FORGETTING TO FORGET: HOW ROUNDING BIAS ALTERS MEMORY IN GATED DELTANET

Anonymous authors
Paper under double-blind review

ABSTRACT

Repeatedly rounding a language model's recurrent state can change what it remembers during decoding. Lower-precision caches save memory, but may discard small updates each time the state is stored. We prove that round-to-nearest (RTN) permanently halts fixed-gate pure decay below an exact format-dependent threshold. The measured active Gated DeltaNet heads do not freeze: writes and erasures keep them moving, while rounding error accumulates across the full update. Using Qwen3.5-0.8B as our main example, we connect this arithmetic to state error and next-token prediction. In teacher-forced evaluation on 72 documents, storing selected heads in fp32 removes 75.4% of the excess negative log-likelihood caused by bf16 round-to-nearest relative to an fp32-state control. In a separate pooled comparison, full-update stochastic rounding removes 91.4% of the RTN excess without enlarging the bf16 cache. Across five architectures, a post-hoc gate diagnostic flags at-risk bf16 heads; separate replays on fixed inputs show substantial state errors in selected small-decay heads. Free-running Qwen3.5-4B also shows changes in generation length under a fixed decoding protocol. The results motivate testing recurrent-state precision over long decoding trajectories, not only long-input prefill. They do not establish equal serving cost at equal cache size.

1 INTRODUCTION

A recurrent state can stop decaying even when its gate requests decay at every token. If each small decrement rounds back to the stored value, repeating the update does not recover the lost decay. This matters for hybrid language models: their Gated DeltaNet and Mamba-2 layers retain information in fixed-size recurrent states (Yang et al., 2025; Dao & Gu, 2024). Lower-precision caches reduce memory use, but long generation repeatedly rounds the same evolving state. The storage format can therefore change how the model retains information, not just how accurately it computes a single update.

Long generation exposes a different numerical problem from reading a long prompt. Decoding stores the recurrent state after every token, whereas chunked, parallel prefill combines updates before storing it. Long-input, short-answer tests such as the original RULER tasks (Hsieh et al., 2024) therefore do not directly test prolonged decode-time accumulation when prefill is chunked. Our focus is this repeated storage during decoding, not a claim that prefill is free of rounding error.

Prior work shows that changing cache rounding can change generated outputs. Nemotron 3 Super reports longer completions after casting its recurrent cache to fp16 and mitigates the effect with stochastic rounding (SR) (NVIDIA, 2026). These results show why cache precision matters, but not exactly when decay is lost or which heads drive prediction errors. For Gated DeltaNet, there is a further question: how does lost decay interact with the writes and erasures that keep an active head moving? We address these gaps by connecting rounding conditions to head-level state error and then to next-token prediction.

We use Qwen3.5-0.8B (Qwen Team, 2026) as the main example, with broader state measurements across architectures and a separate generation test in Qwen3.5-4B. Our contributions are:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

- **An exact limit on forgetting.** We prove that fixed-gate pure decay stalls permanently, above half its initial value, under round-to-nearest (RTN) when $0 < \delta < u/(1+2u)$, where δ is the requested fractional decay and u is the storage format's unit roundoff. This is a pure-decay result, not a claim that active heads freeze (Section 3).
- **A head-level diagnosis across architectures.** We measure gate statistics for 7,920 recurrent heads across five architectures. A post-hoc diagnostic flags 2.2%–16.1% of each model's heads as at risk under bf16 storage. Replays of selected small-decay streams then compare RTN and SR against fp64, separating measured state error from gate-based risk (Section 4).
- **Selective interventions on prediction quality.** On 72 documents of teacher-forced Qwen3.5-0.8B decoding, using bf16-RTN only for heads selected per document by a gate rule with decay cutoff u reproduces 68.7% of the full RTN excess negative log-likelihood relative to an fp32-state control. Keeping those heads in fp32 instead removes 75.4%. In a separate policy comparison, full-update SR removes 91.4% of the RTN excess without enlarging the bf16 cache (Section 5). These are measurements of existing rounding policies, not a new mitigation method.

The central idea is to track both the error a store introduces and how long the head retains it. We first isolate the arithmetic, then replay fixed inputs and split the rounding error into decay-induced and remaining components. This distinguishes literal pure-decay stalling from error accumulation in active heads. Teacher-forced interventions keep document tokens fixed while changing selected heads' storage; free-running generation then lets each policy produce its own tokens (Section 6). The latter tests whether storage changes generation under a fixed decoding protocol, without identifying decay stalling as its specific cause.

2 Recurrent states and rounding

The cache format determines what survives from one update to the next. We define the update, the rounding rules and the time over which an injected error can persist.

The recurrent update. A Gated DeltaNet head updates its memory by decaying, erasing and writing. Its state S_t follows

$$S_t = \alpha_t S_{t-1}(I - \beta_t k_t k_t^\top) + \beta_t v_t k_t^\top, \qquad \delta_t = 1 - \alpha_t. \tag{1}$$

Here k_t is a unit key ($\|k_t\|_2 = 1$), v_t a value, $\alpha_t \in (0, 1]$ the decay gate, and $\beta_t \in [0, 2]$ the write gate. The rank-one factor scales the old state's component along the key by $1 - \beta_t$. The state is $d_v \times d$, where d is the key dimension; in our Qwen3.5-0.8B example both dimensions are 128. Mamba-2 has a scalar decay per head and no corresponding rank-one erase in the configurations studied. Decode stores the state after every token; chunkwise prefill stores it at chunk boundaries. A prefill chunk composes many token updates before one store, so its per-store decrement can exceed the single-token decrement relevant to the stall threshold.

Rounding and reference computations. RTN and SR differ in how they discard information at each store. RTN returns the nearest representable value, with ties to even. SR chooses between the two bracketing values with probabilities that preserve the target in expectation (Gupta et al., 2015; Connolly et al., 2021). The significand carries a number's precision. Unit roundoff u is half its spacing: $u = 2^{-8}$ for bf16.[1] The two rules use the same cache size.

Rounding bias can accumulate faster than zero-mean error. Before forgetting dominates, aligned RTN errors can add up as $O(T)$; conditionally unbiased SR store errors instead have an $O(\sqrt{T})$ root-mean-square bound when per-step error scales are bounded and propagation does not amplify errors (Connolly et al., 2021). These describe absolute store-error accumulation on fixed inputs, not universal relative-error growth: contraction can cause saturation (Figure 1).

[1] bf16 (bfloat16) and fp16 are distinct 16-bit formats; fp32 and fp64 use 32 and 64 bits. Normal values exclude the subnormal range near zero. Gradual underflow retains these smaller values instead of flushing them to zero.

Each experiment holds different inputs fixed. Replays reuse gates, keys and values; teacher-forced tests reuse document tokens; free-running tests reuse prompts but allow generated tokens to differ. Unless noted otherwise, low-precision updates compute the recurrent algebra in fp32 and round only the stored state. The four non-Qwen replays instead compute updates in fp64 and cast through fp32 before low-precision storage. Recurrence replays compare each low-precision path with an fp64 reference of the same algebra on identical inputs. Model-level tests instead compare with an fp32 recurrent-state control in the otherwise low-precision model. Thus an fp32 control is not an entirely fp32 model.

We measure state error relative to the fp64 reference. Specifically, relative state error is $\|\widehat{S}_t - S_t^{\mathrm{ref}}\|_F / \|S_t^{\mathrm{ref}}\|_F$, where $\|\cdot\|_F$ is the Frobenius norm and S_t^{ref} the fp64 reference. A *head-stream* is one head's trajectory on one logged input sequence; multiple streams can measure the same head.

Error persistence. A rounding error injected at one token can survive many subsequent updates. The accumulated error depends both on what each store adds and on how quickly later updates remove it. To describe this persistence, compare reference and rounded replays on the same fixed sequence of gates, keys and values. Before the next rounding injection, their state difference is multiplied by $\alpha_t(I - \beta_t k_t k_t^\top)$: the identical additive writes cancel in the difference. Writes still affect the fresh rounding residue, but decay and rank-one erasure determine how an existing error propagates. Approximating key directions as equally likely (the isotropic-key approximation) and averaging this contraction gives

$$\tau_{\mathrm{err}} = \frac{1}{\bar{\delta} + \bar{e}}, \qquad \bar{e} = \overline{\beta_t(2 - \beta_t)/(2d)}. \tag{2}$$

The error horizon estimates how long an injected error remains in the state. Here τ_{err} is a first-order root-mean-square error-*amplitude* timescale in tokens, valid for small per-step decay and erase rates, not a squared-error timescale. Bars denote time averages. The erase contribution $\bar{e}$ is absent in the Mamba-2 heads. Unlike the gate-only horizon $1/\bar{\delta}$, this estimate includes GDN's erasure along the key. The derivation is given in Appendix A.2.

This horizon is an average approximation, not a deadline for model failure. Trained keys do not cover all directions equally: some error components are erased quickly, while others persist closer to the gate-only horizon $1/\bar{\delta}$.

3 When round-to-nearest stops decay

Repeated rounding can stop a decaying value from shrinking. We first isolate decay by removing writes and erasures; real heads add these operations back.

Theorem 1 (Permanent decay stall)**.** *In a binary format with at least two significand bits and unit roundoff u, let a positive normal stored value be multiplied exactly by $1 - \delta$ in real arithmetic and cast back with round-to-nearest at every step, with no intervening writes and gradual underflow. If*

$$0 < \delta < \delta_\star = \frac{u}{1 + 2u},$$

the iterate reaches a representable fixed point after finitely many steps, at a value exceeding half its initial value, and never changes again. A power-of-two start above the smallest normal may enter the lower binade before locking. The threshold is sharp under round-to-nearest-even: for $\delta_\star \le \delta < u$, no positive normal value above the smallest normal is fixed. Each step moves to its immediate predecessor until it locks at the smallest normal.

Why the threshold is exact. The spacing between stored values determines whether decay survives. If a decrement is smaller than half the gap to the lower neighbor, RTN returns the original value. The next step repeats the same decision. Within a binade, an interval between adjacent powers of two, the last guaranteed locking significand is $1 + 2u$, giving $\delta_\star = u/(1 + 2u)$. The finite descent and boundary cases are proved in Appendix A.1. The threshold is $1/258$ for bf16, $1/2050$ for fp16, $1/18$ for e4m3 and $1/10$ for e5m2. Replacing it by u is especially inaccurate for the float8 formats.

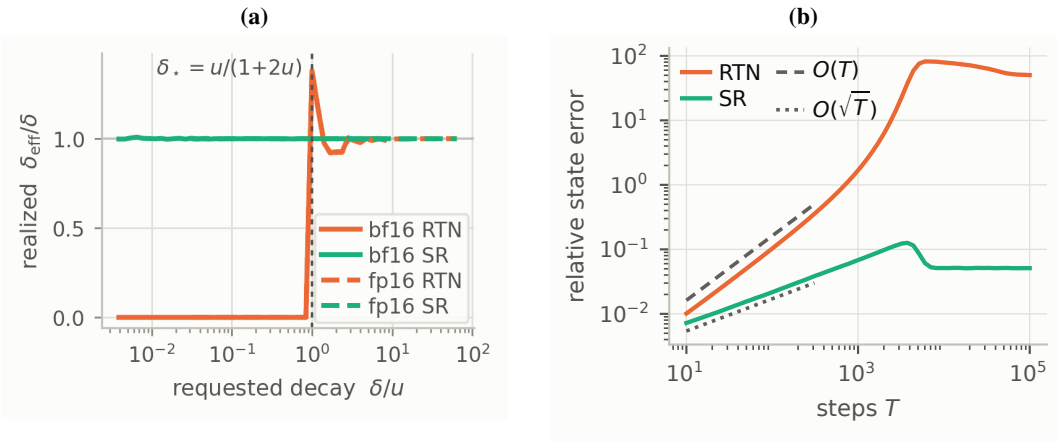

Figure 1: **Pure-decay threshold and synthetic error growth.** (a) Realized/requested decay against δ/u for bf16 (solid) and fp16 (dashed). Below the threshold, RTN's measured late-window effective decay (orange) is zero after transient descent; SR (aqua) tracks the nominal rate. The two normalized thresholds nearly coincide at this scale. (b) Relative state error of bf16 recurrent-state storage against fp64 in a synthetic weak-write regime: $d = 128$, $\delta = 10^{-3}$, $\beta = 0.005$, averaged over three synthetic trajectories. Grey $O(T)$ and $O(\sqrt{T})$ guides illustrate early linear and random-walk growth. SR eventually saturates in this regime, and the guides do not assert all-time scaling.

We test the threshold by comparing exact predictions with floating-point casts. The tests check individual decay steps and repeated-decay trajectories in all four formats. None of the 4,656 single-step tests or 116 trajectories failed (Appendix B.1).

From a stalled scalar to a rounded head. SR can preserve a decay that RTN repeatedly discards. It occasionally rounds to the lower neighbor, delivering the requested decrement on average. To compare the two rules, we first sweep the decay rate in a pure-decay recurrence (Figure 1a). We then replay a synthetic recurrence with weak writes against fp64, using bf16 storage and $\delta = 10^{-3} < \delta_\star = 1/258$ (Figure 1b). RTN's error grows far above SR's and stays there over the run, while SR's error saturates. The exact distributional forgetting result for ideal SR applies to erase-free recurrences with fixed external inputs (Appendix A.3); its monotone-coupling proof does not extend to the GDN rank-one erase.

Real heads keep changing even when most requested decays are small. Writes and erasures move entries across rounding boundaries, so decay can take effect through occasional jumps between stored values. To measure this, we compare each rounded update with the same update with decay switched off. This splits the store error into a decay-off term and the additional error caused by decay; both terms are then propagated through the recurrence (Appendix A.4).

The measured heads do not behave like frozen pure-decay states. On 16 at-risk Qwen3.5-0.8B head-streams from held-out layers, representing nine heads on two logged input sequences, median realized/requested decay is 96%. This measure sums projections onto the requested decay across the stream, so losses and overshoots can cancel. It does not mean that each decrement is accurate: rounding residue accumulates across decay, erase and write together.

4 Prevalence and State Error Across Architectures

We next test whether the small decays studied above occur in trained models. We first measure gate statistics across architectures, then replay selected heads to see whether rounding causes state error.

Selecting heads from their gates. We use two gate statistics to identify heads exposed to persistent rounding error. Define the stall fraction f_{stall} as the fraction of requested per-step decays below $\delta_\star$. We call a head *at risk* when

$$f_{\text{stall}} > 0.99 \quad \text{and} \quad \tau_{\text{err}} > 10^3. \tag{3}$$

Table 1: **Decode-time risk across five architectures.** Heads meeting Equation 3, from roughly one million logged tokens per model.

Model	Recurrence	Heads	bf16 at risk	fp16 at risk
Qwen3.5-0.8B	GDN	288	46 (16.0%)	9 (3.1%)
Falcon-H1-1.5B	Mamba-2	1,152	185 (16.1%)	6 (0.5%)
Granite-4.0-H-Micro	Mamba-2	2,304	287 (12.5%)	189 (8.2%)
Olmo-Hybrid-7B	GDN	720	16 (2.2%)	4 (0.6%)
Nemotron-Nano-9B-v2	Mamba-2	3,456	461 (13.3%)	83 (2.4%)

This rule describes exposure to small decays, not certain damage. We chose the two-condition rule after examining Qwen3.5-0.8B replay. The census uses $\delta_\star$; the original Qwen replay and teacher-forced head selections used the earlier cutoff u, as disclosed below. On the original replay, stall fraction alone selects the same streams. A later exploratory check on held-out data found no benefit from adding the horizon cutoff, which excluded some high-error heads. We therefore do not claim that both conditions independently improve prediction (Appendix B.2).

Gate measurements across models. We compare two recurrent model families to test how widely the diagnostic applies. The census covers GDN models Qwen3.5-0.8B (Qwen Team, 2026) and Olmo-Hybrid-7B (Allen Institute for AI, 2026), and Mamba-2 hybrids Falcon-H1-1.5B-Base (Technology Innovation Institute, 2025), Granite-4.0-H-Micro (Granite Team, IBM, 2025) and Nemotron-Nano-9B-v2 (NVIDIA Corporation, 2025). We log gates using a shared three-domain recipe: PG-19 books (Rae et al., 2020), GovReport government reports (Huang et al., 2021) and code assembled from installed Transformers Python files. Text is retokenized and length-filtered per model; source splits and assembly details are given in Appendix B.2.

All five architectures contain at-risk bf16 heads. Their fractions range from 2.2% to 16.1% across 7,920 recurrent heads (Table 1). The horizon cutoff excludes some stalled heads in both recurrence families; the separate counts are given in Appendix B.2. A further comparison within Qwen3.5 finds at-risk heads from 0.8B to 9B parameters (Table 3). These measurements establish prevalence, not why architectures differ.

Storage frequency matters. Combining updates before storing the state reduces exposure to the stall threshold. To examine the difference between decode and prefill, we estimate the combined decay over different store periods in Qwen3.5-0.8B. Using its original two-domain gate logs, we sample decrements from each head's distribution and multiply the corresponding retention factors. The estimate gives 49 at-risk bf16 heads for every-token storage, four for 64-token periods and none for 256-token periods. Sampling ignores the order of the gates; a comparison that preserves their order on fully logged layers is reported in Appendix B.2. This test measures the decay-stall diagnostic, not overall prefill accuracy.

Replay measures the consequence. Replaying fixed inputs lets us compare rounding policies without changing the logged stream. For each model, we run RTN and SR on the same logged gates, keys and values and compare both with fp64 (Figure 2). The plotted heads satisfy the stall-fraction condition alone: some fail the horizon cutoff, especially in Olmo. At each model's last plotted checkpoint, median RTN state error exceeds SR error.

The longer Qwen replay also shows large errors concentrated in a subset of streams. It covers 128 head-streams from 64 distinct heads in four layers on two input sequences. At 131,072 tokens, the 90th percentile of relative state error is 0.526 under RTN and 0.095 under SR. The diagnostic selects 33 of the 37 streams exceeding 0.1 RTN error, measured on the same replay used to choose the rule.

5 Likelihood damage and mitigation

State error matters for a language model only if it changes its outputs. We test this with teacher-forced decoding: each cache policy receives the same document tokens, but evolves its own hidden states and predicts the next token. We then change storage for selected heads to test which ones contribute to the likelihood loss.

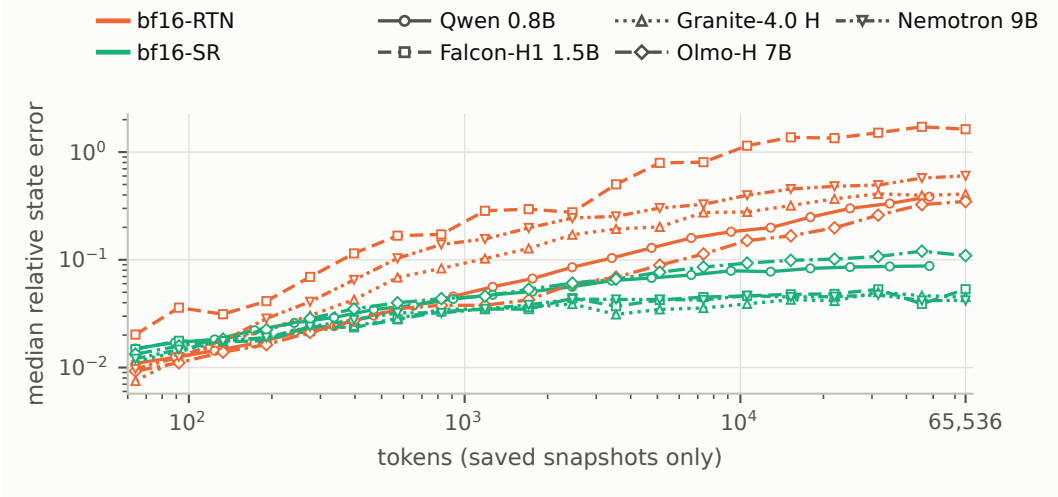

Figure 2: **State-error trajectories across all five architectures.** Median relative state error against fp64 among replayed streams with $f_{\text{stall}} > 0.99$, under bf16-RTN (orange) and bf16-SR (aqua). Dash patterns and markers identify models. Qwen contributes 33 of 128 logged head-streams; Falcon, Granite, Olmo and Nemotron contribute 5, 24, 20 and 27 selected streams, respectively. Qwen ends at its last saved checkpoint below the cap, 48,484 tokens; others end at 65,536. Qwen uses raw-u stall statistics over the full logged stream, while other models use $\delta_\star$ over their replay window. Groups follow measured stall fraction, regardless of initial selection as a stalled or control head. The layer, head and input-sequence sampling frames are given in Appendix B.2; these are not random samples of each architecture's at-risk population. Because the sampled heads differ, the curves cannot rank architectures. Qwen uses fp32 update arithmetic; the other replays use fp64 updates followed by an fp32 cast before bf16 storage. They measure open-loop state error, not output accuracy.

Likelihood degrades at equal cache size. RTN worsens next-token predictions more than SR in this comparison. We decode Qwen3.5-0.8B on 72 documents spanning books, government text and code, scoring 32,768 positions per document. We measure negative log-likelihood (NLL), where lower values mean better predictions, over each document's final quarter. The mean increase over the fp32-state control is 0.02279 nats per token under bf16-RTN, compared with 0.00195 under bf16-SR. Books and government text show larger effects than code (Figure 3b).

Harder documents can affect both gate statistics and prediction quality. A correlation between them therefore does not tell us which heads cause the loss. To test that directly, we change selected heads' cache precision while keeping the document fixed. The domain comparison, pooled policy comparison and head interventions are separate runs; their means, confidence intervals and comparison baselines are given in Appendix B.3.

Selected heads carry most, but not all, of the damage. Changing the selected heads' precision changes much of the likelihood loss. We select heads per document using the legacy rule: a decay below u on more than 99% of tokens and $\tau_{\text{err}} > 10^3$. This uses u, not $\delta_\star$; equivalence of the two selections has not been checked on these documents. We then run two interventions. "Corruption" stores only those heads with bf16-RTN and leaves the others in fp32; "protection" keeps those heads in fp32 and uses bf16-RTN elsewhere. Selected-only corruption reproduces 68.7% of the full RTN NLL increase, while protection removes 75.4% (Figure 3c).

Randomly selected heads produce a much smaller effect. Corrupting random sets with the same head count in each layer reproduces 3.05% of the full increase. These controls do not match the selected heads' error horizons, so the result cannot distinguish stalling from other properties of long-lived states. Corruption and protection are separate tests, not additive shares of a single damage total.

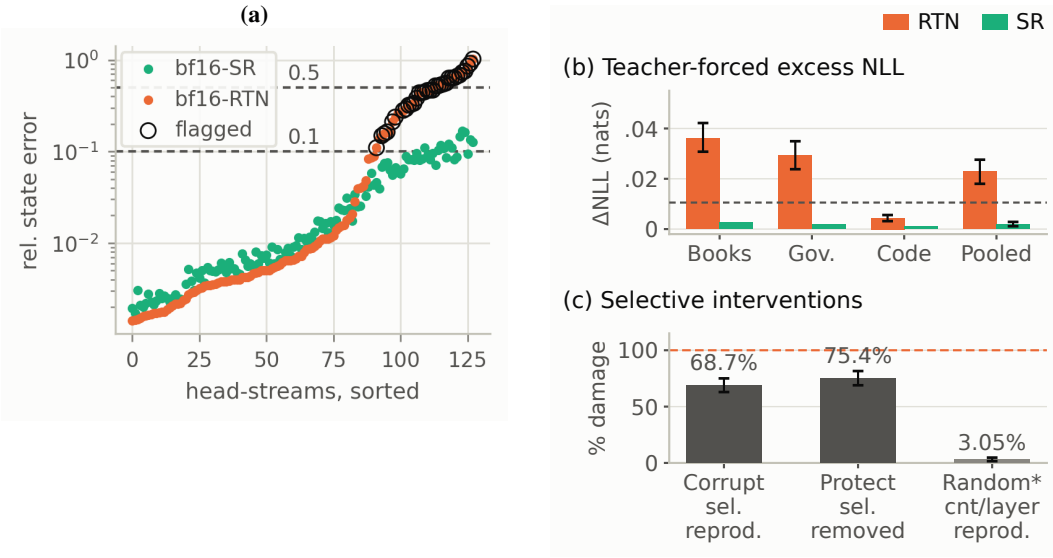

Figure 3: **Open-loop replay and teacher-forced likelihood in Qwen3.5-0.8B.** (a) Relative state error against fp64 at 131,072 tokens for 128 head-streams from 64 distinct heads, sorted by RTN error; selected streams are ringed. RTN has 37 streams above 0.1 error and 15 above 0.5. Repeated measurements of the same head are not independent head samples. (b) Final-quarter excess NLL (nats/token) on 72 documents relative to an fp32 recurrent-state control, with the prespecified damage bar dashed; domain SR bars have no saved intervals. "Pooled" is a separate policy-comparison run, not an aggregate of the adjacent domain bars. (c) Full RTN damage in the intervention run *reproduced* by corrupting selected/random heads or *removed* by protecting selected heads; dashed is 100%. Whiskers are 99% confidence intervals from resampling whole documents, keeping policy comparisons paired. Random* selections match head count and layer, not error horizon. Sampling details and numerical intervals are in Appendix B.3.

Full-update and decay-only mitigation. Applying SR to every recurrent head removes 91.4% of the pooled RTN NLL increase without increasing cache size. This policy changes rounding of the complete update, not just decay. Selective fp32 protection leaves errors in the other heads, so targeting the largest state errors need not give the best output predictions. We also compare error feedback, an existing technique that saves discarded rounding error in a small residual alongside the bf16 value. With an 8-bit or 4-bit residual (3.0 or 2.5 bytes per state entry, against 2.0 for bf16), the 99% interval for its pooled excess NLL includes zero (Table 5).

Correcting only decay tests whether decay error alone explains the damage. We apply a stochastic correction to the decay term while leaving the RTN-rounded erase and write unchanged. On the 16 at-risk streams in Section 3, its median state-error ratio to RTN is 0.77. A 99% bootstrap interval obtained by resampling the nine distinct heads includes no reduction as well as substantial benefit; it remains conditional on the two logged input sequences. By comparison, full-update SR removes about 80% of the state error on these streams. The evidence therefore does not establish decay-only repair, but does not rule it out. The intervals, head-level dependence and readout results are reported in Appendix B.3.

6 Free-running generation

We finally test whether cache precision changes complete generated answers. Unlike teacher forcing, this experiment lets each policy produce its own tokens, so a small difference can change all later inputs.

Generation lengths and completed answers. We compare cache policies on the same set of mathematics problems. Qwen3.5-4B generates solutions to all 30 AIME 2026 problems, with only GDN recurrent-state storage varied between policies. Decoding is greedy with presence penalty

Table 2: **Free-running Qwen3.5-4B at a 32,768-token budget.** Arm names describe only the GDN recurrent state. All 30 AIME 2026 problems use greedy decoding with presence penalty 1.5. $\Delta \log L$ is the paired mean difference in capped log length from fp32, with standard error, not a difference in uncapped termination times. The selected, unmatched longer-cap follow-up is confined to Appendix B.4.

GDN recurrent-state policy	$\Delta \log L \pm$ s.e.	At cap	Correct
32,768-token cap; all 30 problems; batch size 8			
fp32 reference	0	10/30	20/30
bf16 RTN	$+0.288 \pm 0.056$	22/30	8/30
bf16 SR, seed 1	$+0.081 \pm 0.048$	15/30	15/30
bf16 SR, seed 2	$+0.020 \pm 0.045$	10/30	19/30
bf16 SR, seed 3	$+0.115 \pm 0.050$	16/30	13/30
fp32 + uniform noise	$+0.010 \pm 0.041$	9/30	20/30
bf16 + 8-bit residual	-0.032 ± 0.047	9/30	21/30

1.5, changed after calibration from the initially registered penalty-free decoder. Each answer has a 32,768-token budget. We record generation length, budget exhaustion and answer correctness (Table 2).

RTN reaches the token limit more often and answers fewer problems correctly than fp32 or SR in this test. Because 22 of 30 RTN traces reach the limit, the observed length $L = \min(L_{\text{stop}}, C)$ measures budget-limited generation, not the uncapped number of generated tokens L_{stop}, with $C = 32{,}768$. Both the paired log-length comparison and a test that accounts for unfinished traces find longer RTN outputs. Their uncertainty estimates are given in Appendix B.4.

A uniform-noise control asks whether random perturbations without a bf16 store can produce similar generation behavior. Its scale follows the local bf16 spacing. The variance matches an average over uniformly distributed positions within a rounding cell, not SR's conditional variance at each actual update (Appendix B.4).

Interpretation. Most completed answers are correct, so accuracy largely tracks termination in this experiment. Across all eight experimental arms, including the duplicate fp32 control, 136 of 139 finished traces are correct, compared with none of 101 capped traces. Accuracy is therefore a useful diagnostic, not independent evidence for the length effect.

The experiment does not isolate why the generated answers change. We cannot distinguish the uniform-noise control statistically from either fp32 or SR; this does not establish equivalence to either. Across SR seeds, correct answers range from 13 to 19 and cap hits from 10 to 16, so recovery relative to fp32 also varies. The result establishes an effect of recurrent-state storage under this decoder, not decay stalling as its specific cause.

7 Related work

Prior work establishes both rounding-induced stagnation and the usefulness of SR. We place our head-level diagnosis alongside that work and distinguish it from other failures of recurrent memory.

Recurrent models and quantized inference. Gated DeltaNet combines the delta rule with gated forgetting (Yang et al., 2024; 2025); Mamba-2 provides a related recurrent state-space formulation (Dao & Gu, 2024). Quantized GDN inversion (Zhang et al., 2026) studies the chunkwise triangular solve, whereas we study repeated floating-point state storage. Stuffed Mamba identifies a link between under-forgetting, training length and state size, and also studies inference-time forgetting interventions (Chen et al., 2025). Our comparison instead changes cache rounding while keeping the weights and requested update rule fixed.

Stagnation and stochastic rounding. RTN stagnation in low-precision time stepping is an established numerical phenomenon (Croci & Giles, 2023). SR is likewise well established in numerical computation and low-precision training (Gupta et al., 2015; Connolly et al., 2021; Croci et al.,

2022). Nemotron reports coherent RTN cache-error accumulation and deploys SR for Mamba recurrent caches (NVIDIA, 2026). We do not claim either observation as new. Our contribution connects a format-exact pure-decay threshold to head-level measurements, selective likelihood interventions and a decay-off error decomposition for GDN. SR restricted to part of an update also has prior precedent (Zamirai et al., 2020); the decay-only store here is a diagnostic instrument, not a proposed mitigation.

Deterministic alternatives. SR is not the only way to reduce cache error. Our error-feedback arm retains discarded residue alongside the state. Classical compensated accumulation was examined in exploratory synthetic and fixed-input replays, but is not included in the model-level comparisons reported here. Dynamically scaled cache formats are another unevaluated model-level baseline. KIVI quantizes transformer attention keys per channel and values per token (Liu et al., 2024); its KV-cache results do not establish behavior for repeatedly updated GDN states. Comparing adapted deterministic methods with SR at matched memory and measured serving cost remains open.

8 Limitations and conclusion

The theorem covers an idealized recurrence that active heads do not follow literally. It assumes fixed-gate pure decay, exact intermediate multiplication and gradual underflow. Trained heads also write, change their gates and erase some directions more strongly than others. Likewise, the post-hoc selection rule describes susceptible heads but is not a universal damage predictor.

The output-level evidence is narrower than the cross-model state measurements. Selective likelihood tests use one model and do not separate stalling from long error horizons; other architectures have open-loop evidence only. Generation results use one model, a small benchmark and a specific decoder. Many traces hit the token cap, SR seeds differ, and benchmark contamination is possible. The presence penalty may interact with rounding-induced changes; we have not established the same result without it. Standardized long-context retrieval and associative-recall performance remain unmeasured.

Equal cache size does not imply equal serving cost. SR requires random number generation and extra operations; error feedback requires residual storage and arithmetic. We have not measured optimized-kernel latency, throughput or energy, and the reported model-level comparisons do not include classical compensated accumulation or dynamic grouped scaling. The observed SR reductions therefore do not establish a deployment-optimal policy.

Cache precision changes the recurrence that a model actually executes. An apparently stable RTN cache can discard intended updates; in real heads the accumulated bias involves more than decay alone. Full-update SR substantially reduces the measured damage without enlarging the recurrent cache. These results motivate evaluating recurrent-state storage over long decoding trajectories, not only checking a single update or chunked prefill.

AI use statement

Language-model assistants supported the research and writing. Claude Code and GitHub Copilot were used for mathematical formulation and proof checking, experimental design and implementation, numerical analysis and interpretation, literature and citation work, and manuscript drafting and editing. They also supplied internal review feedback. Recorded runs identify the tools and activities; selected claims were checked against saved artifacts, source code, and separate mathematical derivations. These assisted checks and simulated reviews are not external human peer review and do not establish that every claim has been independently verified. The authors take responsibility for the manuscript, including all AI-assisted material.

Reproducibility statement

The appendices provide proofs, rounding semantics, sampled populations, experimental controls, and statistical procedures. Recurrence replays use fp64 references of the same algebra; model-level

tests use fp32 recurrent-state controls. Saved summaries and implementation records support the reported comparisons.

REFERENCES

Allen Institute for AI. Model card for Olmo Hybrid (7B). Hugging Face model card, 2026. URL https://huggingface.co/allenai/Olmo-Hybrid-7B/blob/4f1cc566f9fdf3ce68da2ab6a788a83d89896dcf/README.md. Accessed September 24, 2026.

Yingfa Chen, Xinrong Zhang, Shengding Hu, Xu Han, Zhiyuan Liu, and Maosong Sun. Stuffed Mamba: Oversized states lead to the inability to forget. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2410.07145.

Michael P. Connolly, Nicholas J. Higham, and Theo Mary. Stochastic rounding and its probabilistic backward error analysis. *SIAM Journal on Scientific Computing*, 43(1):A566–A585, 2021. doi: 10.1137/20M1334796.

Matteo Croci and Michael B. Giles. Effects of round-to-nearest and stochastic rounding in the numerical solution of the heat equation in low precision. *IMA Journal of Numerical Analysis*, 43 (3):1358–1390, 2023. doi: 10.1093/imanum/drac012.

Matteo Croci, Massimiliano Fasi, Nicholas J. Higham, Theo Mary, and Mantas Mikaitis. Stochastic rounding: Implementation, error analysis and applications. *Royal Society Open Science*, 9(3):211631, 2022. doi: 10.1098/rsos.211631.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071. PMLR, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026. URL https://arxiv.org/abs/2605.00674.

Granite Team, IBM. Granite-4.0-H-Micro. Hugging Face model card, 2025. URL https://huggingface.co/ibm-granite/granite-4.0-h-micro/blob/d5f01a3ea75f088947be3aae039f4ad52837dfde/README.md. Accessed September 24, 2026.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1737–1746. PMLR, 2015. URL https://proceedings.mlr.press/v37/gupta15.html.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.06654.

Luyang Huang, Shuyang Cao, Nikolaus Parulian, Heng Ji, and Lu Wang. Efficient attentions for long document summarization. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1419–1436. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.naacl-main.112. URL https://aclanthology.org/2021.naacl-main.112/.

Benjamin Jourdain and Gilles Pagès. Quantization and martingale couplings. *ALEA, Latin American Journal of Probability and Mathematical Statistics*, 19(1):1–22, 2022. doi: 10.30757/ALEA.v19-01.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 32332–32344. PMLR, 2024. URL https://proceedings.mlr.press/v235/liu24bz.html.

NVIDIA. Nemotron 3 Super: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for agentic reasoning. *arXiv preprint arXiv:2604.12374*, 2026. URL https://arxiv.org/abs/2604.12374.

NVIDIA Corporation. NVIDIA-Nemotron-Nano-9B-v2. Hugging Face model card, 2025. URL https://huggingface.co/nvidia/NVIDIA-Nemotron-Nano-9B-v2/blob/6533e8de2c68e4536bf7c411d7a3ce5734111476/README.md. Accessed September 24, 2026.

Kaan Ozkara, Tao Yu, and Youngsuk Park. Stochastic rounding for LLM training: Theory and practice. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 4402–4410. PMLR, 2025. URL https://proceedings.mlr.press/v258/ozkara25b.html.

Qwen Team. Qwen3.5-0.8B. Hugging Face model card, 2026. URL https://huggingface.co/Qwen/Qwen3.5-0.8B/blob/2fc06364715b967f1860aea9cf38778875588b17/README.md. Accessed September 24, 2026.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SylKikSYDH.

Technology Innovation Institute. Falcon-H1-1.5B-Base. Hugging Face model card, 2025. URL https://huggingface.co/tiiuae/Falcon-H1-1.5B-Base/blob/06d7330266253c20784f58b0a846dc09a9d12cee/README.md. Accessed September 24, 2026.

Naigang Wang, Jungwook Choi, Daniel Brand, Chia-Yu Chen, and Kailash Gopalakrishnan. Training deep neural networks with 8-bit floating point numbers. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/335d3d1cd7ef05ec77714a215134914c-Abstract.html.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.06484.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated Delta Networks: Improving Mamba2 with delta rule. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2412.06464.

Pedram Zamirai, Jian Zhang, Christopher R. Aberger, and Christopher De Sa. Revisiting BFloat16 training. *arXiv preprint arXiv:2010.06192*, 2020. URL https://arxiv.org/abs/2010.06192.

Luoming Zhang, Yuwei Ren, Kui Zhang, Tian Liu, Lingjuan Ge, Denghao Li, Matthew Harper Langston, Yin Huang, Weiliang Will Zeng, and Liang Zhang. When good enough is optimal: Multiplication-only matrix inversion approximation for quantized Gated DeltaNet. *arXiv preprint arXiv:2606.06034*, 2026. URL https://arxiv.org/abs/2606.06034.

A PROOFS

The proofs separate exact rounding results from approximations used for trained heads. We prove the pure-decay stall threshold and a distributional forgetting result for ideal SR, then derive the approximate error horizon and the decay-error decomposition. We also give the mean and variance of the decay-only correction. The stall theorem excludes intervening writes; the distributional forgetting proof requires a transition that preserves the ordering of states.

A.1 EXACT PERMANENT-STALL THRESHOLD

Why the lattice descent terminates. Below the threshold, a decaying value cannot skip past every possible locking point. For an interior significand m, each step either locks ($\delta < u/m$) or moves down by one representable value. The sequence therefore reaches a locking point after finitely many steps: $\delta < \delta_\star$ puts u/δ above $1 + 2u$, the first significand past a power of two.

We prove Theorem 1 for positive normal values. Symmetry gives the negative case, and zero is already a fixed point. Rescale a binade (an interval $[2^e, 2^{e+1})$ of constant exponent) so that its representable significands are $\mathcal{M} = \{1, 1 + 2u, \ldots, 2 - 2u\}$; every interior value $m \in \mathcal{M} \setminus \{1\}$ then has lower neighbor $m - 2u$. Round-to-nearest returns m after a requested decay whenever

$$m(1 - \delta) > m - u \quad \Longleftrightarrow \quad \delta < \frac{u}{m}. \tag{4}$$

The largest local lock threshold over interior significands is attained at the first one, $m = 1 + 2u$: there it equals $\delta_\star = u/(1 + 2u)$.

Each non-locking step moves to the immediate lower neighbor. Suppose $0 < \delta < \delta_\star$. If an iterate has an interior significand $m < u/\delta$, Equation 4 fixes it. Because $\delta < \delta_\star < u$, the requested decrement δm is smaller than the full gap below m, so the cast returns either m itself or its immediate lower neighbor; the iterate therefore descends the lattice one representable value at a time and cannot step over an intermediate significand. An upward midpoint tie is another way to lock. Any step that does not lock moves to the immediate lower neighbor. The sequence consequently decreases through finitely many lattice points in its current binade. At a binade boundary $m = 1$ above the smallest normal, the predecessor gap is $u2^e$, so a sub-threshold step either locks or moves to the immediate predecessor in the lower binade. At $\delta = u/2$ this is a midpoint tie, which round-to-nearest-even resolves to the starting power of two. In the lower binade it again descends through finitely many lattice points until it reaches an interior $m < u/\delta$. That value exists because $u/\delta > 1 + 2u$. The iteration then locks by Equation 4. Repeating the same real multiplication and deterministic cast returns the identical lattice value on every later step, which proves permanence.

Midpoint ties do not prevent eventual locking below the threshold. An interior tie resolves either to a lock or to a single further step down, so the conclusion is unaffected under any round-to-nearest tie rule; the strict inequality is what makes the guaranteed lock point $m = 1 + 2u$ tie-free. Under round-to-nearest-even, midpoint ties depend on the parity of the candidate significands; equality at the threshold is treated in the sharpness argument below.

The behavior at the smallest normal requires gradual underflow. At that boundary the relative half-gap is u, so every $\delta < u$ fixes the smallest normal itself. Flush-to-zero arithmetic is outside the theorem; behavior at the smallest normal depends on the specified flushing convention. The theorem also excludes a nonzero write that changes the entry between stores.

The locked value remains above half the initial value. An interior start cannot descend past the tie-free locking value $m = 1 + 2u$ in its binade; it may lock earlier, including at a midpoint resolved upward. Thus $1 + 2u$ is a guaranteed latest possible lock, not the actual lock of every trajectory. For a starting significand $m_0 < 2$, the locked-to-initial ratio is at least $(1+2u)/m_0 > 1/2$. A power-of-two start either stays fixed or enters just the lower binade, where the ratio is at least $(1+2u)/2 > 1/2$. The smallest-normal case was handled separately above.

At and above the threshold, the interior locking points disappear. For $\delta \geq \delta_\star$ no interior significand satisfies $\delta < u/m$, and at $\delta = \delta_\star$ the candidate $m = 1+2u$ sits exactly on a midpoint, which round-to-nearest-even resolves downward. For $\delta_\star \leq \delta < u$, a power of two above the smallest normal also descends, since $\delta_\star > u/2$. The decrement is smaller than a full gap, so every step above the smallest normal moves to its immediate predecessor; finite descent therefore terminates at the fixed smallest

normal. Both the location and the converse are needed, since a non-increasing sequence on a finite lattice always terminates.

A.2 GATE-PLUS-ERASE ERROR HORIZON

One-step error propagation. Existing state error is carried forward by decay and erasure. For two trajectories on the same inputs, the additive write $\beta_t v_t k_t^\top$ is identical and cancels in their difference. A perturbation E, such as a rounding error, therefore propagates as

$$E \;\mapsto\; \alpha_t\, E\left(I - \beta_t k_t k_t^\top\right) \;\text{(GDN)}, \qquad E \;\mapsto\; a_t\, E \;\text{(Mamba-2)}, \tag{5}$$

where $a_t \in (0, 1]$ is the scalar per-head decay of a Mamba-2 head, the counterpart of α_t with the β_t-erase absent. A gated state therefore forgets its own error through the decay gate and, in GDN only, the rank-one erase, which scales the component of each row of E along k_t by $1 - \beta_t$ and contracts it for every $\beta_t \in (0, 2)$. Numerical checks on deployed kernels support both Jacobians: the rank-one erase on a GDN kernel, and the pure-decay form on every Mamba-2 kernel censused in Section 4.

We first follow the existing error without adding a new rounding error. Write $E_t = \widehat{S}_t - S_t$ for the state error before the fresh store error; ignoring that injection for one step,

$$E_t = E_{t-1} A_t, \qquad A_t = \alpha_t (I - \beta_t k_t k_t^\top). \tag{6}$$

For a row vector z and a unit key k, the squared norm after the transition is

$$\|z A_t\|_2^2 = \alpha_t^2 \left(\|z\|_2^2 - \beta_t (2 - \beta_t)(z k_t)^2\right). \tag{7}$$

The scalar horizon follows by treating key directions as equally likely. Under a key direction that is isotropic conditional on the row z and the current gates, $\mathbb{E}[(z k_t)^2] = \|z\|_2^2/d$, so averaging Equation 7 gives the squared-error retention factor $\alpha_t^2\big(1 - \beta_t(2 - \beta_t)/d\big) \approx 1 - 2\big(\delta_t + \beta_t(2 - \beta_t)/(2d)\big)$, where the approximation retains first-order terms in the per-step rates. Averaging those rates yields Equation 2.

Trained heads need not erase every error direction at the same rate. Thus τ_{err} summarizes error-amplitude persistence rather than giving an exact rate for each direction. The time-averaged erase operator in production streams is anisotropic: a component orthogonal to every key sees no erase and contracts only through α_t; directions repeatedly aligned with keys can contract faster than the isotropic average. The scalar horizon is therefore neither a directional guarantee nor a worst-case bound. Equation 3 uses it as a descriptive predictor; in-sample fit is not prospective validation. We do not use it to predict failure times.

Accumulation before saturation. Bias and zero-mean store errors accumulate differently even before forgetting matters. Write $E_t = E_{t-1} A_t + \xi_t$, with $E_0 = 0$, on a fixed input stream. The matrices A_t are fixed by those inputs; conditional expectations below are given all preceding rounding draws. If $\|A_t\|_2 \le 1$ and $\|\xi_t\|_F \le \epsilon$, the triangle inequality gives $\|E_T\|_F \le T\epsilon$; aligned errors can reach this linear rate when the transition does not attenuate them. For ideal SR, the store error has conditional mean zero (Connolly et al., 2021, Lemmas 5.1–5.2). If its conditional second moment is bounded by σ^2, the cross term vanishes in expectation, giving $\mathbb{E}\|E_t\|_F^2 \le \mathbb{E}\|E_{t-1}\|_F^2 + \sigma^2$ and hence $(\mathbb{E}\|E_T\|_F^2)^{1/2} \le \sigma\sqrt{T}$. These bounds concern absolute store error, not error divided by a changing reference norm. They exclude biased working-arithmetic errors, and stronger contraction can yield saturation rather than continued growth.

A.3 EXACT DISTRIBUTIONAL FORGETTING UNDER STOCHASTIC ROUNDING

Ideal SR can preserve the rate at which a scalar recurrence forgets its initial state. The following result compares the distributions of two rounded trajectories, rather than requiring their individual paths to stay close. Let $\mathcal{L}$ be an ordered set of representable values (a lattice, possibly nonuniform, such as the bf16 values of one sign). *Ideal* SR maps a target z between adjacent lattice points $l \le z \le h$ to h with probability $(z - l)/(h - l)$ and to l otherwise, draws a fresh uniform variable independent of the past at every step, and stores a target already in $\mathcal{L}$ exactly; hence $\mathbb{E}[\mathrm{SR}(z)] = z$ for every z in the hull of $\mathcal{L}$, the classical unbiasedness of SR (Connolly et al., 2021; Croci et al., 2022). For two laws μ, ν on $\mathcal{L}$, $W_1(\mu, \nu) = \inf \mathbb{E}|X - Y|$ over all couplings with $X \sim \mu$ and $Y \sim \nu$.

Proposition 1 (Exact W_1 forgetting)**.**
Let $X_t = \mathrm{SR}(a_t X_{t-1} + b_t)$ with $0 \le a_t \le 1$, where the stream $(a_t, b_t)_{t\ge1}$ is fixed in advance, is identical for both initial states, and keeps every target inside the hull of $\mathcal{L}$; intermediate arithmetic is exact. Let P_t^x denote the law of X_t given $X_0 = x$. Then for every t,

$$W_1\big(P_t^x, P_t^y\big) = \Big(\prod_{s=1}^{t} a_s\Big)\,|x-y|. \tag{8}$$

Proof. Lower bound. Unbiasedness gives $\mathbb{E}[X_t \mid X_{t-1}] = a_t X_{t-1} + b_t$, so $\mathbb{E}X_t^x - \mathbb{E}X_t^y = (\prod_{s\le t} a_s)(x-y)$. Every coupling satisfies $\mathbb{E}|X-Y| \ge |\mathbb{E}X - \mathbb{E}Y|$, which is the right side of Equation 8. *Upper bound.* Realize SR with an explicit uniform ξ: $\mathrm{SR}(z;\xi) = h$ if $\xi < (z-l)/(h-l)$ and l otherwise. For fixed ξ the map $z \mapsto \mathrm{SR}(z;\xi)$ is nondecreasing: inside one cell the acceptance threshold grows with z, and for $z < z'$ in different cells $\mathrm{SR}(z;\xi) \le h(z) \le l(z') \le \mathrm{SR}(z';\xi)$. Drive both chains with the same ξ_t at each step. Each chain alone still sees fresh uniforms, so the marginals are P_t^x and P_t^y. Because $a_t \ge 0$, $x \le y$ implies $X_t \le Y_t$ almost surely by induction, hence $\mathbb{E}|Y_t - X_t| = \mathbb{E}[Y_t - X_t] = (\prod_{s\le t} a_s)(y-x)$, an upper bound on W_1. The two bounds coincide. □

The shared random numbers are essential to the comparison, not an assumption that independent SR paths stay ordered. The proof uses *stochastic* monotonicity of the transition law: two independent roundings of $z < z'$ may still satisfy $\mathrm{SR}(z) > \mathrm{SR}(z')$ (Connolly et al., 2021; Croci et al., 2022). Only $a_t \ge 0$ is used; $a_t \le 1$ makes the factor a contraction. The one-step kernel coincides with the one-dimensional dual-quantization splitting operator of Jourdain & Pagès (2022). The mean calculation uses the classical conditional unbiasedness of SR (Connolly et al., 2021; Croci et al., 2022). Combining it with the shared-uniform coupling gives Equation 8 for the stated affine recurrence with fixed external writes. Neither the splitting operator nor the coupling method is new.

Scope. *Erase-free recurrences.* Diagonal recurrences retain the same forgetting equality. For a diagonal transition with per-entry gates $a_{i,t}$ (a Mamba-2 head), assume the rounding uniforms at each step have a prescribed joint law with uniform marginals, independent of the past and current state and identical for both chains. Coupling both chains with the same vector of uniforms gives the equality in entrywise ℓ_1 distance, $W_{1,\ell_1}(P_t^x, P_t^y) = \sum_i \big(\prod_{s\le t} a_{i,s}\big)|x_i - y_i|$, which for one gate per head is $(\prod_{s\le t} a_s)\|x-y\|_1$. Independent entrywise draws satisfy this assumption; dependence across entries is also allowed under the common, state-independent joint law. For any coupling, Jensen's inequality gives $\mathbb{E}\|X_t - Y_t\|_1 \ge \|\mathbb{E}X_t - \mathbb{E}Y_t\|_1 = \sum_i (\prod_{s\le t} a_{i,s})|x_i - y_i|$. Under the shared-uniform coupling, each coordinate retains its initial ordering, so its expected absolute difference equals the absolute difference of its means. Summing over coordinates attains this lower bound and proves the equality.

Round-to-nearest. RTN need not forget its initial state, even with nonzero writes: on a grid of spacing $1/8$ with $a = 0.99$ and $b = 0.01$, the state 1 maps to $0.99 + 0.01 = 1$ and the state 2 maps to 1.99, which is nearer to 2 than to 1.875; both are fixed points, so their distance stays 1 forever while SR gives 0.99^t.

The GDN erase. The erase can break the ordering used in this proof. Equation 8 does not immediately extend to the rank-one erase of Equation 1: right-multiplication by $I - \beta kk^\top$ mixes entries and is not entrywise order-preserving, so the shared-uniform monotone coupling used above no longer applies. We do not claim an exact GDN analogue here. The mean lower bound $W_1 \ge \|(x-y)\prod_s A_s\|_1$ survives the erase, since it needs only unbiasedness.

What the proposition says. The equality compares distributions, not individual generated sequences. Both SR chains receive one fixed, open-loop stream. A rounding perturbation injected at step s is an initial condition for the chain from s on; this is how the result bears on "SR forgets its own error." It does not say that independent sample paths stay close, that closed-loop generations agree (once rounding alters future gates and writes the stream is no longer exogenous), or how RTN error accumulates: the RTN fixed-point example shows persistence, not growth. Implementations with finitely many random bits, saturation, or flushed subnormals fall outside the ideal-SR hypothesis.

A.4 EXACT DECAY-ERROR ATTRIBUTION AND A DECAY-RESTORING STORE

We measure decay's contribution by comparing an update with and without decay. Let S be the stored state before a step (a matrix of lattice values) and (k, v, α, β) the current inputs, with $\delta = 1-\alpha$. Write $C = S\,(I-\beta kk^\top)$, $U = C+\beta vk^\top$, $D = \delta\, C$ and $Y = U-D$, so that $Y = \alpha C+\beta vk^\top$ is the exact next state of Equation 1, U is the same update with the decay switched off, and D is what the decay subtracts. Let Q denote entrywise round-to-nearest and define

$$r_0 = Q(U) - U, \qquad r_\delta = Q(Y) - Q(U) + D, \qquad \text{so that} \qquad Q(Y) - Y = r_0 + r_\delta \tag{9}$$

exactly, by adding the two definitions. Here r_0 is the error the step would make without decay and r_δ the change the decay causes: with the realized decay $R = Q(U) - Q(Y)$ we have $r_\delta = D - R$, so if U and $U - D$ round to the same value then $R = 0$ and the whole requested decay is lost, while if a rounding boundary is crossed R is a full lattice gap and may exceed D. Once writes are present, no threshold in δ alone decides which case occurs.

The attribution depends on which counterfactual update we choose. We switch decay off while keeping the state, erase and write fixed. Rounding αC before adding the write would give a different attribution. The identity holds with any stored value in place of $Q(Y)$; we evaluate it on the value the cache actually stores, so for a bf16 cache with fp32 working precision r_δ also absorbs the fp32 working-precision term.

Exact propagation. The two error contributions can be followed separately through later updates. Let $E_t = \widehat{S}_t - S_t$ compare the rounded trajectory with the exact one from the same initial state on the same exogenous stream, and let A_t be as in Equation 6. Applying Equation 9 at the rounded state gives $E_t = E_{t-1}A_t + r_{0,t} + r_{\delta,t}$. Define $Z_{0,t} = Z_{0,t-1}A_t + r_{0,t}$ and $Z_{\delta,t} = Z_{\delta,t-1}A_t + r_{\delta,t}$ from $Z_{0,0} = Z_{\delta,0} = 0$. Both E_t and $Z_{0,t} + Z_{\delta,t}$ obey the same linear recurrence from zero, so

$$E_t = Z_{0,t} + Z_{\delta,t} \qquad \text{for every } t, \tag{10}$$

and the readout error at a query q_t splits as $Z_{0,t}q_t + Z_{\delta,t}q_t$. Squared norms do not add, so we report the projection share and the stream-level realized-decay fraction,

$$\mathrm{share}_\delta = \frac{\langle Z_\delta, E\rangle_F}{\|E\|_F^2}, \qquad \rho = \frac{\sum_t \langle D_t, R_t\rangle_F}{\sum_t \|D_t\|_F^2}.$$

Both fractions require nonzero denominators. Here $\rho = 0$ means zero net realized projection and $\rho \approx 1$ is agreement in this projection; ρ can exceed one and can hide cancellation between loss and overshoot. The identity is retrospective accounting, not a causal intervention: removing one channel would change the states at which later errors are injected. Unrolled linear propagation of per-step cache errors is standard. The Nemotron 3 Super report writes it for a diagonal Mamba recurrence (its eq. 3) (NVIDIA, 2026), without a channel split; the counterfactual split and its two-channel propagation are the definition we add.

Proposition 2 (Decay-restoring store)**.**
Let $S_{\mathrm{next}} = \mathrm{SR}\big(Q(U)-D\big)$ entrywise with ideal SR, and suppose every entry of $\mu = Q(U)-D$ lies in the hull of the lattice. Conditionally on S and the inputs: (i) $\mathbb{E}[S_{\mathrm{next}}] = \mu$, hence $\mathbb{E}[S_{\mathrm{next}} - Y] = Q(U) - U = r_0$; (ii) for each entry, with $l \le \mu \le h$ the adjacent lattice points, $\mathrm{Var}(S_{\mathrm{next}}) = (\mu - l)(h-\mu)$, which is the least variance of any lattice-valued output with mean μ, and $(\mu-l)(h-\mu) \le \mathrm{gap}(\mu)\,|D|$ with $\mathrm{gap}(\mu) = h - l$.

Proof. (i) is unbiasedness of SR together with $\mu - Y = Q(U) - U$. (ii) A two-point law on $\{l, h\}$ with mean μ has variance $(\mu-l)(h-\mu)$. For any lattice-valued X with $\mathbb{E}X = \mu$, no lattice point lies strictly between l and h, so $(X-l)(X-h) \ge 0$; taking expectations gives $\mathrm{Var}X \ge (\mu-l)(h-\mu)$. Finally $(\mu-l)(h-\mu) \le (h-l)\min(\mu-l, h-\mu) \le (h-l)\,|\mu - Q(U)| = \mathrm{gap}(\mu)|D|$, because $Q(U)$ is itself a lattice point. □

The correction removes decay-induced bias for one step, not necessarily over a whole trajectory. It removes r_δ in conditional mean and leaves r_0, including any write that RTN already loses in U, untouched. This is a one-step conditional statement: r_0 remains an RTN error, and a changed trajectory changes future channel injections and their propagation. Removing the conditional bias

of r_δ alone therefore does not imply small multi-step state error. Part (ii) is elementary and closely related to the splitting operator of Jourdain & Pagès (2022) above; we prove it rather than claim it as new or attribute this minimum-variance statement there.

The variance bound depends on the size of the decay term as well as the stored value. For a normal target, $\mathrm{gap}(\mu) \le 2u|\mu|$ and $|D| = \delta|C|$, so the bound is $2u\,\delta\,|\mu||C|$. If $|\mu|$ and $|C|$ are comparable to $|z| = |Q(U)|$, this is $O(u\,\delta\,|z|^2)$, below the $O(u^2|z|^2)$ worst-case full-SR scale for comparable target magnitudes when $\delta \ll u$. This compares upper-bound scales, not pointwise variances: full SR has zero variance whenever its target is representable.

The implementation computes small decay probabilities without subtracting nearly equal state values. It forms the Bernoulli probability $|D|/\mathrm{gap}$ directly from D whenever $|D|$ does not exceed the gap from $z = Q(U)$ to its neighbor in the direction of $-D$ (so a decay below the fp32 resolution of $z - D$ is not cancelled away) and otherwise falls back to SR of $z - D$, giving conditional mean $Q(U_{\mathrm{fp32}}) - D_{\mathrm{fp32}}$, exact up to fp32 working precision. The null control applies zero-mean neighbor perturbations about $Q(Y)$, computing its flip probability from $|D|$ and the local directional gap at that base. It retains the RTN store's conditional mean in ideal arithmetic, but is not generally variance-matched or flip-probability-matched to the decay-restoring store, even within a neighboring cell. It therefore does not isolate mean restoration at fixed noise variance.

Role. SR of complete updates that contain a decay term is established in low-precision training (Gupta et al., 2015; Wang et al., 2018; Ozkara et al., 2025), and SR restricted to one operation of an update exists (Zamirai et al., 2020); the store differs in rounding only the decay increment on top of the round-to-nearest write and erase. Section 5 uses it as a diagnostic instrument that removes one channel with a known conditional bias, not as a proposed mitigation; we claim no novelty for stochastic rounding itself or for its use on decaying states.

B Supporting experimental methods and limitations

These experiments measure different effects of recurrent-state rounding. This appendix explains the numerical controls, sampled populations and statistical limits that separate state error, teacher-forced likelihood loss and changes in generated trajectories. None of these measurements makes active model heads equivalent to the theorem's fixed-gate, pure-decay recurrence.

B.1 Numerical controls

Reference and storage semantics. The numerical reference depends on the experiment. Recurrence replays use an fp64 reference of the same algebra, not exact real arithmetic. Model-level comparisons instead use an fp32 recurrent-state control in the otherwise low-precision model. Qwen recurrence updates compute in fp32 and round the stored state. The Falcon, Granite, Olmo and Nemotron replays instead compute each update in fp64, then cast through fp32 before the low-precision store. Their comparisons therefore concern storage error under that arithmetic, not identical fp32-working production kernels. Decoding stores the state after every token, whereas chunkwise prefill stores it at chunk boundaries. Round-to-nearest (RTN) uses ties to even; stochastic rounding (SR) chooses between the two bracketing representable values with probabilities determined by proximity. The decay-attribution experiment uses a direct fp64-to-bf16 round-to-nearest map to avoid rounding twice through fp32.

Internal checks test whether the likelihood implementation preserves its controls. It reproduced unpatched fp32-control logits bitwise over 256 steps and 4,608 recurrent calls. Masks selecting every head or no head reproduced the corresponding pure policies. These bitwise comparisons require matched batch shapes. They are implementation checks within this harness, not an audit of a deployed serving stack.

The saved runs do not all support bitwise rerunning from a nominal seed. The original Qwen replay derived SR seeds from Python's process-salted string hash and did not save the effective seeds. Its saved measurements remain the evidence for the reported curves; the nominal seed alone cannot recreate the rounding draws. Later attribution runs record stable mode seeds. Their reproduction check compares final relative-error arrays with the saved Qwen replay, not every state in each tra-

jectory. Historical dirty-source and model/tokenizer revisions are not fully pinned. The non-Qwen replay summaries also do not retain their full input streams.

Per-format arithmetic checks. The cast checks test the theorem's arithmetic predictions. The saved PyTorch checks compare casts with exact-rational threshold predictions in 512 two-sided single-step tests for bf16, 4,096 for fp16, 32 for e4m3 and 16 for e5m2, plus 29 trajectory walks per format. All 4,656 single-step checks and 116 walks report zero failures. Locking at the format's exponent floor is distinguished from the interior stall covered by the theorem.

B.2 Populations and selection

Census and scale. The census counts heads that meet the gate-based risk rule. The rule in Equation 3 combines a stall fraction $f_{\text{stall}} > 0.99$ with an approximate error horizon $\tau_{\text{err}} > 10^3$ tokens. The stall fraction measures how often requested decay falls below the format threshold; the horizon estimates how long a perturbation persists. The decode-storage census covers 7,920 recurrent heads in Qwen3.5-0.8B, Falcon-H1-1.5B, Granite-4.0-H-Micro, Olmo-Hybrid-7B and Nemotron-Nano-9B-v2. Head dimensions differ across models. These counts describe gate statistics, not behavioral outcomes.

The censuses share a sampling recipe, not necessarily identical text spans. Each uses 11 single-source rows per domain from three domains: PG-19 test books (Rae et al., 2020), GovReport reports (Huang et al., 2021) and repository-level code from installed Transformers Python sources. Books use the `emozilla/pg19` test split; reports use the `ccdv/govreport-summarization` training split. Code documents concatenate whole files in path order within a package. Texts are retokenized and length-filtered for each model. The corpus builder can also draw from PyTorch, but all code rows selected for these censuses and the likelihood study came from Transformers. Table 1 uses the three-domain Qwen re-census for both precisions.

The within-family comparison tests whether flagged heads persist at larger scales. Table 3 uses one shared tokenizer and three-domain corpus within Qwen3.5. It supports persistence of flagged heads, not a scaling law: the larger models also change head configuration.

Table 3: **Flagged heads persist across the Qwen3.5 scale ladder.** Decode-time bf16 counts use Equation 3 and 1,083,489 shared tokens per model. These are gate statistics, not likelihood or benchmark outcomes.

Model	Linear layers $\times$ heads	Total heads	Flagged
Qwen3.5-0.8B	18×16	288	46 (16.0%)
Qwen3.5-2B	18×16	288	46 (16.0%)
Qwen3.5-4B	24×32	768	86 (11.2%)
Qwen3.5-9B	24×32	768	89 (11.6%)

Table 4: **The horizon cutoff excludes some heads with high stall fractions.** Decode-time bf16 counts compare the stall-only rule with the full conjunction. The horizon clause can exclude stalled heads in either recurrence family. These are gate statistics, not measured damage counts.

Model	$f_{\text{stall}} > 0.99$	Also $\tau_{\text{err}} > 10^3$
Qwen3.5-0.8B	66	46
Falcon-H1-1.5B	185	185
Granite-4.0-H-Micro	287	287
Olmo-Hybrid-7B	89	16
Nemotron-Nano-9B-v2	617	461

Store-period estimator. The store-period calculation estimates how less frequent storage changes the stall diagnostic. It uses the original two-domain Qwen3.5-0.8B gate logs. That original corpus contains a duplicated prose row, so its pooled histograms inherit that weighting; the two rows used for detailed replays are distinct. These original inputs are logged sequences, not necessarily individual source documents. For a chunk of C tokens, with token-level decay decrements δ_t, the

effective decay is $\delta_C = 1 - \prod_{t=1}^{C}(1-\delta_t)$. The full-model estimator samples each δ_t independently from a head's empirical $\log_{10}\delta$ histogram, using bin centers and retaining the zero-decay mass. It estimates $\Pr(\delta_C < \delta_\star)$, where $\delta_\star$ is the format's stall threshold, and applies the same 0.99 cutoff. The horizon cutoff remains 10^3 *tokens*, using the original $1/(\bar{\delta}+\bar{e})$, not 10^3 stores; $\bar{\delta}$ and $\bar{e}$ are the average decay and erase contributions defined in Equation 2. The published implementation uses seed zero and $\min(4096, \max(64, \lfloor 2^{21}/C \rfloor))$ sampled chunks. Independent sampling discards temporal dependence.

A limited cross-check retains the observed order of the gates. It compounds the logged gates in their actual order on four layers and two input sequences, preserving contiguous chunk boundaries; it does not replay state arithmetic. After averaging its stalled counts over the two sequences, the largest discrepancy from the same-layer histogram estimate is two heads. This is a limited cross-check, not a bound on the full-model approximation.

Predictor disclosure. The head-selection rule was chosen after inspecting the original Qwen3.5-0.8B replay. The post-hoc rule requires both stall fraction > 0.99 and error horizon $\tau_{\mathrm{err}} > 10^3$ tokens. Among 128 head-streams, 37 exceeded 0.1 relative state error under bf16-RTN at 131,072 tokens. A head-stream is one head's trajectory on one input sequence: these streams represent 64 distinct heads on two source rows, not 128 independent heads. Stall-only and the conjunction selected the same 33 damaged streams; horizon-only selected 40, including all 37. This comparison is neither prospective nor equal-budget validation and establishes no independent discrimination beyond horizon or stall alone.

The later exploratory check did not show a gain from the horizon cutoff. That check used held-out books and a different damage measure, future-readout error energy. The horizon cutoff gave no selection gain and excluded highly damaged heads when it acted.

Replay sampling. The replay compares RTN and SR on the same logged inputs. Holding those inputs fixed makes this an open-loop test of state error, not a comparison of generated answers. Qwen's 128 head-streams come from layer ordinals 02, 03, 07 and 11 on one prose and one code row, covering 64 distinct layer/head pairs. Falcon replays heads 0–7 in layers 4, 12 and 20 on one input row: 24 distinct heads, of which five satisfy the plotted stall criterion. The streams contributing to the Falcon curves are therefore not a random sample of its 185 at-risk census heads.

The other three models also use selected replay populations. Granite, Olmo and Nemotron each replay one book row, with layers selected from census statistics and within-layer comparison heads: 63 heads in ordinals 01/02/15/33, 54 in 06/12/20, and 66 in 04/12/22, respectively. Figure 2 retains only the measured stalled subset, irrespective of original group labels.

The replay classifications and checkpoint times differ across models. Qwen uses the unit-roundoff threshold u and a different saved checkpoint grid; the other models use the exact threshold $\delta_\star$ and classify heads over their replay windows. No common-grid checkpoint or harmonized Qwen classification is inferred from those summaries. The comparisons support within-model RTN/SR contrasts, not an architecture ranking.

Retrospective methods. The original prespecified criteria failed. The head-selection and SR-rescue criteria remain failures; the post-hoc conjunction does not replace them. A separate prespecified failure-time predictor also failed its error criterion. We therefore report the selection as exploratory and do not claim to predict failure times.

Several quantitative predictions also failed outside the original replay. The registered stalled-to-control error-separation predictions failed for Granite, Olmo and Nemotron. Olmo also missed its registered error-growth and SR-rescue thresholds, and Falcon missed its quantitative growth prediction. Later control-selection amendments do not turn these original failures into passes; the within-model RTN/SR contrasts remain descriptive.

The held-out attribution study likewise missed two registered clauses. Its channel-balance count was five of sixteen streams: the recorded verdict uses the stricter "at least one third" requirement of six, although the registration's parenthesis said five. We retain that conservative failure. The decay-only readout-error ratio also missed its registered threshold, despite indicating improvement relative to RTN. Neither miss is replaced by a favorable exploratory interpretation.

B.3 LIKELIHOOD AND ATTRIBUTION

Teacher-forced design and uncertainty. The likelihood experiment tests whether state rounding changes token prediction. Under teacher forcing, every policy consumes the same token sequence while its recurrent state evolves separately. The Qwen3.5-0.8B design uses 72 documents, 24 each from books, code and government text, with 32,768 scored positions. Excess negative log-likelihood (NLL) is the increase in loss relative to the fp32-state control, measured in nats/token. It is averaged over the final quarter within each document, then over documents.

The likelihood intervals resample whole documents, not individual tokens. The 99% document-bootstrap confidence intervals are obtained by repeatedly resampling whole documents with replacement, keeping paired policy results together. The documents were selected rather than randomly sampled, and the code sequences share a source package; these intervals do not capture all corpus-selection uncertainty. Table 5 gives the pooled estimates and intervals from the separate policy-comparison run on the same document population, not a re-pooling of the displayed domain means.

Table 5: **The pooled likelihood estimates come from a separate policy-comparison run.** Qwen3.5-0.8B excess NLL relative to the fp32-state control is averaged over each document's final quarter of 32,768 scored positions, then over the 72 documents. Both means and 99% document-bootstrap confidence intervals are in nats/token. The RTN and SR rows are the "Pooled" bars in Figure 3(b), not an aggregate of its adjacent domain bars.

State-storage policy	Mean excess NLL	99% confidence interval
bf16-RTN	0.02279	$[0.01800, 0.02759]$
bf16-SR	0.00195	$[0.00114, 0.00285]$
bf16 + 8-bit residual	-0.00008	$[-0.00019, 0.00003]$
bf16 + 4-bit residual	0.00002	$[-0.00017, 0.00023]$

The error-feedback residuals use 3.0 (8-bit) and 2.5 (4-bit) bytes per state entry; the 4-bit packing is emulated in 8-bit storage, and no kernel timing was measured.

Localization and confounding. Selective storage changes test where the likelihood damage is concentrated. Corrupting selected heads applies bf16-RTN only to those heads; protecting them keeps their state in fp32 while the others use bf16-RTN. The selection masks use gate statistics collected before low-precision outcomes. They use the legacy decay cutoff $u = 1/256$, not the exact $\delta_\star = 1/258$, together with the same stall-fraction and horizon cutoffs. The 72 saved masks reproduce from those legacy statistics; their equivalence to exact-threshold masks is unverified. Random controls match head count and layer, not horizon, so they do not separate stalling from other long-horizon properties. Intervals conditional on the fixed masks do not include mask-selection uncertainty.

The domain-level results and correlations have narrower interpretations. The registered damage criterion held for books and government text, not code. Base loss is a stronger marginal correlate than flagged-head count; adjustment leaves a positive association, not causal identification.

The likelihood figure shows results from separate runs. In Figure 3(b,c), "Gov." denotes government text and "sel." the selected heads. Panel (b)'s domain SR means average two seeds; intervals were not saved for those means. Its pooled NLL bars come from the separate error-feedback policy comparison reported in Table 5. Panel (c)'s intervention ratios and their paired-document intervals use the full-RTN denominator from the intervention run, not the pooled bar in panel (b).

Each likelihood run has its own reference values. The displayed domain run has pooled RTN/SR excess NLL of $0.02324/0.00184$ nats/token, whereas the separate policy comparison gives $0.02279/0.00195$. The localization run uses its own full-RTN denominator, 0.02286 nats/token: selected-only corruption gives 0.01571, protecting selected heads gives 0.00563, and the three random-mask arms give 0.001091, 0.000615 and 0.000384. The percentage removed by protection is its reduction relative to full RTN; the corruption percentages are the remaining excess NLL divided by full RTN. They are different quantities, not additive shares of a damage budget.

Selected-head corruption produces more excess loss than the matched random controls. The ratio of pooled excess NLL under selected-only corruption to the mean across the three random-mask arms has point estimate 22.5, with a 99% paired-document bootstrap confidence interval $[14.1, 44.3]$. This interval conditions on the fixed masks and does not include mask or seed uncertainty.

Real-stream decay attribution. The attribution replay tests how decay contributes to state error. It compares each rounded update with a counterfactual update with decay disabled, separates the decay-induced error from the remaining error, and propagates both through the recurrence. The registered open-loop experiment replays 16 at-risk head-streams for 131,072 tokens: nine distinct Qwen3.5-0.8B heads in held-out layers 03 and 11 on the same two logged sequences used for development. The holdout is therefore in heads and layers, not input sequences or models.

The replayed states are not frozen. Across streams, median realized/requested decay is 96%. The median fraction of entry-steps with nonzero requested decay that match the quantized decay-off counterfactual is 99.4%; rare crossings between rounding cells deliver the realized decay. The decay-induced component has a median 58% projection share of the accumulated error, not an additive share of its squared norm.

Decay-only repair remains uncertain in this small, clustered sample. A decay-only stochastic correction has a median per-stream state-error ratio to RTN of 0.77. A post-hoc bootstrap that resamples nine heads and retains every stream belonging to each sampled head gives a 99% interval of $[0.527, 1.233]$ for that same stream-weighted median. The readout-error ratio is 0.714, with head-cluster interval $[0.555, 0.937]$. These intervals remain conditional on the two selected input sequences and layers; they do not support population-wide inference. These observations neither establish complete decay-only repair nor identify a likelihood or accuracy effect.

B.4 Generation design and uncertainty

Free-running methods. The free-running experiment tests whether storage policy changes generated trajectories. Qwen3.5-4B generates answers to all 30 problems from the `train` split of `MathArena/aime_2026`, graded against its integer reference answers (Dekoninck et al., 2026).[2] The run retains a problem/answer manifest, but its dataset-loading call did not pin an upstream revision. Unlike teacher forcing, each policy generates its own tokens. Eight batch rows ($B = 8$) share initial prompts: fp32, its duplicate, bf16-RTN, three bf16-SR seeds, fp32 with ulp-scaled uniform dither, and bf16 with an 8-bit error-feedback residual. Error feedback retains discarded rounding error in a residual. Arm names describe only the Gated DeltaNet (GDN) recurrent state; every arm retains a bf16 convolution cache. The default recurrent state is fp32, making bf16 an intervention. Duplicate fp32 traces were identical on all items.

The uniform-noise control uses a nominal scale, not exact variance matching. It adds $U(-g/\sqrt{2}, g/\sqrt{2})$ to the fp32 state, where $g = \mathrm{ulp}_{\mathrm{bf16}}(Q(S))$ is the local bf16 spacing at its nearest bf16 value. In ideal arithmetic this noise has mean zero and variance $g^2/6$. Ideal SR at a fixed fractional cell position p has variance $p(1-p)g^2$; averaging over uniform p at a constant gap gives $g^2/6$. Actual state phases and binade-boundary gaps need not satisfy that averaging assumption, so the control does not hold conditional perturbation variance fixed. No new matched control was run.

The decoding settings limit the scope of the comparison. The 32,768-token cap reflects the measured 24 GiB memory limit. Decoding is greedy with presence penalty 1.5, deviating from initially registered greedy decoding without penalty. The change followed held-out calibration, before test generation, and increased detection power in calibration. We have not established that the test-set contrast persists without the penalty; an interaction with cache-induced drift remains possible. Checkpoint contamination by competition answers is not excluded.

Endpoints, censoring and seed spread. The primary endpoint compares capped generation lengths. The statistic averages each item's RTN-minus-SR log length, averaging the three SR log lengths within item first. Its mean is $+0.2166$. For this mean-of-three estimator, a two-stage bootstrap resamples problems, then three SR replicates with replacement within each sampled problem. Using 20,000 resamples and seed zero gives a 99% interval $[+0.097, +0.357]$. The one-sided sign-permutation test gives $p = 4.0 \times 10^{-6}$.

[2] Dataset: `https://huggingface.co/datasets/MathArena/aime_2026`.

The interval depends on how seed variation is represented. Resampling only one SR replicate per drawn problem gives the wider $[+0.081, +0.368]$, but targets a different seed-averaging procedure. The original saved interval resampled problems with each observed three-seed mean held fixed. We distinguish these procedures rather than treating them as interchangeable; three fixed seeds provide limited information about other rounding realizations.

Reaching the cap leaves the eventual stopping time unknown. These traces are right-censored and are never dropped; unconstrained lengths remain unknown. The capped log-length difference is not generally a lower bound on the uncapped paired difference, because either arm may be censored. A paired Gehan comparison accounts for censoring: on the preselected primary SR seed, it finds RTN longer on 15 items, shorter on one and indeterminate on 14, $p = 2.6 \times 10^{-4}$. Other seeds give $p = 1.0 \times 10^{-5}$ and 1.1×10^{-2}.

Seed choice affects the size of the observed contrast. SR seed spread spans roughly half the RTN-to-fp32 gap in cap hits and correct answers.

The second endpoint counts budget exhaustion or repetitive generation. It is met by cap exhaustion or three consecutive repeats of a 50-token window and gives nine versus one discordant items, $p = 0.0107$. Both endpoints meet their allocated 0.025 levels.

Accuracy is a diagnostic rather than independent confirmation. Pooled across all eight arms, including duplicate fp32, 136 of 139 finished traces were correct, versus none of 101 capped traces.

Control uncertainty and causal scope. The noise control is not distinguishable from either SR or fp32. Control-minus-SR log length is -0.061 with 99% interval $[-0.166, +0.032]$, and control-minus-fp32 is $+0.010$, $[-0.094, +0.117]$. These saved intervals resample problems while holding the observed SR seed means fixed; they are conditional on those seeds. These comparisons provide only partial support for the bias interpretation: they establish neither equivalence to SR nor that removing rounding bias explains the generation change. The experiment establishes changed trajectories under recurrent-state storage interventions, not that stalling caused them. Selective free-running localization remains unrun.

Separate longer-cap follow-up. The longer-cap follow-up revisits selected fp32–RTN disagreements. An exploratory 65,536-token run revisits the 13 items initially answered correctly by fp32 but not RTN, using only those two arms ($B = 2$). It retains presence penalty 1.5 rather than the initial penalty-free greedy decoder. This is a separate experiment, not a continuation: changing batch shape changes arithmetic, so its lengths are explicitly incomparable to the 32k, $B = 8$ lengths. The selected subset cannot estimate full-benchmark accuracy or isolate a cap-extension effect.

The selected rerun records outcomes only for fp32 and RTN. Here fp32 answers 10 of 13 correctly and hits the cap on two; RTN answers six correctly and hits the cap on seven. Three RTN correct answers arrive within the original budget and three beyond it. There is no SR arm, so this follow-up cannot establish SR behavior at the longer horizon.