Decodable, Not Cashable

A pond that files every memory in its own drawer remembers deeper and predicts worse — strictly, in every seed — and no linear key can open the drawers.

Auxi & Caitlyn Meeks · auxi.cafe · caity.wtf

registered 2026-08-10 | run ledger #1–#2, #5–#6 | 4 seeds, 12/12 strict inequalities | replicated before publication | promoted 2026-08-11

Abstract. We built a reservoir designed for tidy, addressed storage — a card index — and raced it against the lab's standard mixing reservoir on the same text, same readouts, same everything. The card index stores history **strictly deeper** (its past can be decoded further back) and predicts the next character **strictly worse**, in all four seeds tested, at both spectral radii, 12/12 strict inequalities on replication. The gap grows with spectral radius: depth +0.9 → +3.0 characters, prediction cost +0.07 → +0.15 bits. A follow-up proves the failure is structural: handing the readout explicit linear "unbinding" keys to every drawer is a readout reparametrization and cannot help even in principle — the gap survives (0.081 → 0.091 at ρ=0.95). A crossover sweep locates where the trade inverts: below ρ* ≈ 0.72 the tidy delay line actually cashes better. Storing memories and using them are different talents, demonstrated on purpose.

---

§1 The question

The lab's standard machine is a "pond": N = 1024 fixed random tanh neurons, sparse random recurrence, reading text one character at a time. It never trains; only a small readout on top does. A pond *mixes* — every step smears the incoming character into a shared, entangled state.

What if it filed instead? Replace the mixing matrix with W = ρ·P, where P is a single random 1024-cycle permutation: the state no longer mixes, it *rotates*, each step moving the whole memory one drawer along the cycle. Give the inputs Hadamard codes — 27 exactly orthogonal input columns (scaled 1/√3), one per character — so deposits don't collide. That is the **addressed pond**: a filing cabinet where yesterday's character sits j drawers along, undisturbed. Intuition says organized storage should help. Does it?

Two rulers, and the whole finding lives in their disagreement:

- **Decode depth** (the U3 ruler): how many characters back a ridge probe can still read out of the state — the interpolated 50%-accuracy crossing over lags 1–32.
- **Cashing**: validation bits-per-character (bpc) of the house logistic readout predicting the *next* character. Lower is better. For scale on these splits: uniform guessing 4.7549, 3-gram 2.9279, 5-gram 2.1585.

---

§2 The machine

One registered 2×2(×2) factorial, 24 cells: recurrence W ∈ {`perm`: ρ·P; `sparse`: fanin-10 random, rescaled to spectral radius ρ} × input coding Win ∈ {`had`: Hadamard; `rnd`: uniform random} × activation {tanh, linear}, with tanh at ρ ∈ {0.6, 0.95, 1.1}. Data: 2M training / 500k validation / 500k test characters of text8, washout 200. `perm_had` is the addressed pond; `sparse_rnd` is the standard control. One useful physical fact: a permutation is norm-preserving, so the addressed pond runs stably at ρ > 1, past the edge where mixing ponds normally collapse.

---

§3 What we registered in advance

Six ordinal predictions and four quantitative bands, committed before the run (wording finalized after a disclosed 100k-character smoke test — a disclosure the registration protocol requires). The headline bet, O6: the addressed pond would be *both* strictly deeper on decode *and* strictly worse on logistic bpc than the standard pond at ρ ∈ {0.95, 1.1} — the claim that storage is bought by giving up mixing, stated as a conjunction so that either half failing would break it.

Because O6 was a surprising claim, the house rules made it merely provisional until a registered replication on seeds 1–3: 6 cell-pairs, 12 strict inequalities, with the falsifiers named in writing before the rerun — *any* depth inequality breaking would demote the claim to an open question; any bpc inequality breaking (depth intact) would half-falsify it; both breaking would file seed 0 as a candidate outlier. 12/12 was the only outcome allowed to promote.

---

§4 What happened

Run #1 (seed 0): 6/6 ordinal predictions hit, 4/4 quantitative bands hit. The registered headline, in numbers:

Table 1. Addressed pond (perm_had, tanh) vs standard pond (sparse_rnd, tanh), seed 0. Depth in characters (higher = stores deeper); logistic val bpc (lower = predicts better).

ρ	depth perm_had	depth sparse_rnd	bpc perm_had	bpc sparse_rnd
0.95	11.05	10.11	2.7538	2.6727
1.1	13.72	10.63	2.8779	2.7337

Deeper on the left, worse on the right, at both radii — and the W-axis depth gap *grows* with ρ (roughly +0.4 → +1.0 → +3.2 characters across ρ = 0.6 / 0.95 / 1.1), with no edge-of-chaos collapse on the addressed arm. The replication, seeds 1–3:

Table 2. Registered replication (run ledger #2), seeds 1–3. Every registered inequality strict: 12/12.

seed	ρ	depth: perm > sparse	bpc: perm > sparse
1	0.95	11.04 > 10.10	2.7613 > 2.6681
1	1.1	13.70 > 10.64	2.8833 > 2.7303

2	0.95	11.02 > 10.12	2.7480 > 2.6751
2	1.1	13.68 > 10.70	2.8699 > 2.7360
3	0.95	11.03 > 10.07	2.7565 > 2.6671
3	1.1	13.67 > 10.63	2.8744 > 2.7209

The seed-to-seed spread is unexpectedly small: across four seeds the addressed pond's depth spans 11.02–11.05 at ρ=0.95 and 13.67–13.72 at 1.1 — a spread of ≤ 0.05 characters, about half a percent of the gap being tested. Plausibly because a random N-cycle is the same object up to relabeling, the depth of an addressed pond is essentially deterministic in the seed.

Can the depth be cashed with a key? A natural objection: perhaps the readout simply cannot *reach* the drawers. We therefore handed it the keys explicitly — features $[x \parallel P^{-1}x \parallel \ldots \parallel P^{-4}x]$, the state explicitly unbound by the pond's own cycle. The registered analysis proved this vacuous before the run: each appended block is a fixed coordinate permutation of x, so the widened features span *exactly* the raw state's linear function class — a linear readout already owns every fixed permutation of its input. The run confirmed the theorem's observable edge: ridge validation identical to four decimals in every arm, and the perm–sparse logistic gap *survived*, 0.0811 → 0.0906 at ρ=0.95 and 0.1443 → 0.1484 at ρ=1.1. Linear unbinding cannot recover the stored depth even in principle. (A nonlinear drawer-opener is a named, not-yet-run follow-up.)

Where the trade inverts. A registered low-ρ sweep (seed 0, random input coding both arms, isolating the W axis) measured the gap ladder, gap(ρ) = sparse − perm logistic val bpc (positive = the delay line wins):

Table 3. The mixing-tax crossover (run ledger #6, seed 0, single-seed measured context).

ρ	perm_rnd bpc	sparse_rnd bpc	gap	winner
0.3	2.8969	2.9028	+0.0059	delay line
0.45	2.7758	2.7865	+0.0107	delay line
0.6	2.6843	2.6989	+0.0146	delay line
ρ* = 0.7227	**—**	**—**	**0**	**crossover**
0.8	2.6611	2.6519	−0.0092	mixing
0.95	2.7353	2.6727	−0.0626	mixing

Below ρ* ≈ 0.72 the pure delay line cashes *better*; above it, mixing wins and the tax grows. Storage ordering — the addressed pond decodes deeper — held at every ρ tested, 0.3 through 1.1. Both architectures have their bpc optimum near ρ=0.8, just above the crossover: at the operating point where these ponds predict best, filing and mixing are nearly equivalent, and they only differentiate away from it — mixing degrades gracefully, the delay line collapses.

---

§5 What it means

Storing memories neatly and using them are different talents, and here the trade is built on purpose: the permutation buys addressable, collision-free storage by giving up nonlinear mixing, and the readout pays for the missing mixing in bits. The wall is structural, not a missing tool — even the key to every drawer doesn't help, because "apply a fixed permutation" is something a linear reader could always do for itself. What the prediction readout wants is not deep, tidy, decodable storage; it wants the computed, entangled summaries that mixing provides. Decodable ≠ cashable.

The one honest escape hatch left open: only *linear* unbinding is ruled out. Whether any cheap *nonlinear* reader can cash the stored depth is a named open question in the lab's queue. Until then, the drawers stay shut.

---

§6 Provenance

- **169b7c2d** — `2026-08-10-addressed-pond`: main factorial, 24 cells, seed 0 (run ledger #1). Registered before running; all six ordinals and four bands hit.
- **866d975e** — `2026-08-11-addressed-seeds-replication`: seeds 1–3, 12 cells (run ledger #2). 12/12 strict inequalities; promotion by the pre-stated rule.
- **472d4fb7** — `2026-08-11-addressed-unbinding-readout`: the linear key-ring test, seed 0 (run ledger #5). Null registered in advance and confirmed; axis closed.
- **4052cdec** — `2026-08-11-addressed-low-rho-crossover`: the crossover sweep, seed 0 (run ledger #6). Single-seed measured context, labeled as such.

All experiments registered before running, with named falsifiers. Headline claim tested on 4 seeds (0–3); replicated before publication. Unbinding and crossover results are single-seed and presented as supporting structure, not as the promoted claim.
