_1 \gtrsim 0.7$ consistently recovers the target rank.

Modeling a trailing-direction gradient as stationary mean-zero noise with variance s^2 yields

$$\mathrm{Var}(m_i) = \frac{1-\beta_1}{1+\beta_1} s^2.$$

Thus Adam attenuates trailing-direction variance by

$$\frac{1-\beta_1}{1+\beta_1},$$

corresponding to a standard-deviation reduction of

$$\sqrt{\frac{1-\beta_1}{1+\beta_1}} \approx 0.229 \qquad (\beta_1 = 0.9).$$

Proposition 3.1. *Assume the trailing-direction gradient is stationary mean-zero noise with variance s^2 and approximately stationary second-moment estimate $v_t \approx s^2$. Then*

$$\mathrm{Var}(\Delta_i^{\mathrm{Adam}}) = \alpha^2 \frac{1-\beta_1}{1+\beta_1}, \qquad \mathrm{Var}(\Delta_i^{\mathrm{RMSProp}}) = \alpha^2.$$

The proposition relies on an approximately stationary second-moment estimate and should be interpreted as a local characterization of the filtering effect rather than a complete model of Adam's optimization dynamics. Please refer to Appendix A.2 for a more detailed discussion.

Scope of Proposition 3.1. Proposition 3.1 establishes a local property of Adam's momentum dynamics. Under stationary trailing-direction fluctuations, the momentum filter reduces the variance of the trailing component by a factor of $(1-\beta_1)/(1+\beta_1)$ at each update. This result is intentionally local and does not by itself characterize the spectral structure or effective rank of the accumulated weight trajectory W. Our interpretation is that repeatedly suppressing trailing-direction fluctuations causes optimization updates to become increasingly concentrated in a small number of dominant directions, which in turn leads to the low-rank weight updates observed in practice. Proposition 3.1 therefore identifies a plausible mechanism underlying this behavior, while the emergence of low-rank structure reflects the cumulative effect of many optimization steps. This distinction is important empirically. For example, the per-step variance reduction factor is approximately constant (e.g., ≈ 0.05 for $\beta_1 = 0.9$), whereas the observed separation in compression ratio $\rho(W)/\rho(\Delta)$ between Adam and RMSProp spans several orders of magnitude (§3.3). We therefore view Proposition 3.1 as explaining a local effect that is consistent with the observed spectral suppression rather than as a direct proof of low-rank recovery. The empirical results in §4-§5 support this interpretation, while a complete theoretical account of how these local effects accumulate remains an open question (Appendix A.2.2).

3.3. Compression-Ratio Diagnostic

To empirically probe the mechanism, we measure the trailing-energy fraction

$$\rho(M) = \frac{\sum_{i>r^*} \sigma_i^2}{\sum_i \sigma_i^2}$$

on both the per-step update Δ_t and the integrated weights W. The resulting compression ratio

$$\rho(W)/\rho(\Delta)$$

reveals a substantial optimizer-dependent separation.:

$$10^{-7}\ \text{(Adam)}, \qquad 10^{-1}\ \text{(RMSProp)}, \qquad \sim 1\ \text{(SGD)}.$$

Despite only a constant-factor per-step variance reduction, the resulting cumulative compression differs by roughly six orders of magnitude between Adam and RMSProp (refer to Appendix A.2.4 for spectral visualization). In § 5, we show that a similar scale separation is observed during LLM fine-tuning.

Table 1. Controlled falsification experiments for the proposed mechanism ($r^* = 4$). Results are mean values over 3 random seeds. (mom. - momentum)

Prediction	Configuration	NumRk	RecovErr
Baseline	Adam (β_1=0.9)	4.0	0.00069
Baseline	RMSProp	22.3	0.252
(i) remove β_1	Adam (β_1=0)	6.3	0.130
(ii) increase β_1	NAdam (β_1=0.99)	5.7	0.098
(iii) post-norm mom.	RMSProp + mom. 0.9	25.3	0.422
(iv) modify β_2	Adam (β_2=0.99)	4.0	0.0082

4. Falsification and the β_1 Threshold

The mechanism proposed in §3 makes four falsifiable predictions: (i) removing β_1 should break Adam, (ii) post-normalization momentum should fail, (iii) β_2 alone should not explain the effect, and (iv) increasing β_1 should suppress overshoot.

4.1. Falsification Experiments

Table 1 evaluates these predictions using controlled optimizer ablations.

The Adam-family ablations are broadly consistent with the proposed interpretation. Removing β_1 substantially weakens low-rank recovery in Adam, increasing the numerical rank from 4.0 to 6.3 and the recovery error from 7×10^{-4} to 0.13. In contrast, adding momentum after normalization in RMSProp does not reproduce the behavior observed in Adam. Varying β_2 alone has comparatively little effect on the measured rank and recovery metrics. Together, these results suggest that the placement of the β_1 momentum term before normalization plays an important role in the observed behavior. At the same time, removing β_1 from Adam does not fully reproduce the larger overshoot observed in RMSProp, indicating that other differences between the optimizers also contribute. These differences include bias correction, the ϵ term, and the default second-moment decay rate. NAdam provides a further exception to the simple filtering picture. Although it incorporates momentum, its Nesterov formulation includes a direct contribution from the current gradient, and its behavior differs slightly from that of Adam across the β_1 sweep. We therefore view the ablation results as providing evidence that is consistent with the proposed role of pre-normalization momentum within the Adam family, while also suggesting that additional optimizer-specific effects influence the final dynamics. Additional discussion is provided in Appendix A.3.

4.2. The β_1 Threshold

Sweeping β_1 reveals a sharp non-monotonic transition (Fig. 1). Intermediate values produce the largest rank overshoot, while $\beta_1 \geq 0.7$ consistently recovers the target rank across all seeds.

The peak overshoot occurs near $\beta_1 = 0.3$ and exceeds the value observed in the no-momentum case ($\beta_1 = 0$). This behavior is not predicted by Proposition 3.1. Under the assumptions of the proposition, the variance satisfies $\mathrm{Var}(m_t) = \frac{1-\beta_1}{1+\beta_1} s^2$ which decreases monotonically with β_1. The simplified model therefore predicts monotonic suppression of trailing-direction fluctuations as momentum increases. The observed non-monotonicity indicates that additional aspects of the optimization dynamics are not captured by the assumptions underlying Proposition 3.1. Possible contributors include non-stationarity in the trailing directions, temporal correlations across optimization steps, or interactions with the adaptive second-moment estimator. Understanding the origin of this behavior remains an interesting direction for future work. For larger values of β_1, the overshoot disappears and stable low-rank recovery is observed consistently across seeds.

5. Transfer to Real LLM Fine-Tuning

To test whether the mechanism extends beyond deep linear networks, we fine-tune TinyLlama-1.1B-Chat on WikiText-2 for 300 optimization steps under AdamW, NAdam, RMSProp, and SGD-momentum. Rank diagnostics are computed on 12 sampled projection matrices spanning early, middle, and late Transformer layers. Full training details are deferred to Appendix A.4.1.

5.1. Qualitative Consistency with Proposition 3.1

For $\beta_1 = 0.9$, Proposition 3.1 predicts a *per-step, trailing-direction* update-scale ratio

$$\frac{\sigma(\Delta^{\text{RMSProp}})}{\sigma(\Delta^{\text{AdamW}})} = \sqrt{\frac{1+\beta_1}{1-\beta_1}} \approx 4.36.$$

The closest measurable proxy in the LLM setting is the ratio of *cumulative, full-matrix* update norms,

$$\frac{\|\Delta W^{\text{RMSProp}}\|_F}{\|\Delta W^{\text{AdamW}}\|_F} = \frac{0.343}{0.101} \approx 3.4$$

The reported value is averaged over 12 sampled matrices from a single seed. It should not be interpreted as a direct estimate of the trailing-direction variance ratio. The quantity ΔW contains both leading directions, which are fit similarly by the two optimizers, and trailing directions, where the theory predicts a larger separation. As a result, the ratio measured on the full update matrix is expected to be smaller than the ratio associated with the trailing directions alone. The observed value of approximately 3.4 is therefore broadly consistent with the theoretical prediction of 4.36,

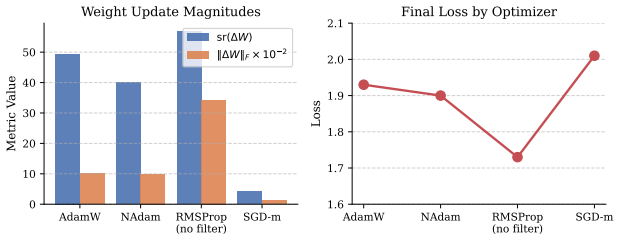

Figure 2. TinyLlama fine-tuning under different optimizers. RMSProp produces substantially larger and higher-rank updates than AdamW, while SGD preserves low-rank structure at the cost of weaker optimization progress.

although the comparison should be viewed as qualitative rather than exact. The result is based on a single seed, and we do not quantify variability across independent runs.

5.2. Optimizer-Dependent Update Spectra

RMSProp produces both larger-magnitude and higher-rank updates than AdamW, consistent with the proposed pre-normalization filtering mechanism. SGD yields dramatically lower-rank updates but substantially weaker optimization progress, mirroring the depth-suppressed behavior observed in the deep linear setting. NAdam closely tracks AdamW in this regime despite exhibiting mild overshoot in DLNs.

The optimizer-dependent rank ordering emerges within the first 50 optimization steps and remains stable throughout training, indicating that the effect is not merely a transient noise-floor phenomenon. Figure 2 summarizes the optimizer-dependent update statistics during fine-tuning.

6. Discussion and Limitations

The proposed mechanism accounts for a substantial portion of the observed Adam–RMSProp separation, but does not fully explain the resulting rank dynamics or the behavior of NAdam. In deep linear networks, NAdam exhibits mild residual rank overshoot that is not eliminated by increasing β_1, suggesting that its lookahead term introduces partially unfiltered gradient contributions. However, this discrepancy does not transfer to the LLM fine-tuning regime, where NAdam closely tracks AdamW both spectrally and in optimization performance.

Our analysis characterizes per-step trailing-direction variance under approximately stationary noise. Empirically, however, the resulting compression ratios differ by several orders of magnitude between Adam and RMSProp, indicating that the local filtering effect compounds constructively with long-timescale optimization dynamics. Formalizing this amplification mechanism remains an important open problem.

More broadly, our results complement recent work on optimizer-induced implicit bias (Cattaneo et al., 2024; **?**; Orvieto & Gower, 2026; Xu, 2026) by isolating the role of pre-normalization momentum filtering in low-rank dynamics. Additional discussion, extended related work, and future directions are deferred to Appendix A.5.

Impact Statement

This work advances the community's understanding of optimizer-induced implicit bias by showing that optimizer architecture, not merely learning rates or normalization can systematically shape the rank structure of learned updates. These findings may influence the design of parameter-efficient fine-tuning methods, optimizer-aware compression strategies, and more predictable training dynamics for large models. By exposing hidden optimizer biases, this work may improve reproducibility, efficiency, and principled optimizer selection in deep learning.

References

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c0c783b5fc0d7d808f1d14a6e9c8280d-Paper.pdf.

Cao, J., Qian, C., Huang, Y., Chen, D., Gao, Y., Dong, J., Guo, D., and Qu, X. A dynamics theory of implicit regularization in deep low-rank matrix factorization, 2023. URL https://arxiv.org/abs/2212.14150.

Cattaneo, M. D., Klusowski, J. M., and Shigida, B. On the implicit bias of Adam. In Salakhutdinov, R., Kolter, Z., Heller, K., Weller, A., Oliver, N., Scarlett, J., and Berkenkamp, F. (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5862–5906. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/cattaneo24a.html.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Orvieto, A. and Gower, R. M. In search of adam's secret sauce. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=CH72XyZs4y.

Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL https://arxiv.org/abs/1312.6120.

Tieleman, T. and Hinton, G. Lecture 6.5—rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

Wang, B., Meng, Q., Zhang, H., Sun, R., Chen, W., Ma, Z.-M., and Liu, T.-Y. Does momentum change the implicit regularization on separable data?, 2022. URL https://arxiv.org/abs/2110.03891.

Xu, Y. Optimizer-induced low-dimensional drift and transverse dynamics in transformer training, 2026. URL https://arxiv.org/abs/2602.23696.

A. Appendix.

A.1. Deep Linear Network Setup

All deep linear network experiments follow the classical matrix factorization framework of Saxe et al. (Saxe et al., 2014) and Arora et al. (Arora et al., 2019). The end-to-end weight matrix is parameterized as

$$W = W_L W_{L-1} \cdots W_1 \in \mathbb{R}^{d\times d},$$

and trained using the regression objective

$$\mathcal{L}(W) = \frac{1}{2}\|XW^\top - Y\|_F^2, \qquad Y = X(M^*)^\top.$$

The target matrix M^* has rank $r^* \ll d$, while the rows of

$$X \in \mathbb{R}^{n\times d}$$

are sampled independently from a Gaussian distribution. Unless otherwise stated, experiments use

$$d = 64, \qquad L = 2, \qquad r^* = 4, \qquad n = 512.$$

Each layer is initialized according to

$$W_k \sim \mathcal{N}(0, \varepsilon^2 I),$$

with initialization scale

$$\varepsilon = 0.01.$$

Training is performed using full-batch optimization for 5000 steps, eliminating finite-sample gradient noise.

The deep linear setting is used because: (i) the global optimum is analytically characterized, (ii) singular directions decouple in the aligned regime, and (iii) the model admits closed-form analysis of depth-dependent spectral suppression.

A.1.1. Optimizer Configurations

The primary experiments study four optimizers:
- AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
- NAdam ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
- RMSProp without momentum ($\alpha = 0.99$)
- SGD with momentum ($\mu = 0.9$)

Unless otherwise specified, each optimizer uses an individually tuned learning rate.

A.1.2. Metrics

For a matrix

$$M \in \mathbb{R}^{d\times d}$$

with singular values

$$\sigma_1 \geq \cdots \geq \sigma_d \geq 0,$$

we report the following spectral metrics.

Stable rank.

$$\mathrm{sr}(M) = \frac{\|M\|_F^2}{\|M\|_2^2}.$$

Numerical rank.

$$\mathrm{nr}_\tau(M) = |\{i : \sigma_i > \tau\sigma_1\}|,$$

using threshold

$$\tau = 0.01.$$

Recovery error. $\mathrm{RecovErr}(W) = \frac{\|W - M^*\|_F}{\|M^*\|_F}$, the relative reconstruction error of the end-to-end matrix against the target M^*.

Trailing-energy fraction.

$$\rho(M) = \frac{\sum_{i>r^*} \sigma_i^2}{\sum_i \sigma_i^2} \in [0, 1].$$

The quantity $\rho(M)$ measures the fraction of spectral energy lying outside the target rank.

Throughout the paper, we evaluate ρ on: the per-step update

$$\Delta_t = W_t - W_{t-1},$$

the cumulative fine-tuning update, and the end-to-end matrix W_t itself.

The compression ratio

$$\rho(W)/\rho(\Delta)$$

serves as the primary diagnostic of optimizer-induced spectral suppression.

A.2. Additional Discussion on the proposed mechanism

A.2.1. Why Post-Normalization Momentum Does Not Recover Adam

The temporal filter must operate on the raw gradient prior to adaptive normalization. Applying momentum after normalization, as in PyTorch's `RMSprop(momentum=0.9)`:

$$b_t = \mu b_{t-1} + \frac{g_t}{\sqrt{v_t}},$$

does not reproduce Adam's behavior. After normalization, the coordinate-wise gradient magnitudes have already been approximately equalized, so the EMA acts primarily on normalized directional fluctuations rather than on raw gradient amplitudes. Consequently, the buffer no longer attenuates trailing-direction fluctuations according to their underlying variance scale.

A.2.2. Full Proposition Statement and Assumptions

We restate Proposition 3.1 for completeness.

Proposition 3.1[Per-step trailing-direction amplitude] Let ξ_t be a stationary mean-zero stochastic process satisfying

$$\mathrm{Var}(\xi_t) = s^2, \qquad \mathrm{Cov}(\xi_t, \xi_{t'}) = 0 \quad (t \neq t').$$

Assume further that the adaptive second-moment estimate remains approximately stationary over the timescale of interest so that

$$v_t \approx s^2.$$

This approximation neglects fluctuations in the adaptive second-moment estimator and is intended only as a local characterization of the filtering effect.

Then

$$\mathrm{Var}(\Delta_i^{\mathrm{Adam}}) = \alpha^2 \frac{1-\beta_1}{1+\beta_1}, \qquad \mathrm{Var}(\Delta_i^{\mathrm{RMSProp}}) = \alpha^2.$$

Starting from Adam's momentum recursion,

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)\xi_t,$$

stationarity implies

$$\mathrm{Var}(m_t) = \beta_1^2 \mathrm{Var}(m_t) + (1-\beta_1)^2 s^2.$$

Solving for the stationary variance yields

$$\mathrm{Var}(m_t) = \frac{(1-\beta_1)^2}{1-\beta_1^2} s^2 = \frac{1-\beta_1}{1+\beta_1} s^2.$$

Assuming

$$v_t \approx s^2,$$

Adam updates satisfy

$$\Delta_t^{\mathrm{Adam}} = \alpha \frac{m_t}{\sqrt{v_t}},$$

and therefore

$$\mathrm{Var}(\Delta_t^{\mathrm{Adam}}) = \alpha^2 \frac{\mathrm{Var}(m_t)}{s^2} = \alpha^2 \frac{1-\beta_1}{1+\beta_1}.$$

For RMSProp,

$$\Delta_t^{\mathrm{RMSProp}} = \alpha \frac{\xi_t}{\sqrt{v_t}},$$

which immediately gives

$$\mathrm{Var}(\Delta_t^{\mathrm{RMSProp}}) = \alpha^2.$$

This establishes the variance ratio claimed in Proposition 3.1.

A.2.3. Per-Step Variance Versus Cumulative Drift

The proposition characterizes instantaneous trailing-direction variance under approximately stationary noise. Empirically, however, the cumulative compression ratio

$$\rho(W)/\rho(\Delta)$$

differs by approximately six orders of magnitude between Adam and RMSProp. This amplification substantially exceeds the constant-factor variance reduction predicted by the proposition, suggesting that the local filtering mechanism interacts nontrivially with long-timescale optimization dynamics.

Formalizing this cumulative amplification effect likely requires modeling temporal correlations in trailing directions together with feedback through the adaptive second-moment estimator v_t. We leave such analysis to future work.

A.2.4. Additional Spectral Visualizations

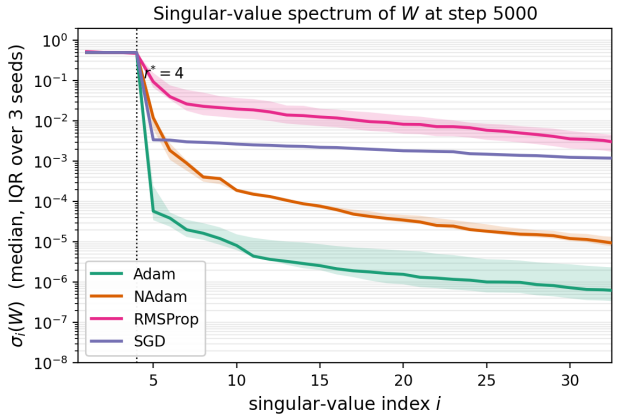

Figure 3. Singular-value spectrum of the end-to-end matrix W at training step 5000. RMSProp exhibits substantial trailing-spectrum inflation beyond the target rank $r^* = 4$, whereas Adam strongly suppresses tail singular directions. NAdam displays intermediate behavior, while SGD preserves low-rank structure through depth-induced suppression. Shaded regions denote interquartile range across three random seeds.

A.3. Additional Falsification Experiments

A.3.1. Extended Analysis of Falsification Experiments

Table 2 reports the full multi-seed falsification results.

Removing β_1 from Adam substantially degrades low-rank recovery, supporting the hypothesis that pre-normalization momentum filtering is essential to the mechanism. Conversely, adding post-normalization momentum to RMSProp fails to recover Adam-like behavior, consistent with the claim that filtering must occur prior to adaptive normalization.

Table 2. Extended falsification results. Mean $\pm$ std over 3 seeds.

Configuration	NumRk(1%)	RecovErr
Adam (β_1=0.9)	4.0 ± 0.0	0.00069
NAdam (β_1=0.9)	4.7 ± 0.5	0.015
RMSProp	22.3 ± 11.3	0.252
Adam (β_1=0)	6.3 ± 2.6	0.130
NAdam (β_1=0.99)	5.7 ± 1.2	0.098
RMSProp + momentum 0.9	25.3 ± 6.1	0.422
Adam (β_2=0.99)	4.0 ± 0.0	0.0082

The only inconsistent prediction arises in NAdam. Increasing β_1 from 0.9 to 0.99 does not eliminate mild overshoot. One possible explanation is that NAdam introduces a current-gradient lookahead term that bypasses the EMA filter. However, because the effect size remains relatively small, alternative explanations cannot presently be excluded.

A.3.2. Full β_1 Sweep Results

Table 3 reports the complete β_1 sweep.

Table 3. Effect of β_1 on numerical rank recovery.

β_1	NumRk(1%)
0.0	6.3 ± 2.6
0.3	21.0 ± 11.2
0.5	11.7 ± 8.0
0.7	4.3 ± 0.5
0.85	4.0 ± 0.0
0.9	4.0 ± 0.0
0.95	4.0 ± 0.0
0.99	4.0 ± 0.0

The dependence on β_1 is strongly non-monotonic. In particular, intermediate values of β_1 produce substantially larger rank overshoot than either the no-momentum setting or larger values of β_1. This behavior is not predicted by the simplified stationary model underlying Proposition 3.1, which implies monotonic variance reduction as β_1 increases.

The observed non-monotonicity therefore suggests that additional aspects of the optimization dynamics influence rank recovery. Possible contributors include temporal correlations, non-stationarity in the trailing directions, or interactions with the adaptive second-moment estimator. Understanding the origin of this behavior remains an open question for future work.

Above approximately $\beta_1 \approx 0.7$, stable low-rank recovery is observed consistently across all seeds.

A.4. LLM Fine-Tuning Details

A.4.1. Experimental Setup

All experiments use TinyLlama-1.1B-Chat (v1.0), a 22-layer decoder-only Transformer trained in bf16 precision.

Dataset and preprocessing. We fine-tune on the WikiText-2 training split (`wikitext-2-raw-v1`) using standard autoregressive language-model preprocessing. Documents are concatenated, tokenized without additional special tokens, and chunked into fixed sequences of length 512. The dataset is capped at 2000 sequences for controlled experimentation.

Training configuration. Training proceeds for 300 optimization steps using per-device batch size 2 with gradient accumulation 4, corresponding to an effective batch size of 8 sequences (4096 tokens) per step. Optimization uses cosine learning-rate decay with 15-step warmup, bf16 mixed precision, and gradient clipping at norm 1.0.

Optimizers. The primary optimizer configurations are:
- AdamW: learning rate 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$
- NAdam: learning rate 10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.999$
- RMSProp: learning rate 10^{-5}, $\alpha = 0.99$, no momentum
- SGD-momentum: learning rate 10^{-3}, momentum 0.9

Weight decay is disabled for all optimizers to isolate optimizer-induced spectral effects from explicit regularization.

A.4.2. Rank Measurements

Rank diagnostics are computed on 12 projection matrices corresponding to layers $\{2, 11, 20\}$ and the Q, V, gate, and down projections.

For each matrix, we store an fp32 initialization snapshot W_0 and evaluate

$$\Delta W = W_t - W_0$$

throughout training.

Measurements are recorded at optimization steps

$$\{0, 5, 10, 25, 50, 75, 100, 150, 200, 250, 300\}.$$

For each checkpoint, we compute stable rank, numerical rank, Frobenius norm, spectral norm, minimum singular value, and the full singular-value spectrum.

A.4.3. Additional Observations

The optimizer-dependent spectral ordering emerges within approximately 50 optimization steps and remains stable through the end of training. This persistence argues against a purely transient noise-floor interpretation.

The observed optimizer dependence of

$$\mathrm{sr}(\Delta W)$$

also suggests a possible connection to parameter-efficient fine-tuning methods such as LoRA. In particular, the intrinsic rank budget required to approximate full fine-tuning may itself depend on the optimizer used during training.

Several caveats remain. The current experiments use a single random seed and relatively short fine-tuning horizon. In addition, the primary comparisons use matched learning rates rather than matched update magnitudes. Finally, because

$$\|\Delta W\|/\|W\| \lesssim 1\%,$$

the fine-tuning regime remains close to the linearization regime, reducing the distinction between instantaneous and cumulative updates relative to training-from-scratch dynamics.

A.5. Extended Discussion and Limitations

A.5.1. The Mechanism Is Incomplete for NAdam

The mechanism developed in §3 explains the Adam-RMSProp gap cleanly but does not fully account for NAdam. Multi-seed experiments in the deep linear setting show that NAdam exhibits mild but persistent rank overshoot:

$$\mathrm{nr}_{0.01} = 4.7 \pm 0.5,$$

compared to

$$4.0 \pm 0.0$$

for Adam. Increasing β_1 to 0.99 does not eliminate the effect:

$$5.7 \pm 1.2.$$

One possible explanation is that NAdam introduces an explicit current-gradient contribution proportional to

$$(1 - \beta_1) g_t,$$

which bypasses the EMA filtering mechanism analyzed in §3. However, because the effect size remains relatively small, alternative explanations, including interactions with NAdam's time-varying momentum schedule, cannot presently be excluded.

Importantly, the mild overshoot observed in deep linear networks does not transfer to the LLM fine-tuning regime studied in §5, where NAdam closely tracks AdamW both spectrally and in optimization performance.

A.5.2. Per-Step Amplitude Versus Cumulative Drift

Proposition 3.1 characterizes the variance of per-step trailing-direction updates under stationary approximately uncorrelated noise. Empirically, however, the cumulative compression ratio

$$\rho(W)/\rho(\Delta)$$

reaches approximately 10^{-7} for Adam and 10^{-1} for RMSProp, corresponding to a six-order-of-magnitude separation.

This amplification substantially exceeds the constant-factor variance reduction predicted by Proposition 3.1, suggesting that the local filtering mechanism compounds constructively with long-timescale optimization dynamics.

A formal treatment would likely require jointly modeling:
- adaptive second-moment dynamics,
- temporal correlations in trailing directions,
- and feedback through v_t.

In the LLM fine-tuning regime,

$$\|\Delta W\|/\|W\| \lesssim 1\%,$$

placing optimization close to the linearization regime where the per-step approximation appears sufficient.

A.5.3. Relation to Recent Work

Several recent works study optimizer-induced implicit bias from complementary perspectives.

Cattaneo et al. (2024) analyze Adam using backward-error methods and show that Adam implicitly anti-penalizes the ℓ_1 norm of the loss gradient. Their analysis is compatible with the mechanism proposed here but does not isolate the role of the pre-normalization β_1 filter.

Wang et al. (2022) prove that momentum does not alter implicit regularization in linearly separable classification settings. Our results instead demonstrate strong momentum dependence in matrix factorization regimes, suggesting qualitatively different behavior across learning geometries.

Orvieto & Gower (2026) argue that coupling between β_1 and β_2 is central to Adam's empirical stability. This interpretation is broadly consistent with the threshold behavior observed in §4.2.

Finally, Xu (2026) report that AdamW induces a low-dimensional drift subspace during Transformer training absent under SGD. Although their metric differs from ours (trajectory PCA versus update-spectrum rank), the conclusions are qualitatively aligned.

A.5.4. Future Directions

Several directions naturally follow from this work.

First, the LLM experiments currently use a single random seed and relatively short fine-tuning horizon. Multi-seed evaluation together with learning-rate-controlled RMSProp baselines would help further disentangle spectral effects from overall update magnitude.

Second, it remains unknown whether the non-monotonic β_1 threshold observed in deep linear networks transfers to large-scale Transformer fine-tuning.

Third, the optimizer dependence of $\mathrm{sr}(\Delta W)$ suggests a previously unexplored connection between optimization dynamics and parameter-efficient fine-tuning. In particular, the rank budget required by LoRA-style methods to approximate full fine-tuning behavior may itself depend systematically on the optimizer used during training.