Rethinking LoRA Initialization for Robust Asymmetric Learning Rates

Disen Liao
University of Waterloo
Vector Institute
d7liao@uwaterloo.ca

Yaoliang Yu
University of Waterloo
Vector Institute
yaoliang.yu@uwaterloo.ca

Abstract

Low-Rank Adaptation (LoRA) is often highly sensitive to learning-rate (LR) choices. Recent empirical studies suggest that, once LRs are carefully tuned, vanilla LoRA can remain competitive with many proposed LoRA variants. This shifts the central practical question from designing ever more variants to reducing the cost and brittleness of LR selection. LoRA+ addresses this issue with asymmetric LRs, using a larger LR for the up-projection matrix B than for the down-projection matrix A. However, the ratio η_B/η_A itself becomes an additional hyperparameter: large ratios can improve adaptation, but under the standard LoRA initialization they can also cause unstable training and ratio-dependent performance collapse. We identify the source of this brittleness through a signal–noise decomposition of LoRA dynamics. Under standard `Init[A]`, the update contains an initialization-induced noise term whose magnitude can be affected by the LR for B, η_B, the initialization scale β, and the adapter rank r. In practical LLM fine-tuning, this non-vanishing noise can limit the performance gains expected from LoRA+ style asymmetric LRs. To suppress this effect, we propose `Init[A`^{2}`]`, a simple initialization that scales the variance of A as $\sigma_A^2 \propto n^{-2}$ while keeping $B_0 = 0$. Under matched rank and training time for both initializations, this reduces the initialization-noise energy by an additional factor of $1/n$, stabilizing asymmetric LoRA training while preserving zero initial adapter output. Empirically, `Init[A`^{2}`]` substantially widens the stable LR region across initialization scales, adapter ranks, and LoRA update multipliers. Across tasks and models, `Init[A`^{2}`]` reduces best-ratio drift, prevents the high-ratio collapse observed with standard `Init[A]`, and makes the width-based guideline $\eta_B \approx n\eta_A$ a useful starting point for finite pretrained models.

1 Introduction

Low-Rank Adaptation (LoRA) [1] is a standard method for parameter-efficient fine-tuning of large language models (LLMs) [2–4]. Given a width-n LoRA layer with pretrained weight $W \in \mathbb{R}^{n\times n}$ and input $\underline{Z} \in \mathbb{R}^n$, LoRA uses $\bar{Z} = W\underline{Z} + \frac{\alpha}{r}BA\underline{Z}$, where $A \in \mathbb{R}^{r\times n}$ and $B \in \mathbb{R}^{n\times r}$ are trainable with learning rates (LRs) η_A and η_B. Here $r \ll n$ is the adapter rank and α/r is the LoRA update multiplier. We use β to denote the initialization-magnitude hyperparameter: standard LoRA initialization, denoted `Init[A]`, initializes $B_0 = 0$ and initializes A_0 randomly with mean zero and standard deviation $\beta n^{-1/2}$, so the adapter preserves the pretrained model output at step zero.

Despite its simplicity, LoRA is sensitive to hyperparameters (HPs). The rank r controls the expressive capacity of the adapter [5], but its effect is inseparable from the update multiplier $\lambda_r = \alpha/r$. Kalajdzievski [6] shows that the conventional choice α/r is not rank-stabilized: it can shrink adapter activations and gradients as r grows, slowing learning at high ranks and motivating the alternative

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

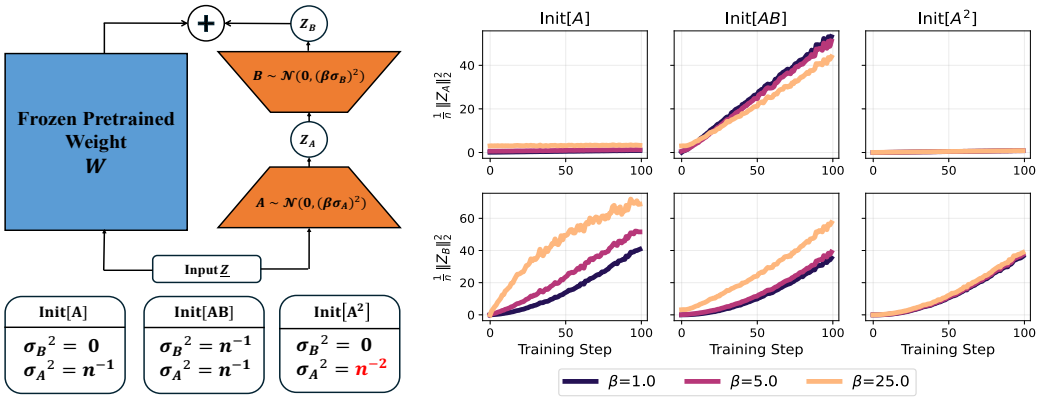


Figure 1: Overview of LoRA initialization and feature-norm stability. **Left**: LoRA branch notation and the initialization scalings compared in this work, including standard `Init[A]`, `Init[AB]`, and the proposed `Init[A`^{2}`]`. **Right**: feature-norm trajectories at width $n = 4096$ for different initialization scales β. `Init[A`^{2}`]` keeps the LoRA features nearly invariant to β, while the alternative scalings exhibit stronger initialization-dependent growth.

$\alpha/\sqrt{r}$. LRs introduce a further source of brittleness. Recent empirical work shows that, after sufficiently broad LR sweeps, vanilla LoRA can remain competitive with many proposed LoRA variants [7]. This shifts the practical question from only designing new adapters to understanding LoRA HP selection.

LoRA+ [8] gives a principled route toward understanding LR selection by assigning different LRs to the two LoRA matrices. Its infinite-width feature learning analysis suggests that efficient LoRA feature updates require an asymmetric regime with $\eta_B \gg \eta_A$ and, under idealized scaling assumptions, a ratio $\eta_B/\eta_A \approx n$. We write this ratio as $\rho = \eta_B/\eta_A$. The asymmetric regime is attractive because B directly controls the LoRA branch output, so increasing η_B can accelerate adaptation. In finite pretrained LLMs, however, ρ often becomes another hyperparameter to tune: large ratios can improve performance, but under `Init[A]` they can also trigger instability. We address the gap between the theoretical LR guidance and practical LLM fine-tuning in this paper.

We show that the fragility of asymmetric LRs comes from an initialization-dependent noise term that survives even though the LoRA branch has zero output at initialization. Although $B_0 = 0$, the random matrix A_0 affects the branch as soon as B moves away from zero. A signal–noise decomposition of the LoRA feature update reveals two components: a task-driven signal from optimizer updates and a noise term from the interaction between the learned B and the random initial A. Both terms scale with η_B, so increasing η_B amplifies useful adaptation and initialization noise at the same time. In the small-LR regime common in pretrained LLM fine-tuning, the signal can remain controlled while the `Init[A]` noise contribution does not vanish with width, making large ρ, large initialization scale β, and large update multiplier α/r especially brittle.

We propose `Init[A`^{2}`]`, a minimal initialization change that suppresses this noise at its source.[1] `Init[A`^{2}`]` keeps the standard zero initialization for B, but initializes A with one additional factor of width scaling: instead of $\sigma_A^2 = \Theta(n^{-1})$, it uses $\sigma_A^2 = \Theta(n^{-2})$. This preserves the LoRA architecture, training objective, and zero initial adapter output. The purpose is not to make the adapter inactive, but to prevent the random initial projection from being amplified by η_B, α/r, or β before a task-driven signal has formed. For comparable rank and time scaling of the optimizer updates, this reduces the initialization-noise energy by a factor of $1/n$. We do not claim that $\rho = n$ is always the finite-model optimum; instead, `Init[A`^{2}`]` reduces the need for exhaustive two-dimensional LR search by making the high-ratio regime substantially more stable.

We validate this claim through LR sweeps and fine-tuning experiments. On a controlled toy problem, `Init[A`^{2}`]` improves hyperparameter transfer across widths and keeps internal and output LoRA feature norms stable as β grows. On encoder-only RoBERTa fine-tuning and decoder-only arithmetic and commonsense benchmarks, `Init[A`^{2}`]` preserves the benefit of $\eta_B > \eta_A$ while widening the

[1] Our code is available at `https://github.com/liaodisen/LoRA-initA2`.

stable LR region and reducing best-ratio drift. These results position `Init[A`^{2}`]` as a lightweight initialization rule for making LoRA+-style asymmetric LR tuning practical in finite pretrained models.

Our contributions are threefold:

- We identify an LR-ratio failure mode in LoRA: asymmetric LRs with $\eta_B > \eta_A$ can improve adaptation, but under standard `Init[A]` the ratio ρ becomes brittle because η_B amplifies both the useful signal and an initialization-dependent noise term.
- We introduce `Init[A`^{2}`]`, a simple LoRA initialization rule that sets $\sigma_A^2 = \Theta(n^{-2})$ while keeping $B_0 = 0$. It preserves the feature-learning property of standard `Init[A]` under the same asymmetric LR scaling, while suppressing the initialization-noise term without changing the LoRA architecture, objective, zero-output initialization, or training cost.
- We show theoretically and empirically that `Init[A`^{2}`]` widens the stable LR region, improves robustness to large initialization scales and LoRA update multipliers, and makes asymmetric LR selection more predictable across models and tasks.

2 Background

2.1 LoRA Parameterization and Features

Following Hayou et al. [8], we consider a general L-layer neural network of the following form:

$$\begin{cases} Y_0 = W_{\text{in}} X, \\ Y_\ell(X) = \mathcal{F}_\ell(W_\ell, Y_{\ell-1}(X)), \ell \in [L], \\ Y_{\text{out}}(X) = W_{\text{out}} Y_L(X), \end{cases} \tag{1}$$

where $X \in \mathbb{R}^{d_{\text{in}}}$ denotes the input vector, n denotes the network width, and $\mathcal{F}$ denotes a nonlinear mapping function. $W_\ell \in \mathbb{R}^{n \times n}$ is the weight matrix of the ℓ-th hidden layer. $W_{\text{in}} \in \mathbb{R}^{n \times d_{\text{in}}}$ and $W_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times n}$ are the input and output layers, respectively. To fine-tune this model, LoRA is applied to a subset of $\{W_{\text{out}}, W_1, \ldots, W_L, W_{\text{in}}\}$. For clarity, the analysis below treats square width-n layers; the rectangular implementation is specified in Appendix C.1. LoRA is defined as:

Definition 1 (LoRA)**.** *For a width-n LoRA layer with frozen pretrained weight $W \in \mathbb{R}^{n \times n}$, we seek an update $W \leftarrow W + \Delta W$ for a specific downstream task. LoRA parameterizes the update as $\Delta W = \frac{\alpha}{r} BA$. Here, only $B \in \mathbb{R}^{n \times r}$ and $A \in \mathbb{R}^{r \times n}$ are trainable. The rank satisfies $r \ll n$, and α/r is the LoRA update multiplier.*

For a given LoRA layer, let $\underline{Z}$ be its input and let $\bar{Z}$ be its output. By Definition 1, $\bar{Z} = W\underline{Z} + \frac{\alpha}{r} BA\underline{Z}$. Since W is frozen during fine-tuning, we focus on the scaled LoRA branch update, $\frac{\alpha}{r} BA\underline{Z}$. Following Hayou et al. [8], we define the LoRA features:

Definition 2 (LoRA Features)**.** *Given a LoRA layer with input $\underline{Z}$ and matrices A, B, we define LoRA features as $Z_A = A\underline{Z}$ and $Z_B = \frac{\alpha}{r} BZ_A = \frac{\alpha}{r} BA\underline{Z}$. The superscript t (e.g., Z_A^t, Z_B^t) denotes the feature value at the t-th step of fine-tuning.*

2.2 Prior Work on LoRA with Scaling Theory

The theoretical analysis of deep neural networks is often impeded by nonlinearities and high-dimensional interactions. To make the behavior tractable, prior work often studies these models in infinite-width or infinite-depth limits, leveraging asymptotic properties governed by the Central Limit Theorem (CLT) and the Law of Large Numbers (LLN), reviewed in Appendix A.1. A critical objective within this framework is to identify scaling laws that avoid the "lazy" kernel regime, thereby preserving the model's ability to perform nontrivial feature learning [9] as dimensions grow. This scaling-limit perspective has facilitated the derivation of feature-preserving HP transfer rules for Transformers [10, 11] and Residual Networks [12, 13], as well as the analysis of low-precision training stability [14]. More recently, these techniques have been adapted to the analysis of LoRA, offering insights into initialization schemes and LR scalings that support feature adaptation [8, 15, 16]. We discuss more related work in Appendix B.

LoRA Feature Learning. To derive optimal scaling rules, Hayou et al. [8] analyze the fine-tuning process through a *coordinate-wise* feature learning perspective. When $\alpha/r = 1$, the update $\Delta Z_B^t := Z_B^{t+1} - Z_B^t$ decomposes into three interaction terms, where $\Delta Z_A^t := Z_A^{t+1} - Z_A^t$:

$$\Delta Z_B^t = \underbrace{B_t \Delta Z_A^t}_{\delta_1^t} + \underbrace{\Delta B_t Z_A^t}_{\delta_2^t} + \underbrace{\Delta B_t \Delta Z_A^t}_{\delta_3^t}. \tag{2}$$

Condition 1. *Hayou et al. [8] propose two conditions on this update:*

1. ***Stability:*** *for all $t > 1$, Z_B^t has $\Theta(1)$ coordinate size as the width $n \to \infty$.*
2. ***Efficiency:*** *for all $t > 1$, both δ_1^t and δ_2^t have $\Theta(1)$ coordinate size as the width $n \to \infty$.*

To systematically analyze these constraints, Hayou et al. [8] employ the γ-calculus, which quantifies the scaling of a variable v: if $v = \Theta(n^\lambda)$, then $\gamma[v] = \lambda$. For a matrix or vector X, $\gamma[X] = \lambda$ means that each coordinate scales as $\Theta(n^\lambda)$ (see Appendix A.2 for details). By applying this framework to Condition 1, Hayou et al. [8] demonstrate that standard symmetric LRs cannot satisfy both stability and efficiency simultaneously. Consequently, they advocate for asymmetric configurations, specifically `Init[A]` ($B_0 = 0, \sigma_A^2 = \Theta(1/n)$) paired with $\eta_A = \Theta(1/n)$ and $\eta_B = \Theta(1)$. Using a similar γ-calculus analysis under the equal-LR setting (i.e., $\eta_A = \eta_B = \eta^*$), Hayou et al. [15] show that `Init[A]` is better than `Init[B]` because it can tolerate a larger LR. Based on related analysis, Li et al. [16] challenge the necessity of zero initialization. Introducing the concept of *robustness*, they investigate whether initialization can preserve stability even when HPs deviate from optimal γ-scales. Their findings suggest that fine-tuning is more stable under $\eta_A = \eta_B$ when both matrices are initialized nonzero (i.e., `Init[AB]`).

3 Analysis

Previous work analyzes LoRA through a feature learning perspective, but achieving feature learning can be challenging in practice. Although Hayou et al. [8] propose the scaling $\eta_B \gg \eta_A$, the best ratio for a finite model still requires costly tuning. Furthermore, practical deployments may use a large LoRA update multiplier α/r to encourage rapid adaptation, even when LRs are small. In this section, we show that standard `Init[A]` then has initialization-induced noise that can accumulate over training, whereas our proposed `Init[A`^{2}`]` suppresses this source of instability.

3.1 Update Decomposition

We consider the LoRA setup defined in Equation (1). We explicitly parameterize the initialization magnitude using a scalar β. Let $\tilde{A}_0$ be a base initialization matrix with entries i.i.d. $\mathcal{N}(0, \sigma_A^2)$. The trainable matrices are initialized as $B_0 = 0$ and $A_0 = \beta \tilde{A}_0$.

Normalized optimizer model. We separate the scalar LR scale from the matrix-valued optimizer direction. Specifically, let $U_A^t \in \mathbb{R}^{r \times n}$ and $U_B^t \in \mathbb{R}^{n \times r}$ denote the directions produced by the optimizer at step t, after any coordinate-wise normalization or adaptive preconditioning but before multiplication by the LoRA-specific LRs:

$$A_{t+1} = A_t - \eta_A U_A^t, \qquad B_{t+1} = B_t - \eta_B U_B^t,$$

where the entries of U_A^t and U_B^t have coordinate size $O(1)$ uniformly in the width n. This formulation abstracts away the optimizer-specific normalization while keeping the LoRA-specific LR scale explicit. The algebraic identities derived below (Lemma 1) hold for any realized sequence of such update directions; optimizer-specific assumptions are needed only when estimating their scaling with width. SignSGD is the special case $U = \operatorname{sign}(\nabla \mathcal{L})$, while Adam and AdamW also fit this normalized-direction model under the non-degeneracy conditions given in Appendix A.4.

Update-Input alignment. We make an assumption on the alignment between block A update and input $\underline{Z}$ following the LoRA+ analysis [8]. To see this, for a single token trained with loss $\mathcal{L}$:

$$\nabla_A \mathcal{L}_t = \frac{\alpha}{r} B_t^\top d\bar{Z}_t \, \underline{Z}^\top,$$

so each row of the A-gradient is a scalar multiplication of the input. We therefore assume optimizer normalization preserves this input coupling:

$$|(U_A^t)_{i,:}\underline{Z}| = \Theta(n), \qquad i \in [r].$$

Sufficient conditions for SignSGD, Adam, and AdamW are given in Appendix A.4. However, this condition is not generally held in the minibatch setting, the measured minibatch alignment in Appendix A.4.6 and Figure 8 varies by module and training stage, so this condition is not asserted generically for minibatch fine-tuning.

Using these update equations, we can unroll A_t and B_t and separate the contribution of the random initialization A_0 (noise) from that of the accumulated optimizer updates in both LoRA factors (signal). The following lemma gives exact decompositions of both the LoRA feature and its one-step update. The alignment condition above is not needed for the decomposition itself and enters only in the subsequent scaling analysis.

Lemma 1 (Signal–noise decompositions of LoRA features)**.** *For any step $t \geq 1$, both the one-step LoRA feature update and the feature itself decompose into initialization-noise and optimizer-driven signal terms:*

$$\Delta Z_B^t = \underbrace{-\frac{\alpha\beta}{r}\eta_B U_B^t \tilde{A}_0 \underline{Z}}_{\Delta T_{noise}^t} + \underbrace{\frac{\alpha}{r}\eta_A\eta_B S_t \underline{Z}}_{\Delta T_{signal}^t},$$

$$Z_B^t = \underbrace{-\frac{\alpha\beta}{r}\eta_B\Big(\sum_{j=0}^{t-1} U_B^j\Big)\tilde{A}_0\underline{Z}}_{T_{noise}^t} + \underbrace{\frac{\alpha}{r}\eta_A\eta_B\Big(\sum_{j=0}^{t-1} U_B^j\Big)\Big(\sum_{i=0}^{t-1} U_A^i\Big)\underline{Z}}_{T_{signal}^t},$$

where $S_t := (\sum_{j=0}^{t-1} U_B^j)U_A^t + U_B^t(\sum_{i=0}^{t-1} U_A^i) + U_B^t U_A^t$.

Lemma 1 separates the LoRA feature Z_B^t into an initialization-dependent component T_{noise}^t and an optimizer-driven component T_{signal}^t. To compare their magnitudes across model widths and training time, we consider the width-normalized squared Euclidean norm $E_{\text{total}}^t := \frac{1}{n}\|Z_B^t\|_2^2$, which is the mean squared magnitude per output coordinate. We refer to this quantity as feature energy. It can be decomposed into three terms:

$$E_{\text{total}}^t = E_N^t + E_S^t + E_X^t.$$

$E_N^t := \frac{1}{n}\|T_{\text{noise}}^t\|_2^2$ is the noise energy, quantifying the contribution carried by the initial random projection $\tilde{A}_0\underline{Z}$. It isolates the component directly targeted by reducing the initialization variance. The signal energy, $E_S^t := \frac{1}{n}\|T_{\text{signal}}^t\|_2^2$, measures the contribution from the interaction of accumulated A- and B-updates. Comparing it with E_N^t reveals whether the optimizer-driven signal dominates the initialization noise. The cross term, $E_X^t := \frac{2}{n}\langle T_{\text{noise}}^t, T_{\text{signal}}^t\rangle$, accounts for reinforcement or cancellation between the two components. Unlike the component energies, it is signed and cannot generally be discarded, since the optimizer directions depend on the initialization.

We can give an upper and a lower bound for the feature energy: by Cauchy–Schwarz, $|E_X^t| \leq 2\sqrt{E_N^t E_S^t}$, giving that

$$(\sqrt{E_N^t} - \sqrt{E_S^t})^2 \leq E_{\text{total}}^t \leq (\sqrt{E_N^t} + \sqrt{E_S^t})^2.$$

These bounds show that when $E_N^t/E_S^t \to 0$, both the noise and cross term become negligible relative to the signal, so $E_{\text{total}}^t/E_S^t \to 1$. Figure 2 illustrates this signal dominance at finite width: under `Init[A²]`, the noise term is dominated by the signal in both Llama models, and the total feature norm closely tracks the signal norm. Measurements of those quantities are reported in Appendix D.6.

3.2 Robustness Analysis

We aim to keep the LoRA feature Z_B bounded as training progresses. To define this form of robustness, we first choose a scaling for the training horizon $T(n)$ as a function of model width n.

According to the neural scaling laws derived by Kaplan et al. [17], optimal training steps S scale very weakly with model size. Specifically, they find $S \propto C_{\min}^{0.03}$ and $N_{\text{param}} \propto C_{\min}^{0.73}$, for some fixed

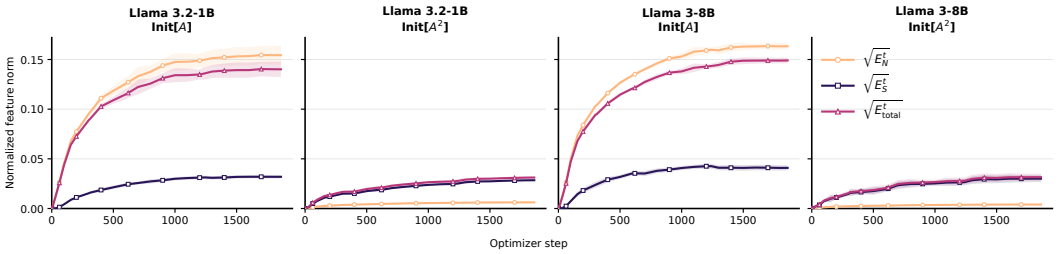


Figure 2: LoRA feature decomposition during fine-tuning of Llama 3.2-1B and Llama 3-8B. Curves show the RMS magnitudes $\sqrt{E_N^t}$ (noise), $\sqrt{E_S^t}$ (signal), and $\sqrt{E_{\text{total}}^t}$ (total). Standard `Init[A]` is noise-dominated, whereas under `Init[A²]` the noise is subdominant and the total norm closely follows the signal norm.

efficient computation budget $C_{\min}$, which implies $S \propto N_{\text{param}}^{0.04}$. Since the parameter count N_{param} scales quadratically with width ($N_{\text{param}} \propto n^2$), the training steps scale as $S \propto n^{0.08}$. Even under less optimal, fixed-compute regimes, training horizons grow sub-linearly. This motivates studying sub-$\sqrt{n}$ horizons, but we do not claim that the pretraining scaling will directly determine the fine-tuning schedules. Thus, we assume the training horizon satisfies $T(n) = o(\sqrt{n})$.

Definition 3 (Robustness of LoRA Dynamics)**.** *Let $T(n)$ be a time horizon scaling as $T(n) = o(\sqrt{n})$. The LoRA dynamics are robust up to horizon $T(n)$ if:*

$$\sup_{0 \le t \le T(n)} \frac{1}{n} \|Z_B^t\|_2^2 = O(1), \quad \text{as } n \to \infty.$$

This condition is similar to the stability condition in Condition 1. At a given step, a large β can make the initialization-dependent component dominate under matched contraction scales.

Theorem 1 (Component Scaling (Informal))**.** *Under the initialization and normalized updates above, fix $\underline{Z}$ with $\|\underline{Z}\|_2^2 = \Theta(n)$ and write $C_t := \sum_{s<t} U_B^s$, $u := \tilde{A}_0 \underline{Z}$, $v_t := (\sum_{s<t} U_A^s)\underline{Z}$, and $\rho_d(x) := \|x\|_2/\sqrt{d}$. For the analyzed steps $t \ge 2$, assume $\rho_r(u) = \Theta(\sigma_A\sqrt{n})$, $\rho_r(v_t) = \Theta(nt)$, $\rho_n(C_t u) = \Theta(tr^{\gamma_N}\rho_r(u))$, and $\rho_n(C_t v_t) = \Theta(tr^{\gamma_S}\rho_r(v_t))$, with fixed exponents γ_N, γ_S and constants uniform in n, r, t. Set $m_N := (\alpha/r)r^{\gamma_N}$ and $m_S := (\alpha/r)r^{\gamma_S}$. Then*

$$E_N^t = \Theta(\eta_B^2\beta^2 m_N^2 \sigma_A^2 n t^2), \qquad E_S^t = \Theta((\eta_A\eta_B m_S n t^2)^2).$$

The full statement and proof are in Appendix A.3.

Interpretation. The exponents γ_N, γ_S describe realized rank contraction for $C_t u$ and $C_t v_t$, respectively: $\gamma = 1$ is coherent, LLN-like scaling and $\gamma = 1/2$ is CLT-like scaling, without asserting either probabilistic limit law. The noise and signal need not share an exponent. The explicit factors of t are separate temporal-accumulation assumptions, not consequences of rank coherence or per-step alignment. The probe measurements in Figure 7 motivate model- and stage-dependent exponents but do not verify these accumulated contractions. At fixed rank and $\alpha/r = \Theta(1)$, both $m_N, m_S = \Theta(1)$, preserving the width comparison. When rank varies, the energy ratio also contains $r^{2(\gamma_N - \gamma_S)}$. These interactions motivate joint tuning of rank, multiplier, and LRs [18] and are consistent with Figure 5.

For fixed rank, $\alpha/r = \Theta(1)$, and finite time, practitioners often take $\eta_A\eta_B \ll n^{-1}$ to keep the signal small under the stated assumptions. Standard `Init[A]` nevertheless leaves a persistent initialization-dependent component because $\sigma_A^2 = \Theta(n^{-1})$:

$$E_{N,\texttt{Init[A]}}^t = \Theta(\eta_B^2\beta^2 m_N^2 t^2).$$

For fixed rank and fixed η_B, β, t, this noise does not vanish with width. At fixed α/r, its rank factor is $r^{2\gamma_N}$: r^2 for LLN-like contraction and r for CLT-like contraction. Because η_B multiplies both components, the LR for block B controls their common scale. We examine this by freezing block A and sweeping only η_B in Appendix D.2 and Figure 10.

Our proposed `Init[A²]` strategy sets $\sigma_A^2 = \Theta(n^{-2})$, giving

$$E_{N,\texttt{Init[A}^2\texttt{]}}^t = \Theta\left(\frac{\eta_B^2\beta^2 m_N^2 t^2}{n}\right).$$

Under matched projection, contraction, and temporal-accumulation scales, this gives an additional factor $1/n$ in noise energy while preserving the signal's scaling order; it does not assert identical optimizer trajectories. Nor does it imply a $1/n$ reduction in total energy: when $E_N^t/E_S^t \to 0$, the total becomes signal-dominated. For $\eta_B, \beta = \Theta(1)$, $m_N T(n) = o(\sqrt{n})$ suffices for noise to vanish uniformly on the analyzed window, reducing to $T(n) = o(\sqrt{n})$ at fixed rank. Robustness additionally requires $\eta_A \eta_B m_S n T(n)^2 = O(1)$ to control the signal. Our finite-width experiments test dominance and stability, not the asymptotic $1/n$ rate itself. Under the finite-time assumptions below, the extra factor $1/n$ makes the initialization noise subdominant to the signal. The same conditional scaling calculation permits stronger suppression with `Init[A^3]` and `Init[A^4]` (i.e., $\sigma_A^2 = 1/n^3$ or $1/n^4$), evaluated in panel (a) of Figure 11.

3.3 `Init[A^2]` Still Enables Feature Learning

Having established conditions for suppressing initialization noise and controlling feature energy, we next verify that `Init[A^2]` still supports nontrivial feature learning, which underlies HP transfer [10] and LoRA rank transfer to full fine-tuning [18]. A potential concern is that reducing σ_A^2 might suppress the signal required for adaptation. In the fixed-rank setting below, the accumulated optimizer updates maintain the signal while `Init[A^2]` suppresses initialization noise.

Theorem 2 (Finite-Time Stability and `Init[A^2]` Efficiency (Informal))**.** *In the fixed-rank setting of Theorem 1, let $t \geq 2$ be finite, $\beta, \alpha/r = \Theta(1)$, $\eta_A = \Theta(n^{-1})$, and $\eta_B = \Theta(1)$. Assume $C_t v_t$, $C_t U_A^t \underline{Z}$, and $U_B^t v_t$ have $\Theta(n)$ coordinate magnitudes. Then `Init[A^2]` makes the initialization-dependent terms $o(1)$ while preserving $\Theta(1)$ feature updates, and therefore satisfies the stability and efficiency conditions in Condition 1. Standard `Init[A]` guarantees bounded feature updates, but a non-degenerate lower bound additionally requires a non-cancellation condition.*

The proof is given in Appendix A.3. Under `Init[A^2]`, the initialization noise satisfies $E_N^t = \Theta(n^{-1})$, whereas the learned signal remains $E_S^t = \Theta(1)$. Hence the signal dominates and $E_{\text{total}}^t/E_S^t \to 1$. Under standard `Init[A]`, both components can remain $\Theta(1)$, so the total feature norm is bounded but may become small through cancellation. Thus `Init[A^2]` guarantees non-degenerate feature learning under the stated assumptions, whereas `Init[A]` requires an additional non-cancellation assumption.

3.4 Internal Stability and Bottleneck Dynamics

A recently proposed alternative, `Init[AB]` [16], seeks to improve robustness by initializing both matrices with $\Theta(n^{-1/2})$ scaling and employing symmetric LRs $\eta_A = \eta_B = \Theta(n^{-1/2})$. While this approach balances the global update norm, it inadvertently compromises the stability of the low-rank bottleneck: it fails to constrain the internal LoRA features Z_A to $\Theta(1)$, leading to feature explosion in the large-width limit.

For stable feature learning, it is crucial that not only the final output Z_B but also the intermediate representation Z_A evolves at a rate independent of the model width n. Consider the update to the internal feature ΔZ_A^t:

$$\Delta Z_A^t = (A_{t+1} - A_t)\underline{Z} = -\eta_A U_A^t \underline{Z}.$$

By the assumed alignment between the processed A-update and the input, $U_A^t \underline{Z}$ has $\Theta(n)$ coordinate size in the low-rank dimension. Consequently, the magnitude of the internal feature shift scales as:

$$\|\Delta Z_A^t\|_2 \propto \eta_A \cdot n.$$

Under the `Init[AB]` regime where $\eta_A = \Theta(n^{-1/2})$, the internal update scales as $\Theta(\sqrt{n})$, leading to feature updates that explode with model width. In contrast, both the standard `Init[A]` and our proposed `Init[A^2]` adopt the asymmetric scaling $\eta_A = \Theta(n^{-1})$. This choice ensures $\|\Delta Z_A^t\|_2 = \Theta(1)$, maintaining stable feature evolution within the interaction bottleneck. We postulate that this key difference, preserving internal stability, is essential for consistent performance across varying model scales, a hypothesis supported by the empirical analysis in Figure 1 and subsequent experiments. In Appendices E.1 and E.2, experimental results demonstrate that `Init[A^2]` outperforms `Init[AB]` in most configurations, even under the condition $\eta_A = \eta_B$. Moreover, `Init[AB]` exhibits notable performance degradation for large values of β.

The analysis above is deliberately stated through the signal-noise decomposition of the LoRA feature dynamics, because this decomposition isolates the initialization-dependent term that causes high-ratio instability. A complementary view is given in Appendix A.5, where the spectral condition

for feature learning [13] leads to the same `Init[A²]` scaling by controlling the worst-case spectral norm. Appendix A.6 explains why γ-calculus can miss aligned directions in which the random initialization is amplified, and why the spectral condition again points to the `Init[A²]` variance scaling $\sigma_A^2 = \Theta(n^{-2})$ as the choice that makes the initialization noise asymptotically subdominant.

4 Experiments

We evaluate whether `Init[A²]` improves HP transfer and robustness in two settings: a controlled toy MLP and LLM fine-tuning. The LLM experiments cover both encoder-only models (RoBERTa-base on GLUE) and decoder-only models (Llama arithmetic and commonsense benchmarks, with additional backbones in the appendix). Across experiments, we vary the initialization scale β, LR ratio η_B/η_A, adapter rank r, and LoRA multiplier α/r; `Init[A]` uses base variance $\sigma_A^2 = 1/n$ and `Init[A²]` uses $\sigma_A^2 = 1/n^2$, both with $B_0 = 0$. Full training details are in Appendix C.

4.1 Toy Example

Setup. To test HP transfer under `Init[A²]`, we consider a simplified MLP equipped with LoRA ($\alpha/r = 1$) on MNIST [19]:

$$Y = W_{\text{out}}\mathcal{F}((W_0 + BA), \mathcal{F}(W_{\text{in}}, X)),$$

where $W_{\text{in}} \in \mathbb{R}^{n\times d}$, $W_{\text{out}} \in \mathbb{R}^{10\times n}$, $W_0 \in \mathbb{R}^{n\times n}$, $A \in \mathbb{R}^{r\times n}$, and $B \in \mathbb{R}^{n\times r}$. Here $\mathcal{F}(W, z) = \text{ReLU}(Wz)$. We fix $W_0 = 0$ and freeze W_{in} and W_{out} after the μP initializations $(W_{\text{out}})_{ij} \sim \mathcal{N}(0, n^{-2})$ and $(W_{\text{in}})_{ij} \sim \mathcal{N}(0, 1)$ [10]. To isolate the clean non-EMA normalized-update regime from our theory, we disable optimizer momentum and moving averages. We compare `Init[A]` and `Init[A²]` across $n \in \{16, 64, 256, 1024, 4096\}$ with a fixed LR ratio $\eta_B = n\eta_A$ and $\eta_A \in \{2^{-7}, \cdots, 2^0\}$. For each width, we report the best validation loss over the LR grid, averaged over three seeds, and track the squared norms of the internal feature Z_A and LoRA feature Z_B to assess stability.

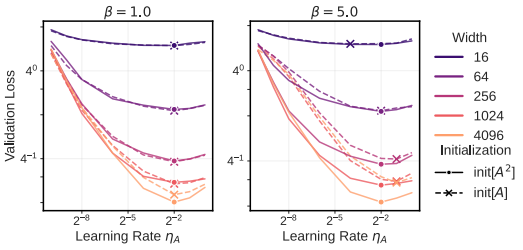


Figure 3: HP transfer on the MNIST toy example, measured by validation loss across widths and LRs.

Results. Figure 3 illustrates the HP transfer capabilities. At a low initialization scale ($\beta = 1.0$), both methods exhibit consistent optimal LRs across widths. However, at a larger scale ($\beta = 5.0$), the optimal LR for `Init[A]` drifts significantly at the largest width ($n = 4096$), and convergence degrades. In contrast, `Init[A²]` maintains stable HP transfer and achieves consistently lower validation loss, with the performance gap widening as β increases. This stability is further corroborated by the feature norm evolution (Figure 1), where `Init[A²]` preserves the stability of Z_A while the evolution of Z_B is not affected by β.

4.2 LLM Experiments

Setup. To validate the effectiveness of our theory for real LLMs, we evaluate both encoder-only and decoder-only LLMs. We use RoBERTa on GLUE for encoder-only results, deferred to Appendix D.1. For decoder-only generation, we fine-tune Llama 3-8B and Llama 3.2-1B [20] on arithmetic and commonsense benchmarks [21], following the evaluation by Li et al. [16]: 10k arithmetic/math examples and 170k commonsense examples. For Llama 3-8B, we set $r = \alpha = 8$ and sweep $\beta \in \{0.25, 0.5, 1, 2, 4, 8\}$ and $\rho = \eta_B/\eta_A \in \{1, 10, 200, 500, 1000, 2000, 4000\}$, fixing $\eta_A = 10^{-5}$ and $\eta_B = \rho\eta_A$. For Llama 3.2-1B, we fix $\beta = 1$, sweep $r \in \{4, 8, 16, 32, 64, 128\}$, and set $\alpha = r$. The loss study trains one epoch on math while varying r, β, equal LRs, and LR ratios.

Results. Figure 4 shows that `Init[A²]` is much less sensitive to β and rank than standard `Init[A]`: it keeps broad high-accuracy regions across LR ratios, while `Init[A]` drifts toward smaller ratios and collapses at high ratios as β or r increases. The training-loss sweep in Figure 5 is consistent with the component decomposition. Here, β appears explicitly only in the initialization term; the signal can still depend on β indirectly through the optimizer trajectory. Empirically, the equal-LR optimum and best loss remain nearly unchanged across β under `Init[A²]`, consistent with suppression of

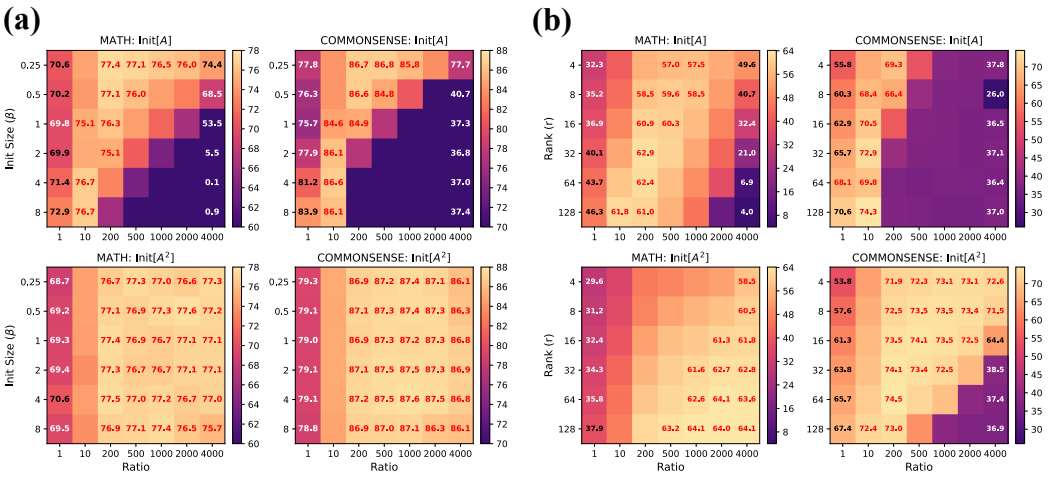


Figure 4: Llama generation results. (a) Llama 3-8B with varying initialization scale β. (b) Llama 3.2-1B with varying rank r. Each panel reports arithmetic and commonsense accuracy under standard `Init[A]` and `Init[A²]` while sweeping the LR ratio $\rho = \eta_B/\eta_A$, implemented by fixing $\eta_A = 10^{-5}$ and setting $\eta_B = \rho\eta_A$. Numerical labels mark endpoint ratios and near-optimal settings.

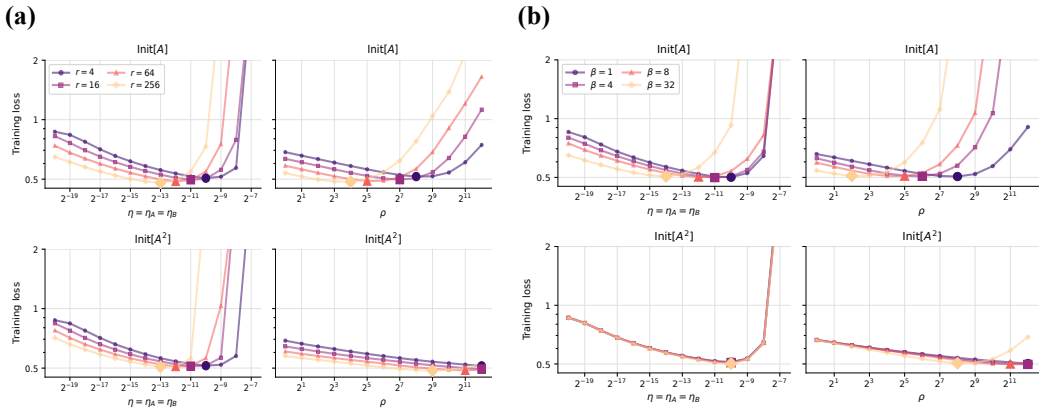


Figure 5: Final training loss reached during one epoch on the math benchmark. The LoRA multiplier is fixed to $\alpha/r = 1$. (a) varies rank r, and (b) varies initialization scale β. The figure includes equal-LR sweeps with $\eta = \eta_A = \eta_B$ and LR-ratio sweeps; in the ratio panels, $\eta_A = 10^{-5}$ and $\eta_B = \rho\eta_A$ for $\rho = \eta_B/\eta_A$.

initialization noise. However, because r appears in both the signal and noise terms, the optimum LR also shifts for `Init[A²]` when sweeping r. Standard `Init[A]` does not suppress the noise, so its low-loss region narrows or shifts. In the ratio sweep, some drift can still appear because very large η_B can introduce its own stability issue, but `Init[A²]` remains broader and more robust. The same qualitative pattern holds for encoder-only RoBERTa and for additional decoder-only Llama, Qwen, and Mistral backbones in Appendix D.1 and Figure 12.

4.3 `Init[A²]` Enables Multiplier Tuning

Setup. In both the analysis and experiments, we set the multiplier $\alpha/r = \Theta(1)$. When this multiplier increases, it can also amplify initialization noise. This explains why many LoRA studies use relatively modest multipliers, often with $\alpha/r \leq 2$ [1, 22, 23]. However, a larger multiplier gives the learned low-rank update more influence on the model output and may therefore increase adaptability on hard decoder-only fine-tuning tasks. Our analysis suggests that the obstacle is not only stronger adaptation: α/r also amplifies the same initialization-dependent noise as β (Lemma 1). If this random-initialization effect is suppressed, larger multipliers need not be intrinsically unstable and may further improve performance. We therefore fix $\beta = 0.5$, $\eta_A = 10^{-5}$, and $\eta_B = 2 \times 10^{-3}$ on Llama 3-8B arithmetic fine-tuning, then vary α/r.

Results. Figure 6 reveals a sharp stability gap. `Init[A]` peaks at $\alpha/r = 3$ (76.4%) and drops to 72.0% at $\alpha/r = 12$, whereas `Init[A²]` improves up to $\alpha/r = 10$ (77.5%) and remains stable at $\alpha/r = 12$ (77.3%). This matches our analysis: larger α/r amplifies both signal and noise, but `Init[A²]` suppresses the initialization-dependent noise enough to exploit stronger low-rank updates without destabilizing the update from the signal term. Thus, `Init[A²]` makes tuning the multiplier α/r a viable way to further improve arithmetic fine-tuning performance. Five-seed multiplier validation, matched LR retuning, and longer-horizon comparisons are in Appendices D.4 and D.5.

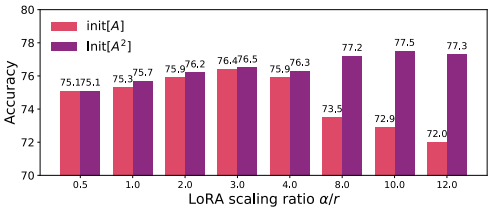


Figure 6: Arithmetic accuracy as the LoRA multiplier α/r increases. `Init[A²]` remains stable at larger multipliers, while `Init[A]` degrades beyond $\alpha/r = 3$.

5 Conclusion and Limitations

We propose `Init[A²]`, a simple LoRA initialization that suppresses initialization-induced noise while preserving zero initial adapter output. The central mechanism is that standard `Init[A]` leaves a random initial projection in A that is amplified as soon as B moves; `Init[A²]` reduces this term by one additional factor of width, making high-ratio asymmetric LRs much less brittle. Practically, this reduces best-ratio drift across initialization scales, lowers dependence on exhaustive HP sweeps, and improves robustness to large update multipliers (α/r), enabling stronger updates and better performance on downstream tasks. These gains are consistent across the controlled HP-transfer experiment, encoder-only RoBERTa fine-tuning, and decoder-only LLM generation benchmarks.

More broadly, the design philosophy only targets the scale of A_0 rather than its distribution. It can therefore be applied to structured initializers: we compress the A_0 scale of Orthogonal and EVA while retaining their selected directions, and use a factorwise gauge transformation for PiSSA that preserves its initialized low-rank product. In selected Llama arithmetic sweeps, these three variance-compressed variants remain stable under aggressive asymmetric LRs for which their uncompressed counterparts degrade (Appendix E.3). This extension is empirical: the $B_0 = 0$ theorem does not directly cover PiSSA, and we do not claim uniform gains over carefully tuned base initializers.

Several limitations remain. Although some configurations are replicated across three to five seeds, most LR-ratio sweeps are single runs because of computational constraints, so small score differences in those sweeps should not be overinterpreted. Our empirical evaluation focuses on supervised fine-tuning of LLMs; extending the study to reinforcement-learning fine-tuning, broader generative-model training, and other adaptation settings is an important next step. Although `Init[A²]` makes asymmetric LRs more robust and our analysis identifies factors that can shift the preferred ratio, including β, r, and α/r, we do not provide a direct rule for setting η_B/η_A across arbitrary models and tasks. Developing such model-aware prescriptions, a fuller theory for rank r and other LoRA HP interactions, and a better understanding of how far initialization scaling can be pushed beyond `Init[A²]` are useful future directions. The limits of initialization scaling may connect to recent observations on the role of initialization in emergent reasoning behavior [24].

Acknowledgments and Disclosure of Funding

Disen Liao is supported by the Queen Elizabeth II Graduate Scholarship in Science and Technology (QEII-GSST). This work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canada CIFAR AI Chairs program and the Ontario Early Researcher program. Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute. We thank Mufan Li for very helpful discussions. The authors declare no competing interests.

References

[1] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

[2] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, pages 2790–2799, 2019.

[3] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[4] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[5] Yuchen Zeng and Kangwook Lee. The expressive power of low-rank adaptation. In *International Conference on Learning Representations*, 2023.

[6] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023.

[7] Yu-Ang Lee, Ching-Yun Ko, Pin-Yu Chen, and Mi-Yen Yeh. Learning rate matters: Vanilla LoRA may suffice for LLM fine-tuning. *arXiv preprint arXiv:2602.04998*, 2026.

[8] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low-rank adaptation of large models. *arXiv preprint arXiv:2402.12354*, 2024.

[9] Greg Yang and Edward J Hu. Tensor programs iv: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pages 11727–11737. PMLR, 2021.

[10] Greg Yang, Edward J Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv preprint arXiv:2203.03466*, 2022.

[11] Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: Completep enables compute-efficient deep transformers. *arXiv preprint arXiv:2505.01618*, 2025.

[12] Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. *arXiv preprint arXiv:2309.16620*, 2023.

[13] Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Feature learning in infinite-depth neural networks. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

[14] Charlie Blake, Constantin Eichenberg, Josef Dean, Lukas Balles, Luke Yuri Prince, Björn Deiseroth, Andres Felipe Cruz-Salinas, Carlo Luschi, Samuel Weinbach, and Douglas Orr. u-μp: The unit-scaled maximal update parametrization. In *International Conference on Learning Representations*, 2025.

[15] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. The impact of initialization on LoRA fine-tuning dynamics. *Advances in Neural Information Processing Systems*, 37:117015–117040, 2024.

[16] Shiwei Li, Xiandi Luo, Xing Tang, Haozhao Wang, Hao Chen, Yuhua Li, Ruixuan Li, et al. Beyond zero initialization: Investigating the impact of nonzero initialization on LoRA fine-tuning dynamics. In *International Conference on Machine Learning*, 2025.

[17] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[18] Nan Chen, Soledad Villar, and Soufiane Hayou. Learning rate scaling across LoRA ranks and transfer to full finetuning. *arXiv preprint arXiv:2602.06204*, 2026. doi: 10.48550/arXiv.2602.06204.

[19] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[20] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

[21] Zhiqiang Hu, Lei Wang, Yihuai Lan, Wanyu Xu, Ee-Peng Lim, Lidong Bing, Xing Xu, Soujanya Poria, and Roy Lee. LLM-Adapters: An adapter family for parameter-efficient fine-tuning of large language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 5254–5276, 2023.

[22] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[23] Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P. Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. URL `https://openreview.net/forum?id=aloEru2qCG`.

[24] Zhongwang Zhang, Pengxiao Lin, Zhiwei Wang, Yaoyu Zhang, and Zhi-Qin John Xu. Initialization is critical to whether transformer fits composite function by inference or memorizing. *arXiv preprint arXiv:2405.05409*, 2024.

[25] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, pages 404–413. PMLR, 2018.

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

[27] Greg Yang, James B Simon, and Jeremy Bernstein. A spectral condition for feature learning. *arXiv preprint arXiv:2310.17813*, 2023.

[28] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[29] Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. LoFTQ: LoRA-fine-tuning-aware quantization for large language models. *arXiv preprint arXiv:2310.08659*, 2023.

[30] Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. *Advances in Neural Information Processing Systems*, 37:121038–121072, 2024.

[31] Fabian Paischer, Lukas Hauzenberger, Thomas Schmied, Benedikt Alkin, Marc Peter Deisenroth, and Sepp Hochreiter. Parameter efficient fine-tuning via explained variance adaptation. *arXiv preprint arXiv:2410.07170*, 2024.

[32] Yibo Yang, Xiaojie Li, Zhongzhu Zhou, Shuaiwen Song, Jianlong Wu, Liqiang Nie, and Bernard Ghanem. CorDA: Context-oriented decomposition adaptation of large language models for task-aware parameter-efficient fine-tuning. *Advances in Neural Information Processing Systems*, 37:71768–71791, 2024.

[33] Kerim Büyükakyüz. OLoRA: Orthonormal low-rank adaptation of large language models. *arXiv preprint arXiv:2406.01775*, 2024.

[34] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[35] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

[36] Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

[37] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

[38] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

[39] Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020.

[40] Edward Milsom, Ben Anson, and Laurence Aitchison. Function-space learning rates. *arXiv preprint arXiv:2502.17405*, 2025.

[41] Tim Large, Yang Liu, Minyoung Huh, Hyojin Bahng, Phillip Isola, and Jeremy Bernstein. Scalable optimization in the modular norm. *Advances in Neural Information Processing Systems*, 37:73501–73548, 2024.

[42] Artem Riabinin, Egor Shulgin, Kaja Gruntkowska, and Peter Richtárik. Gluon: Making muon & scion great again!(bridging theory and practice of lmo-based optimizers for llms). *arXiv preprint arXiv:2505.13416*, 2025.

[43] Moritz Haas, Jin Xu, Volkan Cevher, and Leena Chennuru Vankadara. $\mu\mathrm{P}^2$: Effective sharpness aware minimization requires layerwise perturbation scaling. *Advances in Neural Information Processing Systems*, 37:38888–38959, 2024.

[44] Nolan Dey, Shane Bergsma, and Joel Hestness. Sparse maximal update parameterization: A holistic approach to sparse training dynamics. *Advances in Neural Information Processing Systems*, 37:33836–33862, 2024.

[45] Leena Chennuru Vankadara, Jin Xu, Moritz Haas, and Volkan Cevher. On feature learning in structured state space models. *Advances in Neural Information Processing*, 2024.

[46] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

[47] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

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

[49] Tian Xie, Haoming Luo, Haoyu Tang, Yiwen Hu, Jason Klein Liu, Qingnan Ren, Yang Wang, Wayne Xin Zhao, Rui Yan, Bing Su, et al. Controlled llm training on spectral sphere. *arXiv preprint arXiv:2601.08393*, 2026.

[50] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[51] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

[52] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR, 2020.

[53] Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Liang Zhao, et al. mhc: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015. doi: 10.1109/ICCV.2015.123.

[55] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[56] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[57] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355, 2018.

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

[59] Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[60] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[61] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Appendix Table of Contents

A. Theoretical Analysis and Proofs
- A.1 Two Scaling Laws
- A.2 γ-Calculus
- A.3 Update Decomposition and Noise-Dominance Theorem
- A.4 Adam and AdamW as Normalized LoRA Updates
- A.5 Spectral Derivation of `Init[A`^{2}`]`
- A.6 Spectral Conditions for Robust LoRA Feature Learning

B. Related Literature Review
- B.1 LoRA Initialization
- B.2 Other LoRA Hyperparameters
- B.3 Feature Learning and HP Transfer
- B.4 Training Stability

C. Experimental Setup
- C.1 Initialization Implementation
- C.2 Toy Example
- C.3 Encoder-Only Model Setup
- C.4 Decoder-Only Model Setup

D. Additional Results and Ablations
- D.1 LR Sweep Experiment
- D.2 Sensitivity to the B LR
- D.3 Additional Llama Experiments
- D.4 Long-Horizon Arithmetic Fine-Tuning
- D.5 Multi-Seed Accuracy and Tuning Robustness
- D.6 Signed Signal–Noise Energy Measurements

E. Comparisons with Other LoRA Initializations
- E.1 Comparison with `Init[AB]`
- E.2 Equal-LR Sweep on Math Benchmark
- E.3 Variance-Compressed Structured Initializers under Asymmetric LRs

F. Responsible Research Statements
- F.1 Broader Impact
- F.2 LLM Usage

A Theoretical Analysis and Proofs

A.1 Two Scaling Laws

When working with γ-calculus, we explicitly distinguish between Law of Large Numbers (LLN) and Central Limit Theorem (CLT) scaling. These two scaling regimes are fundamental in maximal-update-parameterization theory [9], which enables HP transfer [10]. Here we give the two basic statistical rules and explain why they matter.

Law of Large Numbers (LLN). If $X_1, X_2, \cdots, X_n$ are i.i.d. random variables with finite expected value $\mathbb{E}[X_i] = \mu$, then

$$\frac{1}{n}\sum_{i=1}^{n} X_i \xrightarrow{\text{a.s.}} \mu, \quad \text{as } n \to \infty.$$

Central Limit Theorem (CLT). If $X_1, X_2, \cdots, X_n$ are i.i.d. random variables with finite expected value $\mathbb{E}[X_i] = \mu$ and finite variance $\mathrm{Var}(X_i) = \sigma^2$, then

$$\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(X_i - \mu) \xrightarrow{d} \mathcal{N}(0, \sigma^2), \quad \text{as } n \to \infty.$$

These two statistical laws reveal two scenarios regarding the size of a sum of X_i when n is large:

$$\sum_{i=1}^{n} X_i \approx \underbrace{n\mu}_{\text{LLN}} + \underbrace{\sqrt{n}\sigma\mathcal{N}(0,1)}_{\text{CLT}}.$$

For $\mu \neq 0$, $|\sum_{i=1}^{n} X_i|/n \to |\mu|$ almost surely. For $\mu = 0$ and $0 < \sigma^2 < \infty$, the sum has RMS magnitude $\sqrt{\mathbb{E}[(\sum_{i=1}^{n} X_i)^2]} = \sigma\sqrt{n}$, but no deterministic lower bound follows.

A.2 γ-Calculus

In the scaling-limit theory of deep learning, we want to track the asymptotic behavior of key quantities such as LRs or initialization as we scale model ingredients such as width or depth. Prior work [8, 15, 16] uses γ-calculus to derive LoRA scaling rules with respect to the model width n.

If a variable v has asymptotically polynomial magnitude in n, i.e., $|v| = \Theta(n^{\lambda})$, we write $\lambda = \gamma[v]$. Its exponent arithmetic follows these rules:

- **Zero:** $\gamma[0] = -\infty$, corresponding to the limit $\gamma[n^{\lambda}]$ when $\lambda \to -\infty$.
- **Product:** For two real-valued variables $v, v' \in \mathbb{R}$, we have $\gamma[vv'] = \gamma[v] + \gamma[v']$.
- **Addition:** Whenever the exponents exist, $\gamma[v + v'] \leq \max(\gamma[v], \gamma[v'])$. Equality holds if the exponents differ or $|v + v'| = \Theta(|v| + |v'|)$; excluding exact cancellation alone is insufficient.

A.3 Update Decomposition and Noise-Dominance Theorem

We first show that the key decomposition only requires width-normalized update directions, not the exact SignSGD form.

Lemma 1 (Update and feature decompositions)**.** *For any step $t \geq 1$, both the one-step LoRA feature update and the LoRA feature itself decompose into noise terms depending on the initialization scale β and signal terms depending on the LRs:*

$$\Delta Z_B^t = \underbrace{-\frac{\alpha\beta}{r}\,\eta_B\, U_B^t \tilde{A}_0 \underline{Z}}_{\Delta T^t_{noise}} + \underbrace{\frac{\alpha}{r}\,\eta_A\eta_B\, S_t\, \underline{Z}}_{\Delta T^t_{signal}},$$

$$Z_B^t = \underbrace{-\frac{\alpha\beta}{r}\eta_B\Big(\sum_{j=0}^{t-1} U_B^j\Big)\tilde{A}_0\underline{Z}}_{T^t_{noise}} + \underbrace{\frac{\alpha}{r}\eta_A\eta_B\Big(\sum_{j=0}^{t-1} U_B^j\Big)\Big(\sum_{i=0}^{t-1} U_A^i\Big)\underline{Z}}_{T^t_{signal}},$$

where

$$S_t := \Big(\sum_{j=0}^{t-1} U_B^j\Big)U_A^t \;+\; U_B^t\Big(\sum_{i=0}^{t-1} U_A^i\Big) \;+\; U_B^t U_A^t.$$

Proof. Let

$$\Delta A_t := A_{t+1} - A_t = -\eta_A U_A^t, \qquad \Delta B_t := B_{t+1} - B_t = -\eta_B U_B^t.$$

Repeatedly applying the update recursion gives

$$A_t = A_0 - \eta_A \sum_{i=0}^{t-1} U_A^i, \qquad B_t = B_0 - \eta_B \sum_{j=0}^{t-1} U_B^j = -\eta_B \sum_{j=0}^{t-1} U_B^j,$$

where the last equality uses $B_0 = 0$.

For the one-step update, expand

$$B_{t+1}A_{t+1} = (B_t + \Delta B_t)(A_t + \Delta A_t) = B_tA_t + B_t\Delta A_t + \Delta B_t A_t + \Delta B_t \Delta A_t.$$

Therefore,

$$\frac{r}{\alpha}(Z_B^{t+1} - Z_B^t) = \underbrace{B_t \Delta Z_A^t}_{\delta_1^t} + \underbrace{\Delta B_t Z_A^t}_{\delta_2^t} + \underbrace{\Delta B_t \Delta Z_A^t}_{\delta_3^t}, \tag{3}$$

since $Z_A^t = A_t\underline{Z}$ and $\Delta Z_A^t = \Delta A_t \underline{Z}$. Substitution into the three terms gives

$$B_t \Delta A_t \underline{Z} = \Big(-\eta_B \sum_{j=0}^{t-1} U_B^j\Big)\Big(-\eta_A U_A^t\Big)\underline{Z} = \eta_A\eta_B\Big(\sum_{j=0}^{t-1} U_B^j\Big)U_A^t\underline{Z}, \tag{4}$$

$$\Delta B_t A_t \underline{Z} = \Big(-\eta_B U_B^t\Big)\Big(A_0 - \eta_A \sum_{i=0}^{t-1} U_A^i\Big)\underline{Z} = -\eta_B U_B^t A_0 \underline{Z} + \eta_A\eta_B\, U_B^t\Big(\sum_{i=0}^{t-1} U_A^i\Big)\underline{Z}, \tag{5}$$

$$\Delta B_t \Delta A_t \underline{Z} = \Big(-\eta_B U_B^t\Big)\Big(-\eta_A U_A^t\Big)\underline{Z} = \eta_A\eta_B\, U_B^t U_A^t \underline{Z}. \tag{6}$$

Summing these identities in Equation (3), defining S_t as above, and substituting $A_0 = \beta\tilde{A}_0$ yields the stated decomposition of ΔZ_B^t.

For the feature itself, use the same closed forms:

$$Z_B^t = \frac{\alpha}{r}\left(-\eta_B \sum_{j=0}^{t-1} U_B^j\right)\left(\beta\tilde{A}_0 - \eta_A \sum_{i=0}^{t-1} U_A^i\right)\underline{Z}.$$

Distributing the two terms and rearranging scalars gives the stated decomposition of Z_B^t. □

Assumption 1 (Width-normalized update directions)**.** *For all training steps $t \le T(n)$, the optimizer directions satisfy*

$$\|U_A^t\|_{\max} = O(1), \qquad \|U_B^t\|_{\max} = O(1).$$

On the analyzed steps after the initial B update, the down-projection update is non-degenerate: for each LoRA row $i \in [r]$, $|(U_A^t)_{i,:}\underline{Z}| = \Theta(n)$, with constants independent of width. This magnitude condition alone does not rule out cancellation between steps.

Under Assumption 1, the component-scaling argument applies to any optimizer in this normalized class. For SignSGD, the coordinate bound is immediate and the fixed-input LoRA A-gradient has the sign-aligned outer-product structure used by Hayou et al. [8]. Appendix A.4 states the additional local conditions under which Adam and AdamW verify the same two optimizer-facing requirements.

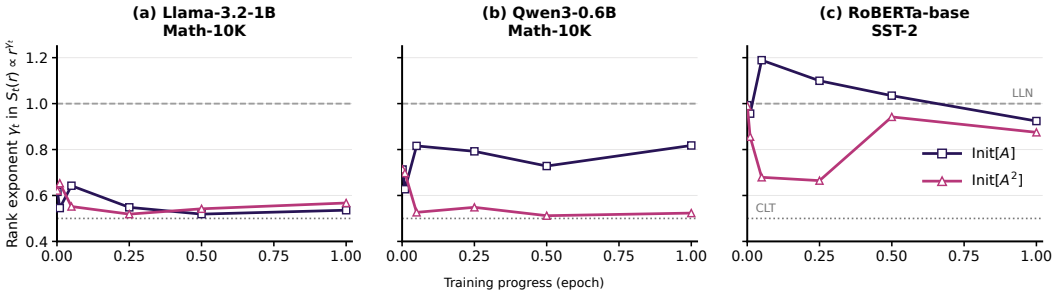


Figure 7: Empirical rank-coherence exponent during Llama-3.2-1B and Qwen3-0.6B fine-tuning on Math-10K, and RoBERTa-base fine-tuning on SST-2. The fitted exponent in $S_t(r) \propto r^{\gamma_t}$ varies with model, initialization, and training stage. Lines mark CLT-like $(1/2)$ and LLN-like (1) rank scaling.

Separate rank and temporal assumptions. Write $\lambda_r := \alpha/r$ and $\rho_d(x) := \|x\|_2/\sqrt{d}$ for the RMS magnitude of a d-vector. For the accumulated directions, define

$$C_t := \sum_{s=0}^{t-1} U_B^s, \qquad u := \tilde{A}_0 \underline{Z}, \qquad v_t := \Big(\sum_{s=0}^{t-1} U_A^s\Big)\underline{Z}.$$

The two vectors entering C_t depend differently on initialization and training. We therefore assign distinct rank exponents γ_N, γ_S and effective multipliers $m_N := \lambda_r r^{\gamma_N}$, $m_S := \lambda_r r^{\gamma_S}$. The values 1 and $1/2$ describe coherent, LLN-like and CLT-like rank scaling, respectively; these are scaling benchmarks, not assertions of a probabilistic limit law. In particular, mean-zero rank terms need not have CLT scaling when they are dependent.

The theorem below assumes linear temporal accumulation separately from the rank exponents. Neither coordinate boundedness nor per-step A–input alignment implies these accumulated lower bounds: signs may cancel across steps. The assumptions are imposed on the realized contractions of u and v_t, allowing optimizer directions to depend on A_0.

Empirical rank coherence. The LLN choice is a conditional benchmark, rather than a universal property of LoRA training. To diagnose rank summation, let $u_t = A_t \underline{Z} \in \mathbb{R}^r$ be the feature on a fixed probe token and $a_t = \|u_t\|_2/\sqrt{r}$ its RMS magnitude. For a sampled output row k, define the normalized contraction magnitude $S_{t,k}(r) := |(U_B^t u_t)_k|/a_t$, with U_B^t taken before the LR and weight decay. After aggregating sampled contractions, fitting $S_t(r) \propto r^{\gamma_t}$ distinguishes CLT-like $(\gamma_t = 1/2)$ from coherent, LLN-like $(\gamma_t = 1)$ behavior. The sweep uses $r \in \{4, 8, 16, 32, 64, 128\}$, $\eta_B/\eta_A \in \{200, 1000, 2000\}$, both initializations, and three seeds per configuration. We fit the exponent using both median and mean aggregation of sampled contractions.

Table 1: Rank-coherence exponents under median / mean aggregation of the normalized contraction. Both aggregation choices show that the exponent depends on model and training stage.

Model	Checkpoint	`Init[A]` median / mean γ	`Init[A²]` median / mean γ
Llama-3.2-1B	First update	0.616 / 0.652	0.621 / 0.655
Llama-3.2-1B	5% epoch	0.642 / 0.665	0.552 / 0.565
Llama-3.2-1B	50% epoch	0.519 / 0.559	0.542 / 0.560
Llama-3.2-1B	End epoch	0.536 / 0.572	0.567 / 0.581
Qwen3-0.6B	First update	0.713 / 0.778	0.714 / 0.779
Qwen3-0.6B	End epoch	0.818 / 0.819	0.523 / 0.533
RoBERTa/SST-2	First update	0.993 / 0.954	0.990 / 0.951
RoBERTa/SST-2	End epoch	0.924 / 0.958	0.875 / 0.877

Figure 7 shows that Llama is largely CLT-like, Qwen depends on the initialization and training stage, and RoBERTa is closer to coherent summation. This motivates exponents appropriate to the model, module, and training window. The probe measures $U_B^t A_t \underline{Z}$, whereas the theorem concerns

$C_t u$ and $C_t v_t$; it therefore does not establish either accumulated exponent or their equality. The theorem allows $\gamma_N \neq \gamma_S$. A shared multiplier $m_{r,\gamma} := \lambda_r r^\gamma$ is recovered only if both accumulated contractions satisfy the same exponent. At fixed rank and $\lambda_r = \Theta(1)$, both multipliers are $\Theta(1)$, preserving the extra $1/n$ noise comparison under matched assumptions.

The optimizer directions are realized quantities and may depend on A_0 through the training trajectory. We therefore do not condition on these directions and then treat $\tilde{A}_0$ as independent. Instead, the following deterministic identity controls the cross term for every realized trajectory.

Lemma 2 (Exact realized energy decomposition)**.** *Define*

$$E_N^t := \frac{1}{n}\|T_{noise}^t\|_2^2, \qquad E_S^t := \frac{1}{n}\|T_{signal}^t\|_2^2, \qquad E_X^t := \frac{2}{n}\langle T_{noise}^t, T_{signal}^t\rangle,$$

and let $E_{total}^t := n^{-1}\|Z_B^t\|_2^2$. Then

$$E_{total}^t = E_N^t + E_S^t + E_X^t, \qquad |E_X^t| \le 2\sqrt{E_N^t E_S^t},$$

and

$$(\sqrt{E_N^t} - \sqrt{E_S^t})^2 \le E_{total}^t \le (\sqrt{E_N^t} + \sqrt{E_S^t})^2.$$

Consequently, $E_{total}^t/E_N^t \to 1$ whenever $E_S^t/E_N^t \to 0$, and $E_{total}^t/E_S^t \to 1$ whenever $E_N^t/E_S^t \to 0$.

Proof. Expanding $n^{-1}\|T_{\text{noise}}^t + T_{\text{signal}}^t\|_2^2$ gives the identity. Cauchy–Schwarz gives $|E_X^t| \le 2\sqrt{E_N^t E_S^t}$, and substituting its two extremal signs gives the displayed bounds. For example, if $q_t := \sqrt{E_S^t/E_N^t} \to 0$, division by E_N^t yields $(1-q_t)^2 \le E_{\text{total}}^t/E_N^t \le (1+q_t)^2$; the other case is symmetric. □

We now give the full conditional statement and proof of Theorem 1.

Theorem 1 (Component scaling with general rank contractions)**.** *Fix $\underline{Z} \in \mathbb{R}^n$ with $\|\underline{Z}\|_2^2 = \Theta(n)$ and positive $\alpha, \beta, \eta_A, \eta_B$. Let $B_0 = 0$, $A_0 = \beta\tilde{A}_0$, with independent $(\tilde{A}_0)_{ij} \sim \mathcal{N}(0, \sigma_A^2)$, and updates $A_{s+1} = A_s - \eta_A U_A^s$, $B_{s+1} = B_s - \eta_B U_B^s$. Define C_t, u, v_t, ρ_d as above. On a set of analyzed steps $t \ge 2$, suppose the realized trajectories satisfy*

$$\rho_r(u) = \Theta(\sigma_A\sqrt{n}), \qquad \rho_r(v_t) = \Theta(nt), \tag{7}$$

$$\rho_n(C_t u) = \Theta(tr^{\gamma_N}\rho_r(u)), \qquad \rho_n(C_t v_t) = \Theta(tr^{\gamma_S}\rho_r(v_t)), \tag{8}$$

for fixed exponents γ_N, γ_S, with constants uniform in n, r and the analyzed steps. Set $m_N = (\alpha/r)r^{\gamma_N}$ and $m_S = (\alpha/r)r^{\gamma_S}$. Then

$$E_N^t = \Theta(\eta_B^2\beta^2 m_N^2\sigma_A^2 nt^2), \qquad E_S^t = \Theta((\eta_A\eta_B m_S nt^2)^2).$$

The total energy satisfies

$$(\sqrt{E_N^t} - \sqrt{E_S^t})^2 \le E_{\text{total}}^t \le (\sqrt{E_N^t} + \sqrt{E_S^t})^2$$

for every realized trajectory, so its ratio to the larger component tends to one whenever the smaller-to-larger energy ratio tends to zero. No independence between initialization and optimizer directions is required.

Proof. The unrolled updates give $B_t = -\eta_B C_t$ and $A_t\underline{Z} = \beta u - \eta_A v_t$. Thus the feature $Z_B^t = (\alpha/r)B_t A_t\underline{Z}$ has the exact components

$$T_{\text{noise}}^t = -\lambda_r\beta\eta_B C_t u, \qquad T_{\text{signal}}^t = \lambda_r\eta_A\eta_B C_t v_t.$$

Using Equations (7) and (8) yields

$$\sqrt{E_N^t} = \lambda_r\beta\eta_B\rho_n(C_t u) = \Theta(\eta_B\beta m_N\sigma_A\sqrt{n}\,t),$$

$$\sqrt{E_S^t} = \lambda_r\eta_A\eta_B\rho_n(C_t v_t) = \Theta(\eta_A\eta_B m_S nt^2).$$

Squaring gives the RMS component estimates without requiring equal output-coordinate magnitudes. The deterministic bounds and dominance conclusions follow from Lemma 2. □

Scope of the assumptions. For fixed $\underline{Z}$, Gaussian initialization gives independent $u_i \sim \mathcal{N}(0, \sigma_A^2 \|\underline{Z}\|_2^2)$ and $\mathbb{E}[\rho_r(u)^2] = \sigma_A^2 \|\underline{Z}\|_2^2$. This motivates Equation (7) but supplies no realized lower bound. The assumption $\rho_r(v_t) = \Theta(nt)$ requires temporal non-cancellation beyond per-step alignment, while the factor t in Equation (8) separately requires B-accumulation. We analyze only steps satisfying these bounds. With $B_0 = 0$ and zero optimizer state, $U_A^0 = 0$ and $v_1 = 0$, so the signal vanishes at the first step; the exact decomposition and deterministic bounds still hold.

Rank and width consequences. The conditional energy ratio is

$$\frac{E_N^t}{E_S^t} = \Theta\left(\frac{\beta^2 \sigma_A^2}{\eta_A^2 n t^2}\, r^{2(\gamma_N - \gamma_S)}\right).$$

The rank factors cancel only if $\gamma_N = \gamma_S$. Taking both exponents equal to 1 recovers the coherent benchmark; taking both equal to $1/2$ gives CLT-like rank scaling under the same temporal assumptions. For fixed r and $\lambda_r = \Theta(1)$, both $m_N, m_S = \Theta(1)$, giving

$$E_N^t = \Theta(\eta_B^2 \beta^2 \sigma_A^2 n t^2), \qquad E_S^t = \Theta((\eta_A \eta_B n t^2)^2).$$

For $\sigma_A^2 = \Theta(n^{-p})$, the noise estimate becomes $\Theta(\eta_B^2 \beta^2 m_N^2 n^{1-p} t^2)$. Thus changing $p = 1$ to $p = 2$ supplies the extra $1/n$ factor under matched contraction scales, without requiring LLN-like rank summation or equality of the noise and signal exponents. This comparison preserves the signal's scaling order under matched assumptions; it does not equate the optimizer trajectories across initializations.

We next record the two terms needed for the finite-time efficiency proof. Recall the decomposition:

$$\frac{r}{\alpha} \Delta Z_B^t = \underbrace{B_t \Delta Z_A^t}_{\delta_1^t} + \underbrace{\Delta B_t Z_A^t}_{\delta_2^t} + \Delta B_t \Delta Z_A^t.$$

Since $B_0 = 0$, the first-step contribution $B_0 \Delta Z_A^0$ vanishes. For standard gradient-based updates with zero optimizer state, $U_A^0 = 0$ as well. For later steps $t \geq 1$, the two terms have the following forms:

Lemma 3. *For $t \geq 1$, if the LoRA weights are updated by*

$$A_{t+1} = A_t - \eta_A U_A^t, \qquad B_{t+1} = B_t - \eta_B U_B^t,$$

then

$$\begin{cases} \delta_1^t = B_t \Delta Z_A^t = \eta_A \eta_B \left(\sum_{j=0}^{t-1} U_B^j\right) U_A^t \underline{Z}, \\ \delta_2^t = \Delta B_t Z_A^t = -\eta_B U_B^t A_0 \underline{Z} + \eta_A \eta_B\, U_B^t \left(\sum_{i=0}^{t-1} U_A^i\right) \underline{Z}. \end{cases}$$

Only δ_2^t explicitly contains A_0; both can depend on it through optimizer directions.

Proof. This follows directly from the identities established in Equations (4) and (5). □

Now, we prove Theorem 2. RMS bounds alone do not give coordinatewise lower bounds or control a one-step contraction, so these are stated explicitly.

Theorem 2 (Finite-time stability and efficiency of `Init[A²]`)**.** *Fix $r = \Theta(1)$ and a finite step $t \geq 2$. Let*

$$\lambda = \alpha/r = \Theta(1), \qquad \beta = \Theta(1), \qquad \eta_A = \Theta(n^{-1}), \qquad \eta_B = \Theta(1),$$

and define

$$C_t = \sum_{s<t} U_B^s, \qquad u = \tilde{A}_0 \underline{Z}, \qquad v_t = \Big(\sum_{s<t} U_A^s\Big) \underline{Z}.$$

Assume coordinate-bounded optimizer directions and, uniformly in output coordinate k,

$$|(C_t v_t)_k| = \Theta(n), \qquad |(C_t U_A^t \underline{Z})_k| = \Theta(n), \qquad |(U_B^t v_t)_k| = \Theta(n).$$

Assume also the initialization projection upper bound from Theorem 1,

$$\|u\|_2 = O(\sigma_A \sqrt{n}),$$

at fixed rank. Then under `Init[A²]`*, for which* $\sigma_A^2 = \Theta(n^{-2})$*, every coordinate of* Z_B^t*,* δ_1^t*, and* δ_2^t *has magnitude* $\Theta(1)$*, and* $E_{\text{total}}^t = \Theta(1)$*. Hence* `Init[A²]` *satisfies both stability and efficiency in Condition 1 at the analyzed step.*

Under standard `Init[A]`*, for which* $\sigma_A^2 = \Theta(n^{-1})$*, the corresponding quantities satisfy*

$$|(Z_B^t)_k| = O(1), \qquad |(\delta_1^t)_k| = \Theta(1), \qquad |(\delta_2^t)_k| = O(1),$$

and $E_{\text{total}}^t = O(1)$*. Lower bounds of* $\Theta(1)$ *for* Z_B^t *and* δ_2^t *require non-cancellation.*

Proof. From Lemmas 1 and 3,

$$Z_B^t = -\lambda\beta\eta_B C_t u + \lambda\eta_A\eta_B C_t v_t,$$

and

$$\delta_1^t = \eta_A\eta_B C_t U_A^t \underline{Z}, \qquad \delta_2^t = -\beta\eta_B U_B^t u + \eta_A\eta_B U_B^t v_t.$$

Thus Z_B^t and δ_2^t each consist of an initialization-dependent term plus a learned signal term, whereas δ_1^t contains only the learned signal.

Initialization-dependent terms. Because r and t are fixed and the optimizer directions are coordinate-bounded, every row of C_t and U_B^t has $O(1)$ Euclidean norm. Therefore

$$|(C_t u)_k|,\ |(U_B^t u)_k| = O(\|u\|_2) = O(\sigma_A\sqrt{n}).$$

For standard `Init[A]`, $\sigma_A = \Theta(n^{-1/2})$, so these terms are $O(1)$. For `Init[A²]`, $\sigma_A = \Theta(n^{-1})$, so they are $O(n^{-1/2}) = o(1)$. Multiplication by the $\Theta(1)$ factors λ, β, and η_B preserves these bounds.

Learned signal terms. By assumption,

$$|(C_t v_t)_k|,\ |(C_t U_A^t \underline{Z})_k|,\ |(U_B^t v_t)_k| = \Theta(n).$$

Since $\eta_A\eta_B = \Theta(n^{-1})$ and $\lambda = \Theta(1)$, each corresponding signal contribution has magnitude $\Theta(1)$.

Combining the two components. Under `Init[A²]`, each coordinate of Z_B^t and δ_2^t is therefore a signal of magnitude $\Theta(1)$ plus an $o(1)$ initialization term, and hence retains magnitude $\Theta(1)$. The same is already true for δ_1^t. Thus the stability and efficiency conditions hold coordinatewise at the analyzed step. Averaging the squared coordinates gives

$$E_{\text{total}}^t = \Theta(1).$$

The component bounds also give $E_N^t = O(n^{-1})$ and $E_S^t = \Theta(1)$, so Lemma 2 yields $E_{\text{total}}^t / E_S^t \to 1$. Under the additional projection and noise-contraction lower bounds of Theorem 1, the noise estimate sharpens to $E_N^t = \Theta(n^{-1})$.

Under standard `Init[A]`, both the signal and initialization-dependent contributions can be $O(1)$. Their sum is therefore $O(1)$, which proves non-explosion, but the two terms may cancel. Consequently, a $\Theta(1)$ lower bound for Z_B^t or δ_2^t requires an additional non-cancellation condition. The term δ_1^t contains no explicit initialization component and is already $\Theta(1)$ by assumption. □

A.4 Adam and AdamW as Normalized LoRA Updates

The signal–noise decomposition in Lemma 1 is algebraic once the optimizer supplies update directions U_A^t and U_B^t. The role of this section is narrower: we explain when common normalized optimizers produce the two optimizer-facing properties used in Assumption 1,

$$\|U_A^t\|_{\max}, \|U_B^t\|_{\max} = O(1), \qquad |(U_A^t)_{i,:}\underline{Z}| = \Theta(n) \quad \text{for each } i \in [r].$$

The first property is coordinate boundedness. The second is the gradient–input alignment condition for the A branch after its initial zero update. This section does not prove the temporal non-cancellation or rank-contraction assumptions in Equations (7) and (8); these remain separate conditions in Theorem 1.

Throughout this section, write

$$\lambda := \frac{\alpha}{r}$$

for the LoRA update multiplier. The LoRA layer output is

$$\bar{Z} = W\underline{Z} + \lambda BA\underline{Z}.$$

We consider a local analysis of a single example or token, in which the input $\underline{Z} \in \mathbb{R}^n$ to the LoRA layer is fixed over the optimizer window. This is the same local viewpoint used in prior LoRA feature-learning analyses [8]. The full-network setting is more complicated because $\underline{Z}$ evolves with depth and training time; our alignment claim is conditional on this fixed single-example input. Minibatch training is assessed empirically in Appendix A.4.6.

Fixed-input chain-rule structure. Let $d\bar{Z}_t := \partial \mathcal{L}_t / \partial \bar{Z}_t$. The gradients of the LoRA matrices are

$$\nabla_B \mathcal{L}_t = \lambda\, d\bar{Z}_t (Z_A^t)^\top, \qquad \nabla_A \mathcal{L}_t = \lambda\, B_t^\top d\bar{Z}_t\, \underline{Z}^\top.$$

Equivalently, for each row $i \in [r]$,

$$(\nabla_A \mathcal{L}_t)_{ij} = s_i^t \underline{Z}_j, \qquad s_i^t := \lambda (B_t^\top d\bar{Z}_t)_i.$$

Thus each row of the A-gradient is a scalar backpropagated signal times the same input vector $\underline{Z}$. This outer-product structure is the source of the $\Theta(n)$ alignment. It is not a generic property of arbitrary update directions.

Since standard LoRA initializes $B_0 = 0$, we have $s_i^0 = 0$, so the first A-update is zero. The non-degeneracy statements below apply after B has received a nonzero update, or more generally at any step for which the scalar signal history is nonzero.

Input small-ball condition. We assume the fixed input has coordinate size $O(1)$ and non-vanishing mass on order-one coordinates. Concretely, there exist constants $p, z_-, z_+ > 0$, independent of width n, such that

$$|\{j : z_- \le |\underline{Z}_j| \le z_+\}| \ge pn, \qquad \max_j |\underline{Z}_j| \le z_+.$$

This implies

$$\|\underline{Z}\|_1 = \Theta(n), \qquad \|\underline{Z}\|_2^2 = \Theta(n).$$

This allows some coordinates to vanish, provided a fixed fraction remain order one.

A.4.1 SignSGD Alignment

We first record the simplest case, since it explains why there is no coordinate-wise cancellation in the fixed-input SignSGD model.

Lemma 4 (SignSGD gives exact input alignment)**.** *Suppose the row gradient has the fixed-input form*

$$(\nabla_A \mathcal{L}_t)_{ij} = s_i^t \underline{Z}_j.$$

Let the optimizer direction be

$$(U_A^t)_{ij} = \operatorname{sign}((\nabla_A \mathcal{L}_t)_{ij}).$$

If $s_i^t \neq 0$ and $\underline{Z}$ satisfies the input small-ball condition, then

$$(U_A^t)_{i,:}\underline{Z} = \operatorname{sign}(s_i^t) \sum_{j=1}^n |\underline{Z}_j| = \Theta(n).$$

Proof. By the row-wise outer-product structure,

$$(U_A^t)_{ij} = \operatorname{sign}(s_i^t \underline{Z}_j) = \operatorname{sign}(s_i^t)\operatorname{sign}(\underline{Z}_j).$$

Therefore

$$(U_A^t)_{i,:}\underline{Z} = \sum_{j=1}^n \operatorname{sign}(s_i^t)\operatorname{sign}(\underline{Z}_j)\underline{Z}_j = \operatorname{sign}(s_i^t) \sum_{j=1}^n |\underline{Z}_j|.$$

The small-ball condition gives $\sum_{j=1}^n |\underline{Z}_j| \ge pnz_-$, and the coordinate upper bound gives $\sum_{j=1}^n |\underline{Z}_j| \le nz_+$. Hence the sum is $\Theta(n)$. □

A.4.2 Adam Coordinate Boundedness

At the zero-based LoRA step $t \geq 0$, Adam forms moments from $g_0, \ldots, g_t$, with $m_{-1} = v_{-1} = 0$:

$$m_t = (1-\beta_1)\sum_{k=0}^{t} \beta_1^k g_{t-k}, \qquad v_t = (1-\beta_2)\sum_{k=0}^{t} \beta_2^k g_{t-k}^2,$$

with bias-corrected moments

$$\widehat{m}_t = \frac{m_t}{1-\beta_1^{t+1}}, \qquad \widehat{v}_t = \frac{v_t}{1-\beta_2^{t+1}},$$

and coordinate update

$$U_t = \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}.$$

Adam hyperparameters $(\beta_1, \beta_2, \epsilon)$ are treated as width-independent constants, with $\epsilon > 0$. In the zero-EMA case $\beta_1 = \beta_2 = 0$, the update is $g_t/(|g_t| + \epsilon)$, which approaches $\operatorname{sign}(g_t)$ as $\epsilon \to 0$ for $g_t \neq 0$ [25, 26]. For the following bound, assume $0 \leq \beta_1 < 1$ and $\beta_1^2 < \beta_2 < 1$.

Proposition 1 (Width-uniform Adam coordinate bound)**.** *For any scalar gradient history $\{g_0, \ldots, g_t\}$, if $0 \leq \beta_1 < 1$ and $\beta_1^2 < \beta_2 < 1$, then*

$$\frac{|\widehat{m}_t|}{\sqrt{\widehat{v}_t} + \epsilon} \leq C_{\beta_1,\beta_2} := \frac{1}{\sqrt{(1-\beta_2)(1-\beta_1^2/\beta_2)}}$$

for every $t \geq 0$ and $\epsilon > 0$. Thus Adam coordinate updates are $O(1)$ uniformly in width.

Proof. If $v_t = 0$, then all gradients in the EMA window are zero, $m_t = 0$, and the update is zero for $\epsilon > 0$. Otherwise, Cauchy–Schwarz gives

$$\begin{aligned} m_t^2 &= (1-\beta_1)^2 \left(\sum_{k=0}^{t} \beta_1^k g_{t-k}\right)^2 \\ &\leq (1-\beta_1)^2 \left(\sum_{k=0}^{t} \left(\frac{\beta_1^2}{\beta_2}\right)^k\right)\left(\sum_{k=0}^{t} \beta_2^k g_{t-k}^2\right) \\ &= \frac{(1-\beta_1)^2}{1-\beta_2}\left(\sum_{k=0}^{t} \left(\frac{\beta_1^2}{\beta_2}\right)^k\right) v_t. \end{aligned}$$

Hence

$$\frac{|m_t|}{\sqrt{v_t}} \leq \frac{1-\beta_1}{\sqrt{1-\beta_2}}\left(\sum_{k=0}^{t} \left(\frac{\beta_1^2}{\beta_2}\right)^k\right)^{1/2} \leq \frac{1-\beta_1}{\sqrt{1-\beta_2}}\frac{1}{\sqrt{1-\beta_1^2/\beta_2}},$$

where the last inequality uses $\beta_1^2/\beta_2 < 1$. The bias-correction multiplier satisfies

$$\frac{\sqrt{1-\beta_2^{t+1}}}{1-\beta_1^{t+1}} \leq \frac{1}{1-\beta_1} \qquad (t \geq 0).$$

Combining these bounds with $\sqrt{\widehat{v}_t} + \epsilon \geq \sqrt{\widehat{v}_t}$ gives a constant independent of width and time. □

A.4.3 Adam as Soft-Sign Gradient–Input Alignment

The coordinate bound above shows that Adam directions are $O(1)$, but Assumption 1 also requires $|(U_A^t)_{i,:}\underline{Z}| = \Theta(n)$. We establish this under scalar EMA non-cancellation and a bound on the relative size of ϵ.

For a fixed row i, suppose the A-gradient history has the form

$$(\nabla_A \mathcal{L}_{t-k})_{ij} = s_i^{t-k}\underline{Z}_j, \qquad k = 0, \ldots, t.$$

Define the bias-corrected scalar EMA moments

$$\bar{s}_i^t := \frac{\sum_{k=0}^t \beta_1^k s_i^{t-k}}{\sum_{k=0}^t \beta_1^k}, \qquad \bar{q}_i^t := \frac{\sum_{k=0}^t \beta_2^k (s_i^{t-k})^2}{\sum_{k=0}^t \beta_2^k}.$$

For $\bar{q}_i^t > 0$, define the Adam signal-to-RMS ratio and epsilon ratio

$$\kappa_i^t := \frac{\bar{s}_i^t}{\sqrt{\bar{q}_i^t}}, \qquad \delta_i^t := \frac{\epsilon}{\sqrt{\bar{q}_i^t}}.$$

The condition $|\kappa_i^t| = \Theta(1)$ means that the first-moment EMA does not cancel relative to the second-moment RMS. The condition $\delta_i^t = O(1)$ prevents ϵ from introducing an additional width-dependent suppression on order-one input coordinates. If $\bar{q}_i^t = 0$, the row update is zero and no alignment lower bound holds.

Lemma 5 (Adam soft-sign alignment)**.** *Suppose the fixed-input gradient history satisfies*

$$(\nabla_A \mathcal{L}_{t-k})_{ij} = s_i^{t-k} \underline{Z}_j.$$

Assume that for row i,

$$0 < c_\kappa \le |\kappa_i^t| \le C_\kappa < \infty, \qquad \delta_i^t \le C_\delta < \infty,$$

with constants independent of width n. If $\underline{Z}$ satisfies the input small-ball condition, then the Adam update satisfies

$$\left|(U_A^t)_{i,:}\underline{Z}\right| = \Theta(n).$$

Proof. Because the gradient factors as $s_i^{t-k}\underline{Z}_j$, the Adam moments also factor coordinate-wise:

$$\widehat{m}_{ij}^t = \bar{s}_i^t \underline{Z}_j, \qquad \widehat{v}_{ij}^t = \bar{q}_i^t \underline{Z}_j^2.$$

Therefore

$$(U_A^t)_{ij} = \frac{\bar{s}_i^t \underline{Z}_j}{\sqrt{\bar{q}_i^t \underline{Z}_j^2} + \epsilon} = \frac{\bar{s}_i^t \underline{Z}_j}{\sqrt{\bar{q}_i^t}|\underline{Z}_j| + \epsilon} = \kappa_i^t \frac{\underline{Z}_j}{|\underline{Z}_j| + \delta_i^t}.$$

Taking the inner product with $\underline{Z}$,

$$(U_A^t)_{i,:}\underline{Z} = \kappa_i^t \sum_{j=1}^n \frac{\underline{Z}_j^2}{|\underline{Z}_j| + \delta_i^t}.$$

Every term in the sum is nonnegative. Hence there is no coordinate-wise sign cancellation. For the at least pn coordinates satisfying $z_- \le |\underline{Z}_j| \le z_+$,

$$\frac{\underline{Z}_j^2}{|\underline{Z}_j| + \delta_i^t} \ge \frac{z_-^2}{z_+ + C_\delta}.$$

Thus

$$\sum_{j=1}^n \frac{\underline{Z}_j^2}{|\underline{Z}_j| + \delta_i^t} \ge pn \frac{z_-^2}{z_+ + C_\delta} = \Omega(n).$$

The coordinate upper bound gives the matching upper bound

$$\sum_{j=1}^n \frac{\underline{Z}_j^2}{|\underline{Z}_j| + \delta_i^t} \le \sum_{j=1}^n |\underline{Z}_j| \le n z_+ = O(n).$$

Multiplying by $|\kappa_i^t| = \Theta(1)$ proves the claim. □

Corollary 1 (Adam verifies the LoRA A-branch non-degeneracy)**.** *Under the conditions of Lemma 5, Adam satisfies the row-wise A-branch requirement in Assumption 1:*

$$|(U_A^t)_{i,:}\underline{Z}| = \Theta(n)$$

Together with Proposition 1, this verifies the Adam coordinate bound and A-branch input product. Temporal non-cancellation and rank contraction remain assumptions in Theorem 1.

Proof. Apply Lemma 5 and Proposition 1. □

The epsilon condition matters even for a sign-consistent history. Let $\underline{Z}_j = 1$, $s_i^0 = 0$, and $s_i^k = 1/n$ for $k \ge 1$. At fixed $t \ge 1$, $|\kappa_i^t| = \Theta(1)$ but $\delta_i^t = \Theta(\epsilon n)$, giving $|(U_A^t)_{i,:}\underline{Z}| = \Theta(n/(1+\epsilon n)) = \Theta(1)$ for fixed $\epsilon > 0$. Thus EMA non-cancellation does not ensure $\Theta(n)$ alignment.

A.4.4 A Weighted-Angle View of Adam Non-Cancellation

The Adam non-cancellation condition has a simple geometric interpretation. Let

$$a_k = \beta_1^k, \qquad b_k = \beta_2^k, \qquad c_{1,t} = \sum_{k=0}^{t} a_k, \qquad c_{2,t} = \sum_{k=0}^{t} b_k.$$

For a fixed row i, write the scalar signal history as

$$s_i = (s_i^{t-k})_{k=0}^{t} \in \mathbb{R}^{t+1}.$$

Then

$$\kappa_i^t = \frac{\bar{s}_i^t}{\sqrt{\bar{q}_i^t}} = \frac{\sqrt{c_{2,t}}}{c_{1,t}} \frac{\sum_{k=0}^{t} a_k s_i^{t-k}}{\sqrt{\sum_{k=0}^{t} b_k (s_i^{t-k})^2}}.$$

Let $D_b = \operatorname{diag}(\sqrt{b_0}, \ldots, \sqrt{b_t})$. Then

$$\sum_{k=0}^{t} a_k s_i^{t-k} = \langle D_b^{-1} a, D_b s_i \rangle,$$

and

$$\sqrt{\sum_{k=0}^{t} b_k (s_i^{t-k})^2} = \|D_b s_i\|_2.$$

Thus

$$\kappa_i^t = \frac{\sqrt{c_{2,t}}}{c_{1,t}} \|D_b^{-1} a\|_2 \cos\theta_i^t,$$

where θ_i^t is the angle between the Adam first-moment kernel $D_b^{-1} a$ and the RMS-weighted scalar signal history $D_b s_i$.

The condition $\beta_1^2 < \beta_2$ guarantees

$$\|D_b^{-1} a\|_2^2 = \sum_{k=0}^{t} \frac{\beta_1^{2k}}{\beta_2^k} = \sum_{k=0}^{t} \left(\frac{\beta_1^2}{\beta_2}\right)^k = \Theta(1).$$

Since $1 \le c_{1,t} \le (1-\beta_1)^{-1}$ and $1 \le c_{2,t} \le (1-\beta_2)^{-1}$, the prefactor of $\cos\theta_i^t$ is bounded above and away from zero uniformly in t. For nonzero histories, $|\kappa_i^t| = \Theta(1)$ is therefore equivalent to $|\cos\theta_i^t| = \Theta(1)$. Under the fixed-input and small-ball assumptions, this and $\delta_i^t = O(1)$ ensure alignment. For $t \ge 1$, it suffices that the entries after the initial zero gradient share a sign and have magnitudes bounded above and below by positive width-independent constants.

A.4.5 AdamW and Effective Horizons

AdamW combines Adam normalized directions with decoupled weight decay:

$$A_{t+1} = (1 - \eta_{A,t}\omega_A) A_t - \eta_{A,t} U_A^t, \qquad B_{t+1} = (1 - \eta_{B,t}\omega_B) B_t - \eta_{B,t} U_B^t,$$

where $\eta_{A,t}, \eta_{B,t} \ge 0$ and $\omega_A, \omega_B \ge 0$. The directions U_A^t, U_B^t are formed from loss gradients, excluding weight decay, along the AdamW trajectory. Define the decay products

$$\Gamma_A(s,t) := \prod_{\ell=s}^{t-1} (1 - \eta_{A,\ell}\omega_A), \qquad \Gamma_B(s,t) := \prod_{\ell=s}^{t-1} (1 - \eta_{B,\ell}\omega_B).$$

We use the convention $\Gamma_A(t,t) = \Gamma_B(t,t) = 1$.

Proposition 2 (Exact AdamW decomposition and noise bound)**.** *Assume $B_0 = 0$ and $A_0 = \beta \tilde{A}_0$. Define*

$$\mathcal{A}_t := \sum_{i=0}^{t-1} \eta_{A,i} \Gamma_A(i+1,t) U_A^i, \qquad \mathcal{B}_t := \sum_{j=0}^{t-1} \eta_{B,j} \Gamma_B(j+1,t) U_B^j.$$

Then

$$Z_B^t = -\lambda\beta\,\Gamma_A(0,t)\mathcal{B}_t\tilde{A}_0\underline{Z} + \lambda\mathcal{B}_t\mathcal{A}_t\underline{Z}.$$

Denote the first and second terms by T^t_{noise} and T^t_{signal}, respectively. If the decay factors are non-expansive,

$$0 \le 1-\eta_{A,t}\omega_A \le 1, \qquad 0 \le 1-\eta_{B,t}\omega_B \le 1,$$

then the noise satisfies

$$\begin{aligned}\|T^t_{\text{noise}}\|_2 &\le \lambda\beta\Gamma_A(0,t)\sum_{j<t}\eta_{B,j}\Gamma_B(j+1,t)\|U_B^j\tilde{A}_0\underline{Z}\|_2 \\ &\le \lambda\beta\sum_{j<t}\eta_{B,j}\|U_B^j\tilde{A}_0\underline{Z}\|_2.\end{aligned}$$

Proof. Unrolling the recursions gives

$$A_t = \Gamma_A(0,t)A_0 - \sum_{i=0}^{t-1}\eta_{A,i}\Gamma_A(i+1,t)U_A^i,$$

and, since $B_0 = 0$,

$$B_t = -\sum_{j=0}^{t-1}\eta_{B,j}\Gamma_B(j+1,t)U_B^j.$$

Using $A_0 = \beta\tilde{A}_0$, this is

$$A_t = \beta\Gamma_A(0,t)\tilde{A}_0 - \mathcal{A}_t, \qquad B_t = -\mathcal{B}_t.$$

Substituting into $Z_B^t = \lambda B_t A_t\underline{Z}$ yields

$$Z_B^t = -\lambda\beta\,\Gamma_A(0,t)\mathcal{B}_t\tilde{A}_0\underline{Z} + \lambda\mathcal{B}_t\mathcal{A}_t\underline{Z}.$$

The first noise bound follows by the triangle inequality. Under non-expansive decay, all decay products lie in $[0,1]$, giving the second bound. □

These upper bounds do not imply that the realized noise decreases with weight decay. For fixed projected contributions $v \neq 0$ and $-v$, the undamped sum is zero, whereas damping the earlier term gives $qv - v = -(1-q)v \neq 0$ for $0 < q < 1$. Decay can therefore reduce cancellation; monotonicity of either component does not follow from bounds on the decay factors.

Under non-expansive decay, define the effective horizons

$$H_A(t) := \sum_{i=0}^{t-1}\eta_{A,i}\Gamma_A(i+1,t), \qquad H_B(t) := \sum_{j=0}^{t-1}\eta_{B,j}\Gamma_B(j+1,t).$$

If the coordinate bounds hold uniformly, then $\|\mathcal{A}_t\|_{\max} = O(H_A(t))$ and $\|\mathcal{B}_t\|_{\max} = O(H_B(t))$. Matching lower bounds require assumptions on the weighted sums themselves.

Conditional component scaling. Write $u := \tilde{A}_0\underline{Z}$, $w_t := \mathcal{A}_t\underline{Z}$, and $\rho_d(x) := \|x\|_2/\sqrt{d}$. On analyzed steps with positive $\Gamma_A(0,t), H_A(t), H_B(t)$, assume the weighted analogues of Equations (7) and (8):

$$\begin{aligned}\rho_r(u) &= \Theta(\sigma_A\sqrt{n}), & \rho_r(w_t) &= \Theta(nH_A(t)),\\ \rho_n(\mathcal{B}_t u) &= \Theta(H_B(t)r^{\gamma_N}\rho_r(u)), & \rho_n(\mathcal{B}_t w_t) &= \Theta(H_B(t)r^{\gamma_S}\rho_r(w_t)),\end{aligned}$$

for fixed exponents γ_N, γ_S and constants uniform in width, rank, and the analyzed steps. Setting $m_N := \lambda r^{\gamma_N}$ and $m_S := \lambda r^{\gamma_S}$, the exact decomposition gives

$$\begin{aligned}E_N^t &= \frac{1}{n}\|T^t_{\text{noise}}\|_2^2 = \Theta\big(\beta^2 m_N^2\Gamma_A(0,t)^2 H_B(t)^2\sigma_A^2 n\big)\,,\\ E_S^t &= \frac{1}{n}\|T^t_{\text{signal}}\|_2^2 = \Theta\big((m_S H_A(t)H_B(t)n)^2\big)\,.\end{aligned}$$

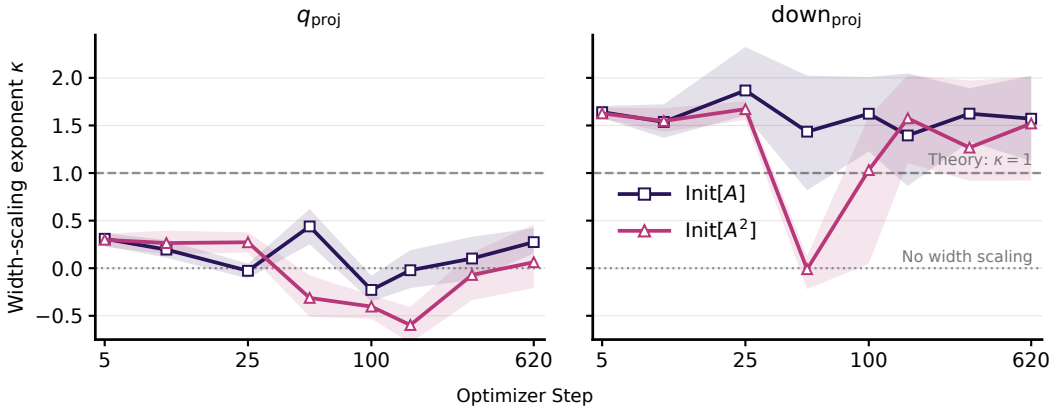


Figure 8: Empirical width-scaling exponent of the processed A-update/input alignment during Qwen2.5 minibatch fine-tuning on Math-10K. Panels compare `q_proj` and `down_proj` under `Init[A]` and `Init[A^2]`. Reference lines mark the single-example prediction $\kappa = 1$ and width-independent alignment $\kappa = 0$. The exponent varies across modules and optimizer steps.

The total energy still satisfies Lemma 2. The assumptions above impose weighted temporal non-cancellation and rank contraction; the scalar horizons alone do not establish them.

For constant LRs, if $\Gamma_A(0,t), \Gamma_B(0,t) = \Theta(1)$ uniformly on the analyzed window, then $H_A(t) = \Theta(\eta_A t)$ and $H_B(t) = \Theta(\eta_B t)$. Under the weighted contraction assumptions, this recovers the component scales in Theorem 1. More generally, changing $\sigma_A^2 = \Theta(n^{-1})$ to $\Theta(n^{-2})$ supplies the extra $1/n$ noise-energy factor when the input, multiplier, β, decay, horizon, and rank-contraction scales are matched across initializations. The optimizer trajectories may differ, and the exact decomposition alone does not establish these matched scales.

A.4.6 Empirical Width Alignment and Single-Example Scope

The alignment derivation above concerns a single example or token with a locally fixed input. With minibatches, each gradient combines multiple inputs and backpropagated signals, so its normalized direction need not retain this outer-product structure. We therefore measure realized alignment rather than assert a general batch extension.

For row i of module ℓ, let $a_{t,\ell,i}(n) := |(U^t_{A,\ell})_{i,:}\underline{Z}_\ell|$. Across Qwen2.5-{0.5B, 1.5B, 3B, 7B}, we fit the module-level alignment magnitude as $a_{t,\ell}(n) \propto n^{\kappa_{t,\ell}}$ at sampled optimizer steps. Both initializations use three seeds, giving 24 runs, with Math-10K, $r = \alpha = 8$, $\eta_A = 10^{-5}$, and $\eta_B/\eta_A = 200$.

We sample four `q_proj` modules and the final-block `down_proj`. The fan-in n is the input width of the measured projection: $n \in \{896, 1536, 2048, 3584\}$ for `q_proj`, and $n \in \{4864, 8960, 11008, 18944\}$ for `down_proj`. We also fit the width exponent of the normalized alignment $a_{t,\ell,i}(n)/(\|(U^t_{A,\ell})_{i,:}\|_2\|\underline{Z}_\ell\|_2)$. The single-example theory predicts a dot-product exponent of 1 and a cosine exponent of 0 under its stated non-degeneracy conditions.

Figure 8 shows predominantly sublinear alignment in `q_proj` and generally superlinear alignment in `down_proj`, with substantial changes during training. Thus $\Theta(n)$ alignment is not generic for minibatch fine-tuning. Our formal signal-scaling and efficiency results remain conditional on the fixed single-example analysis; extending them to minibatches requires the realized alignment scale and its temporal accumulation. This restriction affects the signal estimate, while the separate initialization-noise comparison retains the extra $1/n$ factor under matched input and contraction assumptions.

A.4.7 Summary for the Main Theorem

Corollary 2 (Adam and AdamW instantiate the normalized-update analysis)**.** *Consider a fixed LoRA-layer input $\underline{Z}$ satisfying the small-ball condition and zero initial optimizer moments. Assume Adam hyperparameters $(\beta_1, \beta_2, \epsilon)$ are width-independent, with $\epsilon > 0$, $0 \le \beta_1 < 1$, and $\beta_1^2 < \beta_2 < 1$.*

Table 2: Width-alignment fits at steps 5 and 620. Dot-product exponents include 95% CIs computed over three seeds; the last column gives cosine exponents at the two steps.

Module	Initialization	Step-5 dot κ 95% CI	Step-620 dot κ 95% CI	Cosine exponent step 5 / 620
`q_proj`	`Init[A]`	0.309 [0.235, 0.371]	0.274 [0.155, 0.421]	$-0.477/-0.400$
`q_proj`	`Init[A`^{2}`]`	0.299 [0.223, 0.362]	0.063 [-0.208, 0.459]	$-0.478/-0.353$
`down_proj`	`Init[A]`	1.640 [1.578, 1.706]	1.571 [1.126, 2.021]	$0.172/-0.052$
`down_proj`	`Init[A`^{2}`]`	1.626 [1.573, 1.686]	1.521 [0.922, 2.013]	$0.174/-0.098$

Suppose the A-gradient has the local row-wise form

$$(\nabla_A \mathcal{L}_{t-k})_{ij} = s_i^{t-k} \underline{Z}_j$$

over the complete Adam gradient history. If the following bounds hold uniformly over the analyzed rows and steps,

$$\left| \frac{\bar{s}_i^t}{\sqrt{\bar{q}_i^t}} \right| = \Theta(1), \qquad \frac{\epsilon}{\sqrt{\bar{q}_i^t}} = O(1),$$

then Adam satisfies

$$\|U_A^t\|_{\max} = O(1), \qquad |(U_A^t)_{i,:}\underline{Z}| = \Theta(n).$$

The same coordinate bound applies to U_B^t. Therefore Adam satisfies the optimizer-facing requirements of Assumption 1 under these conditions. Temporal non-cancellation in both branches and the rank contractions remain separate assumptions in Theorem 1.

For AdamW, the same optimizer-facing bounds hold for its loss-gradient directions under the same conditions. Feature scaling uses the exact decay-weighted decomposition in Proposition 2 and additionally requires the weighted temporal and rank-contraction assumptions stated above. The extra $1/n$ noise-energy comparison holds under matched projection, decay, horizon, and contraction scales across initializations; non-expansive decay alone does not guarantee it.

A.5 Spectral Derivation of `Init[A`^{2}`]`

We derive `Init[A`^{2}`]` from spectral conditions for stable feature learning. Spectral and Frobenius norm scalings capture matrix behavior in the large-width limit beyond coordinate-wise estimates.

A.5.1 Matrix Norm and Low-Rank Equivalence

The spectral norm of a matrix A, denoted $\|A\|_{\text{op}}$, is its largest singular value, defined as the maximum stretch applied to any nonzero vector:

$$\|A\|_{\text{op}} = \max_{v \neq 0} \frac{\|Av\|_2}{\|v\|_2}. \tag{9}$$

The spectral norm can be bounded by the Frobenius norm:

$$1/\sqrt{r}\|A\|_F \leq \|A\|_{\text{op}} \leq \|A\|_F. \tag{10}$$

Here $r \geq 1$ is an upper bound on $\text{rank}(A)$. In the fixed-rank LoRA setting of this subsection, the spectral and Frobenius norms are asymptotically equivalent: $\|\cdot\|_F \asymp \|\cdot\|_{\text{op}}$.

A.5.2 General Spectral Condition

A central requirement for feature learning in large-width neural networks is that the feature representations must evolve nontrivially without vanishing or exploding. Following the feature-learning desiderata proposed by Yang et al. [27], we start with the general condition for a weight matrix $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ at layer ℓ.

Condition 2 (Spectral Scaling Condition)**.** *Consider applying a gradient update $\Delta W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$ to the ℓ-th weight matrix W_ℓ. To target the feature learning desideratum $\|h_\ell\|_2 = \Theta(\sqrt{n_\ell})$ and $\|\Delta h_\ell\|_2 = \Theta(\sqrt{n_\ell})$, we use the spectral scaling prescription:*

$$\|W_\ell\|_{\text{op}} = \Theta\left(\sqrt{\frac{n_\ell}{n_{\ell-1}}}\right) \quad \textit{and} \quad \|\Delta W_\ell\|_{\text{op}} = \Theta\left(\sqrt{\frac{n_\ell}{n_{\ell-1}}}\right), \quad \textit{for all layers } \ell. \tag{11}$$

This condition controls the maximum amplification of an input signal. Matching lower bounds for realized features and their updates additionally require alignment and non-cancellation; spectral norms alone do not exclude inputs in a nullspace.

Application to LoRA. In the context of LoRA, we identify two distinct linear transformations: the down-projection $A \in \mathbb{R}^{r\times n}$ and the up-projection $B \in \mathbb{R}^{n\times r}$. For the down-projection A, we have $n_{\ell-1} = n$ and $n_\ell = r$. Substituting these into Condition 2 and noting that $r = \Theta(1)$, we require:

$$\|A\|_{\text{op}} = \Theta\left(\sqrt{\frac{r}{n}}\right) = \Theta(n^{-1/2}).$$

Conversely, for the up-projection B, we have $n_{\ell-1} = r$ and $n_\ell = n$. This yields the requirement:

$$\|B\|_{\text{op}} = \Theta\left(\sqrt{\frac{n}{r}}\right) = \Theta(n^{1/2}).$$

The updates ΔA and ΔB require the same spectral scalings for stable feature evolution.

A.5.3 Implementing Feature Learning via `Init[A²]`

At fixed rank $r = \Theta(1)$, spectral and Frobenius norms have the same asymptotic scaling: $\|A\|_{\text{op}} \asymp \|A\|_F$. We therefore derive initialization variances and LRs from Frobenius norms. For an Adam-like optimizer, we model typical update entries as $(\Delta A)_{ij} \sim \eta_A$ and $(\Delta B)_{ij} \sim \eta_B$.

Derivation for block A ($A \in \mathbb{R}^{r\times n}$). The spectral condition mandates that the Frobenius norm squared scales as $\|A\|_F^2 \asymp \|A\|_{\text{op}}^2 = \Theta(n^{-1})$. With $A_0 = \beta\tilde{A}_0$, $\beta = \Theta(1)$, and independent centered Gaussian entries of variance σ_A^2 in $\tilde{A}_0$, the expected squared Frobenius norm is $\mathbb{E}[\|A_0\|_F^2] = \beta^2 rn\sigma_A^2$. At fixed rank, $\|A_0\|_F^2 = \Theta(n\sigma_A^2)$ with high probability. Matching this to $\Theta(n^{-1})$ gives $\sigma_A^2 = \Theta(n^{-2})$. Similarly, under the assumed update-entry scaling, $\|\Delta A\|_F^2 \approx rn\eta_A^2 = \Theta(n\eta_A^2)$. To satisfy $\|\Delta A\|_F^2 = \Theta(n^{-1})$, we set $\eta_A = \Theta(n^{-1})$.

Derivation for block B ($B \in \mathbb{R}^{n\times r}$). For the up-projection matrix, the spectral condition requires $\|B\|_F^2 \asymp \|B\|_{\text{op}}^2 = \Theta(n)$. Following the same logic, if B_0 were initialized with variance σ_B^2, the expected norm would be $\mathbb{E}[\|B_0\|_F^2] = nr\sigma_B^2 = \Theta(n\sigma_B^2)$. Matching this to $\Theta(n)$ implies $\sigma_B^2 = \Theta(1)$. While standard LoRA initializes $B_0 = 0$, this transient violation is corrected in the first step provided the updates are properly scaled. For the updates, $\|\Delta B\|_F^2 \approx nr\eta_B^2 = \Theta(n\eta_B^2)$. To achieve the target $\Theta(n)$ scaling, we require $n\eta_B^2 = \Theta(n)$, which simplifies to $\eta_B = \Theta(1)$.

The derivation above dictates the following HPs to satisfy the spectral conditions for stable feature learning: $\sigma_A^2 = \Theta(n^{-2})$ with $\eta_A = \Theta(n^{-1})$, and by extension $\eta_B = \Theta(1)$. This configuration corresponds to our proposed `Init[A²]` strategy, using $\Theta(n^{-2})$ variance for the down-projection rather than the standard $\Theta(n^{-1})$ scale. It controls spectral magnitudes; non-degenerate feature learning additionally requires the alignment and non-cancellation assumptions stated above.

A.6 Spectral Conditions for Robust LoRA Feature Learning

Prior work [8, 15, 16] provides useful intuition for LoRA scaling, but its coordinate-wise, average-case γ-calculus can obscure a failure mode in practical fine-tuning. We explain how a spectral, worst-case perspective [27] leads to the `Init[A²]` scaling.

A.6.1 Why γ-Calculus Is Not Sufficient

The γ-calculus tracks the coordinate-wise magnitude of random variables. The framework is implicitly average-case, assuming that summations exhibit typical cancellations and that individual coordinates are representative of global behavior. This assumption leads to fundamental limitations.

For a matrix-vector product Wx, the scaling also depends on cancellation among the products $W_{ij}x_j$. Under suitable independence and moment assumptions, two benchmark rules are:

- **CLT-like (centered summands):** $\gamma[Wx] = \gamma[W] + \gamma[x] + 1/2$
- **LLN-like (nonzero-mean summands):** $\gamma[Wx] = \gamma[W] + \gamma[x] + 1$

Neither rule follows from the marginal mean of W alone when W and x are dependent. In LoRA, the initialization-dependent and optimizer-driven components can have different rank contractions, as captured by γ_N and γ_S in Theorem 1. Tracking only the final feature's coordinate exponent does not distinguish these components or their signed cross term. The explicit decomposition is therefore needed to diagnose noise dominance even when the total feature satisfies a bounded scaling condition.

A.6.2 Spectral Condition: Worst-Case Control

Following Yang et al. [27], the worst-case spectral condition is:

Condition 3 (Spectral Condition for LoRA Feature Learning)**.** *For $n \to \infty$ and fixed rank $r = \Theta(1)$, the LoRA factors and their updates satisfy:*

$$\begin{cases} \|A\|_{\mathrm{op}} = \Theta(n^{-1/2}) \\ \|\Delta A\|_{\mathrm{op}} = \Theta(n^{-1/2}) \end{cases} \quad \textit{and} \quad \begin{cases} \|B\|_{\mathrm{op}} = \Theta(n^{1/2}) \\ \|\Delta B\|_{\mathrm{op}} = \Theta(n^{1/2}) \end{cases}$$

For $\|\underline{Z}\|_2 = \Theta(\sqrt{n})$ aligned with a principal singular vector, $\|A\|_{\mathrm{op}} = \Theta(n^{-1/2})$ gives an order-one bottleneck feature $Z_A = A\underline{Z}$. Conversely, $\|B\|_{\mathrm{op}} = \Theta(n^{1/2})$ maps an aligned order-one bottleneck vector to output scale $\Theta(\sqrt{n})$. These worst-case scales do not by themselves give lower bounds for arbitrary inputs.

By Lemma 1, $T^t_{\text{noise}} = -(\alpha\beta/r)\eta_B(\sum_{i<t} U_B^i)\tilde{A}_0\underline{Z}$. Applying a spectral bound gives

$$\|T^t_{\text{noise}}\|_2 \le \frac{\alpha\beta}{r}\eta_B \| \sum_{i=0}^{t-1} U_B^i\|_{\mathrm{op}} \|\tilde{A}_0\|_{\mathrm{op}} \|\underline{Z}\|_2.$$

At fixed rank, coordinate-bounded directions give $\| \sum_{i<t} U_B^i\|_{\mathrm{op}} = O(t\sqrt{n})$, while Gaussian initialization gives $\|\tilde{A}_0\|_{\mathrm{op}} = \Theta(\sigma_A\sqrt{n})$ with high probability. Therefore the worst-case noise RMS satisfies

$$\frac{\|T^t_{\text{noise}}\|_2}{\sqrt{n}} = O\left(\frac{\alpha\beta}{r}\eta_B t\sigma_A n\right).$$

For fixed t and $\alpha/r, \beta, \eta_B = \Theta(1)$, this bound is $O(\sqrt{n})$ under `Init[A]` and $O(1)$ under `Init[A²]`. Thus the spectral argument establishes bounded worst-case noise RMS for `Init[A²]`, but does not establish vanishing noise or signal dominance for all aligned inputs.

Vanishing noise and signal dominance require the fixed-input projection and contraction assumptions in Theorems 1 and 2; we derive the spectral initialization rule in Appendix A.5.

B Related Literature Review

B.1 LoRA Initialization

Standard LoRA sets $B_0 = 0$ and initializes A_0 with mean-zero entries of variance $\Theta(n^{-1})$, denoted `Init[A]`; PEFT uses Kaiming-uniform initialization (Appendix C.1). Hayou et al. [15] show that the symmetric configuration `Init[B]`, with $A_0 = 0$ and B_0 initialized at entrywise variance $\Theta(1)$, has inferior optimization dynamics and stability compared with `Init[A]`. PEFT [28] also offers initialization strategies to accelerate convergence or enhance stability. LoftQ [29] minimizes the quantization error between original and quantized weights. PiSSA [30] and EVA [31] adopt spectral approaches: PiSSA uses principal singular components of the weight matrix to approximate full-rank updates, while EVA uses a singular value decomposition of layer input activations. CorDA [32] uses context-oriented decomposition to preserve world knowledge or enhance task-specific learning. OLoRA [33] and orthogonal initialization use orthonormal decompositions for stable training.

B.2 Other LoRA Hyperparameters

Beyond initialization, LoRA introduces sensitivity to several coupled HPs, including the rank r, scaling factor α, and LRs η_A, η_B. The rank controls the expressivity of the adapter: Zeng and Lee [5] characterize the expressive power of low-rank adaptation and clarify how the choice of r constrains the class of attainable updates. However, increasing rank does not automatically improve vanilla LoRA. Kalajdzievski [6] show that the conventional multiplier α/r can make the LoRA branch weaker as r grows, so higher ranks may yield diminishing gains in practice. Their rank-stabilized alternative replaces this scaling with $\alpha/\sqrt{r}$, preserving a more consistent update scale across ranks.

LR selection introduces a separate source of brittleness. In the feature-learning view, useful transfer requires both the hidden states and their updates to remain $\Theta(1)$ as width increases. Hayou et al. [8] analyze LoRA in the infinite-width limit and derive stability conditions under which non-degenerate feature updates require a large LR separation, with $\eta_B \gg \eta_A$ and an idealized ratio that scales with width. Recent work further studies the joint width-and-rank limit, showing that optimal LR-rank scaling depends on the initialization and LoRA update multiplier [18]. These theories provide guidance, but pretrained LLMs typically do not use the corresponding width-scaling parameterization. Empirical studies find that vanilla LoRA can be competitive after careful LR search [7], while the best asymmetric LR ratios observed in practice are often far smaller than the model width [8]. We address this theory-to-practice gap through initialization: standard `Init[A]` injects persistent noise under aggressive asymmetric LRs, whereas `Init[A`^{2}`]` suppresses it and improves robustness.

B.3 Feature Learning and HP Transfer

Feature learning [9] is a central mechanism in modern deep learning, wherein neural networks are trained not merely to match target outputs, but to acquire internal representations that encode useful invariances and hierarchical abstractions. This stands in contrast to the Neural Tangent Kernel (NTK) or "lazy" training regime [34–36], in which an infinite-width network behaves as a kernel machine with features effectively frozen at their random initialization. In the feature-learning regime, by contrast, the network's internal feature map evolves substantially over the course of training. To facilitate scalable feature learning, Yang and Hu [9] introduced the maximal update parametrization (μP), which enables nontrivial feature-learning dynamics even in the infinite-width limit. Unlike the NTK parametrization, μP ensures that pre-activations, gradients, and parameter updates remain $\Theta(1)$ as the network width tends to infinity. Closely related approaches based on mean-field theory [37–39] likewise analyze the evolution of the distribution of neurons and demonstrate that learned features can emerge outside the kernel regime.

While early scaling analyses focused on width, recent work has extended these ideas to the depth scaling limit, where the number of layers $L \to \infty$. Naively deep networks are prone to vanishing activations or exploding gradients, but these issues can be resolved by scaling residual blocks as $O(1/\sqrt{L})$ [12, 13].

A key use case is HP transfer across model scales [10]. Because μP maintains consistent training dynamics across width and depth, practitioners can tune LRs and other HPs on small proxy models for use in larger networks. Empirical methods [40–42] similarly normalize layer outputs and updates for width transfer. Related work studies transformer width-to-depth ratios [11], sharpness-aware optimization [43], sparse training [44], and state space models [45].

B.4 Training Stability

While feature learning offers insight into representation adaptation, such analyses typically depend on a specific choice of parameterization to guarantee well-behaved limits as network width grows. However, not all models are parameterized in this manner, and identifying an appropriate parameterization can be challenging for complex architectures. This motivates a complementary line of research that elevates training stability to a primary objective, often treated as orthogonal to, or only weakly coupled with, feature-learning optimality.

Historically, stability concerns were most prominent in recurrent neural networks, where unconstrained temporal composition of states led to exploding or vanishing gradients. Architectures such as Long Short-Term Memory (LSTM) [46] and Gated Recurrent Units (GRUs) [47] mitigated these issues by introducing gated, additive state updates that constrain the hidden dynamics to a

contractive, well-conditioned manifold, thereby enabling stable gradient propagation over long time horizons. From a geometric perspective, these mechanisms effectively bound the recurrent Jacobian and attenuate sensitivity to HPs.

Muon [48] and the Spectral Sphere Optimizer [49] constrain optimization trajectories to spectrally controlled manifolds, regulating operator norms for stability. Layer Normalization [50], Post-LN Transformers [51], and Pre-LN Transformers [52] can also be interpreted as imposing implicit spectral control. DeepSeek's manifold hyperconnections [53] constrain learned residual mixing matrices to the doubly stochastic manifold, yielding convex, well-conditioned layer compositions.

C Experimental Setup

C.1 Initialization Implementation

For transformer fine-tuning (RoBERTa, Llama, Qwen, and Mistral), we follow the `PEFT` convention [28]: Kaiming-uniform A [54] and $B_0 = 0$. Under Kaiming-uniform,

$$A_{ij} \sim \mathcal{U}(-b, b), \qquad b = \sqrt{3\ \mathrm{Var}(A_{ij})},$$

so that the uniform bound b and the variance are related by $\mathrm{Var}(A_{ij}) = b^2/3$.

We parameterize the standard deviation of A by β. For a LoRA layer with input dimension fan_in, we set

$$\mathrm{Var}(A_{ij}) = \frac{\beta^2}{(\text{fan_in})^{2s}},$$

where `Init[A]` uses $s = \frac{1}{2}$ and `Init[A²]` uses $s = 1$. Thus `Init[A]` has variance $\beta^2/\text{fan_in}$, while `Init[A²]` has variance $\beta^2/\text{fan_in}^2$ and suppresses initialization noise by another factor of $1/\text{fan_in}$. For rectangular layers, n denotes the fan-in of A.

C.2 Toy Example

Dataset. We use MNIST, flattening each image to $x \in \mathbb{R}^{784}$ with one of 10 labels.

Model. We use $Y = W_{\text{out}}\mathcal{F}(W_0 + BA, \mathcal{F}(W_{\text{in}}, X))$ with $\mathcal{F}(W, z) = \mathrm{ReLU}(Wz)$, $W_{\text{in}} \in \mathbb{R}^{n\times 784}$, $W_{\text{out}} \in \mathbb{R}^{10\times n}$, $A \in \mathbb{R}^{r\times n}$, and $B \in \mathbb{R}^{n\times r}$. We set $r = 8$ and sweep $n \in \{16, 64, 256, 1024, 4096\}$.

Training. We train only A and B for 20 epochs at batch size 256 using Adam [55] with $\beta_1 = \beta_2 = 0$ (no moving averages). We fix $W_0 = 0$, draw and freeze $W_{\text{in}} \sim \mathcal{N}(0, 1)$ and $W_{\text{out}} \sim \mathcal{N}(0, 1/n^2)$, compare `Init[A]` and `Init[A²]` with $\eta_B = n\eta_A$, and sweep $\eta_A \in \{2^{-7}, 2^{-6}, \cdots, 2^0\}$. The released reconstruction uses ReLU subgradient 1 at zero (`clamp_min(0)`); subgradient 0 would prevent the first update when $W_0 = B_0 = 0$. The original toy source is not archived, and exact numerical reproduction of the plots has not been verified.

C.3 Encoder-Only Model Setup

Setup. We fine-tune RoBERTa-base [56] on GLUE MNLI, QQP, SST-2, and QNLI [57], applying LoRA to the `query` and `value` projections. We use $r = \alpha = 8$ and the Kaiming-uniform implementation from Appendix C.1. Fixed-ratio sweeps set $\eta_A = 10^{-5}$ and vary $\eta_B/\eta_A \in \{1, 50, 200, 400, 800\}$; 2D sweeps vary η_A, η_B directly at $\beta \in \{0.25, 8\}$.

Optimization. We use AdamW [58] with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.99, 10^{-8})$, a linear schedule without warmup, sequence length 128, batch size 32, and 3 epochs. Runs use NVIDIA L40S GPUs and each individual run takes roughly one hour.

C.4 Decoder-Only Model Setup

Datasets and models. We use the arithmetic/math and commonsense benchmarks from Hu et al. [21], following Li et al. [16]: 10k arithmetic/math examples and 170k commonsense examples.

The main sweeps use Llama 3-8B for initialization scale and Llama 3.2-1B for rank [20]; appendix sweeps add QLoRA, stronger A-scalings, Llama 3.2-3B, Qwen2.5-3B [59], Qwen3-0.6B [60], and Mistral-v0.3 [61]. Long-horizon, multi-seed multiplier, and energy-decomposition checks use Llama 3-8B (Appendices D.4 to D.6). By default, we apply LoRA to `q_proj`, `k_proj`, `v_proj`, `up_proj`, and `down_proj`.

LoRA HPs. For Llama 3-8B, we use $r = \alpha = 8$ and sweep $\beta \in \{0.25, 0.5, 1, 2, 4, 8\}$ and $\rho = \eta_B/\eta_A \in \{1, 10, 200, 500, 1000, 2000, 4000\}$. For decoder-only ratio sweeps, unless stated otherwise, we fix $\eta_A = 10^{-5}$ and set $\eta_B = \rho\eta_A$. The Llama 3.2-1B rank sweep fixes $\beta = 1$, varies $r \in \{4, 8, 16, 32, 64, 128\}$, and sets $\alpha = r$. The QLoRA comparison uses the same Llama 3-8B grid; the model-family sweep uses $r = \alpha = 8$, $\beta \in \{0.5, 1, 2, 4, 8\}$, and the ratio grid in Figure 12; the initialization-power sweep varies $\beta \in \{4, 8, 12, 16\}$ and $\rho \in \{1, 10, 50, 200, 500, 1000, 2000, 4000, 8000, 10000, 20000\}$.

Optimization. We use AdamW [58] with $(\beta_1, \beta_2, \epsilon) = (0.9, 0.999, 10^{-8})$, zero weight decay, global batch size 16, micro-batch size 8, and sequence length 256. We train 1 epoch on commonsense and 3 epochs on arithmetic using NVIDIA L40S GPUs; each run takes roughly three hours. The arithmetic diagnostics in Appendix D.4 compare separately trained horizons of up to 10 epochs.

D Additional Results and Ablations

D.1 LR Sweep Experiment

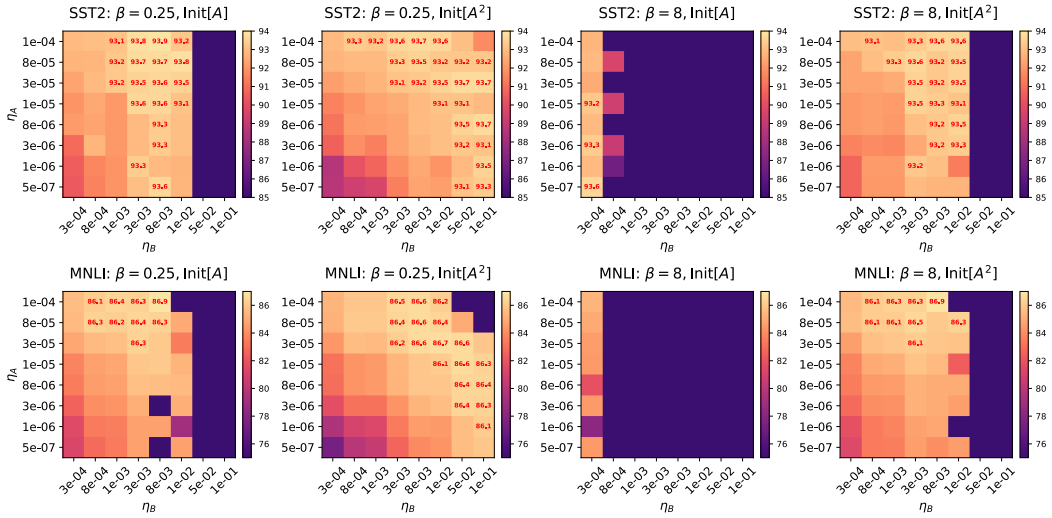


Figure 9: RoBERTa accuracy on SST-2 (top) and MNLI (bottom) under `Init[A]` and `Init[A`^{2}`]`. The grids sweep LRs η_A and η_B at two initialization scales β.

We sweep the individual LRs η_A and η_B at initialization scales $\beta \in \{0.25, 8\}$ to compare the usable regions of `Init[A]` and `Init[A`^{2}`]`. In Figure 9, both methods have broad high-accuracy regions at $\beta = 0.25$. At $\beta = 8$, the high-accuracy region for `Init[A]` narrows to low η_B values, whereas `Init[A`^{2}`]` retains a broad region across the tested LRs. These results support reduced sensitivity to initialization scale over the tested LR grid.

D.2 Sensitivity to the B LR

Under the assumptions of Theorem 1, η_B multiplies both the signal and noise components, while η_A appears explicitly only in the signal. At fixed projection and contraction scales, increasing β under `Init[A]` requires a smaller η_B to keep the noise energy at the same level. To examine the role of η_B, we freeze block A ($\eta_A = 0$) in the RoBERTa MNLI and SST-2 setup and sweep $\eta_B \in \{5\times10^{-5}, 1\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}, 8\times10^{-3}, 1\times10^{-2}\}$ for both initializations. Figure 10

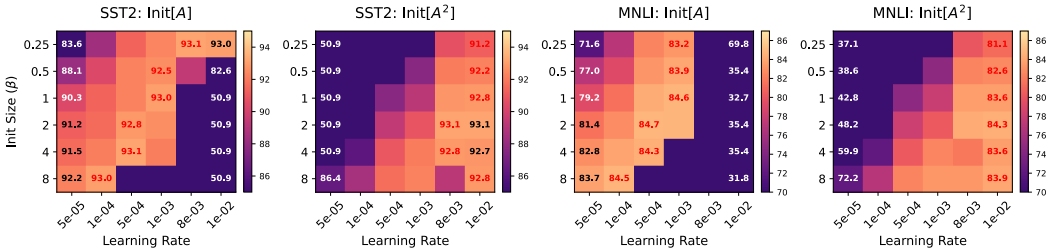


Figure 10: LoRA accuracy on MNLI and SST-2 with frozen A and varying B LR.

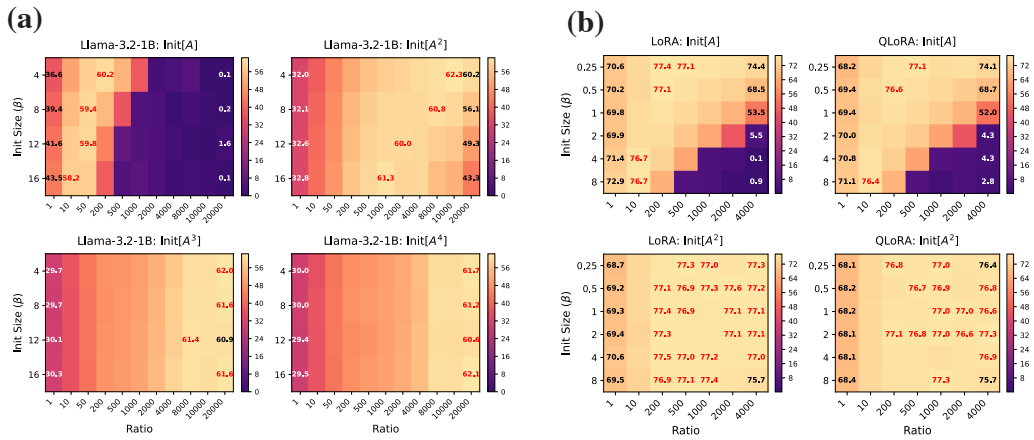


Figure 11: Additional Llama arithmetic experiments. (a) Llama 3.2-1B heatmaps compare `Init[A]`, `Init[A`^{2}`]`, `Init[A`^{3}`]`, and `Init[A`^{4}`]` over initialization scale $\beta \in \{4, 8, 12, 16\}$ and LR ratio $\rho = \eta_B/\eta_A \in \{1, 10, 50, 200, 500, 1000, 2000, 4000, 8000, 10000, 20000\}$. (b) Llama 3-8B heatmaps compare full-precision LoRA and QLoRA under `Init[A]` and `Init[A`^{2}`]` over $\beta \in \{0.25, 0.5, 1, 2, 4, 8\}$ and $\rho \in \{1, 10, 200, 500, 1000, 2000, 4000\}$. In both panels, each ratio uses $\eta_A = 10^{-5}$ and $\eta_B = \rho\eta_A$. QLoRA exhibits the same high-ratio sensitivity under `Init[A]` as full-precision LoRA, while `Init[A`^{2}`]` preserves a broad high-accuracy region in both settings.

shows that `Init[A`^{2}`]` reaches high accuracy at the largest tested η_B values across β, whereas the best η_B for `Init[A]` shifts downward as β increases.

D.3 Additional Llama Experiments

We further test whether the initialization behavior persists under quantized fine-tuning, additional model families, and smaller Llama backbones. Unless stated otherwise, these runs follow Appendix C.4 with $r = \alpha = 8$, AdamW, the same LoRA modules, and arithmetic evaluation. For QLoRA [22], the pretrained backbone is quantized while gradients update the LoRA adapters.

Panel (b) of Figure 11 shows similar LR-ratio sensitivity for LoRA and QLoRA. Under `Init[A]` at $\eta_B/\eta_A = 2000$, raising β from 0.5 to 8 lowers accuracy from 71.9% to 1.8% for LoRA and from 70.7% to 4.5% for QLoRA. Under `Init[A`^{2}`]`, LoRA reaches 80.2% at $\beta = 0.5$ and stays within 77.7–79.3% at ratio 2000 across tested β; QLoRA reaches 79.5% and retains roughly 77–79% at large ratios. Quantization thus preserves the high-ratio failure under `Init[A]` and the broad high-accuracy region under `Init[A`^{2}`]`.

Figure 12 shows the same effect across various models. Under standard `Init[A]`, increasing the LR ratio becomes fragile as β grows: Llama 3.2-1B, Qwen2.5-3B, Qwen3-0.6B, and Mistral-v0.3 all collapse to near-zero accuracy in the high-ratio, large-β region, and Llama 3.2-3B shows the same degradation pattern at $\beta \geq 4$. In contrast, `Init[A`^{2}`]` removes most of this diagonal failure region. The high-ratio columns remain competitive across the model families, reaching roughly 58–60% on Llama 3.2-1B, 72–74% on Llama 3.2-3B, 57–66% on Qwen2.5-3B, 72–74% on Qwen3-0.6B, and

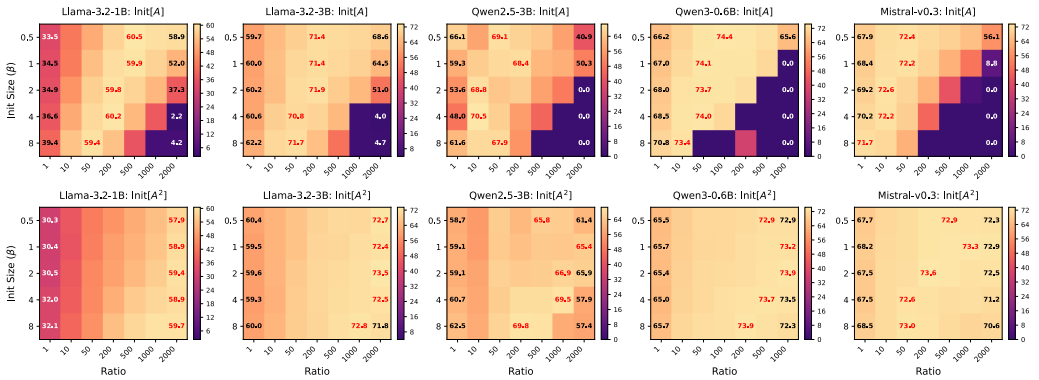


Figure 12: Arithmetic accuracy across model families under standard `Init[A]` (top) and `Init[A`^{2}`]` (bottom). Columns correspond to Llama 3.2-1B, Llama 3.2-3B, Qwen2.5-3B, Qwen3-0.6B, and Mistral-v0.3. Each heatmap sweeps initialization scale $\beta \in \{0.5, 1, 2, 4, 8\}$ on the vertical axis and the LR ratio $\rho = \eta_B/\eta_A$ on the horizontal axis, with ratio values shown in each panel; each ratio uses $\eta_A = 10^{-5}$ and $\eta_B = \rho\eta_A$.

71–73% on Mistral-v0.3. These results indicate that the noise-suppression benefit of `Init[A`^{2}`]` is not specific to one backbone, but appears consistently across Llama, Qwen, and Mistral models.

Panel (a) of Figure 11 repeats the same stress test on Llama 3.2-1B and extends the ratio grid to 20000. Standard `Init[A]` again exhibits a drifting optimum and severe high-ratio collapse: the best cells lie around ratios 10–200, while the ratio-20000 column is near zero for all tested β. `Init[A`^{2}`]` substantially improves the stable region, with peak accuracies around 60–62% at large ratios, but very large β still degrades at the extreme ratio. More aggressive scalings, `Init[A`^{3}`]` and `Init[A`^{4}`]`, further flatten the dependence on β and keep accuracy near 61–62% at the largest ratios. The tradeoff is visible at small ratios: these stronger scalings start near 30% at ratio 1, indicating that excessive suppression of the initial A scale requires a larger η_B/η_A ratio to recover signal learning. Overall, the smaller-model experiment supports the same mechanism: lower initialization variance suppresses high-ratio noise, while the exponent balances noise control and adaptation speed.

D.4 Long-Horizon Arithmetic Fine-Tuning

The long-horizon runs compare training budgets of up to 10 epochs. Each horizon is trained separately from initialization, with its LR schedule set for that horizon. We first use a Llama 3-8B arithmetic stress configuration with $r = \alpha = 8$, $\beta = 1$, $\eta_A = 10^{-5}$, and $\rho = \eta_B/\eta_A = 4000$; other settings follow Appendix C.4.

Table 3: Separate training-horizon runs on Llama 3-8B arithmetic at $\rho = 4000$. Entries report final accuracy (%) / training loss; the gap is `Init[A`^{2}`]` minus `Init[A]` accuracy.

Epoch	`Init[A]`	`Init[A`^{2}`]`	Accuracy gap
1	55.06 / 0.687	75.61 / 0.348	+20.55
3	56.71 / 0.697	77.57 / 0.268	+20.86
6	55.73 / 0.687	77.30 / 0.182	+21.58
10	55.58 / 0.661	75.68 / 0.128	+20.10

Across the four horizons in Table 3, the reported accuracy gap is about 20–22 points. Standard `Init[A]` remains below `Init[A`^{2}`]` in the 10-epoch run, while `Init[A`^{2}`]` reaches lower final training loss at longer horizons. Thus, the gap persists across the tested training budgets; these runs do not establish that all `Init[A]` configurations behave similarly.

We next compare training horizons on Llama 3-8B, using the nominal LRs and initialization scale from Figure 6: $\beta = 0.5$, $\eta_A = 10^{-5}$, and $\eta_B = 2 \times 10^{-3}$ with $\alpha/r \in \{4, 12\}$. Each run has its own horizon-dependent schedule.

Table 4: Training-horizon comparison for the Llama 3-8B multiplier experiment. Each entry reports final arithmetic accuracy (%) / training loss from a separate run.

Epoch	$\alpha/r = 4$ `Init[A]`	$\alpha/r = 4$ `Init[A²]`	$\alpha/r = 12$ `Init[A]`	$\alpha/r = 12$ `Init[A²]`
1	75.40 / 0.360	74.43 / 0.360	71.33 / 0.419	75.24 / 0.356
3	75.27 / 0.256	77.40 / 0.294	71.18 / 0.339	76.89 / 0.279
5	74.39 / 0.199	76.34 / 0.239	70.57 / 0.298	75.75 / 0.216
10	68.72 / 0.135	75.63 / 0.144	68.33 / 0.249	72.64 / 0.132

At $\alpha/r = 4$, `Init[A]` is 0.97 points higher in the 1-epoch run, but `Init[A²]` is higher at the longer reported horizons. At $\alpha/r = 12$, `Init[A²]` is higher at every horizon. Both methods reach lower training loss with longer budgets, while an accuracy gap remains. These separate runs vary both duration and schedule and lack seed-level uncertainty; they do not isolate early stopping as a cause. They provide a limited horizon comparison, with the broader robustness claim supported by the full sweeps.

D.5 Multi-Seed Accuracy and Tuning Robustness

Five-seed multiplier validation. We repeat selected multiplier configurations on Llama 3-8B arithmetic fine-tuning with five seeds, $r = 8$, $\beta = 0.5$, $\eta_A = 10^{-5}$, and $\rho = \eta_B/\eta_A = 200$. Table 5 reports accuracy and the 95% confidence interval for the `Init[A²]`-minus-`Init[A]` difference.

Table 5: Five-seed Llama 3-8B arithmetic accuracy (%) at fixed LR ratio $\rho = 200$. Accuracy entries are mean $\pm$ SD. The final column gives the reported 95% CI for the accuracy difference, in percentage points.

α/r	`Init[A]`	`Init[A²]`	95% CI for `Init[A²]` − `Init[A]`
1	76.35 ± 0.63	76.42 ± 0.67	$[-0.28, 0.43]$
3	75.71 ± 0.23	76.71 ± 0.70	$[0.15, 1.84]$
10	71.69 ± 0.70	76.42 ± 0.16	$[3.80, 5.65]$
12	71.44 ± 0.52	76.87 ± 0.48	$[4.68, 6.20]$

At $\alpha/r = 1$, the interval includes zero, so the small difference does not support an improvement claim. At $\alpha/r = 10$ and 12, the intervals remain above zero: the large-multiplier gap persists across seeds. These fixed-LR results support stress-regime robustness without establishing best-tuned performance.

Five-seed usable-region validation. For RoBERTa-base on SST-2, we run five seeds at each of 25 LR-grid cells per initialization with $\beta = 8$. Table 6 reports peak accuracy and usable-cell counts.

Table 6: Five-seed RoBERTa/SST-2 LR-grid summary at $\beta = 8$. Best accuracy (%) is mean $\pm$ SD. Counts are reported out of 25 cells; the near-best threshold is one percentage point below each method's own best.

Initialization	Best accuracy	Within 1 point of own best	Above 90% accuracy	Chance-level cells
`Init[A]`	93.28 ± 0.64	5/25	8/25	14/25
`Init[A²]`	93.53 ± 0.52	8/25	16/25	5/25

The best scores are similar relative to the reported uncertainty, while `Init[A²]` increases near-best cells from 5 to 8, doubles cells above 90% accuracy, and reduces chance-level cells from 14 to 5. This supports a wider usable LR region rather than a universal gain in peak accuracy.

Three-seed matched LR retuning. To distinguish initialization robustness from a fixed-LR mismatch, we retune η_B in the multiplier experiment while keeping $\eta_A = 10^{-5}$. Both methods search the same ratios $\rho \in \{0.1, 0.5, 1, 50, 100\}$ with three seeds per cell; Table 7 reports the best settings.

Table 7: Three-seed matched LR retuning at two LoRA multipliers. Accuracy (%) is mean ± SD; the best ratio is selected from the same grid for both methods.

α/r	Initialization	Best ρ	Best accuracy
4	`Init[A]`	50	76.39 ± 0.62
4	`Init[A`^{2}`]`	50	76.83 ± 0.36
12	`Init[A]`	50	75.51 ± 0.34
12	`Init[A`^{2}`]`	100	76.84 ± 0.31

At $\alpha/r = 4$, the best scores are similar after retuning. At $\alpha/r = 12$, retuning recovers much of the fixed-ratio loss of `Init[A]`, while `Init[A`^{2}`]` remains 1.33 points higher within the searched grid. Together, these experiments separate robustness to fixed aggressive HPs from the best attainable score after matched tuning. Most other LR-ratio sweeps remain single-run, so small differences in those sweeps should not be interpreted as established improvements.

D.6 Signed Signal–Noise Energy Measurements

We measure the realized noise, signal, and signed cross energies defined after Lemma 1 on Llama 3.2-1B and Llama 3-8B, using Math-10K, three seeds, and three epochs. Table 8 reports each component's final mean percentage contribution to E_{total}. The identity was checked to relative error below 6×10^{-7}, confirming numerical consistency but not the mechanism alone.

Table 8: Final mean energy contributions (%) over three seeds. Noise and signal energies are nonnegative, while the cross term is signed; noise can exceed 100% when the cross term is destructive. Rounded entries need not sum to exactly 100%.

Model	Initialization	Noise	Signal	Cross
Llama-3.2-1B	`Init[A]`	117.2	4.9	-22.1
Llama-3.2-1B	`Init[A`^{2}`]`	3.1	85.9	$+11.1$
Llama 3-8B	`Init[A]`	117.4	8.1	-25.5
Llama 3-8B	`Init[A`^{2}`]`	1.5	90.4	$+8.1$

Under `Init[A]`, initialization noise dominates and the cross term is destructive. Under `Init[A`^{2}`]`, the signal contributes approximately 86–90% of total energy and the cross term is constructive. The cross term is nonzero under both initializations, supporting the use of deterministic dominance bounds rather than cancellation. The noise/signal ordering emerges within the first 5% of an epoch and persists through training, consistent with Figure 2. These two finite widths show a transition to signal dominance, without directly verifying the asymptotic $1/n$ noise-suppression rate.

E Comparison with Other LoRA Initializations

We report matched-setting comparisons that avoid method-specific tuned LR advantages.

E.1 Comparison with `Init[AB]`

Li et al. [16] apply the γ-calculus to derive another initialization strategy, `Init[AB]`, that can be more robust to suboptimal LRs. The robustness is attributed to the condition:

$$\gamma[A_0] + \gamma[B_0] = -1.$$

Strictly speaking, since $B_0 = 0$, $\gamma[B_0] = -\infty$ under the γ-calculus convention. The relevant comparison is instead with the first nonzero B update, whose target/update scale under $\eta_B = \Theta(1)$ is $\gamma[\Delta B] = 0$. Thus `Init[A`^{2}`]` attains the desired post-update scaling, $\gamma[A_0] = -1$ and $\gamma[\Delta B] = 0$, with zero initial adapter output. Its two advantages over `Init[AB]` are:

1. **Internal Stability**: `Init[A`^{2}`]` maintains bounded internal features Z_A, preventing the feature explosion characteristic of `Init[AB]` (Figure 1)

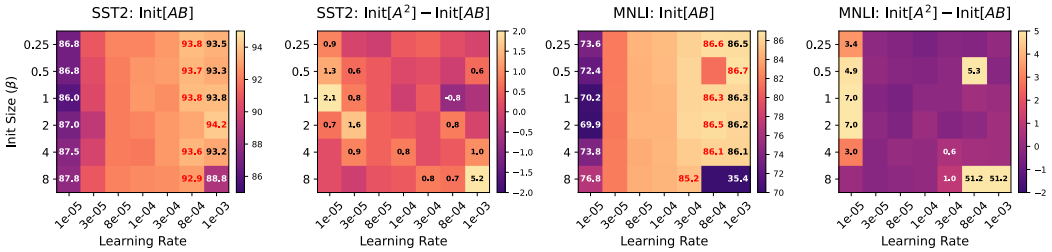


Figure 13: MNLI and SST-2 accuracy under `Init[A²]` and `Init[AB]` with symmetric LRs ($\eta_A = \eta_B$). Annotated cells have gaps $|\Delta| > 0.5\%$.

Table 9: Average accuracy (%) on the math benchmark across initialization methods under equal LRs $\eta_A = \eta_B = \eta$. Best result in each column is bolded.

Initialization	1e−6	3e−6	1e−5	3e−5	1e−4	3e−4	1e−3	3e−3	8e−3
EVA	24.23	29.36	34.01	38.82	46.01	55.95	61.78	60.15	0.00
OLoRA	29.19	30.29	33.19	38.61	52.13	58.23	60.07	58.22	7.28
PiSSA	30.36	**34.03**	**40.25**	47.19	54.77	58.42	60.97	58.80	**22.30**
`Init[A]` ($\beta = 1$)	23.51	28.69	34.74	40.91	48.26	57.83	**62.15**	60.72	2.41
`Init[A]` ($\beta = 8$)	**30.88**	32.85	39.56	**47.90**	**56.08**	**60.27**	60.15	57.10	0.53
`Init[A²]` ($\beta = 1$)	24.13	28.33	31.02	35.22	45.51	51.48	60.77	**60.98**	2.14
`Init[A²]` ($\beta = 8$)	23.35	28.57	31.84	36.72	45.62	52.81	61.93	60.43	5.43
`Init[AB]` ($\beta = 1$)	23.00	31.49	35.38	42.38	52.43	58.78	61.64	60.22	4.09
`Init[AB]` ($\beta = 8$)	16.42	0.91	2.86	0.00	2.01	4.13	5.43	4.19	1.25

2. **Implementation Efficiency**: Unlike `Init[AB]`, which necessitates nonzero initialization for both matrices and subsequent weight subtraction to preserve the pretrained mapping, `Init[A²]` allows $B = 0$. This simplifies implementation and avoids inference overhead.

To validate these advantages, we compare the two strategies on SST-2 and MNLI across varying initialization scales $\beta \in \{0.25, 0.5, 1, 2, 4, 8\}$ and symmetric LRs $\eta_A = \eta_B = \eta \in \{1 \times 10^{-5}, 3 \times 10^{-5}, 8 \times 10^{-5}, 1 \times 10^{-4}, 3 \times 10^{-4}, 8 \times 10^{-4}, 1 \times 10^{-3}\}$. In Figure 13, the methods perform similarly in standard regimes, while `Init[A²]` has higher accuracy at the extreme LRs $(10^{-5}, 10^{-3})$ when $\beta = 8$. This is consistent with greater internal stability under `Init[A²]`.

E.2 Equal-LR Sweep on Math Benchmark

Table 9 shows that no initialization dominates across equal LRs; the best method varies with the LR. The smaller initial A scale can slow early adaptation for `Init[A²]`, but it remains competitive near the best equal-LR results and avoids the severe large-β degradation of `Init[AB]`.

E.3 Variance-Compressed Structured Initializers under Asymmetric LRs

The `Init[A²]` philosophy is not tied to Gaussian initialization. For a layer with fan-in n_ℓ, define

$$\mathrm{RMS}(A_{0,\ell}) := \frac{\|A_{0,\ell}\|_F}{\sqrt{r n_\ell}}.$$

An A^2-style variant targets $\mathrm{RMS}(A_{0,\ell}) = \beta/n_\ell$ while retaining the base initializer's structure. We apply this to three initializers:

- **Orthogonal-A^2.** For $Q_A \in \mathbb{R}^{r \times n_\ell}$ with orthonormal rows, set $A^{(2)}_{0,\ell} = (\beta/\sqrt{n_\ell})Q_A$ and $B^{(2)}_{0,\ell} = 0$. Then $\mathrm{RMS}(A^{(2)}_{0,\ell}) = \beta/n_\ell$ with orthogonal A-directions and zero initial adapter output.

- **EVA-A^2.** Rescale EVA's activation-derived $A_{0,\ell}$ [31] by $c_\ell = (\beta/n_\ell)/\operatorname{RMS}(A_{0,\ell})$. Set $A^{(2)}_{0,\ell} = c_\ell A_{0,\ell}$ and $B^{(2)}_{0,\ell} = 0$, preserving its row space.
- **PiSSA-A^2.** PiSSA initializes both factors from leading singular components of the pretrained weight and compensates for their initial product in the stored base weight [30]. We use the same c_ℓ but apply the gauge transformation $A^{(2)}_{0,\ell} = c_\ell A_{0,\ell}$ and $B^{(2)}_{0,\ell} = c_\ell^{-1} B_{0,\ell}$. Hence $B^{(2)}_{0,\ell} A^{(2)}_{0,\ell} = B_{0,\ell} A_{0,\ell}$: the PiSSA subspaces, initialized low-rank product, and initial model function are preserved, although the optimization geometry changes.

We evaluate the three base initializers and their variance-compressed variants on Llama 3-8B arithmetic fine-tuning using the setup in Appendix C.4. We sweep $\rho = \eta_B/\eta_A \in \{1, 200, 4000\}$ and use $\beta \in \{0.5, 1\}$ for the compressed variants.

Table 10: Arithmetic accuracy (%) for structured LoRA initializers and their variance-compressed A^2 variants under asymmetric LR ratios. Results are single-run values; the best result within each initializer family and ratio is bolded.

Initializer	β	$\rho = 1$	$\rho = 200$	$\rho = 4000$
`Init[A²]`	0.5	**71.44**	**76.72**	76.70
`Init[A²]`	1	71.24	76.63	**77.93**
PEFT Orthogonal	–	**72.73**	70.06	2.49
Orthogonal-A^2	0.5	71.35	76.76	77.72
Orthogonal-A^2	1	71.66	**77.23**	**77.74**
EVA	–	**75.19**	74.26	5.10
EVA-A^2	0.5	71.56	**77.28**	**77.19**
EVA-A^2	1	71.82	76.85	76.53
PiSSA	–	**72.87**	**74.80**	48.10
PiSSA-A^2	0.5	68.90	69.52	69.90
PiSSA-A^2	1	72.74	73.59	**73.80**

At $\rho = 4000$, the uncompressed Orthogonal, EVA, and PiSSA initializers fall to 2.49%, 5.10%, and 48.10%, while their best compressed variants retain 77.74%, 77.19%, and 73.80%. Compression is not uniformly beneficial at $\rho \in \{1, 200\}$. These single-run results suggest that controlling A_0 scale helps structured initializers tolerate aggressive asymmetric LRs; small differences remain uncertain.

F Responsible Research Statements

F.1 Broader Impact

We study LoRA initialization to improve the understanding and robustness of parameter-efficient fine-tuning. We identify no direct negative societal impacts unique to this initialization; broader impacts depend on the underlying pretrained models and their downstream uses.

F.2 LLM Usage

We used large language models to polish the language and improve presentation clarity. We also use Codex to clean up the code and maintain the code base. The core research ideas, technical content, theoretical analysis, experimental design, and scientific conclusions were generated by the authors.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the proposed `Init[A`^{2}`]` initialization, the signal–noise mechanism, and the empirical claims. The analysis and experiments in the main paper and appendix support these claims within the stated assumptions and evaluated settings.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The conclusion discusses open limitations, including the focus on supervised fine-tuning, the lack of a direct rule for choosing asymmetric LR ratios across arbitrary models and tasks, the need for a fuller theory of other LoRA HPs such as rank r, and the uncertainty around more aggressive initialization scaling.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The main text states the informal theorem assumptions, and the appendix provides the full decomposition, normalized-update assumptions, Adam/AdamW conditions, and proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The experimental setup, datasets, model choices, LoRA HPs, optimizer settings, and sweep ranges are described in the main experiment section and in the experimental-details appendix. Reconstruction details and limitations are documented in the public experiment guide and Appendix C.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper uses public datasets and models. Code, setup instructions, and data-access recipes are publicly available at `https://github.com/liaodisen/LoRA-initA2`. The experimental setup and hyperparameter sweeps are documented in Appendix C; the public experiment guide documents reconstruction details and limitations.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The main text and appendix specify the benchmark tasks, models, LoRA modules, rank, scaling factor, LR sweeps, optimizers, batch sizes, sequence lengths, and training epochs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The appendix reports selected three- and five-seed accuracy summaries and confidence intervals, while most main language-understanding and generation heatmaps remain single-run. The reported uncertainty types and interval-construction methods are not yet fully documented.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The appendix reports the GPU type used for the experiments. Each individual fine-tuning run is relatively cheap and can be completed within roughly three hours on a single L40s GPU, but the full set of reported experiments requires substantially more total compute because we sweep many combinations of initialization scale, LR ratio, rank, and LoRA multiplier.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work studies initialization and optimization behavior for parameter-efficient fine-tuning and does not involve data collection from human subjects, unsafe data release, or intentionally harmful applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader-impact section states that the work aims to improve understanding and robustness of LoRA and that any downstream impacts are inherited from the underlying pretrained models and applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release new pretrained models, high-risk datasets, or other assets requiring release safeguards.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [No]

Justification: The paper cites the existing datasets, models, and software used in the experiments, but the current draft does not explicitly list their licenses or terms of use.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: We do not release a new dataset, pretrained model, or benchmark. The public repository provides the implementation of the proposed method.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The research does not involve crowdsourcing experiments or human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The research does not involve crowdsourcing experiments or human subjects, so IRB approval is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are the experimental backbone models, but LLM usage is not an original or non-standard component of the proposed method development.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.