The Hidden Ratio in Adam: Stable Structure, Compression, and Sign Dynamics

Yihe Zhou[1] **Tongtian Zhu**[2] **Yingxiao Huo**[1] **Satya Prakash Dash**[1]
Can Wang[2] **Samuel Kaski**[3,4,1] **Mingfei Sun**[1]

[1]Department of Computer Science, The University of Manchester, Manchester, UK
[2]College of Computer Science and Technology, Zhejiang University, Hangzhou, China
[3]ELLIS Institute Finland
[4]Aalto University

{yihe.zhou,yingxiao.huo,satyaprakash.dash}@postgrad.manchester.ac.uk
{samuel.kaski,mingfei.sun}@manchester.ac.uk
{raiden,wcan}@zju.edu.cn

Abstract

Adam is the default optimizer for training modern deep neural networks, yet its adaptive behavior remains poorly understood due to the complex interaction between its first- and second-moment exponential moving averages (EMAs). We study Adam in the tied-β regime, where the two EMA decay rates are equal, and show that its adaptive dynamics can be expressed through a transformed ratio with approximately scale-stable behavior. Empirically, this transformed ratio exhibits a stable, heavy-tailed distribution across tasks, model scales, and training stages, in contrast to the variability of raw moment magnitudes. This empirical stability has both practical and conceptual consequences. First, we derive a recurrence for the transformed ratio, yielding a reparameterization of Adam that replaces the second moment with a compressible state. Leveraging its stable distribution, we show that a fixed 4-bit codebook is sufficient in our experiments to store this state without auxiliary scaling, achieving performance competitive with full-precision Adam. Second, the transformed ratio view clarifies Adam's connection to sign-based methods: Adam reduces to sign-based momentum modulated by the transformed ratio, and replacing it with a constant recovers Signum as a limiting case. This perspective further provides a simple rule for transferring learning rates between the two methods. Together, these results suggest that tied-β Adam admits a simple and approximately stable ratio structure underlying its adaptive behavior and demonstrate its utility for both analysis and efficient implementation. Code is available at https://github.com/agent-lab/hidden-ratio-adam.

1 Introduction

Adam [23, 30] is the default optimizer for training deep neural networks, from vision models [14, 28] to large language models [10, 5, 41, 38, 34, 1]. Yet, despite its ubiquity, a fundamental question remains unresolved: *what determines Adam's adaptive behavior across scales, tasks, and training regimes?* In particular, the interaction between its two exponential moving averages (EMAs) remains difficult to interpret. Specifically, Adam maintains EMAs of the gradient mean m_t and second moment v_t, controlled by β_1 and β_2. The update normalizes a momentum-like direction by a scale estimate of recent gradients, producing coordinate-wise step sizes. While this mechanism is simple, its dynamics depend intricately on gradient scale and noise, obscuring a clean, invariant characterization.

Recent large-scale training increasingly uses closer decay rates, e.g., $(0.9, 0.95)$ [19, 22, 35, 7], motivating the *tied-β* regime $\beta_1 = \beta_2$. This regime is both practically relevant and structurally

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

simpler, as Adam admits a clearer interpretation: its update direction resembles a smoothed sign-based momentum, while the adaptive denominator modulates the step magnitude through a noise-to-signal ratio of stochastic gradients [33]. Complementary analyses suggest that matching the two decay rates introduces useful structural properties, including robustness to gradient scaling and improved stability under nonstationary noise [12, 16, 6].

In this work, we show that tied-β Adam admits an unexpected simplification: its adaptive dynamics can be reparameterized through an approximately scale-stable ratio. Specifically, decomposing the second moment and factoring out the shared β-dependent scale yields a natural-scale ratio y_t (i.e., *the hidden ratio*) that isolates stochastic fluctuations from deterministic scaling. Our key finding is that y_t follows a remarkably stable distribution across models, tasks, and training stages. Unlike raw moment magnitudes, which vary with gradient scale, the distribution of y_t is concentrated with a persistent heavy right tail. This suggests a simple underlying structure: tied-β Adam combines a deterministic scale with an approximately stable law governing adaptive attenuation.

This perspective has two immediate consequences. The first is representational. In the tied-β regime, Adam can be equivalently parameterized by (m_t, y_t) instead of (m_t, v_t). Since y_t has a stable and compressible distribution, it admits aggressive quantization. We show that a fixed 4-bit codebook (16 levels) suffices to nearly match full-precision Adam performance, without requiring additional block-wise or tensor-wise scaling factors. To our knowledge, this is the first result demonstrating that a transformed second-moment-derived state in Adam can be quantized to 4-bit precision with a fixed codebook while retaining strong performance. The second consequence is both practical and conceptual. The ratio formulation clarifies Adam's relationship to Signum (SignSGD with momentum) [40, 3]. In the tied-β regime, Adam can be viewed as a sign-based momentum method whose step magnitude is attenuated by y_t. Averaging this attenuation under the stable distribution of y_t yields a simple prediction for the effective learning-rate ratio between Adam and its sign-dominated counterpart, which aligns with our empirical observations. Furthermore, replacing y_t with a constant reduces Adam to a Signum-type method with fixed attenuation, showing that a constant-state approximation recovers a Signum-like limit of tied-β Adam. This view provides a principled way to transfer or initialize learning rates between the two methods.

We do not claim that tied-β Adam universally dominates standard Adam configurations or that the transformed-ratio law fully characterizes optimizer dynamics globally. Rather, our contributions are:

- We identify a scale-stable ratio y_t in tied-β Adam with a stable, heavy-tailed distribution across tasks and models. Truncating or coarsely discretizing y_t incurs negligible performance loss.
- We identify and derive a recursive characterization of y_t, enabling a (m_t, y_t) parameterization and a 4-bit fixed-codebook implementation without auxiliary scaling.
- The ratio view yields a learning-rate transfer rule and recovers Signum as a constant-state limit.

2 Background and Related Work

Adam [23] and Signum [3, 40]. As an iterative adaptive optimizer, **Adam** maintains exponential moving averages of the gradient g_t and its square at each iteration t,

$$m_t := \beta_1 m_{t-1} + (1-\beta_1) g_t, \qquad v_t := \beta_2 v_{t-1} + (1-\beta_2) g_t^2. \tag{1}$$

Since both averages are initialized to zero, Adam uses the bias-corrected elementwise scaling:

$$\hat{m}_t := \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t := \frac{v_t}{1-\beta_2^t}, \qquad \Delta\theta_t := -\eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}, \tag{2}$$

where ϵ is a small positive constant for numerical stability. Thus, Adam can be viewed as normalizing an exponential estimate of the gradient mean by an exponential estimate of its root second moment. In contrast, **Signum** maintains a first-moment EMA and applies a sign update:

$$m_t := \beta m_{t-1} + (1-\beta) g_t, \qquad \Delta\theta_t := -\eta \operatorname{sign}(m_t). \tag{3}$$

The update keeps only the direction of the momentum and uses a fixed coordinatewise step magnitude. A growing line of work has sought to understand Adam through structural interpretations of its update rule, rather than only through empirical tuning heuristics. Early analyses separated the roles of momentum and adaptive scaling, interpreting Adam as combining a sign-based directional component

with a variance-controlled magnitude term [2], or more generally disentangling momentum from adaptivity in Adam-style methods [29]. More recent work further expresses Adam's adaptive scaling through a local signal-to-noise ratio, giving the denominator a direct statistical interpretation [33]. Under this view, Signum [3, 40] can be related to steepest descent under an ℓ_∞ trust-region geometry, while Adam corresponds to a signed-momentum direction whose effective step magnitude is modulated by local signal-to-noise structure. Our work follows this structural perspective, but identifies within Adam's scaling term a comparatively stable distributional quantity that can be represented directly.

Tied-β regime. Our analysis focuses on the tied-β case, $\beta_1 = \beta_2 = \beta$. Ignoring bias correction and ϵ for clarity, the adaptive factor reduces to $\frac{m_t}{\sqrt{v_t}}$. Because m_t and v_t are then computed with the same exponential weights, $v_t - m_t^2$ has a natural variance-like interpretation. In fact, $v_t \geq m_t^2$ for all gradient sequences and all t if and only if $\beta_1 = \beta_2$; the proof is deferred to Appendix B. In this case, Adam admits a particularly clean online mean–variance interpretation [33], and satisfies a first-order gradient scale-invariance property that singles out tied β values as a structurally meaningful choice [16]. Analyses of mini-batch noise further suggest that the preferred relation between β_1 and β_2 depends on the noise regime, with larger batch sizes often favoring β_1 closer to β_2 in terms of validation performance [6]. Complementary empirical studies in nonstationary optimization and reinforcement learning indicate that mismatched first- and second-moment timescales can weaken normalization and destabilize updates [15, 12], whereas tied or near-tied decay rates can improve stability in practice [32, 13, 18]. Motivated by these observations, we focus on $\beta_1 = \beta_2$ when analyzing Adam's ratio recursion and distributional behavior, since this is the regime where prior work suggests both cleaner dynamics and improved stability.

3 Hidden Ratio: Theoretical and Empirical Properties

In the tied-β regime, define

$$r_t := \frac{v_t - m_t^2}{m_t^2}, \qquad \frac{m_t}{\sqrt{v_t}} = \frac{\operatorname{sign}(m_t)}{\sqrt{1 + r_t}}.$$

Thus the update direction is determined by the sign of the momentum, while r_t controls the magnitude attenuation. Since $r_t \geq 0$ under tied β values, the denominator is bounded below by 1 [33]. We now define the **transformed ratio**

$$y_t := \frac{1-\beta}{\beta} r_t, \qquad \frac{m_t}{\sqrt{v_t}} = \frac{\operatorname{sign}(m_t)}{\sqrt{1 + \frac{\beta}{1-\beta} y_t}}. \tag{4}$$

This reparameterization identifies y_t as the central quantity in the Adam update, thereby motivating a closer examination of its statistical structure. Intuitively, y_t measures the variance-like residual in Adam's denominator after removing the deterministic scale induced by tied exponential averaging.

3.1 Statistical Structure of y_t: Stability and Heavy Tails

Since mini-batches are sampled independently, we follow the diffusion-approximation view of mini-batch SGD and model the local gradient noise by independent Gaussian fluctuations [31, 20, 25].

Assumption 1 (Local Gaussian window)**.** For one coordinate in a local time window, with $\sigma > 0$,

$$g_{t-j} = \mu + \sigma \xi_j, \qquad \xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,1), \qquad j \geq 0.$$

We use an infinite-history local reference, omitting initialization transients. Fix $0 < \beta < 1$ and let $w_j = (1-\beta)\beta^j$, $m_t = \sum_{j\geq 0} w_j g_{t-j}$, $v_t = \sum_{j\geq 0} w_j g_{t-j}^2$, $U_t = v_t - m_t^2$, $y_t = \frac{1-\beta}{\beta}\frac{U_t}{m_t^2}$, $s_\beta^2 = \sum_{j\geq 0} w_j^2 = \frac{1-\beta}{1+\beta}$, $\sigma_m = \sigma s_\beta$, $a = \mu/\sigma_m$, and $Z = s_\beta^{-1} \sum_{j\geq 0} w_j \xi_j$. Under Assumption 1, $Z \sim \mathcal{N}(0,1)$ and $m_t = \sigma_m(Z + a)$. When comparing many local windows, let $\mathcal{I}$ index the observed coordinate-window instances and let a_i be the local shift at index i. Treating these instances as equally weighted samples gives the empirical pooled-shift distribution

$$\widehat{P}_A := \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} \delta_{a_i}, \qquad A \sim \widehat{P}_A, \qquad \mathbb{E}_{\widehat{P}_A}[A^2] = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} a_i^2.$$

Equivalently, $A \sim \widehat{P}_A$ means drawing I uniformly from $\mathcal{I}$ and setting $A = a_I$. More generally, a pooled-shift distribution P is any distribution over local shifts used in this sampling role; the empirical case is $P = \widehat{P}_A$. For any fixed scalar shift b, write $Y_b := 2/(Z+b)^2$, so a single local window uses Y_a. If $A \sim P$ is independent of Z, the pooled denominator reference is $Y_A := 2/(Z+A)^2$.

Proposition 1 (Shifted inverse-square reference for the ratio)**.** *Under Assumption 1 and the notation above, the transformed ratio satisfies*

$$y_t = Y_a \left(1 + \frac{\eta_t}{2}\right), \qquad \eta_t = \eta_t^{\text{coup}} + \eta_t^{\text{fluc}}, \tag{5}$$

where $\gamma_\beta = \frac{1-\beta}{1+\beta+\beta^2}$, $\eta_t^{\text{coup}} := \gamma_\beta(Z^2-1)$, *and* $\eta_t^{\text{fluc}} := \frac{1-\beta}{\beta}\frac{U_t - \mathbb{E}[U_t|Z]}{\sigma_m^2}$. *Let P be a pooled-shift distribution with finite fourth moment, and let $A \sim P$ be independent of Z. Then the fixed shifted reference*

$$Y_c = \frac{2}{(Z+c)^2}, \qquad Z \sim \mathcal{N}(0,1), \qquad c^2 := \mathbb{E}_P[A^2] \tag{6}$$

matches the mixed denominator reference $Y_A = 2/(Z+A)^2$ through second order: for every fixed $k > 0$, writing $q = \sqrt{2/k}$, and denoting the standard normal CDF and density by Φ and ϕ,

$$\begin{aligned} \Pr(Y_A > k) &= 2\Phi(q) - 1 - q\phi(q)\mathbb{E}_P[A^2] + O(\mathbb{E}_P[A^4]), \\ \Pr(Y_c > k) &= 2\Phi(q) - 1 - q\phi(q)c^2 + O(c^4). \end{aligned} \tag{7}$$

In particular, for a family P_ε of pooled-shift distributions with $A_\varepsilon \sim P_\varepsilon$, $c_\varepsilon^2 := \mathbb{E}_{P_\varepsilon}[A_\varepsilon^2]$, and $\mathbb{E}_{P_\varepsilon}[A_\varepsilon^4] = O(\varepsilon^4)$, the corresponding references satisfy $\Pr(Y_{A_\varepsilon} > k) - \Pr(Y_{c_\varepsilon} > k) = O(\varepsilon^4)$.

Remark. Equation 5 is a denominator-reference decomposition: for a fixed local shift a, $Y_a = 2/(Z+a)^2$ carries the inverse-square denominator, while η_t keeps the numerator as an exact relative correction. The pooled-shift distribution P is the formal version of flattening many observed local shifts; drawing $A \sim P$ selects one representative local shift, so the denominator reference becomes $Y_A = 2/(Z+A)^2$. Equation 7 shows that the survival curve of Y_A depends to second order only on $\mathbb{E}_P[A^2]$. Choosing $c = \sqrt{\mathbb{E}_P[A^2]}$ keeps this leading shift effect and the same inverse-square singularity, which is why we use Y_c in Equation (6) as the reference for the empirical y_t.

Figure 1 illustrates this point for the reference distribution. When the shift c is small, the CDF and histogram of $Y_c = 2/(Z+c)^2$ remain close to those of the unshifted reference $2/Z^2$, preserving both the bulk and the right-tail shape. This suggests that the inverse-square denominator singularity drives the dominant distributional form, while local shifts and numerator fluctuations introduce only secondary distortions. The full derivation is given in Appendix A. This picture is reflected in the empirical distributions in Figure 2. Across pre-training, SFT, RLHF, and $\beta \in \{0.90, 0.92, 0.95\}$, the transformed ratio y_t consistently shows a compact single-digit bulk with a persistent right tail. Within each setting, the early-, middle-, and late-stage curves remain close, indicating only mild distributional drift during training. The empirical CDFs also stay close to the $2/Z^2$ reference over a broad range. Together, these observations support the view that the rescaled ratio y_t captures a natural-scale denominator state whose distribution is more stable than the raw moment magnitudes m_t and v_t.

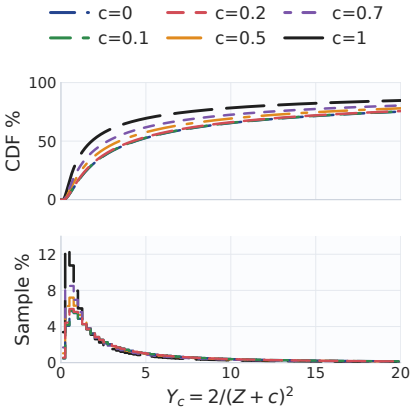


Figure 1: Shifted inverse-square reference $Y_c = 2/(Z+c)^2$. For $c \le 0.2$, the CDF and histogram are nearly unchanged; the shape is driven mainly by the inverse-square form.

3.2 Robustness of y_t to coarse quantization

Given that the distribution of y_t exhibits a clearly concentrated bulk together with a long right tail, we now investigate whether a very coarse quantization of y_t is sufficient to retain behavior and performance close to those of the original Adam. To isolate its role, we keep the original Adam

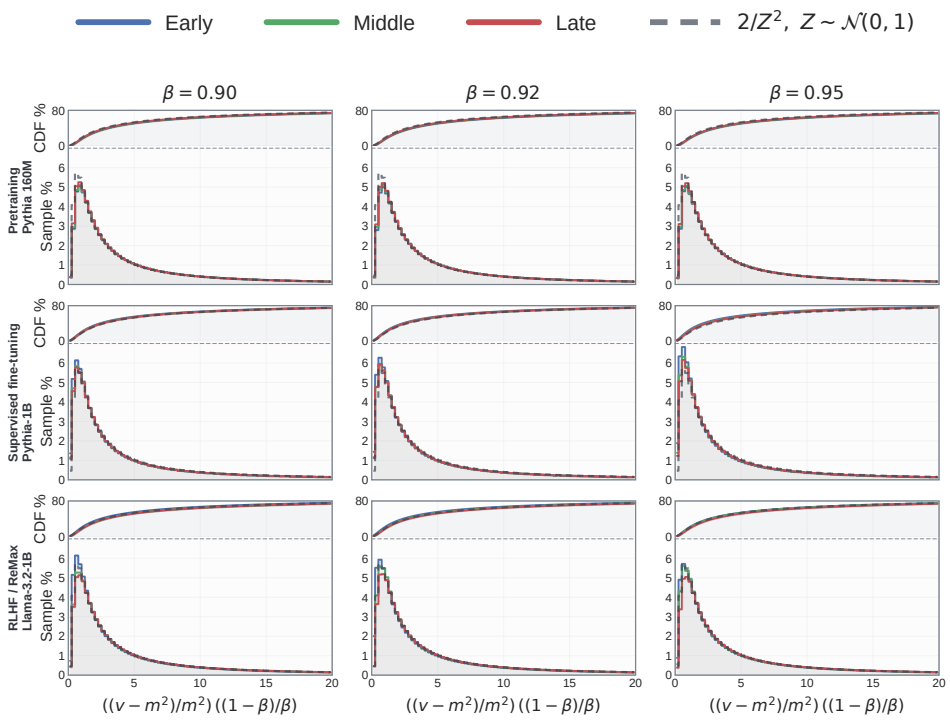


Figure 2: Distribution of the transformed ratio y_t across training regimes. Columns correspond to $\beta = 0.90, 0.92, 0.95$, while rows compare pretraining on Pythia-160M, supervised fine-tuning on Pythia-1B, and RLHF on Llama-3.2-1B-Instruct. Within each block, the top panel shows the CDF and the bottom panel shows the histogram; colors denote early, middle, and late training stages.

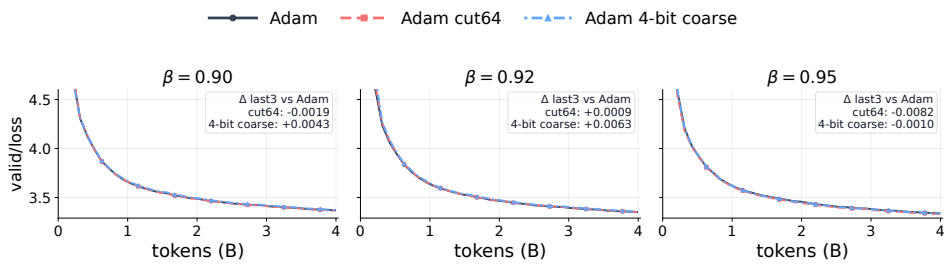


Figure 3: Comparison of the training performance of standard Adam and its modified variants based on the transformed ratio y_t, including long-tail truncation and coarse 4-bit quantization. Even after truncating the right tail and representing the remaining bulk with only 16 quantization levels, the resulting optimizer still closely tracks the behavior and final performance of the standard Adam.

recursions for m_t and v_t unchanged, compute y_t from the resulting states, and replace only this variable by a coarse low-precision approximation before reconstructing the adaptive denominator.

We begin with a particularly simple modification: truncating the upper tail of y_t. The cut64 variant replaces y_t by $\min(y_t, 64)$ before reconstructing the adaptive denominator, thereby removing only rare far-tail values. Empirically, this truncation has little effect, and the resulting optimizer still tracks standard Adam closely.

We then take a more aggressive step and replace y_t itself by a coarse low-precision approximation. After the same cut64 truncation, we quantize y_t using the fixed 4-bit codebook $\mathcal{C} = \{0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64\}$, an FP4-like non-uniform codebook with levels concentrated around the empirical bulk of y_t. The quantized value is obtained by nearest-neighbor projection: $\tilde{y}_t = Q(y_t) := \arg\min_{q \in \mathcal{C}} |y_t - q|$.

Taken together, the results in Figure 3 suggest that Adam does not require precise pointwise fidelity in y_t in order to realize its adaptive effect. What appears to matter is not the exact value of the transformed ratio at each step, but the coarse attenuation information that it carries. This is consistent

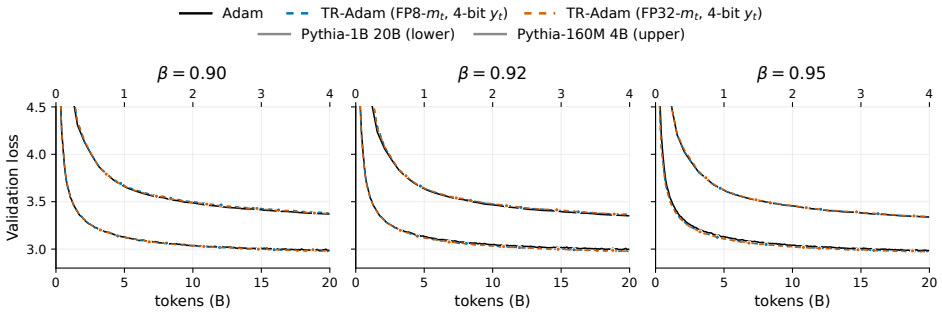


Figure 4: Pre-training comparison between full Adam and low-precision ratio representation.

with the distributional structure described above: most of the mass of y_t lies in a relatively compact regime, while the update depends on it only through the smooth factor $\left(1 + \frac{\beta}{1-\beta} y_t\right)^{-1/2}$. As a result, neither removing rare tail events nor coarsening the representation within the bulk of the distribution significantly perturbs the effective step size. This robustness motivates the ratio-based reformulation developed in the next section. If Adam's essential adaptive behavior is already preserved under severe truncation and coarse quantization of the transformed ratio, then the more natural object to analyze is not the raw moment pair (m_t, v_t), but the induced dynamics of the transformed ratio itself. We emphasize that this is only a diagnostic experiment in Figure 3: the original Adam recursions for m_t and v_t are unchanged, and only the induced y_t is coarsened before reconstructing the denominator.

4 Recursive Reformulation of Adam

Unlike the diagnostic coarsening above, we now maintain y_t recursively and use it as the stored second state. Given the stable structure of the transformed ratio, we now show that Adam can be equivalently parameterized by (m_t, y_t) instead of (m_t, v_t), motivating the low-precision state representation evaluated below.

4.1 A scalar recursion for the transformed ratio

We now derive a recursion for the transformed ratio y_t. Starting from the recursion $m_t = \beta m_{t-1} + (1-\beta) g_t$, we can rewrite the current gradient as

$$g_t = \frac{m_t - \beta m_{t-1}}{1-\beta} = \frac{m_t}{1-\beta}(1 - \beta x_t), \quad x_t := \frac{m_{t-1}}{m_t}. \tag{8}$$

Substituting this into the second-moment recursion $v_t = \beta v_{t-1} + (1-\beta) g_t^2$ gives

$$v_t = \beta v_{t-1} + \frac{m_t^2}{1-\beta}(1 - \beta x_t)^2. \tag{9}$$

Using $v_s = m_s^2 (1 + \frac{\beta}{1-\beta} y_s)$ for $s \in \{t, t-1\}$, together with $m_{t-1} = x_t m_t$, Equation (9) yields

$$y_t = \beta x_t^2 y_{t-1} + (1 - x_t)^2. \tag{10}$$

Equation 10 shows that, once the sign of m_t is fixed, Adam's adaptive behavior is well described by a one-dimensional transformed ratio recursion: the sign determines the update direction, while y_t controls the attenuation magnitude. Additional algebraic details are provided in Appendix C.

4.2 Coarse representation through the recursion

Equation 10 suggests a natural target for coarse representation. Instead of compressing the coupled raw states (m_t, v_t) directly, we represent the transformed ratio y_t, which is the scalar quantity controlling the adaptive denominator. Concretely, after obtaining y_t, we replace it by the nearest value in a fixed FP4-like codebook: $\tilde{y}_t = Q(y_t) := \arg\min_{q \in \mathcal{C}} |y_t - q|$. Here, $\mathcal{C} = \{0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64\}$. This 16-level codebook allocates

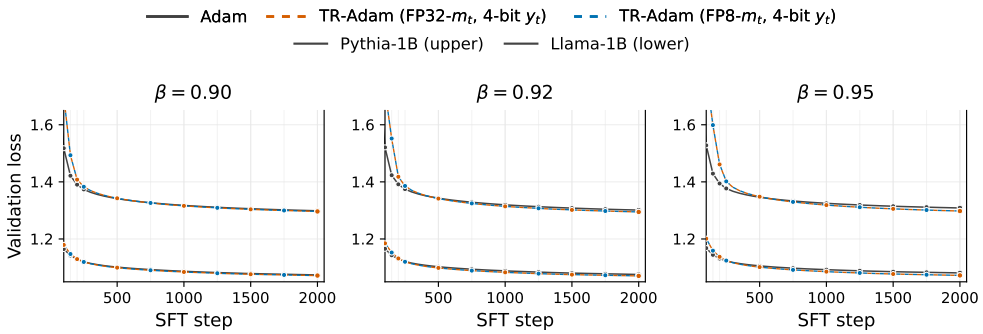


Figure 5: SFT comparison between full Adam and low-precision transformed-ratio representation.

more resolution to the moderate range where the transformed ratio distribution concentrates, while still retaining several larger values for the right tail. The adaptive denominator is then reconstructed from $\tilde{y}_t$ through the same transformed expression as in Equation (4), with y_t replaced by $\tilde{y}_t$. Thus, coarse representation acts directly on the transformed-ratio attenuation rather than separately perturbing the raw moment states m_t and v_t. Notably, this representation uses no per-block scaling factors, learned quantizers, or auxiliary normalization metadata.

4.3 Experiments on low-precision quantization of the transformed ratio

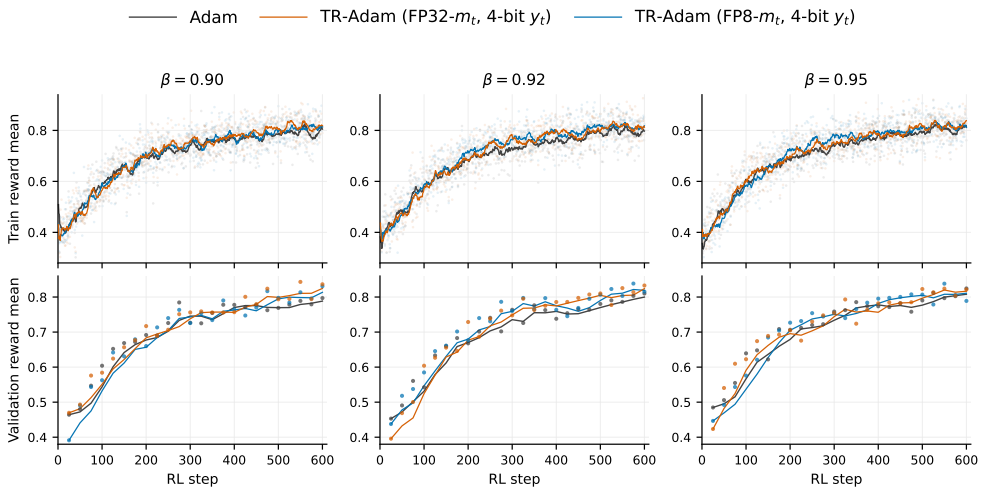


Figure 6: RLHF comparison between full Adam and low-precision transformed-ratio representation.

We next examine whether the transformed ratio representation remains effective across the full language-model training pipeline. We compare tied-β Adam with its low-precision transformed ratio variant in three representative stages: pre-training, supervised fine-tuning (SFT), and reinforcement-learning-based alignment (RLHF). The training recipe is held fixed while we vary the representation of the ratio and first-moment states. In addition to storing y_t with a 4-bit codebook, we also evaluate variants in which the first-moment state m_t is stored in FP8 [37, 8, 17]. Low-precision states are emulated in FP32 tensors: these experiments assess quantization effects, not packed-state memory or runtime savings. This is a useful stress test for the transformed-ratio formulation: although the recursion depends on ratios involving m_t, the combination of FP8 first-moment storage and 4-bit transformed-ratio storage still preserves strong performance in our experiments. Figures 4 to 6 show that the low-precision transformed-ratio variant closely tracks full Adam across all three stages. In pre-training, the two validation-loss curves nearly overlap throughout optimization and reach essentially the same final loss, suggesting that the transformed ratio preserves the denominator-side information needed for long-horizon large-scale training. In SFT and RLHF, the low-precision variant may initially lag the full-Adam baseline, but it improves steadily and often matches or slightly exceeds Adam later in training. Thus, the effect of coarse transformed-ratio representation is not merely harmless compression: in post-training regimes, it can behave like a mild smoothing or regularization of the adaptive denominator dynamics.

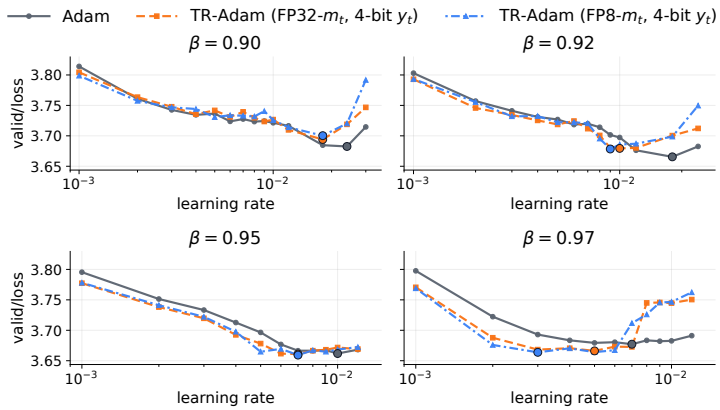


Figure 7: Learning-rate sweeps on Llama-20M pre-training comparing tied-β Adam with transformed-ratio Adam using FP32 or FP8 first-moment storage and 4-bit y_t.

We further examine whether the low-precision transformed-ratio representation preserves Adam's learning-rate sensitivity. Figure 7 shows learning-rate sweeps on a Llama-20M pre-training setup under four tied-β values. Across $\beta = 0.90, 0.92, 0.95, 0.97$, both transformed-ratio variants closely track the Adam sweep profile: their favorable learning-rate regions remain aligned with Adam, and performance degradation at overly large learning rates occurs at similar scales. This indicates that coarse transformed-ratio storage does not simply succeed after careful retuning at a single operating point; rather, it largely preserves Adam's effective learning-rate structure.

Overall, the results show that the transformed ratio y_t is a robust target for coarse representation: in pre-training it recovers Adam-like behavior almost exactly, while in SFT and RLHF it remains competitive and can sometimes improve late-stage performance. This stage-consistent behavior suggests that the essential adaptive dynamics of tied-β Adam are captured by a low-complexity transformed-ratio structure rather than by high-precision values of the raw second-moment state. More experimental results are provided in Appendix D.

5 The Signum-like Limit

5.1 A constant-state approximation and the Signum-like limit

The transformed-ratio representation provides a simple interpretation of Adam as a sign-momentum method with stochastic attenuation. From Equation (4), once the sign of the momentum is fixed, Adam's adaptive behavior is controlled by the scalar attenuation factor $(1 + (\beta/(1-\beta))y_t)^{-1/2}$. This motivates a constant-state approximation in which the transformed ratio y_t is replaced by a representative constant value $y_{\text{const}} \geq 0$. The resulting Signum-type update and the corresponding learning-rate matching rule are

$$\Delta\theta_t \approx -\eta_{\text{adam}} \left(1 + \frac{\beta}{1-\beta} y_{\text{const}}\right)^{-\frac{1}{2}} \operatorname{sign}(m_t), \quad \eta_{\text{sign}} \approx \eta_{\text{adam}} \left(1 + \frac{\beta}{1-\beta} y_{\text{const}}\right)^{-\frac{1}{2}}. \tag{11}$$

Here, the first expression is the constant-y_t Signum-type update, while the second gives the learning-rate scale needed to write it as $\Delta\theta_t^{\text{sign}} = -\eta_{\text{sign}} \operatorname{sign}(m_t)$. This relation should be viewed as a matching rule: the full Adam update still retains the coordinate-wise fluctuations and temporal variation of y_t, whereas the constant-state approximation preserves only its dominant effective scale.

To choose y_{const}, we match the expected attenuation, or equivalently the expected effective step size, under the transformed-ratio distribution:

$$\mathbb{E}\big[(1 + \tfrac{\beta}{1-\beta} y_t)^{-\frac{1}{2}}\big] = (1 + \tfrac{\beta}{1-\beta} y_{\text{const}})^{-\frac{1}{2}}. \tag{12}$$

Solving for the attenuation-matched constant gives

$$y_{\text{const}} = \frac{1-\beta}{\beta} \Big(\mathbb{E}\big[(1 + \tfrac{\beta}{1-\beta} y_t)^{-\frac{1}{2}}\big]^{-2} - 1 \Big). \tag{13}$$

Table 1: $y_{\text{const}}^{\text{eff}}$ under $Y_c = 2/(Z+c)^2$.

c	$\beta = 0.90$	$\beta = 0.92$	$\beta = 0.95$	$\beta = 0.97$
0	3.36	3.31	3.25	3.21
0.1	3.33	3.28	3.22	3.17
0.2	3.23	3.19	3.13	3.08

This calibration is useful because it connects the constant-state approximation directly to Adam's native learning-rate scale, rather than treating Signum as requiring an unrelated learning-rate search.

To make the scale of y_{const} more concrete, we substitute the shifted inverse-square reference $Y_c = 2/(Z+c)^2$ from Proposition 1 into this expression. This yields

$$y_{\text{const}}^{\text{eff}}(\beta, c) = \frac{1-\beta}{\beta}\left(\mathbb{E}_{Z\sim\mathcal{N}(0,1)}\left[\left(1+\frac{2\beta}{(1-\beta)(Z+c)^2}\right)^{-\frac{1}{2}}\right]^{-2} - 1\right). \tag{14}$$

Numerically, this value is stable for the small shifts considered in the distributional model. As shown in Table 1, the attenuation-matched constant remains slightly above 3 across $\beta \in [0.90, 0.97]$ and $c \in \{0, 0.1, 0.2\}$. Thus, under the same shifted-reference law used to describe the empirical transformed-ratio distribution, a moderate constant transformed-ratio value should already capture Adam's typical attenuation scale.

A practical advantage of this view is that it expresses a Signum-like optimizer in Adam's native learning-rate parameterization. In modern language-model training, Adam learning-rate heuristics are well established, whereas Signum often requires a separate learning-rate search. Replacing y_t with a moderate constant therefore yields a simplified sign-style update that remains calibrated to the Adam learning-rate scale, making it possible to reuse Adam-style tuning rather than introducing a separate optimizer-specific search. We therefore test several representative values of y_{const} and compare them against Adam and Signum in learning-rate sweeps across different values of β.

5.2 Learning-rate sweeps across β: empirical observations

Using the Llama-style 20M model, we now test the constant-state proxy by comparing learning-rate sweeps of Adam, Signum, and several constant-y_t variants across a range of β; see Figure 8. Two empirical trends consistently appear.

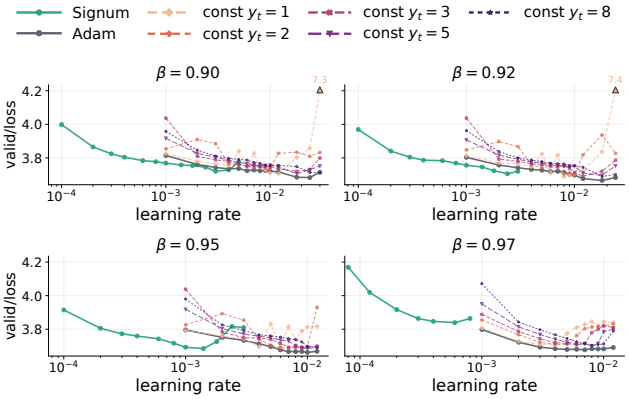


Figure 8: Learning-rate sweeps: Signum shifts to smaller learning rates as β increases, while constant-state proxies $y_{\text{const}} \in \{3, 5\}$ best preserve Adam's learning-rate scale.

First, Signum's preferred learning-rate region shifts systematically to smaller values as β increases. This behavior is consistent with the learning-rate scale predicted by Equation (11). If Adam's preferred learning-rate scale is viewed as relatively stable across β, then a pure Signum update must compensate for the transformed-ratio attenuation that Adam applies. Since the prefactor $\beta/(1-\beta)$ grows rapidly with β, the effective attenuation in Equation (11) becomes stronger at larger β. Thus, matching Adam's effective step size requires a smaller Signum learning rate. This matches the trend observed empirically in Figure 8: as β increases, the favorable learning-rate region for Signum moves markedly left relative to Adam. Second, the constant-y_t variants preserve Adam's learning-rate scale much better than pure Signum. Among the tested constants, $y_{\text{const}} \in \{3, 5\}$ gives the closest overall match to Adam's preferred learning-rate region. This is broadly consistent with the step-size-matching calculation in Equation (14) and the numerical values reported above, which place the effective constant slightly above 3 under the shifted-reference model. Although the true transformed ratio y_t is stochastic and heavy-tailed, these results suggest that its dominant effect on the optimizer's learning-rate scale can be captured by a single moderate constant.

This observation is practically useful. Directly using Signum requires a separate learning-rate search, and the appropriate learning rate changes noticeably with β. By contrast, the constant-y_t proxy remains in Adam's native learning-rate parameterization: replacing y_t by a moderate constant yields a Signum-like update whose effective step size is already calibrated to Adam. Taken together, the sweeps show that tied-β Adam behaves like a sign-based method modulated by the transformed-ratio attenuation, and that a constant approximation can preserve much of Adam's learning-rate scale.

6 Conclusion and Limitations

In this paper, we studied Adam in the tied-β regime, $\beta_1 = \beta_2$, through a transformed ratio y_t. By factoring out the explicit β-dependent scale, this representation exposes a comparatively stable ratio structure in Adam's adaptive denominator that is obscured in the raw moment states m_t and v_t. Empirically, the distribution of y_t remains consistent across training settings: most coordinates concentrate in a moderate range, while a persistent heavy tail captures rare large ratio-state values. By isolating an approximately stable state inside Adam's adaptive denominator, the ratio view yields two practical consequences. First, y_t provides a compact replacement for the raw second-moment state v_t: across the language-model training regimes studied here, a fixed 4-bit codebook for y_t preserves performance close to that of full-precision Adam. Second, the same representation clarifies Adam's connection to Signum: tied-β Adam can be viewed as a sign-momentum update attenuated by y_t, and replacing y_t by a constant yields a Signum-type limit with an explicit learning-rate scaling. Overall, these results suggest that tied-β Adam's adaptive behavior can be captured to a surprising extent through coarse but stable ratio information. Our main optimizer-performance evaluation covers Transformer language models up to 1B parameters; Appendix D.3 adds short-budget MLP and ResNet experiments and an early-stage 8B ratio-distribution check. Full-training optimizer comparisons beyond 1B and broader architecture coverage remain future work. The additional seed, decay-rate, and numerical-stability studies in Appendix D further delimit the scope of the empirical claims.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) through the AI Hub in Generative Models [grant number EP/Y028805/1]. Samuel Kaski is supported by UKRI Turing AI World-Leading Researcher Fellowship, EP/W002973/1 and EU, ERC ODD-ML 101201120.

References

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[2] Lukas Balles and Philipp Hennig. Dissecting adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, 2018.

[3] Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Anima Anandkumar. signsgd: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, 2018.

[4] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, 2023.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

[6] Matias D. Cattaneo and Boris Shigida. The effect of mini-batch noise on the implicit bias of adam. *arXiv preprint arXiv:2602.01642*, 2026.

[7] Kuang-Ming Chen, Jenq-Neng Hwang, and Hung yi Lee. InstructionCP: A simple yet effective approach for transferring large language models to target languages. In *Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, 2025.

[8] Kamran Chitsaz, Quentin Fournier, Gonçalo Mordido, and Sarath Chandar. Exploring quantization for efficient pre-training of transformer language models. In *Findings of the Association for Computational Linguistics: EMNLP*, 2024.

[9] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019.

[11] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *Conference on Empirical Methods in Natural Language Processing*, 2023.

[12] Shibhansh Dohare, Qingfeng Lan, and A. Rupam Mahmood. Overcoming policy collapse in deep reinforcement learning. In *Sixteenth European Workshop on Reinforcement Learning*, 2023.

[13] Shibhansh Dohare, J. Fernando Hernandez-Garcia, Qingfeng Lan, Parash Rahman, A. Rupam Mahmood, and Richard S. Sutton. Loss of plasticity in deep continual learning. *Nature*, 2024.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[15] Benjamin Ellis, Matthew T. Jackson, Andrei Lupu, Alexander D. Goldie, Mattie Fellows, Shimon Whiteson, and Jakob N. Foerster. Adam on local time: Addressing nonstationarity in rl with relative adam timesteps. In *Advances in Neural Information Processing Systems*, 2024.

[16] Alberto Fernández-Hernández, Cristian Pérez-Corral, Jose I. Mestre, Manuel F. Dolz, and Enrique S. Quintana-Ortí. Why adam works better with $\beta_1 = \beta_2$: The missing gradient scale invariance principle. *arXiv preprint arXiv:2601.21739*, 2026.

[17] Maxim Fishman, Brian Chmiel, Ron Banner, and Daniel Soudry. Scaling fp8 training to trillion-token llms. In *International Conference on Learning Representations*, 2025.

[18] Alexander D. Goldie, Chris Lu, Matthew T. Jackson, Shimon Whiteson, and Jakob N. Foerster. Can learned optimization make reinforcement learning less difficult? In *Advances in Neural Information Processing Systems*, 2024.

[19] Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. OLMo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

[20] Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[22] Hugging FaceTB. SmolLM3: Smol, multilingual, long-context reasoner. https://huggingface.co/blog/smollm3, 2025.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[24] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[25] Zhiyuan Li, Tianhao Wang, and Sanjeev Arora. What happens after SGD reaches zero loss? –a mathematical framework. In *International Conference on Learning Representations*, 2022.

[26] Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. ReMax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *International Conference on Machine Learning*, 2024.

[27] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

[28] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[29] Ziyin Liu, Zhikang T. Wang, and Masahito Ueda. Laprop: Separating momentum and adaptivity in adam. *arXiv preprint arXiv:2002.04839*, 2020.

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

[31] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

[32] Skander Moalla, Andrea Miele, Daniil Pyatko, Razvan Pascanu, and Caglar Gulcehre. No representation, no trust: Connecting representation, collapse, and trust issues in ppo. In *Advances in Neural Information Processing Systems*, 2024.

[33] Antonio Orvieto and Robert M. Gower. In search of adam's secret sauce. In *Advances in Neural Information Processing Systems*, 2025.

[34] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022.

[35] Jupinder Parmar, Sanjev Satheesh, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Reuse, don't retrain: A recipe for continued pretraining of language models. *arXiv preprint arXiv:2407.07263*, 2024.

[36] Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *arXiv preprint arXiv:2406.17557*, 2024.

[37] Houwen Peng, Kan Wu, Yixuan Wei, Guoshuai Zhao, Yuxiang Yang, Ze Liu, Yifan Xiong, Ziyue Yang, Bolin Ni, Jingcheng Hu, Ruihang Li, Miaosen Zhang, Chen Li, Jia Ning, Ruizhe Wang, Zheng Zhang, Shuguang Liu, Joe Chau, Han Hu, and Peng Cheng. Fp8-lm: Training fp8 large language models. *arXiv preprint arXiv:2310.18313*, 2023.

[38] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*, 2020.

[39] Jianlin Su. Are $\beta_1 = \beta_2$ the optimal hyperparameters for Adam? https://kexue.fm/archives/11593, 2026. Blog post in Chinese; title translated.

[40] Tao Sun, Qingsong Wang, Dongsheng Li, and Bao Wang. Momentum ensures convergence of signsgd under weaker assumptions. In *International Conference on Machine Learning*, 2023.

[41] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Appendix

A Distributional properties of the transformed ratio state 14
B Additional Motivation and Related Work for the Tied-β Regime 16
C Details for the transformed-state recursion and implementation 18
D Additional Experimental Results 19
E Datasets and Implementation Details 30
F Broader Impact 31

A Distributional properties of the transformed ratio state

Appendix A records the derivation behind Proposition 1. The main text states the local Gaussian model and the formal result; here we keep only the short intuition needed to read the proof. In the tied-beta regime, the transformed ratio state is

$$y_t = \frac{1-\beta}{\beta}\frac{v_t - m_t^2}{m_t^2}.$$

Under a local Gaussian window, the EMA denominator has the form $m_t = \sigma_m(Z+a)$, so the small-denominator event $Z+a \approx 0$ naturally produces the shifted inverse-square reference $2/(Z+a)^2$. The numerator does not disappear; the proof below shows that it enters as the explicit relative correction η_t in Equation (5).

When the local shift varies across coordinates or nearby windows, let $\mathcal{I}$ index the observed coordinate-window instances and let each instance contribute its shift a_i. Flattening these shifts gives the empirical pooled-shift distribution

$$\widehat{P}_A = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}} \delta_{a_i}, \qquad A \sim \widehat{P}_A.$$

Thus $A \sim \widehat{P}_A$ is the shift obtained by selecting one observed instance uniformly. More generally, for a pooled-shift distribution P, drawing $A \sim P$ gives the mixed reference $Y_A = 2/(Z+A)^2$. The fixed reference $Y_c = 2/(Z+c)^2$, with $c^2 := \mathbb{E}_P[A^2]$, keeps the leading shift effect; the shifted-reference calculation below proves the corresponding second-order survival matching.

Proof of Proposition 1 and shifted-reference calculation. We justify Proposition 1 under the local Gaussian window used in the main text. The calculation is not a global model of training; it isolates the inverse-square denominator that controls the ratio state.

Exact factorization. Let $L = \sum_{j\ge 0} w_j \xi_j$, $Q = \sum_{j\ge 0} w_j \xi_j^2$, and $s_\beta^2 = \sum_{j\ge 0} w_j^2 = (1-\beta)/(1+\beta)$. Then

$$\begin{aligned} m_t &= \mu + \sigma L = \sigma_m(Z+a), \\ U_t &= v_t - m_t^2 = \sigma^2(Q - L^2), \qquad Z = L/s_\beta. \end{aligned}$$

Since $\mathbb{E}[Q] = 1$ and $\mathbb{E}[L^2] = s_\beta^2$,

$$\begin{aligned} \mathbb{E}[U_t] &= \sigma^2(1 - s_\beta^2) = \frac{2\beta}{1+\beta}\sigma^2, \\ \frac{1-\beta}{\beta}\frac{\mathbb{E}[U_t]}{\sigma_m^2} &= 2. \end{aligned}$$

Now decompose the numerator around the denominator coordinate,

$$U_t = \mathbb{E}[U_t] + \big(\mathbb{E}[U_t \mid Z] - \mathbb{E}[U_t]\big) + \big(U_t - \mathbb{E}[U_t \mid Z]\big).$$

For Gaussian variables ξ_j, projecting onto Z gives

$$\begin{aligned} \mathbb{E}[U_t \mid Z] - \mathbb{E}[U_t] &= \sigma^2 d_\beta (Z^2 - 1), \\ d_\beta &= \frac{\sum_{j\ge 0} w_j^3}{s_\beta^2} - s_\beta^2 = \frac{\beta(1-\beta)}{(1+\beta)(1+\beta+\beta^2)}. \end{aligned}$$

Substituting into $y_t = \frac{1-\beta}{\beta} U_t / m_t^2$ yields

$$\begin{aligned} y_t &= \frac{2+\eta_t}{(Z+a)^2} = Y_a \left(1 + \frac{\eta_t}{2}\right), \\ \eta_t &= \eta_t^{\text{coup}} + \eta_t^{\text{fluc}}, \\ \eta_t^{\text{coup}} &= \gamma_\beta (Z^2 - 1), \qquad \gamma_\beta = \frac{1-\beta}{1+\beta+\beta^2}, \\ \eta_t^{\text{fluc}} &= \frac{1-\beta}{\beta} \frac{U_t - \mathbb{E}[U_t \mid Z]}{\sigma_m^2}, \end{aligned}$$

where $Y_a = 2/(Z+a)^2$. This proves the factorization in Equation (5).

Auxiliary correction estimates. The fluctuation term is conditionally centered by construction:

$$\mathbb{E}[\eta_t^{\text{fluc}} \mid Z] = 0.$$

For its variance, write $B = \operatorname{diag}(w) - ww^\top$, so that $U_t = \sigma^2 \xi^\top B \xi$. The Gaussian quadratic-form identity gives

$$\operatorname{Var}(U_t) = 2\sigma^4 \operatorname{tr}(B^2) = 2\sigma^4 \left(s_\beta^2 - 2 \sum_{j \geq 0} w_j^3 + s_\beta^4 \right).$$

Since $\mathbb{E}[U_t \mid Z] - \mathbb{E}[U_t] = \sigma^2 d_\beta (Z^2 - 1)$, $\operatorname{Var}(\mathbb{E}[U_t \mid Z]) = 2\sigma^4 d_\beta^2$, we have

$$\operatorname{Var}(\eta_t^{\text{fluc}}) = \frac{2(1+\beta)^2}{\beta^2} \left(s_\beta^2 - 2 \sum_{j \geq 0} w_j^3 + s_\beta^4 - d_\beta^2 \right) = \frac{2(1+\beta)}{\beta^2} (1-\beta) + O((1-\beta)^2).$$

Also, since $y_t / Y_a - 1 = \eta_t / 2$, for any $0 < u < 2\delta$,

$$\left\{ \left| \frac{y_t}{Y_a} - 1 \right| > \delta \right\} \subseteq \{ |\eta_t^{\text{coup}}| > u \} \cup \{ |\eta_t^{\text{fluc}}| > 2\delta - u \}.$$

Taking probabilities of both sides and applying Chebyshev's inequality gives the corresponding relative-error control.

Tail order. Let $S_t = 2 + \eta_t \geq 0$. Then

$$\Pr(y_t > k) = \int_{\mathbb{R}} \phi(z) \, \Pr\big(S_t > k(z+a)^2 \mid Z = z\big) \, dz.$$

With $x = \sqrt{k}(z+a)$, this becomes

$$\frac{1}{\sqrt{k}} \int_{\mathbb{R}} \phi\left(-a + \frac{x}{\sqrt{k}}\right) \Pr\left(S_t > x^2 \mid Z = -a + \frac{x}{\sqrt{k}}\right) dx.$$

We justify domination using the Gaussian quadratic-form structure. Set $\alpha = w/s_\beta$, $C = (1-\beta)/(\beta s_\beta^2)$, and decompose $\xi = \alpha Z + \xi_\perp$, with $\xi_\perp$ independent of Z. Since $B = \operatorname{diag}(w) - ww^\top$ is positive semidefinite and trace class, $V = \sqrt{C} B^{1/2} \xi_\perp$ has finite second moment, and $d = \sqrt{C} B^{1/2} \alpha$ has finite norm. Conditionally on $Z = z$, $\sqrt{S_t}$ has the representation $\|V + dz\|$. Consequently $h(z) = \mathbb{E}\|V + dz\|$ is finite and Lipschitz continuous. Let $D = \|V\| + |a|\|d\|$. For $\sqrt{k} \geq 2\|d\|$, $\|V + d(-a + x/\sqrt{k})\| > |x|$ implies $D > |x|/2$. The integrand above is therefore bounded by the integrable function

$$(2\pi)^{-1/2} \min\{1, 4\mathbb{E}[D^2]/x^2\},$$

with its value at zero defined by continuity. The conditional tail probabilities converge for almost every x, since $\|V + d(-a + x/\sqrt{k})\| \to \|V - da\|$ almost surely. Dominated convergence now gives

$$\Pr(y_t > k) = \frac{\phi(a)}{\sqrt{k}} \int_{\mathbb{R}} \Pr(S_t > x^2 \mid Z = -a) \, dx + o(k^{-1/2}) = \frac{2\phi(a)}{\sqrt{k}} h(-a) + o(k^{-1/2}).$$

Shifted-reference matching. Let P be a pooled-shift distribution with finite fourth moment, and draw $A \sim P$. If P is empirical, then $P = |\mathcal{I}|^{-1} \sum_{i \in \mathcal{I}} \delta_{a_i}$, and

$$\mathbb{E}_P[A^2] = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} a_i^2.$$

Assume $Z \perp A$ and $Z \sim \mathcal{N}(0,1)$. Fix $k > 0$. Writing $q = \sqrt{2/k}$, the mixed denominator reference $Y_A = 2/(Z + A)^2$ satisfies

$$\Pr(Y_A > k \mid A) = \Phi(q - A) + \Phi(q + A) - 1.$$

A Taylor expansion in the shift gives

$$\Phi(q - A) + \Phi(q + A) - 1 = 2\Phi(q) - 1 - q\phi(q)A^2 + O(A^4).$$

Therefore

$$\begin{aligned} \Pr(Y_A > k) &= 2\Phi(q) - 1 - q\phi(q)\mathbb{E}_P[A^2] + O(\mathbb{E}_P[A^4]), \\ \Pr(Y_c > k) &= 2\Phi(q) - 1 - q\phi(q)c^2 + O(c^4), \qquad c^2 := \mathbb{E}_P[A^2]. \end{aligned}$$

For a family P_ε of pooled-shift distributions, let $A_\varepsilon \sim P_\varepsilon$ and $c_\varepsilon^2 := \mathbb{E}_{P_\varepsilon}[A_\varepsilon^2]$. If $\mathbb{E}_{P_\varepsilon}[A_\varepsilon^4] = O(\varepsilon^4)$, Jensen's inequality gives $c_\varepsilon^4 = (\mathbb{E}_{P_\varepsilon}[A_\varepsilon^2])^2 \le \mathbb{E}_{P_\varepsilon}[A_\varepsilon^4] = O(\varepsilon^4)$. Hence the two survival functions differ only at fourth order:

$$\Pr(Y_{A_\varepsilon} > k) - \Pr(Y_{c_\varepsilon} > k) = O(\varepsilon^4).$$

This proves the second-order matching for the reference Y_c in Equation (6).

B Additional Motivation and Related Work for the Tied-β Regime

This appendix provides additional motivation for studying the tied-β regime, $\beta_1 = \beta_2$. Our goal is not to claim that this choice is universally optimal, but rather to explain why it is a structurally distinguished regime for analyzing Adam. The classical Adam default $(\beta_1, \beta_2) = (0.9, 0.999)$ uses very different memories for the first and second moving averages. However, recent analyses and large-scale training practice have increasingly considered settings where these two memories are closer, including tied or nearly tied choices. This motivates asking what becomes simpler or more stable when the two exponential memories are matched.

Several recent works point in this direction from complementary perspectives. Orvieto and Gower [33] show that tied-β Adam preserves much of Adam's empirical behavior in language-model training while admitting a cleaner interpretation in terms of smoothed sign momentum and a noise-to-signal correction. Fernández-Hernández et al. [16] identify a structural principle behind this regime, showing that Adam has first-order gradient-scale invariance precisely when $\beta_1 = \beta_2$. A separate line of work studies how mini-batch noise interacts with Adam's memory parameters, reporting that moving β_1 closer to β_2 can improve validation behavior in some larger-batch, multi-epoch settings [6]. A concise informal overview of several related perspectives on the tied-β regime is also provided by Su [39]. These results suggest that tied or near-tied memories are not merely algebraically convenient, but can reveal meaningful structure in Adam's adaptive dynamics.

Our use of the tied-β regime is more specific. It is exactly the setting in which Adam's adaptive denominator admits a clean sign-over-ratio reformulation with a nonnegative variance-like residual. Ignoring bias correction and ϵ for simplicity, Adam evolves coordinate-wise as

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \tag{15}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2, \tag{16}$$

and the adaptive factor is $m_t / \sqrt{v_t}$. When $\beta_1 = \beta_2 = \beta$, a direct expansion gives

$$v_t - m_t^2 = \beta\big(v_{t-1} - m_{t-1}^2\big) + \beta(1 - \beta)(g_t - m_{t-1})^2. \tag{17}$$

Therefore, under the standard zero initialization,

$$v_t - m_t^2 \ge 0 \qquad \text{for all } t. \tag{18}$$

This makes $v_t - m_t^2$ a genuine variance-like residual. Whenever $m_t \neq 0$, we can define

$$r_t := \frac{v_t - m_t^2}{m_t^2} \geq 0, \qquad v_t = m_t^2(1 + r_t), \tag{19}$$

which yields

$$\frac{m_t}{\sqrt{v_t}} = \frac{\operatorname{sign}(m_t)}{\sqrt{1 + r_t}}. \tag{20}$$

Thus, in the tied-β regime, Adam's adaptive factor is a sign-momentum direction multiplied by an attenuation factor. Since $r_t \geq 0$, this attenuation is at most one; the update cannot be amplified beyond the corresponding sign-momentum magnitude by a negative residual.

This nonnegativity property is in fact special to the tied-β regime. For $0 < \beta_1, \beta_2 < 1$, fix a time t and write the zero-initialized moment estimates as

$$m_t = \sum_{i=0}^{t-1} a_i g_{t-i}, \qquad a_i := (1 - \beta_1)\beta_1^i, \tag{21}$$

$$v_t = \sum_{i=0}^{t-1} b_i g_{t-i}^2, \qquad b_i := (1 - \beta_2)\beta_2^i. \tag{22}$$

By weighted Cauchy–Schwarz,

$$m_t^2 = \left(\sum_{i=0}^{t-1} a_i g_{t-i}\right)^2 \leq \left(\sum_{i=0}^{t-1} \frac{a_i^2}{b_i}\right)\left(\sum_{i=0}^{t-1} b_i g_{t-i}^2\right) = S_t v_t, \tag{23}$$

where

$$S_t := \sum_{i=0}^{t-1} \frac{a_i^2}{b_i} = \frac{(1 - \beta_1)^2}{1 - \beta_2} \sum_{i=0}^{t-1} \left(\frac{\beta_1^2}{\beta_2}\right)^i. \tag{24}$$

Moreover, the bound is tight: equality in Equation (23) is attained by choosing $g_{t-i} \propto a_i/b_i$. Consequently, $v_t \geq m_t^2$ for all gradient sequences at time t holds if and only if $S_t \leq 1$.

When $\beta_1 = \beta_2 = \beta$, Equation (24) gives

$$S_t = (1 - \beta) \sum_{i=0}^{t-1} \beta^i = 1 - \beta^t \leq 1, \tag{25}$$

recovering the nonnegativity above. Conversely, suppose $\beta_1 \neq \beta_2$. If $\beta_1^2 \geq \beta_2$, then the sum in Equation (24) grows without bound as t increases, so $S_t > 1$ for some finite t. If $\beta_1^2 < \beta_2$, then

$$\lim_{t \to \infty} S_t = \frac{\beta_2(1 - \beta_1)^2}{(1 - \beta_2)(\beta_2 - \beta_1^2)}. \tag{26}$$

A direct comparison gives

$$\frac{\beta_2(1 - \beta_1)^2}{(1 - \beta_2)(\beta_2 - \beta_1^2)} > 1 \qquad \Longleftrightarrow \qquad (\beta_2 - \beta_1)^2 > 0, \tag{27}$$

which holds whenever $\beta_1 \neq \beta_2$. Thus $S_t > 1$ for some finite t. By the tightness of Equation (23), there exists a gradient sequence for which $m_t^2/v_t = S_t > 1$, and therefore

$$v_t - m_t^2 < 0. \tag{28}$$

Hence the tied-β regime is not only sufficient but necessary for the residual $v_t - m_t^2$ to remain nonnegative for all gradient sequences.

A simple spike calculation illustrates this failure mode for the classical mismatched setting. For fixed previous state (m_{t-1}, v_{t-1}),

$$1 + r_t = \frac{v_t}{m_t^2} = \frac{\beta_2 v_{t-1} + (1 - \beta_2) g_t^2}{\big(\beta_1 m_{t-1} + (1 - \beta_1) g_t\big)^2}. \tag{29}$$

As $|g_t| \to \infty$, the lower-order terms vanish and

$$1 + r_t \longrightarrow \frac{1-\beta_2}{(1-\beta_1)^2}. \tag{30}$$

For the classical Adam default $(\beta_1, \beta_2) = (0.9, 0.999)$, this limiting value is

$$\frac{1-\beta_2}{(1-\beta_1)^2} = \frac{0.001}{0.1^2} = 0.1. \tag{31}$$

So in the large-spike regime $1 + r_t \approx 0.1$. Thus, away from the tied-β regime, the sign-over-ratio residual is not protected by the same nonnegativity mechanism and can become substantially negative.

This observation also connects to work on nonstationary training dynamics, especially in reinforcement learning. In such settings, gradient statistics can shift rapidly, and mismatched first- and second-moment memories may cause the adaptive denominator to lag behind the signal term. This can temporarily weaken the attenuation and produce unexpectedly large updates. Related reinforcement-learning studies have connected Adam's timescale choices to policy collapse or instability under nonstationary data streams, and report that matching the first- and second-moment decay rates, or bringing them closer together, can mitigate some of these failures [12, 32]. From another angle, Ellis et al. [15] show that nonstationary changes in gradient magnitude can cause Adam to produce overly large updates, motivating a relative-timestep correction. These works do not establish tied β values as a universally optimal choice, but they support the broader view that the relative timescales of Adam's two moving averages are central to stability under changing gradient statistics.

Taken together, these perspectives motivate the tied-β regime as a useful analytical setting. It is the regime in which the adaptive denominator admits a clean decomposition into a squared mean and a nonnegative variance-like residual; it aligns with recent structural analyses of Adam's scale behavior; and it provides a natural starting point for our ratio-state formulation.

C Details for the transformed-state recursion and implementation

This appendix collects additional details for the transformed-state recursion used in the main text. We first give a step-by-step derivation of the scalar recursion for y_t, then describe its initialization and the practical handling of numerically unstable regions, and finally summarize the ablations used to evaluate the ratio-recursive formulation.

C.1 Step-by-step derivation of the transformed-state recursion

Recall that in the tied-beta regime $\beta_1 = \beta_2 = \beta$, we define

$$r_t := \frac{v_t - m_t^2}{m_t^2}, \qquad y_t := \frac{1-\beta}{\beta} r_t = \frac{1-\beta}{\beta} \frac{v_t - m_t^2}{m_t^2}, \tag{32}$$

whenever $m_t \neq 0$. Equivalently,

$$v_t = m_t^2 \left(1 + \frac{\beta}{1-\beta} y_t\right). \tag{33}$$

We also define the momentum ratio

$$x_t := \frac{m_{t-1}}{m_t}. \tag{34}$$

Starting from the first-moment recursion

$$m_t = \beta m_{t-1} + (1-\beta) g_t, \tag{35}$$

we solve for the current gradient:

$$g_t = \frac{m_t - \beta m_{t-1}}{1-\beta} = \frac{m_t}{1-\beta}(1 - \beta x_t). \tag{36}$$

Substituting Equation (36) into the second-moment recursion

$$v_t = \beta v_{t-1} + (1-\beta) g_t^2 \tag{37}$$

gives

$$v_t = \beta v_{t-1} + \frac{m_t^2}{1-\beta}(1-\beta x_t)^2. \tag{38}$$

Next, express both v_t and v_{t-1} through the transformed state:

$$v_t = m_t^2\left(1+\frac{\beta}{1-\beta}y_t\right), \qquad v_{t-1} = m_{t-1}^2\left(1+\frac{\beta}{1-\beta}y_{t-1}\right). \tag{39}$$

Using $m_{t-1} = x_t m_t$, Equation (38) becomes

$$m_t^2\left(1+\frac{\beta}{1-\beta}y_t\right) = \beta x_t^2 m_t^2\left(1+\frac{\beta}{1-\beta}y_{t-1}\right) + \frac{m_t^2}{1-\beta}(1-\beta x_t)^2. \tag{40}$$

Dividing by m_t^2 yields

$$1+\frac{\beta}{1-\beta}y_t = \beta x_t^2\left(1+\frac{\beta}{1-\beta}y_{t-1}\right) + \frac{(1-\beta x_t)^2}{1-\beta}. \tag{41}$$

Multiplying both sides by $1-\beta$ and simplifying gives

$$(1-\beta) + \beta y_t = \beta(1-\beta)x_t^2 + \beta^2 x_t^2 y_{t-1} + (1-\beta x_t)^2, \tag{42}$$

and therefore

$$y_t = \beta x_t^2 y_{t-1} + (1-x_t)^2. \tag{43}$$

This is the scalar recursion stated in the main text. Algorithm 1 gives the detailed pseudocode for TR-Adam, including the quantized ratio-state storage and bias-corrected parameter update.

Numerical representation. Algorithm 1 describes the FP32-momentum variant. The experiments emulate low-precision states in FP32 tensors rather than using packed storage. The current ratio is quantized before the parameter update. In the FP8-momentum variant, the fresh momentum working value is used for the current update, then quantized to FP8 E4M3 using a per-tensor absolute-maximum scale for state writeback and use in the next step. Thus the comparisons evaluate numerical effects of state quantization; they do not measure memory or throughput gains from a packed implementation.

D Additional Experimental Results

D.1 Original language-model comparisons

This section provides additional quantitative and distributional evidence for the robustness of the transformed ratio-state representation. Figure 9 first shows that the transformed variable y_t has a broadly consistent bulk-and-tail distribution across five representative settings, including pre-training, SFT, and ReMax. We then report final validation metrics in Tables 2 to 4, averaging over the final three validation points to reduce noise from individual checkpoints.

Across these settings, TR-Adam remains close to full Adam in pre-training and often improves over Adam in post-training, suggesting that the coarse transformed state preserves the main adaptive behavior while introducing a mild stabilizing effect.

Algorithm 1 TR-Adam with quantized transformed ratio

Require: Parameters θ_0, objective $f_t(\theta)$, learning rate η, tied momentum coefficient β, numerical constant ϵ, weight decay λ, ratio-state codebook $\mathcal{Q}$.

Initialize $m_0 \leftarrow 0$ for all parameters.
for $t = 1, \ldots, T$ **do**
  $g_t \leftarrow \nabla f_t(\theta_{t-1})$
  $m_t \leftarrow \beta m_{t-1} + (1-\beta) g_t$
  if $t = 1$ **then**
    $\widetilde{y}_t \leftarrow 1$ ▷ initial transformed ratio
  else
    $x_t \leftarrow m_{t-1}/m_t$
    $\widetilde{y}_t \leftarrow \beta x_t^2 y_{t-1} + (1-x_t)^2$
  end if
  if $\widetilde{y}_t$ is non-finite or $\widetilde{y}_t > \max \mathcal{Q}$ **then**
    $\widetilde{y}_t \leftarrow \max \mathcal{Q}$ ▷ saturate before quantization
  end if
  $y_t \leftarrow Q_{\mathcal{Q}}(\widetilde{y}_t)$ ▷ nearest codebook value, FP4-style quantization
  $\widehat{m}_t \leftarrow m_t/(1-\beta^t)$ ▷ first-moment bias correction
  $r_t \leftarrow \frac{\beta}{1-\beta} y_t$
  $d_t \leftarrow |m_t| \sqrt{\frac{1+r_t}{1-\beta^t}} + \epsilon$ ▷ denominator bias correction
  $\theta_{t-\frac{1}{2}} \leftarrow (1-\eta\lambda)\theta_{t-1}$ ▷ decoupled weight decay
  $\theta_t \leftarrow \theta_{t-\frac{1}{2}} - \eta \widehat{m}_t / d_t$
end for

Table 2: Pre-training validation loss averaged over the final three validation points. For Llama-style 20M, the learning rate is selected from the full learning-rate sweep; for Pythia-style 160M, it is selected from $\{1, 3, 5, 8, 10\} \times 10^{-3}$. Lower is better. Here $\Delta = \text{TR-Adam} - \text{Adam}$.

Model	β	Adam	TR-Adam (FP32-m_t, 4-bit y_t)	TR-Adam (FP8-m_t, 4-bit y_t)	Δ FP32	Δ FP8
Llama-style 20M	0.90	**3.694**	3.706	3.710	+0.012	+0.016
Llama-style 20M	0.92	**3.676**	3.689	3.688	+0.013	+0.012
Llama-style 20M	0.95	3.672	3.672	**3.670**	-0.000	-0.002
Llama-style 20M	0.97	3.692	3.677	**3.675**	-0.014	-0.016
Pythia-style 160M	0.90	**3.373**	3.378	3.387	+0.004	+0.013
Pythia-style 160M	0.92	**3.355**	3.365	3.369	+0.010	+0.014
Pythia-style 160M	0.95	**3.342**	3.347	3.346	+0.005	+0.004
Pythia-1B	0.90	2.985	2.979	**2.979**	-0.006	-0.007
Pythia-1B	0.92	2.996	**2.977**	2.980	-0.019	-0.016
Pythia-1B	0.95	2.982	**2.973**	2.974	-0.009	-0.008

Table 3: SFT validation loss averaged over the final three validation points. Lower is better. Here $\Delta = \text{TR-Adam} - \text{Adam}$.

Model	β	Adam	TR-Adam (FP32-m_t, 4-bit y_t)	TR-Adam (FP8-m_t, 4-bit y_t)	Δ FP32	Δ FP8
Pythia-1B	0.90	1.299	**1.297**	1.297	-0.002	-0.002
Pythia-1B	0.92	1.302	**1.295**	1.295	-0.007	-0.006
Pythia-1B	0.95	1.309	**1.299**	1.299	-0.011	-0.011
Llama-3.2-1B	0.90	1.075	**1.072**	1.072	-0.002	-0.002
Llama-3.2-1B	0.92	1.076	**1.071**	1.071	-0.005	-0.005
Llama-3.2-1B	0.95	1.082	**1.073**	1.073	-0.009	-0.009

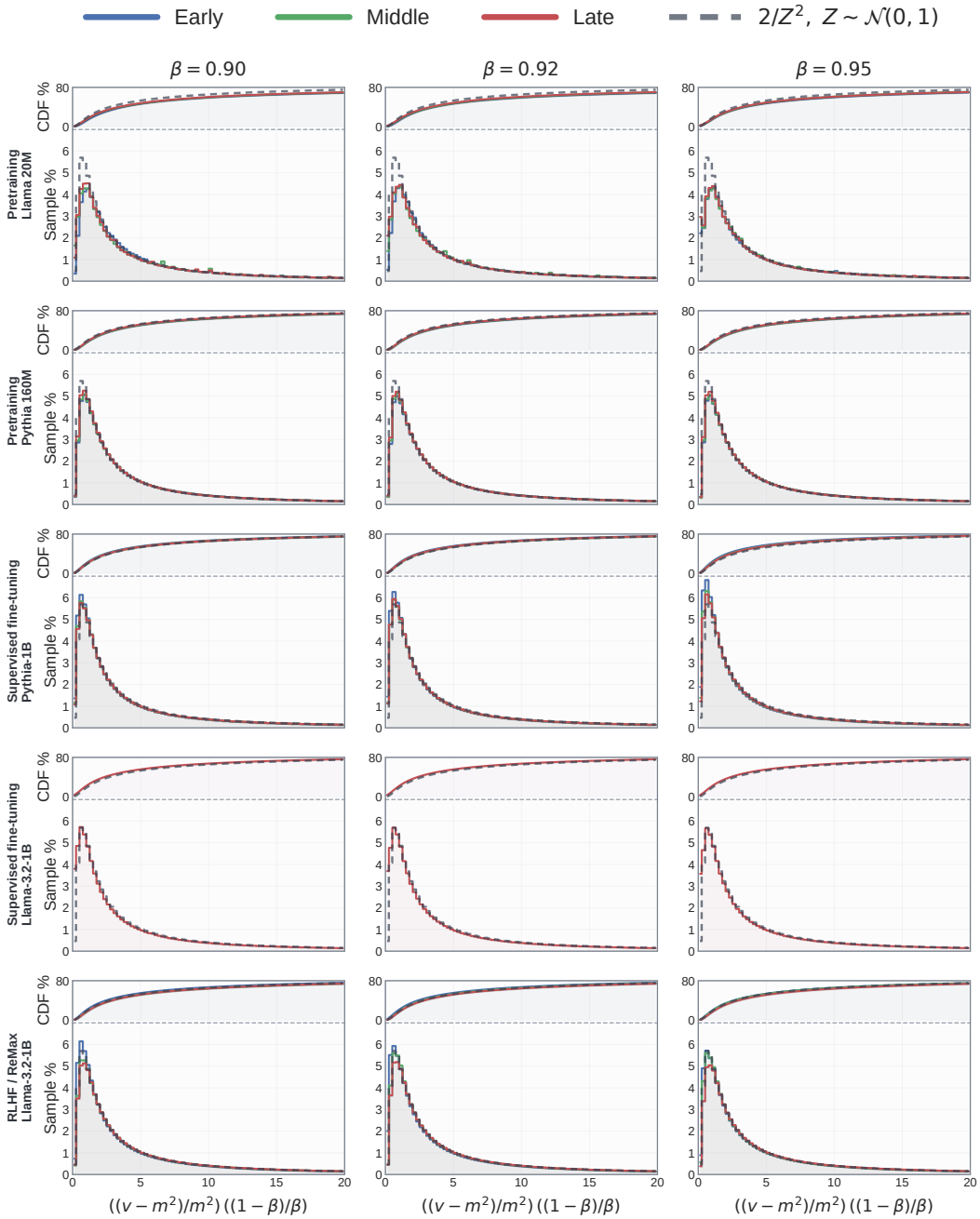


Figure 9: Distributions of the Adam-derived transformed variable y_t across five representative training settings: Llama-style 20M and Pythia-style 160M pre-training, Pythia-1B and Llama-3.2-1B supervised fine-tuning, and Llama-3.2-1B-Instruct ReMax training. Each block shows the empirical CDF at the top and the histogram at the bottom for different training stages. Across model families, scales, and training regimes, the transformed state exhibits a consistent heavy-tailed shape: a concentrated bulk near small values together with a long right tail. The dashed reference curve $2/Z^2$, with $Z \sim \mathcal{N}(0,1)$, is included to highlight the shared inverse-square-like structure.

Table 4: ReMax validation reward averaged over the final three validation points. Higher is better. Here $\Delta = \text{TR-Adam} - \text{Adam}$.

β	Adam	TR-Adam (FP32-m_t, 4-bit y_t)	TR-Adam (FP8-m_t, 4-bit y_t)	Δ FP32	Δ FP8
0.90	0.788	**0.825**	0.813	+0.036	+0.025
0.92	0.800	**0.824**	0.819	+0.024	+0.019
0.95	0.808	**0.817**	0.811	+0.009	+0.003

D.2 Reproducibility across random seeds

We repeat the Llama-style 20M pre-training and Llama-3.2-1B-Instruct ReMax RLHF comparisons using three random seeds. Tables 5 and 6 retain the original single-seed results alongside the repeated experiments. For three-seed rows, we report the mean and standard deviation across runs, rounded to three decimal places; these are standard deviations, not standard errors or confidence intervals. The FP32 and FP8 variants refer to momentum precision, with a 4-bit transformed-ratio state in both cases. Each pre-training score averages validation loss at steps 2,400, 2,600, and 2,800; each ReMax score averages validation reward at steps 550, 575, and 600.

The repeated results preserve the main qualitative trends. On Llama-style 20M, TR-Adam has slightly higher loss at $\beta = 0.90, 0.92$ and slightly lower loss at $\beta = 0.95, 0.97$. In the RLHF comparison, both TR-Adam variants have higher mean reward than Adam at the three tested decay rates. These small samples characterize run-to-run variation in selected settings; they do not establish statistical significance or replace repeated evaluation of every experiment.

Table 5: Llama-style 20M pre-training validation loss; lower is better. Single-seed entries are the original results; three-seed entries are mean $\pm$ standard deviation.

β	Runs	Adam	TR-Adam FP32 momentum	TR-Adam FP8 momentum
0.90	1 seed	3.694	3.706	3.710
0.90	3 seeds	3.697 ± 0.012	3.711 ± 0.004	3.711 ± 0.003
0.92	1 seed	3.676	3.689	3.688
0.92	3 seeds	3.676 ± 0.002	3.691 ± 0.004	3.694 ± 0.006
0.95	1 seed	3.672	3.672	3.670
0.95	3 seeds	3.673 ± 0.002	3.671 ± 0.002	3.670 ± 0.002
0.97	1 seed	3.692	3.677	3.675
0.97	3 seeds	3.693 ± 0.005	3.672 ± 0.005	3.676 ± 0.003

Table 6: Llama-3.2-1B-Instruct ReMax RLHF evaluation reward; higher is better. Single-seed entries are the original results; three-seed entries are mean $\pm$ standard deviation.

β	Runs	Adam	TR-Adam FP32 momentum	TR-Adam FP8 momentum
0.90	1 seed	0.788	0.825	0.813
0.90	3 seeds	0.802 ± 0.013	0.811 ± 0.017	0.818 ± 0.004
0.92	1 seed	0.800	0.824	0.819
0.92	3 seeds	0.798 ± 0.002	0.820 ± 0.005	0.818 ± 0.005
0.95	1 seed	0.808	0.817	0.811
0.95	3 seeds	0.801 ± 0.015	0.835 ± 0.019	0.805 ± 0.009

D.3 Distributional structure across architectures and model scales

D.3.1 Non-Transformer performance and training-stage distributions

We evaluate two multilayer perceptrons (MLPs; 4.73M and 427.7K parameters) on CIFAR-10 and an 11.22M-parameter ResNet-18 [21] on CIFAR-100 [24]. Each model is trained for 3,000 optimizer steps with learning rate 2×10^{-4} and batch size 512, $\epsilon = 10^{-8}$, and no weight decay. TR-Adam uses FP32 momentum and the 16-value ratio codebook in Algorithm 1. In these experiments the quantized ratio values are represented in FP32 tensors; the 4-bit description refers to the codebook cardinality, not measured packed-state memory. For each seed, test loss is averaged over evaluations at steps 2,842, 2,940, and 3,000. Table 7 reports the mean and standard deviation of this statistic over seeds 1 and 2. A displayed standard deviation of 0.000 reflects rounding to three decimal places, not necessarily zero variation.

TR-Adam gives losses close to Adam across all nine settings. The differences are small and not uniformly favorable: for example, ResNet-18 at $\beta = 0.90$ has a slightly higher mean loss with TR-Adam. The evidence supports applicability beyond Transformers under the tested budget, without establishing a universal performance advantage.

Table 7: Non-Transformer test loss (mean $\pm$ standard deviation over two seeds; lower is better).

Model / dataset	β	Adam	TR-Adam
MLP 4.73M / CIFAR-10	0.90	1.121 ± 0.005	1.119 ± 0.006
MLP 4.73M / CIFAR-10	0.92	1.124 ± 0.003	1.122 ± 0.000
MLP 4.73M / CIFAR-10	0.95	1.128 ± 0.003	1.117 ± 0.008
MLP 427.7K / CIFAR-10	0.90	1.289 ± 0.003	1.280 ± 0.010
MLP 427.7K / CIFAR-10	0.92	1.290 ± 0.004	1.276 ± 0.007
MLP 427.7K / CIFAR-10	0.95	1.298 ± 0.000	1.275 ± 0.000
ResNet-18 / CIFAR-100	0.90	1.576 ± 0.009	1.588 ± 0.026
ResNet-18 / CIFAR-100	0.92	1.569 ± 0.001	1.568 ± 0.025
ResNet-18 / CIFAR-100	0.95	1.557 ± 0.012	1.545 ± 0.008

Using Adam moment states and seed 1, we separately measure the empirical distribution $F(c) = \Pr(y_t \leq c)$ during early (steps 1–1,000), middle (1,001–2,000), and late (2,001–3,000) training, pooling all parameter entries at every optimizer step within each interval. Tables 8–10 compare the measured CDF values with the zero-shift reference $2/Z^2$, where $Z \sim \mathcal{N}(0, 1)$. The entries are probabilities, not percentages. These distributional measurements and the optimizer-performance comparisons answer distinct questions: similarity of the observed ratio distribution does not by itself establish the accuracy of a recursively quantized optimizer.

The later-stage CDFs remain close across the tested decay rates. The smaller MLP exhibits a clearer early-stage shift, while its middle and late distributions are similar. The reference is an approximate description: systematic differences remain, and the tabulated thresholds do not constitute a goodness-of-fit test or characterize the full tail.

Table 8: Training-stage hidden-ratio CDF for the 4.73M-parameter MLP on CIFAR-10.

β	Stage	$F(1)$	$F(2)$	$F(4)$	$F(7)$	$F(10)$	$F(15)$	$F(20)$
–	$2/Z^2$ reference	0.157	0.317	0.480	0.593	0.655	0.715	0.752
0.90	Early	0.142	0.295	0.456	0.570	0.633	0.695	0.733
0.90	Middle	0.143	0.297	0.460	0.575	0.639	0.701	0.739
0.90	Late	0.148	0.303	0.466	0.581	0.644	0.705	0.743
0.92	Early	0.143	0.296	0.456	0.571	0.633	0.695	0.732
0.92	Middle	0.143	0.296	0.458	0.574	0.638	0.700	0.738
0.92	Late	0.148	0.302	0.464	0.579	0.642	0.704	0.742
0.95	Early	0.145	0.297	0.455	0.568	0.629	0.689	0.726
0.95	Middle	0.143	0.294	0.455	0.570	0.633	0.695	0.734
0.95	Late	0.150	0.301	0.462	0.577	0.640	0.702	0.740

Table 9: Training-stage hidden-ratio CDF for the 427.7K-parameter MLP on CIFAR-10.

β	Stage	$F(1)$	$F(2)$	$F(4)$	$F(7)$	$F(10)$	$F(15)$	$F(20)$
–	$2/Z^2$ reference	0.157	0.317	0.480	0.593	0.655	0.715	0.752
0.90	Early	0.184	0.344	0.501	0.610	0.670	0.727	0.762
0.90	Middle	0.143	0.302	0.466	0.582	0.645	0.707	0.745
0.90	Late	0.141	0.300	0.464	0.580	0.644	0.706	0.744
0.92	Early	0.190	0.349	0.506	0.614	0.672	0.729	0.764
0.92	Middle	0.142	0.300	0.464	0.580	0.643	0.705	0.743
0.92	Late	0.140	0.298	0.463	0.579	0.643	0.705	0.743
0.95	Early	0.209	0.367	0.520	0.626	0.683	0.738	0.772
0.95	Middle	0.141	0.300	0.464	0.580	0.643	0.705	0.743
0.95	Late	0.139	0.296	0.461	0.577	0.641	0.703	0.742

Table 10: Training-stage hidden-ratio CDF for the 11.22M-parameter ResNet-18 on CIFAR-100.

β	Stage	$F(1)$	$F(2)$	$F(4)$	$F(7)$	$F(10)$	$F(15)$	$F(20)$
–	$2/Z^2$ reference	0.157	0.317	0.480	0.593	0.655	0.715	0.752
0.90	Early	0.157	0.317	0.480	0.593	0.655	0.715	0.752
0.90	Middle	0.147	0.306	0.470	0.585	0.648	0.709	0.747
0.90	Late	0.148	0.308	0.471	0.586	0.649	0.710	0.748
0.92	Early	0.159	0.319	0.481	0.594	0.656	0.716	0.753
0.92	Middle	0.144	0.303	0.468	0.583	0.646	0.708	0.746
0.92	Late	0.145	0.304	0.468	0.584	0.647	0.708	0.746
0.95	Early	0.165	0.326	0.486	0.599	0.660	0.719	0.755
0.95	Middle	0.140	0.298	0.462	0.578	0.642	0.704	0.743
0.95	Late	0.141	0.299	0.463	0.579	0.643	0.705	0.743

D.3.2 An early-stage distributional check at 8B scale

Table 11 reports early-stage measurements from Llama-3.1-8B-Instruct under ReMax RLHF. Here "Early" means the first third of a 600-step budget: the statistics pool snapshots at steps 10, 20, . . ., 200 from both FSDP ranks, weighted by their finite-entry counts. The runs use seed 42, tied-β AdamW, a prompt batch size of 64, and maximum prompt/response lengths of 768/1,024 tokens. This is a step-based interval; variable response lengths do not define a fixed token count. At the reported thresholds, the CDFs for $\beta = 0.90$ and 0.95 are close to the $2/Z^2$ reference. This extends the distributional observation to an 8B model, but the available measurements cover only the early stage. They do not establish stability throughout training, nor do they constitute an 8B comparison of TR-Adam performance, memory consumption, or runtime.

Table 11: Early-stage hidden-ratio CDF for Llama-3.1-8B-Instruct under ReMax RLHF, pooling snapshots every 10 steps through step 200.

β	Stage	$F(1)$	$F(2)$	$F(4)$	$F(7)$	$F(10)$	$F(15)$	$F(20)$
–	$2/Z^2$ reference	0.157	0.317	0.480	0.593	0.655	0.715	0.752
0.90	Early	0.150	0.315	0.478	0.591	0.651	0.710	0.747
0.95	Early	0.168	0.334	0.495	0.605	0.664	0.722	0.758

D.4 Ratio distributions at large decay rates

We examine the distribution of the exact Adam-derived ratio at tied decay rates $\beta = 0.99$ and 0.999 on Llama-style 20M.

Table 12 measures the exact Adam-derived ratio on Llama-style 20M, using learning rates 0.004 for $\beta = 0.99$ and 0.0015 for $\beta = 0.999$. The first three intervals pool every optimizer step and parameter entry: steps 1–954, 955–1,908, and 1,909–2,861, with 524,288 tokens per step (approximately 0–0.5B, 0.5B–1B, and 1B–1.5B tokens). CDFs are normalized by finite ratio entries; undefined or non-finite ratios are excluded (about 1.1–3.4% across these stages). At $\beta = 0.99$, the early interval differs from the reference, while the later intervals are closer. At $\beta = 0.999$, the CDF is substantially shifted toward smaller values throughout the first 1.5B tokens; for example, $F(1)$ decreases from 0.686 to 0.476, compared with the reference value 0.157. The extended run gives $F(1) = 0.213$ in the 10B–12B interval. This row pools 38 snapshots at steps 19,100–22,800, spaced 100 steps apart (10.014B–11.954B tokens); 1.036% of ratio entries are non-finite and excluded.

This evolution is consistent with the shifted reference $Y_c = 2/(Z + c)^2$ and a changing local mean-to-noise ratio. The CDFs alone do not identify the cause of the shift or prove convergence to the zero-shift law. They show that approximate distributional stability should not be assumed uniformly across decay rates and training stages. The extended run provides distributional measurements, not a new optimizer-performance comparison.

Table 12: Hidden-ratio CDF at large tied decay rates, including the 10B–12B interval of extended training at $\beta = 0.999$.

β	Token interval	$F(0)$	$F(1)$	$F(2)$	$F(4)$	$F(7)$	$F(10)$	$F(15)$	$F(20)$
$2/Z^2$	Reference	0	0.157	0.317	0.480	0.593	0.655	0.715	0.752
0.99	0–0.5B	0	0.319	0.457	0.582	0.668	0.715	0.763	0.792
0.99	0.5B–1.0B	0	0.158	0.313	0.467	0.574	0.633	0.692	0.728
0.99	1.0B–1.5B	0	0.152	0.309	0.467	0.578	0.638	0.698	0.734
0.999	0–0.5B	0	0.686	0.792	0.853	0.889	0.907	0.924	0.934
0.999	0.5B–1.0B	0	0.561	0.677	0.769	0.824	0.852	0.878	0.895
0.999	1.0B–1.5B	0	0.476	0.599	0.704	0.773	0.809	0.843	0.863
0.999	10B–12B	0	0.213	0.372	0.524	0.629	0.686	0.741	0.774

D.5 Unequal decay rates: distributional changes and scope

Appendix B explains why tied decay rates are structurally special: the nonnegative ratio-state property holds for all gradient histories in the tied setting, but is not guaranteed for unequal decays. To examine the distribution empirically, define

$$\bar{\beta} = \frac{\beta_1 + \beta_2}{2}, \qquad r_t^{\mathrm{u}} = \frac{v_t^{\mathrm{u}}}{(m_t^{\mathrm{u}})^2} - 1, \qquad \widetilde{y}_t^{\mathrm{u}} = \frac{1-\bar{\beta}}{\bar{\beta}} r_t^{\mathrm{u}}.$$

The measurements use Pythia-style 160M. Here $(m_t^{\mathrm{u}}, v_t^{\mathrm{u}})$ are the exact Adam moments with the specified unequal decays; the ratio is defined where $m_t^{\mathrm{u}} \neq 0$. We measure $F(c) = \Pr(\widetilde{y}_t^{\mathrm{u}} \leq c)$, rather than applying the tied-β TR-Adam recursion to unequal decays. The unequal-decay rows pool all parameter entries at steps 1–2,000; the tied baseline is the previously recorded Early snapshot at step 2,000. These aggregation windows differ, so the comparison is descriptive rather than a matched isolation of the decay-rate effect. The histogram column $F(0^-)$ counts strictly negative finite ratios; other columns include this mass and the histogram bins below the indicated threshold.

For context, let $(\bar{m}_t, \bar{v}_t)$ be moments computed from the same gradient history using tied decay $\bar{\beta}$. Whenever the relative deviations are defined,

$$\varepsilon_{m,t} = \frac{m_t^{\mathrm{u}} - \bar{m}_t}{\bar{m}_t}, \qquad \varepsilon_{v,t} = \frac{v_t^{\mathrm{u}} - \bar{v}_t}{\bar{v}_t}, \qquad 1 + r_t^{\mathrm{u}} = (1 + \bar{r}_t)\frac{1 + \varepsilon_{v,t}}{(1 + \varepsilon_{m,t})^2}.$$

This identity shows why small relative moment changes can preserve the ratio away from small-momentum events, while closeness of decay rates alone does not guarantee a uniformly small perturbation.

Table 13: Pythia-style 160M CDF of the exact Adam-derived transformed ratio for tied and unequal decay rates. $F(0^-)$ measures strictly negative mass.

(β_1, β_2)	$F(0^-)$	$F(1)$	$F(2)$	$F(4)$	$F(7)$	$F(10)$	$F(15)$	$F(20)$
$2/Z^2$	0	0.157	0.317	0.480	0.593	0.655	0.715	0.752
$(.90, .90)$	0	.131	.290	.456	.573	.637	.699	.738
$(0.90, 0.92)$	6.56×10^{-6}	0.145	0.290	0.445	0.559	0.623	0.686	0.725
$(0.90, 0.95)$	1.71×10^{-3}	0.170	0.309	0.456	0.565	0.627	0.688	0.727
$(0.90, 0.999)$	3.66×10^{-2}	0.201	0.277	0.364	0.438	0.485	0.537	0.572

At $(\beta_1, \beta_2) = (0.90, 0.92)$ and $(0.90, 0.95)$, the positive-range CDF values remain broadly similar to the tied baseline, but negative mass appears. At $(0.90, 0.999)$, $F(0^-)$ rises to 0.0366 and the positive-range CDF differs much more strongly; for example, $F(20) = 0.572$ versus 0.738 for the tied $(0.90, 0.90)$ baseline. These diagnostics delimit the tied-regime claims. They do not validate a low-bit unequal-decay optimizer or establish equivalence of the resulting training trajectories.

D.6 Numerical stability and the role of epsilon

D.6.1 Optimization sensitivity

On Llama-style 20M with FP32 momentum and the 16-value ratio codebook, Table 14 varies the numerical constant ϵ around the default 10^{-6}. Each loss averages validation at steps 2,400, 2,600, and 2,800. Learning rates are held at their default-ϵ values: Adam/TR-Adam use $0.024/0.018$, $0.010/0.006$, and $0.005/0.005$ for $\beta = 0.90, 0.95, 0.97$, respectively. The 10^{-6} column retains the original paper results; other cells use seed 100, except the matched $\epsilon = 10^{-4}, \beta = 0.95$ pair, which uses seed 10,000. This is a single-run sensitivity check, not a fully seed-matched sweep. NaN denotes a non-finite outcome rather than a valid loss. Standard Adam produces NaNs at $\epsilon = 0$ in all three tested decay settings. TR-Adam produces finite losses using the special $\epsilon = 0$ implementation described below; directly substituting zero into the denominator form also failed. For positive values from 10^{-12} to 10^{-6}, the losses change modestly, whereas 10^{-4} gives consistently worse losses for both optimizers. These observations concern the tested implementations and settings, and do not imply that ϵ can be removed safely in general.

Table 14: Validation loss under numerical-constant ablation. Lower is better; the default is $\epsilon = 10^{-6}$.

Optimizer	β	10^{-6}	0	10^{-12}	10^{-8}	10^{-4}
Adam	0.90	3.694	NaN	3.700	3.702	3.791
Adam	0.95	3.672	NaN	3.673	3.678	3.737
Adam	0.97	3.692	NaN	3.697	3.693	3.795
TR-Adam	0.90	3.706	3.722	3.727	3.721	3.804
TR-Adam	0.95	3.672	3.683	3.683	3.678	3.764
TR-Adam	0.97	3.677	3.680	3.676	3.669	3.786

For $\epsilon = 0$, the implementation cancels $|m_t|$ algebraically before evaluating the update. Writing $b_t = 1 - \beta^t$ and $r_t = \frac{\beta}{1-\beta}\widehat{y}_t$, it computes

$$u_t = \frac{\operatorname{sign}(m_t)}{\sqrt{b_t(1 + r_t)}}, \qquad \operatorname{sign}(0) = 0.$$

This equals the bias-corrected denominator form when $m_t \neq 0$ and defines a zero update at $m_t = 0$, avoiding its explicit $0/0$. For positive ϵ, the implementation retains Algorithm 1's denominator form. A nonnegative quantized state ensures $\sqrt{1 + r_t} \geq 1$, but this bound alone does not resolve the zero-momentum case. The ratio recurrence separately saturates NaN or positive-infinite candidates at the maximum codebook value and clips the candidate before quantization. These are implementation choices, distinct from adding a positive constant to the denominator.

D.6.2 Layer-wise second-moment magnitudes

Tables 15 and 16 measure uncorrected Adam second moments on Llama-style 20M, with seed 100 and a 2,861-step budget. They pool every parameter entry at every optimizer step across Early (1–954), Middle (955–1,908), and Late (1,909–2,861). Each reports the arithmetic mean $\mathbb{E}[\sqrt{v_t}]$ and the percentage of entries for which $\sqrt{v_t} \leq \epsilon$. The means are on the original scale, not means of $\log_{10}\sqrt{v_t}$. Percentages displayed as 0.000% are rounded observations.

The condition $\sqrt{v_t} \gg \epsilon$ is not uniform across layers. With $\epsilon = 10^{-6}$, roughly 40–71% of embedding entries and 38–63% of LM-head entries are at or below ϵ. The fractions are much smaller in the Transformer blocks and final normalization layer. Even with $\epsilon = 10^{-8}$, non-negligible fractions of embedding and LM-head entries remain below the threshold. Thus, a small change in aggregate validation loss does not establish that ϵ is negligible for every coordinate. The measured sensitivity qualifies the epsilon-free theoretical interpretation rather than providing a uniform justification for dropping the stabilizer.

Table 15: Layer-wise $\sqrt{v_t}$ magnitudes with $\epsilon = 10^{-6}$: arithmetic means and percentages at or below the stabilizer. Measurements are aggregated across training stages.

Layer	$\beta = 0.90$ Mean	$\beta = 0.90$ $\leq \epsilon$ (%)	$\beta = 0.95$ Mean	$\beta = 0.95$ $\leq \epsilon$ (%)	$\beta = 0.97$ Mean	$\beta = 0.97$ $\leq \epsilon$ (%)
Embedding	1.428×10^{-6}	70.829	2.464×10^{-6}	50.863	3.666×10^{-6}	40.316
Block 0	2.496×10^{-5}	3.513	4.978×10^{-5}	0.143	8.711×10^{-5}	0.052
Block 1	1.110×10^{-5}	0.415	1.981×10^{-5}	0.273	3.047×10^{-5}	0.397
Block 2	1.301×10^{-5}	0.345	2.113×10^{-5}	0.352	3.113×10^{-5}	0.523
Block 3	1.285×10^{-5}	0.260	2.434×10^{-5}	0.323	3.401×10^{-5}	0.435
Final norm	1.575×10^{-3}	0.000	5.875×10^{-4}	0.000	4.229×10^{-4}	0.001
LM head	2.372×10^{-6}	62.816	3.857×10^{-6}	46.659	5.542×10^{-6}	37.638

Table 16: Layer-wise $\sqrt{v_t}$ magnitudes with $\epsilon = 10^{-8}$: arithmetic means and percentages at or below the stabilizer. Measurements are aggregated across training stages.

Layer	$\beta = 0.90$ Mean	$\beta = 0.90$ $\leq \epsilon$ (%)	$\beta = 0.95$ Mean	$\beta = 0.95$ $\leq \epsilon$ (%)	$\beta = 0.97$ Mean	$\beta = 0.97$ $\leq \epsilon$ (%)
Embedding	1.114×10^{-6}	20.784	2.040×10^{-6}	14.831	3.183×10^{-6}	11.174
Block 0	2.420×10^{-5}	0.000	4.926×10^{-5}	0.000	8.524×10^{-5}	0.000
Block 1	1.219×10^{-5}	0.000	2.260×10^{-5}	0.000	3.139×10^{-5}	0.000
Block 2	1.206×10^{-5}	0.000	2.084×10^{-5}	0.000	3.314×10^{-5}	0.000
Block 3	1.133×10^{-5}	0.000	2.131×10^{-5}	0.001	3.653×10^{-5}	0.001
Final norm	2.155×10^{-3}	0.000	6.438×10^{-4}	0.000	4.429×10^{-4}	0.000
LM head	2.067×10^{-6}	19.282	3.497×10^{-6}	12.924	5.179×10^{-6}	8.934

E Datasets and Implementation Details

Across all optimizer comparisons, we keep the dataset, model architecture, batch size, learning-rate schedule, and evaluation protocol fixed, and only change the optimizer update. For short-run supervised experiments, including `plainLM`[1] pre-training ablations and `VERL`[2] SFT, we use zero weight decay to isolate optimizer-state effects and provide a cleaner comparison with Signum-style updates. For long-run supervised experiments, such as the 20B-token setting, we use weight decay 0.1. For SFT and RLHF, we follow the default `VERL` learning-rate settings, 1×10^{-5} and 1×10^{-6}, respectively; for RLHF, we also keep the default actor weight decay 0.01.

For learning-rate selection, SFT and RLHF use the default `VERL` learning rates described above. For pre-training, we tune learning rates separately at the smaller scales before reporting the best result for each method. For the Llama-style 20M experiments, we sweep the learning-rate grid shown in Fig. 7 and Fig. 8. For the Pythia-style 160M experiments, we sweep $\{1, 3, 5, 8, 10\} \times 10^{-3}$ and select the best learning rate for each optimizer variant and β. For the Pythia-1B 20B-token experiment, a full sweep is too expensive; we therefore run a short 5B-token sweep for Adam at $\beta = 0.95$ over $\{0.001, 0.003, 0.005, 0.008\}$, select 0.001, and use this learning rate for all β values and optimizer variants.

For the small-scale pre-training models, we use compact decoder-only Transformer architectures implemented in `plainLM`. The Llama-style 20M model is our own `plainLM` implementation based on Hugging Face's `LlamaForCausalLM/LlamaConfig` interface, with its scale and layout adapted from a community 20M Llama-style checkpoint[3]; we do not use its pretrained weights. The Pythia-style 160M model follows a Pythia-like architecture [4] constructed in the same codebase. For the 1B-scale experiments, we use the official Llama-3.2-1B and Pythia-1B architectures; in pre-training experiments, all models are trained from scratch by reinitializing the parameters.

Pre-training. For pre-training, we use subsets of the FineWeb corpus [36] and implement all experiments in `plainLM`.

Table 17: Pre-training experimental setup.

Dataset	FineWeb 10B / 10B / 100B
Models	Llama-style 20M / Pythia-style 160M / Pythia-1B
Training budget	1.5B / 4B / 20B tokens
Sequence length	2048
Batch size	256
Warmup	512 / 1024 / 1024 steps
LR schedule	constant after warmup
Implementation	`plainLM`

Supervised Fine-Tuning (SFT). For SFT, we use UltraChat [11], formatted as multi-turn conversations and trained with the standard teacher-forcing objective. We fine-tune 1B-scale backbones, including Pythia-1B and Llama-3.2-1B, using the `VERL` SFT trainer.

Table 18: SFT experimental setup.

Dataset	UltraChat
Models	Pythia-1B, Llama-3.2-1B
Training budget	2000 steps
Maximum sequence length	2048
Batch size	128
Learning rate	1×10^{-5}
LR schedule	constant after warmup
Implementation	`VERL` SFT trainer

[1] https://github.com/Niccolo-Ajroldi/plainLM

[2] https://github.com/volcengine/verl

[3] https://huggingface.co/alan918727/TinyStories-LLaMA2-20M-256h-4l-colab

RLHF. For RLHF, we use UltraFeedback [9] as the prompt and preference source. We use Llama-3.2-1B-Instruct as the actor model and Skywork-Reward-V2-Llama-3.2-3B [27] as the reward model. The RL training pipeline is implemented with `VERL`, with our optimizer integrated into the actor update.

Table 19: RLHF/ReMax experimental setup.

Dataset	UltraFeedback
Actor model	Llama-3.2-1B-Instruct
Reward model	Skywork-Reward-V2-Llama-3.2-3B
Algorithm	ReMax [26]
Training budget	600 steps
Batch size	64 prompts
Maximum prompt length	768
Maximum response length	1024
Rollout temperature	1.0
Actor learning rate	1×10^{-6}
Actor LR schedule	constant, no warmup
Implementation	`VERL` RL trainer

The original 1B-scale experiments, including pre-training, SFT, and RLHF, are run on $8\times$H200 GPUs, while the original smaller-scale experiments are run on $16\times$RTX 4090 GPUs.

F Broader Impact

This work primarily aims to improve our understanding of Adam's internal adaptive dynamics. By identifying a natural-scale ratio-state representation in the tied-β regime, it provides a more transparent view of how the mean estimate and adaptive denominator interact during training. This mechanistic perspective may help guide optimizer analysis, hyperparameter choices, and the design of simpler Adam-like methods. This paper is methodological and does not involve data from human subjects, user studies, or deployment of a user-facing system. We do not directly study downstream model behavior; our focus is on optimizer dynamics and training efficiency.

A future packed implementation of the low-precision representation could reduce optimizer-state memory and make training more accessible; the present experiments do not measure such savings. Lower training costs could support beneficial applications but could also lower barriers to harmful downstream uses. These effects depend on the trained models and deployment context, which are outside our experimental scope.

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: Abstract and introduction accurately reflect the paper's contributions and scope

   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: Discussed in paper

   Guidelines:

   - The answer [N/A] means that the paper has no limitations 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 paper provides the full set of assumptions and a complete proof

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 paper specifies the optimizer, training settings, and evaluation definitions in Appendices C and D.2, together with the dataset and implementation details. Public reference implementations provide an avenue to reproduce the algorithm; the release is not an exhaustive archive of every historical run.

   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: [No]

Justification: The implementation, example training configurations, and setup instructions are public at https://github.com/agent-lab/hidden-ratio-adam. This initial release provides pre-training, SFT, ReMax, and CIFAR examples, but does not provide a complete per-result manifest and all historical distribution-analysis scripts, so we do not claim fully scripted reproduction of every main result.

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: Mentioned in Appendix

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: Appendix D.2 reports means and standard deviations over three seeds for selected pre-training and RLHF comparisons, and Appendix D.3 reports two-seed means and standard deviations for the CIFAR experiments. However, repeated-run uncertainty is not available for all main experiments or the additional diagnostic studies, so we retain a conservative No answer. These standard deviations describe variability across runs, not confidence intervals or significance tests.

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 rather 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: [No]

   Justification: The appendix identifies GPU types and counts for the original experiments and specifies training budgets. Per-run execution times, complete resource details for all additional studies, and total research compute are not available, so the disclosure is incomplete.

   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 research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics

   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 appendix discusses potential accessibility benefits of a future packed implementation and the possibility of lowering barriers to harmful downstream uses. It distinguishes these potential effects from measured results.

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 introduce or release a new high-risk pretrained model, image generator, or scraped dataset. The work focuses on optimizer-state representations and training methodology rather than releasing a deployable generative model.

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: Original sources are cited and the code release preserves component license files. However, the paper does not provide a complete inventory of licenses and usage terms for every dataset, model, and software asset.

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: [Yes]

    Justification: We provide the source code and README documentation at `https://github.com/agent-lab/hidden-ratio-adam`.

    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 paper does not involve crowdsourcing experiments or research with 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 paper does not involve crowdsourcing, user studies, or research with human subjects, so IRB approval or equivalent review 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 not an important, original, or non-standard component of the proposed optimizer methodology.

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.