Understanding Quantization of Optimizer States in LLM Pre-training: Dynamics of State Staleness and Effectiveness of State Resets

Anonymous ACL submission

Abstract

Quantizing optimizer states is becoming an important ingredient of memory-efficient large-scale pre-training, but the resulting optimizer dynamics remain only partially understood. We study low-precision exponential moving average (EMA) optimizer states and show how quantization can cause many nominal updates to round back to the same stored value, making the state effectively stale and slowing adaptation beyond what the nominal decay would suggest. We then develop a simple predictive model of stalling that estimates one-step stalling probabilities and characterizes how stalling builds up over time after the initialization. This perspective provides a mechanistic explanation for why optimizer-state resets help in low precision: once a quantized EMA becomes effectively stale, resetting it can temporarily restore responsiveness. Motivated by this picture, we derive a simple theory-guided heuristic for choosing useful reset periods, showing that in low precision the key question is not only whether resets help, but when they should be applied. Experiments in controlled simulations and LLM pre-training show that suitable reset schedules recover the performance lost to low-precision state storage while substantially reducing optimizer-state memory.

1 Introduction

Modern large-scale pre-training relies heavily on optimizers such as AdamW (Kingma and Ba, 2017; Loshchilov and Hutter, 2019), Lion (Chen et al., 2023), Shampoo (Gupta et al., 2018), SOAP (Vyas et al., 2025), and Muon (Liu et al., 2025), which maintain running statistics, typically exponential moving averages (EMAs), to smooth gradients, adapt step sizes, or construct preconditioners. At the same time, modern training pipelines increasingly store these states in reduced precision to lower memory cost and improve throughput (Dettmers et al., 2022; Peng et al., 2023; Xi et al., 2025; Wang et al., 2024). This makes optimizer-state quantization practically important, but its effect on the actual dynamics of the optimizer remains insufficiently understood.

A related empirical observation is that resetting optimizer states can improve pre-training (Huang et al., 2025b,a; Glentis et al., 2025). These resets are usually motivated through spike mitigation or generic training stability. However, in our experiments their benefit is strongly precision-dependent: resets help substantially when EMA states are stored in BF16 or lower precision, but are often much less useful in full precision. This suggests that state quantization and state resets should not be viewed separately: understanding why quantized states become ineffective is also the key to understanding when resets help.

In this paper, we study that connection through the dynamics of quantized EMAs. Consider a generic recursion

$$s_t = \beta s_{t-1} + (1-\beta)\Delta_t,$$

where s_t is an optimizer state, β is a decay coefficient close to 1, and Δ_t denotes the new update information. This form captures the state evolution of many widely used optimizers. In low precision, each step first forms a higher-precision update and then maps it back to the storage format. When the increment is too small relative to the local spacing of the floating-point grid, the update rounds back to the same stored value, so the state does not change. We refer to this phenomenon as *state-update stalling*, also known as signal swamping and unchanged-state dynamics in low-bit EMA updates (Higham, 1993; Wang et al., 2018; Xu et al., 2025). In high-decay recursions, repeated stalling slows adaptation, effectively lengthens the memory of the EMA beyond what the nominal decay suggests, and makes quantized states substantially less responsive than their full-precision counterparts.

This viewpoint also helps explain several recurring features of low-precision training. In particular, it clarifies why large β_2 values become more problematic as precision decreases, why stochastic rounding (Gupta et al., 2015; Ozkara et al., 2025; Croci et al., 2022) helps but does not fully remove the problem when stalling is severe, and why the first and second moments of Adam-like methods respond differently to quantization. Most importantly, it explains why resets can help and why their timing matters: once a quantized EMA has become effectively stale, resetting it can restore responsiveness, but resetting too early discards useful averaging. The key question is therefore not only whether resets help, but when they should be applied.

We formalize these effects through a simple predictive model of quantized EMA dynamics. The model combines local floating-point grid geometry with a short-horizon approximation to estimate one-step stalling probabilities and to characterize the finite post-initialization period during which a quantized EMA remains meaningfully responsive. This responsive *startup window* provides a natural basis both for understanding when resets should help and for choosing useful reset periods as a function of decay and precision. We validate the resulting predictions in controlled simulations and LLM pre-training, showing that suitable reset schedules recover the performance lost to low-precision state storage while substantially reducing optimizer-state memory.

2 Related Work

Optimizers for large-scale pre-training. Modern LLM pre-training increasingly follows compute-optimal scaling and open training recipes (Hoffmann et al., 2022; Touvron et al., 2023a,b; Groeneveld et al., 2024), most of which rely on optimizers with persistent state. Beyond momentum SGD and Adam/AdamW (Kingma and Ba, 2017; Loshchilov and Hutter, 2019), this includes memory-reduced adaptive methods such as Adafactor (Shazeer and Stern, 2018), SM3 (Anil et al., 2019) and Adam-mini (Zhang et al., 2025), structured preconditioners such as Shampoo (Gupta et al., 2018) and SOAP (Vyas et al., 2025), and more recent alternatives such as Lion (Chen et al., 2023) and Muon (Liu et al., 2025).

Low-precision and quantized pre-training. Recent low-precision pre-training builds on FP8 formats (Micikevicius et al., 2022), with systems such as FP8-LM (Peng et al., 2023) and COAT (Xi et al., 2025) demonstrating end-to-end FP8 LLM training, and more recent work extending to FP4 through mixed-precision or native low-bit stacks (Wang et al., 2025; Chmiel et al., 2025; Castro et al., 2025). Closest to our setting is direct quantization of optimizer states: 8-bit and 4-bit states for Adam-like methods (Dettmers et al., 2022; Li et al., 2023), 4-bit Shampoo preconditioners (Wang et al., 2024), and 8-bit quantized Muon states (Gupta et al., 2026). Our contribution is complementary: rather than proposing a new quantization or scaling scheme, we analyze a specific failure mode of quantized EMA states, state-update stalling under requantization, and use it to explain when stochastic rounding and periodic resets help.

Optimizer state resetting. Restarting or state resetting have a long history in optimization (O'Donoghue and Candès, 2012) and were later adapted to neural network training (Wang et al., 2022). More recent work shows that clearing optimizer states can help in non-stationary settings such as reinforcement learning (Asadi et al., 2023). In LLM training, methods such as SPAM (Huang et al., 2025b,a) and (Glentis et al., 2025) use momentum resets mainly for spike suppression and stabilization, while (Topollai and Choromanska, 2026) show that implicit resets via adapting the momentum coefficient can also improve performance. In contrast, we show that in low precision the benefit of resets is instead tied to quantized-state stalling: resets help by temporarily restoring responsiveness once EMA states become effectively stale.

3 State update stalling under quantized EMA states

Preliminaries. We consider exponential moving averages (EMAs) of the form

$$x_t = \beta x_{t-1} + (1-\beta) s_t, \tag{1}$$

where s_t is a stochastic signal (e.g., g_t or g_t^2) and $\beta \in (0,1)$ is the decay coefficient. In the quantized setting, the EMA state is stored in low precision but updated in high. Let x_t^q denote the stored state, let $D(\cdot)$ denote dequantization (equivalently, an exact cast to FP32), and let $Q(\cdot)$ denote quantization back to the storage format. One update therefore reads $x_{t-1} = D(x_{t-1}^q)$, forms the FP32 update $x_t =$

$\beta x_{t-1} + (1-\beta)s_t$, and writes back $x_t^q = Q(x_t)$. Since $D(\cdot)$ is just a cast, no new quantization error is introduced during the read phase; all new errors enter on the write phase through $Q(\cdot)$.

Definition 3.1 (State update stalling)**.** *We say that state update stalling occurs at step t if*

$$x_t^q = x_{t-1}^q. \tag{2}$$

Equivalently, the high precision update fails to move the stored state to a new representable value.

Quantization therefore acts as a nonlinear gate in the EMA recursion, suppressing sufficiently small updates. We focus on the standard Adam setting, where the optimizer maintains first- and second-moment EMAs of the gradient $g_t = \nabla f(\theta_t)$, namely the momentum $m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$ and the variance estimate $v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2$. Although the same gating mechanism applies to both moments, it is typically more pronounced for the second moment, whose quantization effects are often more visible in practice (Fishman et al., 2025; Tang et al., 2026; Yu et al., 2024). Moreover, the second moment admits a simple probabilistic model that matches empirical behavior well. We therefore focus the analysis on the second-moment EMA.

3.1 One-step stalling as a geometry problem

We analyze a single coordinate of the second-moment EMA:

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

Our analysis is coordinate-wise and does not depend on the global training objective; it requires only a short-window approximation for one gradient coordinate. Concretely, over the window of interest we assume that the coordinate has an approximately fixed local scale σ, and we analyze the EMA in the regime where v_{t-1} is of the same order as σ^2. To obtain closed-form expressions, we further use the local Gaussian approximation

$$\frac{g_t^2}{\sigma^2} \overset{d}{\approx} \chi_1^2, \qquad \text{equivalently} \qquad \frac{g_t}{\sigma} \overset{d}{\approx} \mathcal{N}(0,1). \tag{3}$$

This is a tractable short-window approximation for the step-wise distribution of a single gradient coordinate, and is in line with recent evidence that stochastic gradients exhibit substantial Gaussian structure in this sense (Xie et al., 2023).

The high precision update increment is then

$$\Delta_t := v_t - v_{t-1} = (1-\beta_2)\big(g_t^2 - v_{t-1}\big). \tag{4}$$

Let $u_t := \mathrm{ulp}(v_{t-1}^q)$ denote the local spacing between adjacent representable numbers around the stored value. Under nearest rounding, state-update stalling occurs whenever the high precision proposal remains within the rounding cell of the current stored value, i.e.

$$|\Delta_t| < \frac{u_t}{2}. \tag{5}$$

Equivalently, the stored value changes only if $|\Delta_t| \geq u_t/2$.

Let ε denote the characteristic relative grid spacing of the number format used for the EMA state under consideration. For the second moment, which is nonnegative, we use an unsigned E2M2 FP4 format, giving $\varepsilon = 2^{-2}$; for BF16 and E4M3 FP8 we use $\varepsilon = 2^{-7}$ and $\varepsilon = 2^{-3}$, respectively. The precise relation between u_t and ε, including the mantissa-dependent factor, is derived in Section A.1. Using the local-stationarity approximation $v_{t-1} \approx \sigma^2$ and defining

$$z_t := \frac{g_t^2}{\sigma^2} \sim \chi_1^2, \tag{6}$$

the update-to-spacing ratio can be summarized by a single effective precision parameter introduced next.

Under the effective-spacing approximation, we summarize the update-to-spacing ratio by a single effective precision parameter. Here ε denotes a format-dependent effective spacing parameter, and $\bar{m} := \frac{1}{\ln 2}$ is the mean normalized mantissa under the log-uniform mantissa model used here as a simple single-scalar approximation to the exact mantissa dependence in floating-point spacing (Goldberg, 1991); see Section A.1 for details.

Definition 3.2 (Effective precision ratio)**.** *We define*

$$\hat{\rho} := \frac{\varepsilon}{2(1-\beta_2)\bar{m}}. \tag{7}$$

With this convention,

$$\frac{|\Delta_t|}{u_t} \approx \frac{|z_t - 1|}{2\hat{\rho}}, \tag{8}$$

so all stalling statements below are controlled by $\hat{\rho}$.

3.2 Nearest rounding induces hard-gated stalling

Under nearest rounding (NR), the condition (5) becomes, under the effective-spacing approximation,

$$|z_t - 1| < \hat{\rho}. \tag{9}$$

Equivalently, the stored value changes only when $|z_t - 1| \geq \hat{\rho}$.

Proposition 3.3 (NR stalling probability)**.** *Let $z_t \sim \chi^2_1$. Under the effective-spacing approximation, the one-step probability of state-update stalling under nearest rounding is*

$$P^{\mathrm{NR}}_{\mathrm{stall}}(\hat{\rho}) \approx F_{\chi^2_1}(1+\hat{\rho}) - F_{\chi^2_1}(\max(0, 1-\hat{\rho})). \tag{10}$$

Corollary 3.4 (High-stalling regime under nearest rounding)**.** *When $\hat{\rho} \geq 1$, the lower term in (10) vanishes, so*

$$P^{\mathrm{NR}}_{\mathrm{stall}}(\hat{\rho}) \approx F_{\chi^2_1}(1+\hat{\rho}) \approx 1. \tag{11}$$

Thus, once the EMA reaches its steady magnitude, most updates are blocked by the rounding gate.

For $\beta_2 = 0.999$, the effective ratios from (7) give the steady-state stalling probabilities in Table 1 which are also verified empirically in Figure 2.

Format	ε	$\hat{\rho}$	$P^{\mathrm{NR}}_{\mathrm{stall}}(\hat{\rho})$
BF16	2^{-7}	2.71	≈ 0.946
FP8 (E4M3)	2^{-3}	43.3	≈ 1.000
FP4 (E2M2)	2^{-2}	86.6	≈ 1.000

Table 1: Theoretical steady-state stalling probabilities under nearest rounding (NR) from (7).

These values already reveal the core problem: at FP8 and FP4, once the second moment reaches its steady-state scale, *almost every step stalls*. In this regime the EMA rarely changes, so the optimizer effectively stops updating its estimate of gradient magnitude and instead operates with nearly frozen statistics. Importantly, this effect is largely independent of model or training scale. The analysis is expressed in terms of the normalized variable $z_t = g_t^2/\mathbb{E}[g^2]$, whose distribution does not depend on the absolute magnitude of gradients, while the ratio $\hat{\rho}$ governing stalling depends only on the precision format and decay β_2. As a result, changing the overall gradient scale, model width, or dataset shifts the EMA magnitude and floating-point spacing together, leaving the stalling probability essentially unchanged.

3.3 Empirical sensitivity to simulated moment stalling

To isolate the effect of stale EMA updates, we run a controlled intervention during LLaMA-60M pre-training on WikiText: after warmup, each update of a chosen moment buffer is independently skipped with probability p. This is not meant as a literal model of quantization-induced stalling, but as a direct stress test of optimizer sensitivity to persistent unchanged-state behavior.

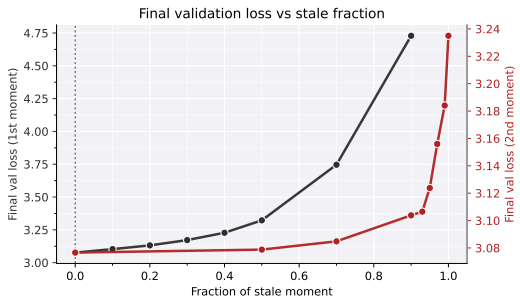

Figure 1: Simulated EMA stalling during LLaMA-60M pre-training on WikiText. After warmup, each moment update is skipped independently with probability p.

Figure 1 shows a clear trend: second-moment stalling degrades convergence gradually, mainly through worse final loss, whereas first-moment stalling is far more destabilizing. This is consistent with their roles in Adam-like methods: the second moment mainly controls effective learning-rate scaling, while the first moment carries directional information and is therefore more sensitive to missed updates.

4 What stalling does to the EMA dynamics

4.1 A coarse-grained effective decay approximation

A useful summary of stalling is through an *effective decay* approximation. If a step stalls, the stored state remains unchanged ($v_t^q = v_{t-1}^q$), which is equivalent to using decay 1 on that step rather than the nominal decay β_2. If the empirical stalling rate is approximately constant at P_{stall} over a short window, only a fraction $1 - P_{\mathrm{stall}}$ of steps effectively apply the nominal EMA contraction, giving

$$\beta_2^{\mathrm{eff}} \approx 1 - (1-\beta_2)(1 - P_{\mathrm{stall}}), \tag{12}$$

and, for $\tau := \frac{1}{1-\beta_2}$,

$$\tau_{\mathrm{eff}} \approx \frac{1}{(1-\beta_2)(1-P_{\mathrm{stall}})} = \frac{\tau}{1 - P_{\mathrm{stall}}}. \tag{13}$$

This is not an exact reduction of the quantized recursion, since stalling events are state dependent and correlated across steps, but a local mean-field summary of how a high stalled fraction slows EMA response. The same persistence also makes successive optimizer updates more temporally correlated, a behavior linked to instability in Adam-like methods (Molybog et al., 2023).

4.2 A finite startup window before stalling dominates

Stalling does not occur immediately after state initialization. When the EMA state is small, the quantization grid is fine relative to the typical update, so many updates pass through. As the state grows toward its stationary scale, the probability of a stalled update rises, creating a finite *startup window* during which the EMA remains meaningfully responsive.

Under the same idealized model as above, after j updates have been accumulated from a zero-initialized state, the reference high precision trajectory is

$$v_j^{\mathrm{ref}} := \sigma^2 \phi_j, \qquad \phi_j := 1 - \beta_2^j. \tag{14}$$

Proposition 4.1 (Transient NR stalling probability)**.** *With z_t defined in (6), the transient stalling probability under nearest rounding is approximated by*

$$P_{\mathrm{stall}}^{\mathrm{NR}}(j) \approx F_{\chi_1^2}\big((1+\hat{\rho})\phi_j\big), \tag{15}$$

for the formats of interest, where $\hat{\rho} \geq 1$.

This expression centers the stalling event at the *current* state scale ϕ_j rather than the steady-state value, and therefore captures the increase in stalling as the EMA grows away from initialization.

Definition 4.2 (Startup window length)**.** *For a tolerance $P_0 \in (0,1)$, define the* startup window length *as the smallest j such that $P_{\mathrm{stall}}(j) \geq P_0$.*

Proposition 4.3 (Ideal startup window)**.** *Let*

$$\phi^\star(P_0) := \frac{F_{\chi_1^2}^{-1}(P_0)}{1+\hat{\rho}}.$$

Whenever $\phi^\star(P_0) < 1$, the ideal startup window is

$$j_{\mathrm{ideal}}^\star(P_0) = \left\lceil \frac{\log\big(1-\phi^\star(P_0)\big)}{\log \beta_2} \right\rceil. \tag{16}$$

In practice, measured stalling often starts from a nonzero floor P_{init}, so we use

$$P_{\mathrm{total}}(j) = P_{\mathrm{init}} + (1 - P_{\mathrm{init}})\, P_{\mathrm{stall}}(j), \tag{17}$$

which is equivalent to replacing P_0 by the effective threshold

$$P_0^{\mathrm{eff}} := \frac{P_0 - P_{\mathrm{init}}}{1 - P_{\mathrm{init}}}, \tag{18}$$

and therefore

$$j^\star(P_0) = \begin{cases} 0, & P_0 \leq P_{\mathrm{init}}, \\ j_{\mathrm{ideal}}^\star(P_0^{\mathrm{eff}}), & P_0 > P_{\mathrm{init}}. \end{cases} \tag{19}$$

Details on the empirical floor P_{init} and numerical startup-window values are deferred to Sections A.4 and A.5. Figure 2 illustrates two main trends. First, the stalled fraction is already nonzero immediately after initialization, and this empirical floor P_{init} is itself larger at lower precision; we discuss its origin and estimate it in Section A.4. Second, the stalled fraction rises as the second moment grows toward its steady-state scale, producing a finite responsive startup window before the EMA becomes largely stale. This window is longest in BF16 and substantially shorter in FP8; for FP4, accounting for the empirical floor is essential to separate the reset-sensitive buildup of stalling from the large format-dependent baseline. This precision-dependent startup behavior will later provide the key intuition for why state resets can help and why their timing matters.

4.3 First moment stalling

The first moment behaves qualitatively similar to the second even though its update scale and ulp scale do not normalize in the same way. For the second moment, the update is driven by g_t^2 and the stored state is of the same order as the local variance, so the variance scale largely cancels from the update-to-ulp ratio, yielding the nearly universal approximation developed above. For the first moment, $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$, the relevant scale is set directly by the signed gradient. As a result, first-moment stalling depends on the local signal-to-noise ratio between the mean gradient and its fluctuations, and therefore varies across coordinates rather than being determined primarily by the precision format. This makes the first moment less amenable to a simple closed-form analysis of the kind developed above for the second moment. Empirically, however, its behavior follows a qualitatively similar pattern, as we show in the Appendix Section A.6.

5 A recipe for stalling mitigation

Quantized EMA states fail primarily because stalling makes the effective EMA dynamics too

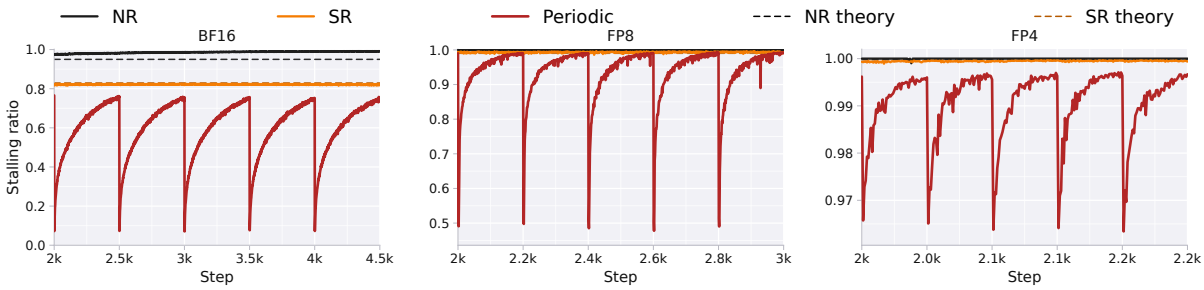

Figure 2: Measured stalling fraction during LLM pre-training for quantized EMA states. As the second moment grows away from initialization, the stalled fraction rises and then plateaus near its steady-state level. Lower-precision formats enter the stalled regime earlier and remain there longer. 100M-parameter LLama model trained on C4.

slow. We discuss two complementary ways to mitigate this.

5.1 Stochastic rounding

Stochastic rounding (SR) rounds to one of the two neighboring representable values with probabilities proportional to distance, replacing the NR hard gate by a soft one.

Proposition 5.1 (SR stalling probability)**.** *Under the effective-spacing approximation at steady state, with z_t as in (6), the conditional probability that SR returns to the current stored value is*

$$p^{\mathrm{SR}}_{\mathrm{stall}}(z_t;\hat{\rho}) = \max\Big(0,\ 1 - \frac{|z_t - 1|}{2\hat{\rho}}\Big), \qquad (20)$$

and hence $P^{\mathrm{SR}}_{\mathrm{stall}}(\hat{\rho}) = \mathbb{E}\big[p^{\mathrm{SR}}_{\mathrm{stall}}(z_t;\hat{\rho})\big]$

SR is only a partial fix. At $\beta_2 = 0.999$, the steady-state stalling probabilities remain high (Table 2) and are consistent with the empirical behavior in Figure 2.

Format	BF16	FP8	FP4
$P^{\mathrm{SR}}_{\mathrm{stall}}$	0.825	0.989	0.994

Table 2: Theoretical steady-state stalling probabilities under SR for $\beta_2 = 0.999$.

Thus SR materially helps at BF16, but at FP8 and FP4 the stalled regime remains dominant. This analysis isolates unchanged-state probabilities only; in practice SR may also help by reducing directional quantization bias (Croci et al., 2022).

5.2 Periodic state resetting

When steady-state stalling is high, a quantized EMA eventually becomes nearly frozen, so resetting the state is a natural way to restore responsiveness. The transient analysis above shows, however, that this responsiveness is not lost immediately: after each initialization there is a finite startup window during which the EMA remains adaptive before stalling dominates. Since the length of this window is governed primarily by the effective precision ratio $\hat{\rho}$ in Definition 3.2, it is largely stable throughout training for a fixed precision regime. This both motivates periodic resets and suggests how to time them: resets should occur once the startup window is largely exhausted, but not so early that they discard useful averaging.

We compare two quantities over a cycle of length K: the accumulated *reset-sensitive* staleness, and the statistical benefit still left to gain from continuing the EMA. Removing the reset-insensitive floor P_{init}, we define the normalized stalling progress

$$S(j) := \frac{P_{\mathrm{total}}(j) - P_{\mathrm{init}}}{P^{\mathrm{ss}}_{\mathrm{total}} - P_{\mathrm{init}}} = \frac{P_{\mathrm{stall}}(j)}{P^{\mathrm{ss}}_{\mathrm{stall}}}, \qquad (21)$$

where $P^{\mathrm{ss}}_{\mathrm{total}} = P_{\mathrm{init}} + (1 - P_{\mathrm{init}})\, P^{\mathrm{ss}}_{\mathrm{stall}}$. Rather than using the endpoint $S(K)$ alone, we average over the cycle and count only the excess above a tolerance level s_0:

$$\bar{S}_{s_0}(K) := \frac{1}{K}\sum_{j=1}^{K}\left[\frac{S(j) - s_0}{1 - s_0}\right]_+, \qquad (22)$$

and compare it with the remaining statistical error of the bias-corrected EMA, whose derivation is given in the Appendix Section A.7,

$$E(K) := \frac{2\beta_2^K}{1 + \beta_2^K}. \qquad (23)$$

Proposition 5.2 (Heuristic reset period)**.** *Define*

$$K^\star := \min\big\{K \geq 1 :\ \bar{S}_{s_0}(K) \geq E(K)\big\}. \qquad (24)$$

Using $P_{\mathrm{stall}}(j) \approx F_{\chi^2_1}\big((1+\hat{\rho})(1-\beta_2^j)\big)$ *and* $P^{\mathrm{ss}}_{\mathrm{stall}} \approx F_{\chi^2_1}(1+\hat{\rho})$ *the condition in (24) is a one-dimensional monotone test in K. We use $s_0 = 0.6$ throughout.*

Intuitively, we reset once the cycle spends enough time in a highly stalled regime that the remaining benefit of further averaging is no longer

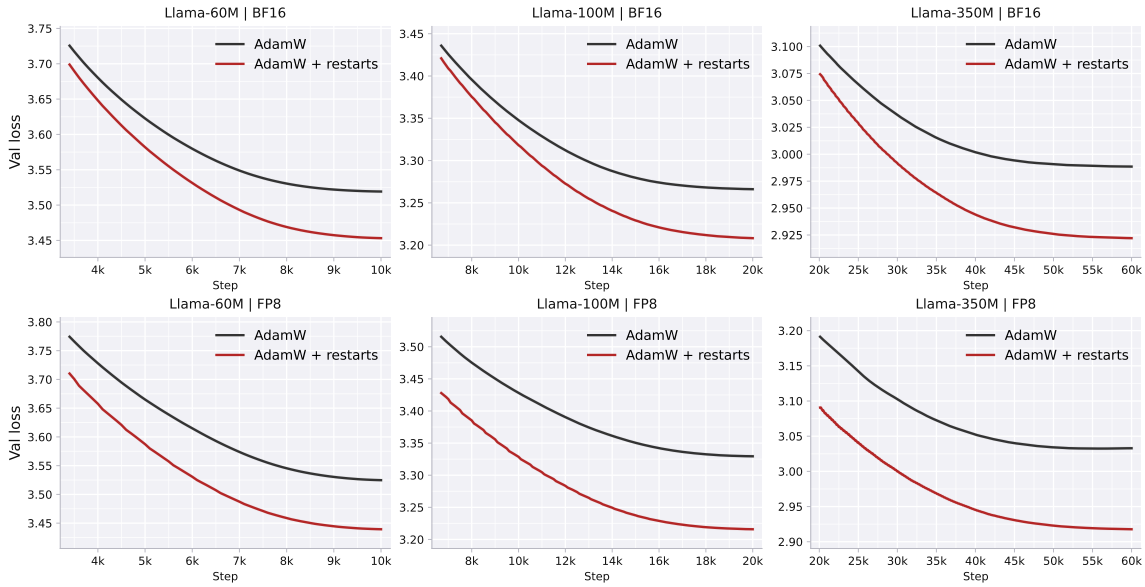

Figure 3: Training loss curves for the 3 model scales under 2 different precisions

worth the cost. Solving (24) for the precisions considered in this work gives the periods in Table 3, which should be viewed as selecting a good operating region rather than a uniquely optimal period.

	BF16	FP8	FP4
$K^\star$	≈ 1116	≈ 320	≈ 224

Table 3: Heuristic reset periods from (24) with $\beta_2 = 0.999$ and $s_0 = 0.6$.

Resetting the first moment is mainly useful in very aggressive regimes such as FP4, since at higher precisions its steady-state stalling rate is often below the initial post-reset level P_{init}, as also shown in Section A.6.

6 Experiments

6.1 Settings

We follow the pre-training protocol of (Zhao et al., 2024; Lialin et al., 2024) on C4 (Raffel et al., 2020) using the LLaMA architecture family (Touvron et al., 2023a), matching the small-scale GaLore setup for the 60M, 100M, and 350M models. Concretely, the models are trained for 10K/20K/60K steps on approximately 1.3B/2.6B/7.8B tokens, with maximum sequence length 256, token batch size 131K, linear warmup over the first 10% of training, cosine decay to 10% of the initial learning rate, and no data repetition. We evaluate three optimizer-state precision regimes: BF16, FP8 using E4M3 for both moments, and FP4 with the first moment in FP4 (E2M1) and the unsigned second moment in FP4 (E2M2). For FP8 we use per-tensor scaling; for FP4 we use block-wise scaling with block size 128 and exclude the zero point from the second-moment grid, following (Li et al., 2023). Hyperparameters are given in Section A.9.

For each regime, we compare vanilla AdamW against AdamW with stochastic rounding and periodic EMA resets. Unless otherwise stated, the second-moment reset period is chosen using Equation (24), with P_{init} estimated empirically from the stalled fraction at the first step after state initialization and $s_0 = 0.6$. Although this heuristic is derived under nearest rounding, we use it as a practical proxy under stochastic rounding as well: SR lowers unchanged-state probabilities but preserves the same qualitative dependence on precision and decay, so the resulting periods still identify a good reset regime. This gives periods of 1000 for BF16, 300 for FP8, and 200 for FP4. The first moment is reset with the same period.

6.2 Results

We first evaluate whether periodic EMA resets improve training under quantized optimizer-state storage across several practical low-precision regimes. Table 4 and Figure 3 show that stochastic rounding and periodic EMA resets consistently improve final performance in BF16, FP8, and FP4. Although implementation choices such as scaling affect how severe stalling becomes, they do not remove the underlying problem of staleness.

6.3 Ablation Studies

All ablations are conducted on the 100M model.

Configuration	60M	100M	350M
BF16 AdamW	3.52	3.27	2.99
BF16 AdamW + Resets	3.45	3.21	2.92
FP8 AdamW	3.53	3.33	3.02
FP8 AdamW + Resets	3.44	3.22	2.92
FP4 AdamW	3.52	3.32	3.03
FP4 AdamW + Resets	3.48	3.26	2.97

Table 4: Validation loss across model scales.

Sensitivity to the Reset Period. We vary the second-moment reset period over a wide range while keeping all other hyperparameters fixed. Figure 4 shows that performance is robust over a broad range, with the best results attained near the period predicted by Equation (24). This supports the view that the useful reset window is governed by quantized EMA stalling.

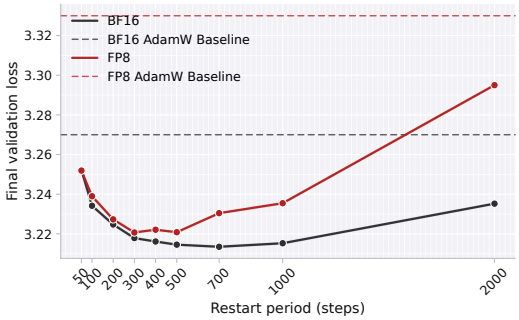

Figure 4: Sensitivity to the reset period.

Contribution of Different Components. We further isolate each mitigation component by keeping training otherwise in FP32 and quantizing only the optimizer states. Table 5 shows that both stochastic rounding and resets help on their own in quantized regimes, while their combination gives the best overall performance. By contrast, applying the same reset schedule in full FP32 slightly hurts performance, reinforcing our main claim that the benefit of resets is tied to low-precision state staleness rather than to resets being universally useful.

Configuration	FP32	BF16	FP8	FP4
NR only	3.155	3.181	3.193	3.199
SR only	–	3.156	3.160	3.182
Resets only	3.161	3.158	3.159	3.166
SR + Resets	–	3.155	3.155	3.160
Memory savings	0%	50%	75%	87.5%

Table 5: Component ablation with FP32 training and quantized optimizer states only. In the FP32 column, we apply resets only, with period 1000 for both moments.

Asymmetric and Adaptive Resets. We next consider two extensions of the basic reset strategy: asymmetric reset periods for the two moments, and an adaptive rule based on Proposition 5.2 and detailed in Section A.8. In the latter case, we set $P_{\text{stall}}^{\text{ss}} = 1$ and $s_0 = 0.6$, estimate $P_{\text{stall}}(j)$ online, and trigger a reset once $\bar{S}_{s_0}(K) - E(K) > 0$. As shown in Table 6, using no resets for the first moment and period 300 for the second slightly improves on the symmetric baseline, consistent with the fact that in FP8 first-moment steady-state stalling is typically the same as its initial floor P_{init}, see Section A.6. The adaptive rule is comparable while avoiding a fixed reset period.

Reset Strategy	Validation Loss
Symmetric Resets	3.223
Asymmetric Resets	3.212
Adaptive Resets	3.225

Table 6: Different resetting policies under FP8.

Extensions to Other EMA-Based Optimizers. Finally, we test the same reset strategy on SOAP, also resetting the EMA states of its left and right preconditioners using the same period as in AdamW and following the setup of (Wang et al., 2024). Table 7 shows that resets again improve low-precision performance.

Configuration	BF16	FP8
SOAP	3.22	4.15
SOAP + Resets	3.19	3.23

Table 7: SOAP with and without EMA resets.

7 Conclusion

Low-precision optimizer states do not just save memory; they change the dynamics of optimization. Our central message is that quantized EMA states can become effectively stale for long stretches of training, and that this staleness is a useful lens for understanding both the failure modes of low-precision training and the surprising effectiveness of state resets. Viewed this way, resets are not merely a heuristic borrowed from successful training recipes: they are a targeted way to restore responsiveness once the optimizer state has stopped meaningfully evolving. Just as importantly, our analysis suggests that good resets are not arbitrary, their timing matters. We hope this perspective helps shift the discussion of low-precision optimizer states from storage alone to dynamics, and motivates future work on optimizers and quantization schemes designed explicitly to remain responsive under quantization.

8 Limitations

Our analysis is designed to isolate a specific and practically important mechanism, state-update stalling under quantized EMA storage, rather than to model every aspect of low-precision training. As a result, some components are intentionally approximate, including the local scalar analysis of the EMA dynamics and the heuristic used to choose reset periods. These simplifications still match the main empirical trends well in our experiments, but the exact quantitative behavior can depend on implementation choices such as scaling strategy, rounding mode, and optimizer variant. In addition, our empirical study focuses on the LLaMA pre-training settings and EMA-based optimizers considered here. Extending the analysis to broader training regimes, larger scales, and additional classes of stateful optimizers is a natural direction for future work.

References

Rohan Anil, Vineet Gupta, Tomer Koren, and Yoram Singer. 2019. Memory efficient adaptive optimization. *Advances in Neural Information Processing Systems*, 32.

Kavosh Asadi, Rasool Fakoor, and Shoham Sabach. 2023. Resetting the optimizer in deep rl: An empirical study. In *Advances in Neural Information Processing Systems*, volume 36, pages 72284–72324. Curran Associates, Inc.

Roberto L. Castro, Andrei Panferov, Soroush Tabesh, Oliver Sieberling, Jiale Chen, Mahdi Nikdan, Saleh Ashkboos, and Dan Alistarh. 2025. Quartet: Native fp4 training can be optimal for large language models. *Advances in neural information processing systems*.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and 1 others. 2023. Symbolic discovery of optimization algorithms. *Advances in neural information processing systems*, 36:49205–49233.

Brian Chmiel, Maxim Fishman, Ron Banner, and Daniel Soudry. 2025. Fp4 all the way: Fully quantized training of llms. *Advances in neural information processing systems*.

Matteo Croci, Massimiliano Fasi, Nicholas J Higham, Theo Mary, and Mantas Mikaitis. 2022. Stochastic rounding: implementation, error analysis and applications. *Royal Society Open Science*, 9(3).

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. *International Conference on Learning Representations*.

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

Athanasios Glentis, Jiaxiang Li, Qiulin Shang, Andi Han, Ioannis Tsaknakis, Quan Wei, and Mingyi Hong. 2025. Scalable parameter and memory efficient pretraining for llm: Recent algorithmic advances and benchmarking. *arXiv preprint arXiv:2505.22922*.

David Goldberg. 1991. What every computer scientist should know about floating-point arithmetic. *ACM computing surveys (CSUR)*, 23(1):5–48.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Aman Gupta, Rafael Celente, Abhishek Shivanna, DT Braithwaite, Gregory Dexter, Shao Tang, Hiroto Udagawa, Daniel Silva, Rohan Ramanath, and S Sathiya Keerthi. 2026. Effective quantization of muon optimizer states. *arXiv preprint arXiv:2509.23106*.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. 2015. Deep learning with limited numerical precision. In *International conference on machine learning*, pages 1737–1746. PMLR.

Vineet Gupta, Tomer Koren, and Yoram Singer. 2018. Shampoo: Preconditioned stochastic tensor optimization. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1842–1850. PMLR.

Nicholas J Higham. 1993. The accuracy of floating point summation. *SIAM Journal on Scientific Computing*, 14(4):783–799.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, and 1 others. 2022. An empirical analysis of compute-optimal large language model training. *Advances in neural information processing systems*, 35:30016–30030.

Tianjin Huang, Haotian Hu, Zhenyu Zhang, Gaojie Jin, Xiang Li, Li Shen, Tianlong Chen, Lu Liu, Qingsong Wen, Zhangyang Wang, and 1 others. 2025a. Stablespam: How to train in 4-bit more stably than 16-bit adam. In *First Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*.

Tianjin Huang, Ziquan Zhu, Gaojie Jin, Lu Liu, Zhangyang Wang, and Shiwei Liu. 2025b. Spam: Spike-aware adam with momentum reset for stable llm training. *International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization. *Preprint*, arXiv:1412.6980.

Bingrui Li, Jianfei Chen, and Jun Zhu. 2023. Memory efficient optimizers with 4-bit states. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Vladislav Lialin, Sherin Muckatira, Namrata Shivagunde, and Anna Rumshisky. 2024. ReloRA: High-rank training through low-rank updates. In *The Twelfth International Conference on Learning Representations*.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, and 9 others. 2025. Muon is scalable for llm training. *Preprint*, arXiv:2502.16982.

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

Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep K. Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, Naveen Mellempudi, Stuart F. Oberman, Mohammad Shoeybi, Michael Siu, and Hao Wu. 2022. Fp8 formats for deep learning. *ArXiv*, abs/2209.05433.

Igor Molybog, Peter Albert, Moya Chen, Zachary DeVito, David Esiobu, Naman Goyal, Punit Singh Koura, Sharan Narang, Andrew Poulton, Ruan Silva, and 1 others. 2023. A theory on adam instability in large-scale machine learning. *arXiv preprint arXiv:2304.09871*.

Brendan O'Donoghue and Emmanuel J. Candès. 2012. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, 15:715 – 732.

Kaan Ozkara, Tao Yu, and Youngsuk Park. 2025. Stochastic rounding for llm training: Theory and practice. In *International Conference on Artificial Intelligence and Statistics*.

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. 2023. Fp8-lm: Training fp8 large language models. *Preprint*, arXiv:2310.18313.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4596–4604. PMLR.

Xuan Tang, Jichu Li, and Difan Zou. 2026. A convergence analysis of adaptive optimizers under floating-point quantization. *International Conference on Learning Representations*.

Kristi Topollai and Anna Ewa Choromanska. 2026. Adaptive memory momentum via a model-based framework for deep learning optimization. In *The 29th International Conference on Artificial Intelligence and Statistics*.

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. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Nikhil Vyas, Depen Morwani, Rosie Zhao, Itai Shapira, David Brandfonbrener, Lucas Janson, and Sham Kakade. 2025. Soap: Improving and stabilizing shampoo using adam for language modeling. In *International Conference on Learning Representations*, volume 2025, pages 93423–93444.

Bao Wang, Tan Nguyen, Tao Sun, Andrea L. Bertozzi, Richard G. Baraniuk, and Stanley J. Osher. 2022. Scheduled restart momentum for accelerated stochastic gradient descent. *SIAM Journal on Imaging Sciences*, 15(2):738–761.

Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. 2018. Training deep neural networks with 8-bit floating point numbers. *Advances in neural information processing systems*, 31.

Ruizhe Wang, Yeyun Gong, Xiao Liu, Guoshuai Zhao, Ziyue Yang, Baining Guo, Zheng-Jun Zha, and Peng Cheng. 2025. Optimizing large language model training using FP4 quantization. In *Proceedings of the*

42nd International Conference on Machine Learning, volume 267 of *Proceedings of Machine Learning Research*, pages 62937–62957. PMLR.

Sike Wang, Pan Zhou, Jia Li, and Hua Huang. 2024. 4-bit shampoo for memory-efficient network training. *Advances in Neural Information Processing Systems*, 37:126997–127029.

Haocheng Xi, Han Cai, Ligeng Zhu, Yao Lu, Kurt Keutzer, Jianfei Chen, and Song Han. 2025. Coat: Compressing optimizer states and activation for memory-efficient fp8 training. *International Conference on Learning Representations*.

Zeke Xie, Qian-Yuan Tang, Mingming Sun, and Ping Li. 2023. On the overlooked structure of stochastic gradients. *Advances in Neural Information Processing Systems*, 36:66257–66276.

Cong Xu, Wenbin Liang, Mo Yu, Anan Liu, Ke-Yue Zhang, Shunli Wang, Lizhuang Ma, Jianyong Wang, Jun Wang, and Wei Zhang. 2025. Pushing the limits of low-bit optimizers: A focus on ema dynamics. *Preprint*, arXiv:2505.00347.

Tao Yu, Gaurav Gupta, Karthick Gopalswamy, Amith Mamidala, Hao Zhou, Jeffrey Huynh, Youngsuk Park, Ron Diamant, Anoop Deoras, and Luke Huan. 2024. Collage: light-weight low-precision strategy for llm training. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik P Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. 2025. Adam-mini: Use fewer learning rates to gain more. *International Conference on Learning Representations*.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61121–61143. PMLR.

A Appendix

A.1 Floating-point grid spacing and the effective precision ratio

For a normalized floating-point number,

$$x = m\,2^e, \qquad m \in [1, 2), \quad e \in \mathbb{Z},$$

the spacing between adjacent representable numbers within a fixed exponent binade is

$$\mathrm{ulp}(x) = 2^{e-p},$$

where p is the number of stored mantissa bits. Hence

$$\mathrm{ulp}(x) = 2^{e-p} = \frac{2^{-p}}{m}\,x, \tag{25}$$

so the exact local relative spacing is

$$\frac{\mathrm{ulp}(x)}{x} = \frac{2^{-p}}{m}.$$

In the main text we do not carry this binade- and mantissa-dependent quantity through all formulas. Instead, we summarize each format by a single parameter ε, writing

$$\mathrm{ulp}(x) \approx \frac{\varepsilon}{m}\,x.$$

For the normalized regimes considered here, we take $\varepsilon = 2^{-p}$; in particular, $\varepsilon = 2^{-7}$ for BF16, $\varepsilon = 2^{-3}$ for E4M3 FP8, and $\varepsilon = 2^{-2}$ for the unsigned E2M2 FP4 format used for the second moment.

In our setting, the current EMA state is v_{t-1} and the high precision proposal is

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

Hence the update increment is

$$\Delta_t := v_t - v_{t-1} = (1-\beta_2)(g_t^2 - v_{t-1}).$$

Writing $v_{t-1} = m_t 2^{e_t}$ and using (25), the local spacing around the stored value is approximated by

$$u_t = \mathrm{ulp}(v_{t-1}^q) \approx \mathrm{ulp}(v_{t-1}) = \frac{\varepsilon}{m_t} v_{t-1}.$$

Under nearest rounding, state-update stalling occurs whenever $|\Delta_t| < u_t/2$; equivalently,

$$|\Delta_t| \geq \frac{u_t}{2}$$

is required for the stored value to change. Therefore

$$\frac{|\Delta_t|}{u_t} = \frac{(1-\beta_2)|g_t^2 - v_{t-1}|}{(\varepsilon/m_t)\,v_{t-1}}. \tag{26}$$

Under the local-stationarity approximation $v_{t-1} \approx \sigma^2$, and with

$$z_t := \frac{g_t^2}{\sigma^2} \sim \chi_1^2,$$

this becomes

$$\frac{|\Delta_t|}{u_t} \approx \frac{(1-\beta_2) m_t |z_t - 1|}{\varepsilon}. \tag{27}$$

Hence the nearest-rounding gate $|\Delta_t| \geq u_t/2$ is equivalent to

$$|z_t - 1| \geq \frac{\varepsilon}{2(1-\beta_2) m_t}. \tag{28}$$

The threshold depends on the normalized mantissa m_t, which varies across coordinates. To reduce this dependence to a single scalar, we adopt a log-uniform mantissa model and treat $\log_2 m_t$ as approximately uniform on $[0, 1]$, corresponding to

$$f(m) = \frac{1}{m \ln 2}, \qquad m \in [1, 2].$$

Under this model,

$$\bar{m} := \mathbb{E}[m] = \int_1^2 m \cdot \frac{1}{m \ln 2}\, dm = \frac{1}{\ln 2}. \tag{29}$$

We then replace the coordinate-dependent threshold in (28) by its single-scalar surrogate obtained from $\bar{m}$. This is an effective-spacing approximation rather than an exact averaging of the stalling probability, and it is used only to obtain the compact predictor $\hat{\rho}$:

$$\hat{\rho} := \frac{\varepsilon}{2(1-\beta_2)\bar{m}}, \qquad \bar{m} = \frac{1}{\ln 2}. \tag{30}$$

Thus the gate is approximated by $|z_t - 1| \geq \hat{\rho}$, or equivalently

$$\frac{|\Delta_t|}{u_t} \approx \frac{|z_t - 1|}{2\hat{\rho}}.$$

A.2 NR stalling probability

We derive Proposition 3.3. Under nearest rounding (NR), the stored state stalls if and only if the high precision proposal rounds back to the current stored value. Under the hard-gate approximation from the main text,

$$|\Delta_t| < \frac{u_t}{2} \quad \Longleftrightarrow \quad |z_t - 1| < \hat{\rho}, \qquad z_t \sim \chi_1^2.$$

Equivalently, the stored value changes only when $|z_t - 1| \geq \hat{\rho}$. Since $z_t \geq 0$, the stalling region is

$$z_t \in \big(\max(0,\, 1-\hat{\rho}),\, 1+\hat{\rho}\big).$$

Therefore,

$$P_{\text{stall}}^{\text{NR}}(\hat{\rho}) = \mathbb{P}(\max(0, 1-\hat{\rho}) < z_t < 1+\hat{\rho}) \tag{31}$$

$$= F_{\chi_1^2}(1+\hat{\rho}) - F_{\chi_1^2}(\max(0, 1-\hat{\rho})). \tag{32}$$

Useful closed form for the χ_1^2 CDF. If $z = X^2$ with $X \sim \mathcal{N}(0,1)$, then for $x \geq 0$,

$$F_{\chi_1^2}(x) = \mathbb{P}(|X| \leq \sqrt{x}) = 2\Phi(\sqrt{x}) - 1.$$

This identity is convenient for numerical evaluation and inversion of the stalling thresholds.

A.3 SR stalling probability

We derive Proposition 5.1. Fix a grid with spacing u around the current stored value $\bar{v}$, and suppose the higher-precision proposal is $\tilde{v} = \bar{v} + \delta$. Under stochastic rounding (SR), if $\tilde{v}$ lies between $\bar{v}$ and an adjacent grid point at distance u, then the probability of rounding back to $\bar{v}$ is proportional to proximity:

$$\mathbb{P}\big(\mathcal{Q}_{\text{SR}}(\tilde{v}) = \bar{v} \mid \delta\big) = \begin{cases} 1 - |\delta|/u, & |\delta| < u, \\ 0, & |\delta| \geq u. \end{cases}$$

At steady state, writing $\alpha := 1 - \beta_2$, using

$$\delta \approx \alpha\bar{v}(z_t - 1), \qquad u \approx (\varepsilon/\bar{m})\bar{v}, \qquad z_t \sim \chi_1^2,$$

gives

$$\frac{|\delta|}{u} \approx \frac{\alpha\bar{m}}{\varepsilon}|z_t - 1| = \frac{|z_t - 1|}{2\hat{\rho}}.$$

Hence the conditional SR stalling probability is

$$p_{\text{stall}}^{\text{SR}}(z_t; \hat{\rho}) = \max\Big(0,\ 1 - \frac{|z_t - 1|}{2\hat{\rho}}\Big), \tag{33}$$

and averaging over z_t yields

$$P_{\text{stall}}^{\text{SR}}(\hat{\rho}) = \mathbb{E}_{z_t \sim \chi_1^2}\left[\max\Big(0,\ 1 - \frac{|z_t - 1|}{2\hat{\rho}}\Big)\right]. \tag{34}$$

Large-$\hat{\rho}$ approximation. When $\hat{\rho}$ is large, the truncation at 0 is rarely active, since $|z_t - 1|$ is typically $O(1)$. Dropping the truncation gives

$$P_{\text{stall}}^{\text{SR}}(\hat{\rho}) \approx 1 - \frac{1}{2\hat{\rho}}\,\mu_1, \qquad \mu_1 := \mathbb{E}[|z_t - 1|].$$

For $z_t = X^2$ with $X \sim \mathcal{N}(0,1)$, μ_1 has a closed form.

Lemma A.1 (Closed form for μ_1)**.** *If $X \sim \mathcal{N}(0,1)$ and $z_t = X^2$, then*

$$\mu_1 = \mathbb{E}[|X^2 - 1|] = 4\phi(1) = \frac{4}{\sqrt{2\pi}}e^{-1/2} \approx 0.9679,$$

where ϕ is the standard normal density.

Proof. By symmetry,

$$\mu_1 = \mathbb{E}[|X^2 - 1|] = 2\int_0^\infty |x^2 - 1|\phi(x)\,dx$$
$$= 2\Big(\int_0^1 (1 - x^2)\phi(x)\,dx + \int_1^\infty (x^2 - 1)\phi(x)$$

Since $\mathbb{E}[X^2 - 1] = 0$, the positive and negative parts have equal mass:

$$\int_1^\infty (x^2 - 1)\phi(x)\,dx = \int_0^1 (1 - x^2)\phi(x)\,dx.$$

Hence

$$\mu_1 = 4\int_0^1 (1 - x^2)\phi(x)\,dx.$$

Using $\phi'(x) = -x\phi(x)$,

$$\int_0^1 x^2\phi(x)\,dx = \big[-x\phi(x)\big]_0^1 + \int_0^1 \phi(x)\,dx$$
$$= -\phi(1) + \int_0^1 \phi(x)\,dx,$$

so

$$\int_0^1 (1 - x^2)\phi(x)\,dx = \phi(1),$$

which gives $\mu_1 = 4\phi(1)$. □

This subsection isolates the unchanged-state probability under SR. It does not attempt to capture the full benefit of SR, which may also arise from the conditional unbiasedness of the quantization error even on steps where the stored state changes.

A.4 Empirical initial floor P_{init}

Empirically, the stalled fraction immediately after initialization is already nonzero. In the main text we therefore treat P_{init} as an empirical correction; here we explain why such a floor is expected and why it depends only weakly on model scale.

At the first update after a zero-initialized second moment, the higher-precision proposal is

$$\tilde{v}_1 = \alpha g_1^2, \qquad \alpha := 1 - \beta_2.$$

A coordinate remains at its initialized stored value in two main ways: (i) the corresponding gradient entry is exactly zero, and (ii) the proposal is nonzero in higher precision but is mapped back to the initialized quantized value by low-precision storage. Let

$$p_{\text{zero}} := \Pr(g_{1,i} = 0)$$

denote the fraction of coordinates with exactly zero gradient at the first step. This contribution is architecture- and data-dependent, but largely format-independent.

The second contribution depends on the quantization scheme. Consider a scaling group of size B (the full tensor under per-tensor scaling, or one block under block-wise scaling). If the maximum proposal magnitude in the group is mapped to the format maximum $x_{\max}$, then coordinate i is represented by

$$x_i^{\text{work}} = \frac{g_i^2}{\max_{j \le B} g_j^2}\, x_{\max}.$$

Under nearest rounding, a nonzero coordinate is mapped back to the initialized stored value whenever

$$\frac{g_i^2}{\max_{j \le B} g_j^2} < \tau, \qquad \tau := \frac{s_{\min}}{2x_{\max}}, \tag{35}$$

where $s_{\min}$ is the smallest positive representable spacing of the format. For stochastic rounding, the hard threshold is replaced by the corresponding soft gate, so the probability of returning to the initialized value is smaller but controlled by the same scale ratio.

To obtain a simple approximation, assume that within one scaling group

$$\frac{g_i^2}{\sigma^2} \overset{d}{\approx} \chi_1^2.$$

If the group maximum is replaced by a typical upper order statistic

$$M_B := F_{\chi_1^2}^{-1}\left(1 - \frac{1}{B}\right),$$

then the fraction of nonzero coordinates crushed back to the initialized value under nearest rounding is approximately

$$f_{\text{crush}}^{\text{NR}}(\tau, B) \approx F_{\chi_1^2}\big(\tau M_B\big). \tag{36}$$

This is a heuristic approximation: it replaces the random group maximum by a typical high quantile. For stochastic rounding, the analogous crush fraction is obtained by averaging the soft gate over the same distribution, and is therefore no larger than the NR value.

Combining exact zeros with quantization-induced crushing gives

$$P_{\text{init}} \approx p_{\text{zero}} + (1 - p_{\text{zero}})\, f_{\text{crush}}(\tau, B). \tag{37}$$

This explains two qualitative features seen in practice. First, BF16 typically has little additional crushing beyond structural zeros, whereas narrower-range formats can exhibit a much larger initial floor. Second, the dependence on model size is weak: under per-tensor scaling it enters only through the typical order statistic M_B, which grows only logarithmically in B, while under fixed block-wise scaling it is governed primarily by the block size itself.

Because subsequent stalling slows the growth of the EMA, the ideal predictor $j^\star_{\text{ideal}}$ can still rise somewhat too quickly after initialization. Rather than introducing a fully self-consistent transient model, in the main text we account for the dominant effect through the phenomenological correction

$$P_{\text{total}}(j) = P_{\text{init}} + (1 - P_{\text{init}})\, P_{\text{stall}}(j),$$

which preserves the closed-form startup-window expression while incorporating the empirical post-initialization floor.

A.5 Deriving the startup window length $j^\star(P_0)$

We derive the transient stalling law under an *ideal-trajectory approximation*. After initialization (or reset) to zero, the unquantized EMA evolves as

$$v_j = \beta_2 v_{j-1} + \alpha g_j^2, \qquad v_0 = 0, \qquad \alpha := 1 - \beta_2.$$

Over the short window relevant to the startup analysis, we assume that a single gradient coordinate has an approximately fixed local scale σ, and use

$$\frac{g_j^2}{\sigma^2} \overset{d}{\approx} \chi_1^2.$$

In particular, this implies

$$\mathbb{E}[g_j^2] \approx \sigma^2,$$

which we use only to motivate a deterministic reference trajectory. By linearity of expectation,

$$\mathbb{E}[v_j] = \alpha \sum_{k=1}^{j} \beta_2^{j-k}\, \mathbb{E}[g_k^2] \approx \alpha\sigma^2 \sum_{k=1}^{j} \beta_2^{j-k} = \sigma^2(1-\beta_2^j).$$

This motivates the reference path

$$\bar{v}_j := \sigma^2\phi_j, \qquad \phi_j := 1-\beta_2^j. \tag{38}$$

We index j as the number of post-initialization updates already accumulated in the EMA. Thus the *next* squared-gradient update is tested against the current stored state $\bar{v}_j$. Define

$$z := \frac{g^2}{\sigma^2} \sim \chi_1^2, \qquad W := \frac{g^2}{\bar{v}_j} = \frac{z}{\phi_j}.$$

Under nearest rounding (NR), stalling occurs when the innovation is smaller than half a ulp of the current state, i.e.

$$|W-1| < \hat{\rho}.$$

Equivalently,

$$-\hat{\rho} < \frac{z}{\phi_j} - 1 < \hat{\rho} \iff \phi_j(1-\hat{\rho}) < z < \phi_j(1+\hat{\rho}).$$

Therefore

$$P_{\text{stall}}^{\text{NR}}(j) = F_{\chi_1^2}\big(\phi_j(1+\hat{\rho})\big) - F_{\chi_1^2}\big(\max\{0, \phi_j(1-\hat{\rho})\}\big). \tag{39}$$

For the formats of interest in the main text, $\hat{\rho} \geq 1$, so the lower endpoint is nonpositive and

$$P_{\text{stall}}^{\text{NR}}(j) = F_{\chi_1^2}\big((1+\hat{\rho})\phi_j\big). \tag{40}$$

Conservativeness of the ideal-trajectory approximation. More generally, if the current state scale is $v = \sigma^2\phi$, then the same derivation gives

$$P_{\text{stall}}^{\text{NR}}(\phi) = F_{\chi_1^2}\big((1+\hat{\rho})\phi\big), \qquad (\hat{\rho} \geq 1),$$

which is monotone increasing in ϕ. Therefore, whenever the quantized EMA lags behind the ideal reference trajectory, its instantaneous stalling probability is smaller than the ideal-trajectory prediction at the same step. This does not yield a pathwise bound for the full quantized process, since nearest rounding can also perturb successful updates upward, but it explains why the ideal-trajectory predictor is typically conservative early in the startup phase.

Why the naive substitution is incorrect. A tempting shortcut is to replace $\hat{\rho}$ by the scaled quantity $\hat{\rho}_j = \hat{\rho}\,\phi_j$ inside the steady-state formula. This gives a stalling interval centered at $z = 1$ for every j, namely

$$1 - \hat{\rho}\phi_j < z < 1 + \hat{\rho}\phi_j,$$

which is correct only at steady state ($\phi_j = 1$). During the transient, the relevant ratio is $W = z/\phi_j$, so the correct interval is instead centered at the *current state scale* $z = \phi_j$, as in (39). This distinction matters especially for small ϕ_j, because the χ_1^2 density is largest near zero.

Closed-form startup window. Fix a tolerance $P_0 \in (0,1)$ and define

$$j^\star(P_0) := \min\{j : \; P_{\text{stall}}^{\text{NR}}(j) \geq P_0\}.$$

When $\hat{\rho} \geq 1$, using (40),

$$F_{\chi_1^2}\big((1+\hat{\rho})\phi_j\big) \geq P_0 \iff (1+\hat{\rho})\phi_j \geq F_{\chi_1^2}^{-1}(P_0).$$

Hence

$$\phi_j \geq \phi^\star(P_0), \; \phi^\star(P_0) := \frac{F_{\chi_1^2}^{-1}(P_0)}{1+\hat{\rho}}.$$

Since $\phi_j = 1-\beta_2^j$ is monotone increasing in j, the smallest such j is

$$j^\star(P_0) = \left\lceil \frac{\log\big(1-\phi^\star(P_0)\big)}{\log \beta_2} \right\rceil, \; \phi^\star(P_0) = \frac{F_{\chi_1^2}^{-1}(P_0)}{1+\hat{\rho}}, \tag{41}$$

defined whenever $\phi^\star(P_0) < 1$, i.e.

$$F_{\chi_1^2}^{-1}(P_0) < 1 + \hat{\rho}.$$

Including the empirical initial floor. The closed-form expression above captures only the precision-induced transient. In practice, the measured stalled fraction also contains an empirical floor P_{init}, due to effects not modeled by the idealized scalar EMA analysis, including structural zeros in the gradient and, for narrow-range formats, dynamic-range crushing under scaled quantization. We therefore use the phenomenological mixture model

$$P_{\text{total}}(j) = P_{\text{init}} + (1-P_{\text{init}})\, P_{\text{stall}}^{\text{NR}}(j). \tag{42}$$

For a target P_0, this is equivalent to replacing P_0 by the effective target

$$P_0^{\text{eff}} = \frac{P_0 - P_{\text{init}}}{1-P_{\text{init}}}, \tag{43}$$

with the convention that $j^\star(P_0) = 0$ whenever $P_0 \le P_{\text{init}}$. Thus the startup window reported in the main text should be interpreted as a theory-plus-floor predictor rather than a fully first-principles quantity:

$$j^\star(P_0) = j^\star_{\text{ideal}}\big(P_0^{\text{eff}}\big).$$

Table 8: Startup window under nearest rounding for $\beta_2 = 0.999$ after incorporating the empirical floor P_{init}. Theory uses (16)–(18); empirical values are measured in LLM pretraining. "**0**" means the target is already met immediately because $P_0 \le P_{\text{init}}$. "–" means the threshold was not reached empirically within the observation window.

	BF16	FP8	FP4
P_{init}	0.17	0.53	0.97
$j^\star(P_0{=}0.5)$	76 / **50**	**0 / 0**	**0 / 0**
$j^\star(P_0{=}0.8)$	464 / **360**	15 / **15**	**0 / 0**
$j^\star(P_0{=}0.9)$	1051 / **700**	36 / **43**	**0 / 0**
$j^\star(P_0{=}0.95)$	3044 / –	61 / **79**	**0 / 0**

Startup-window values. Table 8 gives the corresponding numerical startup-window values. The main trend is consistent with Figure 2: BF16 retains the longest responsive window, FP8 remains responsive only over a much shorter range, and in FP4 the *total* stalled fraction already exceeds practical thresholds immediately after reset once the empirical floor P_{init} is incorporated.

A.6 Empirical observations of stalling in the first moment

Figure 5 shows that the first moment exhibits the same basic qualitative pattern as the second: the stalled fraction is already nonzero after initialization and then quickly approaches a steady regime. What differs is the severity. In BF16, the steady-state stalled fraction is very small, confirming that first-moment stalling is largely negligible in this regime. Moreover, the initial floor P_{init} is slightly higher than the steady-state level, which explains the mild non-monotonic behavior and suggests that repeatedly resetting the first moment is not especially useful. In FP8, the steady-state stalled fraction is still relatively low, and is close to the initial floor; accordingly, resetting the first moment has little effect, either positive or negative. In FP4, by contrast, the steady-state stalled fraction becomes substantially larger and clearly harmful, while remaining above P_{init}, so resetting can recover part of the performance lost to persistent first-moment stalling. Finally, as in the second moment, stochastic rounding lowers the steady-state stalled fraction, but its practical importance depends strongly on the underlying precision regime.

A.7 Derivation of the reset-period heuristic

This section derives the theory-guided heuristic used in Section 5.2. The goal is not to identify a uniquely optimal reset period, but rather to preselect a useful operating region: long enough to let the EMA accumulate statistical information, but not so long that the *reset-sensitive* part of stalling dominates a substantial portion of the cycle. Compared to a purely endpoint-based rule, the heuristic below incorporates two empirical facts: (i) moderate second-moment staleness is often harmless, and (ii) good reset periods typically form a broad window rather than a sharp optimum.

Step 1: normalize the controllable stalling buildup. After a reset, the total measured stalled fraction at step j is modeled as

$$P_{\text{total}}(j) = P_{\text{init}} + (1 - P_{\text{init}})\, P_{\text{stall}}(j),$$

where P_{init} is the baseline floor and $P_{\text{stall}}(j)$ is the precision-induced stalling on the responsive subset. Since P_{init} is already present immediately after reset and is not removed by resetting, it should not determine the reset period. We therefore define the normalized stalling progress

$$S(j) := \frac{P_{\text{total}}(j) - P_{\text{init}}}{P^{\text{ss}}_{\text{total}} - P_{\text{init}}}. \tag{44}$$

Using

$$P^{\text{ss}}_{\text{total}} = P_{\text{init}} + (1 - P_{\text{init}})\, P^{\text{ss}}_{\text{stall}},$$

this simplifies to

$$S(j) = \frac{P_{\text{stall}}(j)}{P^{\text{ss}}_{\text{stall}}}. \tag{45}$$

Under the transient model, $S(j)$ starts at 0 and approaches 1 as $j \to \infty$. The key point is that this normalization removes the reset-insensitive floor P_{init} and isolates the part of stalling that accumulates within a cycle and can be mitigated by resetting.

Step 2: measure cycle-averaged excess staleness. A rule based only on the endpoint $S(K)$ is often too conservative: it penalizes a cycle as soon as the *last* step becomes highly stalled, even if much of

Figure 5: Measured stalling fraction for the first moment during pretraining of the 100M-parameter LLaMA model on C4.

the cycle remains in a regime where training is still robust. To reflect cumulative exposure to harmful staleness, we average over the cycle and count only the excess above a tolerance level $s_0 \in [0, 1)$:

$$\bar{S}_{s_0}(K) := \frac{1}{K} \sum_{j=1}^{K} \left[\frac{S(j) - s_0}{1 - s_0} \right]_+, \tag{46}$$

$$[x]_+ := \max(x, 0). \tag{47}$$

This quantity is 0 when the entire cycle stays below the tolerated staleness level s_0, and approaches 1 only when a substantial fraction of the cycle is spent near fully saturated controllable stalling. Thus s_0 encodes the empirically observed tolerance to moderate staleness: larger s_0 delays the point at which staleness begins to count against a longer cycle.

Step 3: quantify the remaining statistical error of the EMA. Consider the bias-corrected EMA after K steps, where the local age of the EMA is reset at the reset boundary together with the moment state itself. Its normalized weights are

$$w_t^{(K)} = \frac{(1 - \beta_2)\beta_2^{K-t}}{1 - \beta_2^K}, \qquad t = 1, \ldots, K.$$

The corresponding effective sample size is

$$N_{\text{stat}}(K) = \frac{1}{\sum_{t=1}^{K} (w_t^{(K)})^2} = \frac{(1 + \beta_2)(1 - \beta_2^K)}{(1 - \beta_2)(1 + \beta_2^K)}. \tag{48}$$

Its infinite-horizon limit is

$$N_{\text{stat}}^{\infty} = \frac{1 + \beta_2}{1 - \beta_2}.$$

We define the normalized remaining statistical error as

$$E(K) := 1 - \frac{N_{\text{stat}}(K)}{N_{\text{stat}}^{\infty}} = \frac{2\beta_2^K}{1 + \beta_2^K}. \tag{49}$$

This quantity decreases monotonically from 1 to 0 as the EMA accumulates samples, and measures how much averaging benefit is still left to gain by extending the current cycle.

Step 4: define the crossing heuristic. We choose the reset period as the first cycle length for which the cycle-averaged excess staleness matches the remaining statistical error:

$$K^\star := \min\{K \geq 1 : \bar{S}_{s_0}(K) \geq E(K)\}. \tag{50}$$

This yields a direct tradeoff. Before the crossing, the EMA is still gaining substantial statistical accuracy relative to the harmful staleness accumulated within the cycle. After the crossing, the additional gain from a longer cycle is increasingly outweighed by the time spent in a highly stalled regime.

Step 5: substitute the transient stalling law. For the precision regimes considered in the main text, $\hat{\rho} \geq 1$, so the transient NR approximation from Section A.5 gives

$$P_{\text{stall}}(j) \approx F_{\chi_1^2}\big((1+\hat{\rho})(1-\beta_2^j)\big), \; P_{\text{stall}}^{\text{ss}} \approx F_{\chi_1^2}(1+\hat{\rho}).$$

Substituting into (45) yields

$$S(j) \approx \frac{F_{\chi_1^2}\big((1 + \hat{\rho})(1 - \beta_2^j)\big)}{F_{\chi_1^2}(1 + \hat{\rho})}. \tag{51}$$

Therefore the cycle-averaged excess staleness becomes

$$\bar{S}_{s_0}(K) \approx \frac{1}{K} \sum_{j=1}^{K} \left[\frac{\dfrac{F_{\chi_1^2}\big((1 + \hat{\rho})(1 - \beta_2^j)\big)}{F_{\chi_1^2}(1 + \hat{\rho})} - s_0}{1 - s_0} \right]_+. \tag{52}$$

Combining (52) with (49), the reset period is the smallest integer K such that

$$\frac{1}{K} \sum_{j=1}^{K} \left[\frac{\dfrac{F_{\chi_1^2}\big((1 + \hat{\rho})(1 - \beta_2^j)\big)}{F_{\chi_1^2}(1 + \hat{\rho})} - s_0}{1 - s_0} \right]_+ \geq \frac{2\beta_2^K}{1 + \beta_2^K}. \tag{53}$$

The left-hand side is nondecreasing in K, since it is the average of a nondecreasing sequence, while the right-hand side is strictly decreasing and tends to 0. Moreover, because $S(j) \to 1$, the left-hand side tends to 1. Hence the crossing in (53) exists and is unique.

Numerical values. For $\beta_2 = 0.999$ and the effective precision ratios used in the main text,

$$\hat{\rho}_{\mathrm{BF16}} = 2.71, \quad \hat{\rho}_{\mathrm{FP8}} = 43.3, \quad \hat{\rho}_{\mathrm{FP4}} = 86.6,$$

we solve (53) numerically. For the tolerance level used in the main text, $s_0 = 0.6$, this gives

$$K^{\star}_{\mathrm{BF16}} \approx 1116, \quad K^{\star}_{\mathrm{FP8}} \approx 320, \quad K^{\star}_{\mathrm{FP4}} \approx 224.$$

Table 9 also reports nearby values of s_0 to show that the heuristic is stable across a reasonable range of tolerance levels. Overall, the predicted reset periods lie in the same broad regime as the empirically robust windows observed in our experiments, namely roughly 500–1500 for BF16, 300–600 for FP8, and 50–250 for FP4.

Table 9: Reset periods from the cycle-averaged thresholded heuristic (53) under NR with $\beta_2 = 0.999$. We report several tolerance levels s_0 to show the sensitivity of the rule to the tolerated amount of reset-sensitive staleness.

		$K^{\star}$ for tolerance level s_0			
Format	$\hat{\rho}$	0.5	0.6	0.7	Empirical window
BF16	2.71	≈ 1004	≈ 1116	≈ 1262	500–1500
FP8	43.3	≈ 295	≈ 320	≈ 351	300–600
FP4	86.6	≈ 206	≈ 224	≈ 246	50–250

In the main text we use $s_0 = 0.6$, which places the predicted periods near the middle of the empirically robust range while remaining stable to moderate changes in the tolerance level.

A.8 Adaptive resetting policies

While periodic resets are simple and effective, one can also design *adaptive* reset rules by monitoring stalling online during training. This avoids committing to a fixed period in advance. Our adaptive rule follows the same principle as the period heuristic in Proposition 5.2, but evaluates it online rather than precomputing a reset interval.

Concretely, at each iteration we estimate the empirical stalled fraction $P_{\mathrm{stall}}(k)$ by comparing the newly quantized state with the previously stored state before writeback. We then form the same cycle-averaged excess staleness quantity used in the periodic heuristic, but replace the model-based transient predictor by the observed stalled fraction. In the adaptive version, we fix $P^{\mathrm{ss}}_{\mathrm{stall}} = 1$ and $s_0 = 0.6$, and trigger a reset once

$$\bar{S}_{s_0}(k) - E(k) > 0. \tag{54}$$

Intuitively, this condition detects when the observed buildup of stalling within the current cycle outweighs the remaining statistical benefit of continuing the EMA.

Thus, the adaptive policy implements the same reset principle as the periodic heuristic, but replaces a fixed reset interval with an online test based on the actual behavior of the quantized EMA states. In practice, periodic resets provide a simple default strategy, while adaptive resetting offers a lightweight extension that can respond to the realized stalling dynamics during training.

A.9 Experimental Setting

We introduce the LLaMA architecture and training setup used in our experiments. All models use a maximum sequence length of 256, a global token batch size of 131K, and BF16 model weights and activations. We apply gradient clipping with maximum norm 1.0, linear warmup over the first 10% of training steps, and cosine decay to 10% of the peak learning rate.

We train all models on C4 (Raffel et al., 2020) using AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and AdamW epsilon $\epsilon_{\mathrm{adam}} = 10^{-6}$. The peak learning rates used for each model are reported in Table 10 and are kept fixed across precision regimes. SOAP-specific parameters for the ablation include the preconditioner update frequency every 10 steps, maximum preconditioner dimension 10,000, and preconditioner EMA coefficient $\beta = 0.999$

We evaluate three optimizer-state precision settings: BF16, FP8 (E4M3 for both moments with per-tensor scaling), and FP4 (E2M1 for the first moment and unsigned E2M2 for the second moment with block size 128). Model weights and activations remain in BF16.

Because current hardware does not provide native support for all of the low-precision state formats and rounding modes considered here, we simulate quantized optimizer-state storage in software. Concretely, optimizer states are dequantized to high precision for the update, then quantized and stored in the target format after each step, using either nearest or stochastic rounding depending on the experiment.

Params	Hidden	Intermediate	Heads	Layers	Steps	Tokens	LR
60M	512	1376	8	8	10K	1.3B	0.001
100M	640	1708	10	12	20K	2.6B	0.001
350M	1024	2736	16	24	60K	7.8B	0.0004

Table 10: Model configurations used in our experiments. Data amounts are given in tokens.

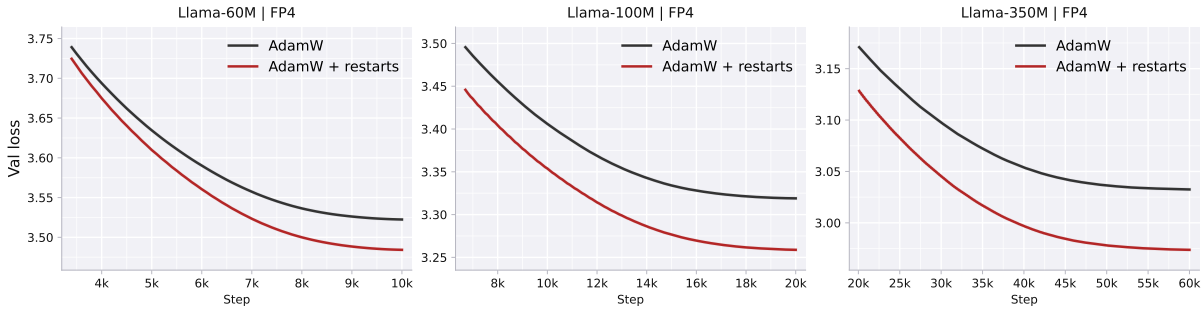

Figure 6: Training loss curves for the three model scales with FP4 optimizer-state storage.

When enabled, stochastic rounding is applied to all quantized states. Periodic EMA resets are applied to both moments unless stated otherwise. The reset period for the second moment is chosen using Equation (24), yielding periods of 1000 (BF16), 300 (FP8), and 200 (FP4). The first moment uses the same period unless otherwise specified.

A.10 Additional Results

We report additional results and plots omitted from the main text.

To better understand the role of resets relative to scaling, we also evaluate FP4 with per-tensor scaling instead of the block-wise scaling used in the main text. As expected, per-tensor scaling leads to substantially worse performance, since the effective quantization grid is much coarser. In this regime, resets should be viewed as complementary to scaling rather than a substitute for it: even with resets, the validation loss reaches only ≈ 3.8, compared to 3.45–3.55 with block-wise scaling.

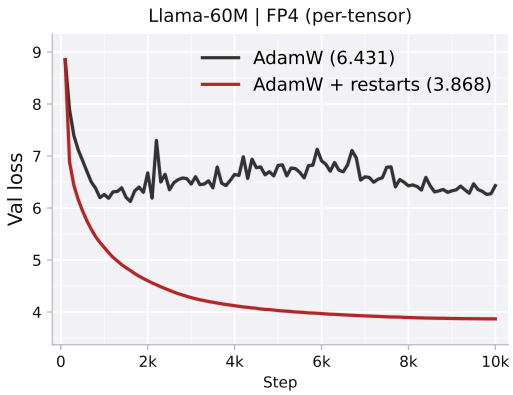

Figure 7: Training loss curves for LLaMA-60M with FP4 optimizer-state storage under per-tensor scaling.

However, resets remain critical in this setting. Without resets, training fails to converge, whereas with resets it reaches a reasonable loss. This further supports our interpretation that resets act by restoring responsiveness in highly stalled regimes, even when the underlying precision is too limited to match stronger scaling strategies.

A.11 Algorithms

Here s_t may denote a single EMA state or a tuple of optimizer states; when multiple states are present, Φ, Ψ, $\mathcal{Q}$, and $\mathcal{D}$ are understood componentwise.

Algorithm 1 Stateful Optimizer with Quantized State Storage and Periodic Resets

Require: Update maps Φ, Ψ; hyperparameters h; quantizer $\mathcal{Q}$ (NR or SR); dequantizer $\mathcal{D}$; reset period K; initial state s_{init} (often 0); (optional) resettable schedule state ξ_{init}.

Initialize θ_0; $\bar{s}_0 \leftarrow \mathcal{Q}(s_{\text{init}})$; $k \leftarrow 0$; (optional) $\xi \leftarrow \xi_{\text{init}}$
for $t = 1, 2, \ldots$ **do**
  $g_t \leftarrow \nabla f_t(\theta_{t-1})$
  $s_{t-1}^{\text{hp}} \leftarrow \mathcal{D}(\bar{s}_{t-1})$ ▷ load quantized state
  $k \leftarrow k + 1$ ▷ local step within the current reset cycle
  $s_t^{\text{hp}} \leftarrow \Phi\big(s_{t-1}^{\text{hp}}, g_t, \theta_{t-1}; h, k, \xi\big)$
  $\Delta\theta_t \leftarrow \Psi\big(s_t^{\text{hp}}, g_t, \theta_{t-1}; h, k, \xi\big)$
  $\theta_t \leftarrow \theta_{t-1} + \Delta\theta_t$ ▷ parameter update in higher precision
  $\bar{s}_t \leftarrow \mathcal{Q}\big(s_t^{\text{hp}}\big)$ ▷ requantize state for storage
  if $k = K$ **then**
    $\bar{s}_t \leftarrow \mathcal{Q}(s_{\text{init}})$ ▷ often $\bar{s}_t \leftarrow 0$
    $k \leftarrow 0$
    (optional) $\xi \leftarrow \xi_{\text{init}}$ ▷ reset bias correction / schedules
  end if
end for

Algorithm 2 Stateful Optimizer with Quantized State Storage and Adaptive Resets

Require: Update maps Φ, Ψ; hyperparameters h; quantizer $\mathcal{Q}$; dequantizer $\mathcal{D}$; decay β_2; reset tolerance s_0; steady-state stalled fraction $P_{\text{stall}}^{\text{ss}}$ (we use $P_{\text{stall}}^{\text{ss}} = 1$ in practice); initial state s_{init} (often 0); optional resettable schedule state ξ_{init}.

Initialize parameters θ_0
Initialize quantized state $\bar{s}_0 \leftarrow \mathcal{Q}(s_{\text{init}})$
Initialize cycle counter $k \leftarrow 0$
Initialize accumulated excess staleness $A \leftarrow 0$
(optional) initialize schedule state $\xi \leftarrow \xi_{\text{init}}$
for $t = 1, 2, \ldots$ **do**
  $g_t \leftarrow \nabla f_t(\theta_{t-1})$
  $s_{t-1}^{\text{hp}} \leftarrow \mathcal{D}(\bar{s}_{t-1})$
  $k \leftarrow k + 1$
  $s_t^{\text{hp}} \leftarrow \Phi(s_{t-1}^{\text{hp}}, g_t, \theta_{t-1}; h, k, \xi)$
  $\Delta\theta_t \leftarrow \Psi(s_t^{\text{hp}}, g_t, \theta_{t-1}; h, k, \xi)$
  $\theta_t \leftarrow \theta_{t-1} + \Delta\theta_t$
  $\bar{s}_t \leftarrow \mathcal{Q}(s_t^{\text{hp}})$
  Compute the empirical stalled fraction

$$P_{\text{stall}}(k) \leftarrow \frac{1}{d} \sum_{i=1}^{d} \mathbf{1}[(\bar{s}_t)_i = (\bar{s}_{t-1})_i]$$

  where d is the number of entries in the state tensor
  Compute normalized stalling progress

$$S(k) \leftarrow \frac{P_{\text{stall}}(k)}{P_{\text{stall}}^{\text{ss}}}$$

  Update accumulated excess staleness

$$A \leftarrow A + \left[\frac{S(k) - s_0}{1 - s_0}\right]_+$$

  Compute cycle-averaged excess staleness

$$\bar{S}_{s_0}(k) \leftarrow \frac{A}{k}$$

  Compute remaining statistical error

$$E(k) \leftarrow \frac{2\beta_2^k}{1 + \beta_2^k}$$

  if $\bar{S}_{s_0}(k) \geq E(k)$ **then**
    reset: $\bar{s}_t \leftarrow \mathcal{Q}(s_{\text{init}})$
    $k \leftarrow 0$
    $A \leftarrow 0$
    (optional) $\xi \leftarrow \xi_{\text{init}}$
  end if
end for