# Periodic Versus Random Block Resets in a Reservoir: Bounded Staleness Beats Random Forgetting

A Preprint

**Caitlyn Meeks**
AuxiLab Tenerife — research@auxi.cafe

August 11, 2026

registered 2026-08-11 | internal run log #20, #21, #24 | 3 seeds × 6 comparisons, zero reversals | replicated before publication | promoted 2026-08-11



**Abstract**

This study operationalizes an open-ended design brief as a forced block-reset schedule. The program's reference 1024-unit reservoir has its state partitioned into equal contiguous blocks, and at every timestep exactly one block is forcibly zeroed — under a **periodic** schedule (block t mod n, round-robin) or a **random** one (a uniformly random block each step), the two removing identical total state volume per step. The finding, replicated on three independently drawn reservoirs with zero reversals in 18 registered comparisons: **the periodic schedule never costs more, and at heavier reset volume costs measurably less**. At coarse resets (1/8 of the state zeroed per step) periodic beats random on ridge bpc, logistic bpc, *and* decode depth in every seed (logistic gap 0.0644–0.0661 bits); at fine resets (1/64) the two schedules nearly tie (0.0053–0.0056) — a ~12× larger periodic advantage under heavier resets, stable across all three seeds. Both schedules cost something relative to the reset-free reservoir: forced forgetting is never free. But a bounded worst-case staleness — no block ever waits more than n steps, and none is ever reset twice in succession — is worth something that a text stream, which has no true period at 8 or 64 characters, would not be expected to reward under a naive entropy argument.

***Keywords*** reservoir computing · state resets · forgetting schedules · structured vs random perturbation · memory · pre-registration

---

## §1 Introduction

Forced forgetting in a recurrent reservoir is usually treated as a question of volume: how much state can be discarded per step? Zeroing one block of the state per step sharpens it into a question of *schedule*: holding the volume of forgetting exactly fixed, does its temporal *structure* matter? A round-robin schedule and a uniformly random one destroy the same number of units per step; they differ only in structure — round-robin selection guarantees every block a full cycle of accumulation time, while random selection occasionally leaves a block untouched for a long interval, or selects the same block twice in succession. Random per-step zeroing of units is a familiar object in the training literature [2]; the question here is whether imposing deterministic structure on that zeroing, at matched volume, is worth anything for inference-time dynamics.

## §2 Method

The program's reference reservoir [1] (N=1024, sparse tanh, spectral radius ρ=0.95, fanin 10, the standard input map, leak 1.0 — wiring completely unchanged from the archived reference cell of the addressed-memory ESN study) is used with one intervention: its units are partitioned into n_blocks equal contiguous blocks, and immediately after each normal update, exactly one block is set to zero:

```
# periodic: block (t mod n_blocks) is zeroed at step t
#   — bounded staleness: no block waits > n_blocks steps,
#     none is ever reset twice in succession
# random:   a uniformly random block is zeroed at step t
#   — identical total reset volume at matched n_blocks
x[block] ← 0   # after the leaky-tanh update, every step
```

Two schedules × two block counts (n_blocks ∈ {8, 64}: coarse resets, 1/8 of the state per step; fine, 1/64) = 4 cells per seed. The reset-free reference cell (ridge 3.1472, logistic 2.6727, depth 10.11) is cited from the archive, not rerun. Standard text8 splits [3], 2M train characters; instruments: ridge and logistic validation bits per character (bpc), and U3 decode depth — a linear decode-depth probe, so designated in the program's internal instrument numbering.



## §3 Pre-registered predictions

- **W1** (volume costs monotonically): coarse resets are worse than fine on all three instruments, at both schedules — 6/6.
- **W2** (manipulation check): all four reset cells are worse than the reset-free reference on all three instruments — 12/12. Any forced zeroing costs something.
- **W3** (the headline): periodic beats or ties random at both block counts on all three instruments (tie bands ridge/logistic ±0.005, depth ±0.2) — 6/6. Named alternative: any reversal means the periodic schedule's regularity is actively harmful, e.g. a synchronized reset pattern interacting badly with the data's own structure.
- **W4** (the gap scales with reset volume): the periodic-vs-random logistic gap is larger at 1/8 than at 1/64.

Bands Q1–Q3 put numbers on the levels and the gaps. The registered follow-up policy promoted only W3, and only after replication; W1, W2, and W4 are supporting context even where they came in clean. The seed-1 rerun re-registered all four claims in the same wording; a third seed was subsequently commissioned, re-registering the headline and gap claims (W3/W4) in identical wording — W1/W2 were supporting context, not re-registered a third time — under the pre-stated rule that a clean third result appends a replication count and that a reversal would be flagged rather than silently absorbed.

## §4 Results

**Table 1.** All three seeds (internal run log #20 / #21 / #24; values seed 0 / 1 / 2). Reset-free reference cell, cited from the archive: ridge 3.1472, logistic 2.6727, depth 10.11. Zero consistency-gate failures, and no activation of the divergence tripwire (the program's fit-invalidating divergence rule).

| cell | ridge val bpc | logistic val bpc | U3 depth |
|---|---|---|---|
| **periodic, 1/8** | **3.3712 / 3.3706 / 3.3740** | **2.8905 / 2.8937 / 2.8980** | **7.42 / 7.39 / 7.43** |
| random, 1/8 | 3.4257 / 3.4249 / 3.4300 | 2.9564 / 2.9581 / 2.9641 | 7.06 / 7.02 / 7.06 |
| **periodic, 1/64** | **3.2718 / 3.2684 / 3.2756** | **2.7975 / 2.7950 / 2.8026** | **9.16 / 9.17 / 9.26** |
| random, 1/64 | 3.2760 / 3.2723 / 3.2801 | 2.8031 / 2.8003 / 2.8080 | 9.14 / 9.13 / 9.21 |

**W3, the headline: 6/6 in every seed, with no reversals.** At coarse reset volume the periodic schedule wins outright on all three instruments, in every seed, well outside the tie bands. At fine reset volume the detailed structure replicated, not just the direction: a ridge tie (gaps 0.0042/0.0039/0.0045, inside ±0.005), a genuine logistic win just past the band (0.0056/0.0053/0.0054), and a depth tie — the same tie-win-tie pattern on all three independently drawn reservoirs. W1 came in 6/6 and W2 came in 12/12 as registered on seeds 0 and 1; seed 2's table shows the same pattern in every comparison, though only W3/W4 carried registered bars there. Forced forgetting always costs something relative to the reset-free reference.

**W4: the periodic advantage scales with reset volume, stably.** Periodic-vs-random logistic gaps: 0.0659 / 0.0644 / 0.0661 bits at 1/8 against 0.0056 / 0.0053 / 0.0054 at 1/64 — ratios 11.8×, 12.2×, 12.2×. The advantage of the periodic schedule is ~12× larger at the heavier reset volume, to within a percentage point, on three reservoirs that share no wiring and (in the random arms) no reset sequence.

**Registered bands.** At seed 0: Q1 hit; Q2 missed — the coarse periodic cell cost less than the pilot-informed band predicted (2.8905 against a floor of 2.95); Q3 missed at 1/8 and hit at 1/64 — the coarse periodic advantage (0.0659) came in at more than double the band's upper end (0.03). Both misses run in the beat-expectations direction. The replication bands — held against the earlier seeds' actuals rather than the original conservative estimates — then came in 3/3 on both later seeds, with every bpc cell within a hundredth of a bit of its counterpart at the earlier seed.

---

## §5 Discussion

**Bounded staleness beats random selection.** The stream being modeled has no true period at 8 or 64 characters, so a content-blind, input-independent reset schedule has no obvious phase structure to exploit — the same total information is destroyed either way. The schedule matters anyway. The plausible mechanism, recorded with the finding: worst-case staleness and repeated resets of the same block. Random selection occasionally leaves a block unreset for many steps (its contents grow stale, then are destroyed all at once) or resets the same block twice in succession (a wasted reset); round-robin selection structurally forbids both. That risk bites harder when the per-step surviving budget is smaller — consistent with the measured ~12× scaling.

**Forgetting is never free, but its schedule is a real design axis.** Both schedules cost real bits and real depth relative to the reset-free reference at every setting — the periodic schedule does not convert forced loss into a net gain. What it shows is that at fixed reset volume, *structure* in the forgetting schedule is worth up to 0.066 bits — roughly the size of effects this program treats as headline findings elsewhere — and that the worth is predictable from the reset rate. Follow-up axes named in the program's register but not run: finer block-count ladders, and non-contiguous block partitions.

---

## §6 References


1. H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," GMD Report 148, German National Research Center for Information Technology, 2001.
2. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research* 15(56), 2014.
3. M. Mahoney, "Large text compression benchmark" (text8), mattmahoney.net/dc/textdata.


---

## §7 Provenance

- **eb8cf008** — the initiating design brief; 4 cells, seed 0 (internal run log #20). Registered before running, pilot run disclosed. Headline hit 6/6 ⇒ provisional, replication auto-queued before any confirmed-findings entry.
- **ab4f82a2** — seed 1, fresh reservoir wiring and a fresh random reset sequence (run internal run log #21). 6/6 again, zero reversals; promoted to the program's confirmed-findings record per the pre-stated rule.
- **79ae8b93** — seed 2, commissioned as a direct third-seed confirmation; the headline and gap claims re-registered in identical wording (run internal run log #24). 6/6 a third time; the findings entry annotated 3/3 seeds.

**Audit status: audit pending.** This program requires every promoted finding to be independently audited by a party that did not run the science, re-deriving its numbers from the raw result files with freshly written code. That audit has not yet been carried out for this finding. Replication and audit are different guarantees, and this result currently has the first and not the second.

All claims judged strictly against the registered wording, misses included above; 3 seeds; replicated before publication. Internal designation: waterwheel.



---