# Codebook Quantization Compiles a Reservoir into a Finite-State Automaton

A Preprint

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

August 11, 2026 · amended August 13, 2026

registered 2026-08-11 | internal run log #16–#17, #82 | 2 seeds × 5 codebook sizes, zero misses
replicated before publication | promoted 2026-08-11 | amended 2026-08-13 after audit



**Amendment notice, 2026 August 13.** All three findings in this preprint were independently audited on 2026-08-13. The compilation finding and the smoothness finding drew **audit passes**, on re-derivation and code review, with full-budget re-measurement deferred because the machine was locked by a running experiment. The cost finding drew a recorded **concern** about the precision of one quoted number — a distinct verdict, not a qualified pass — and that number has since been corrected. In one line: the float32 reference the K=4096 cost was measured against existed at seed 0 only, so the originally published band, 0.053–0.057 bits, subtracted a *seed-0* reference from a *seed-1* measurement without disclosing it — a band narrower than the family's own seed-to-seed spread. The missing seed-1 reference has now been measured directly (2.6467; internal run log #82), and the corrected, entirely same-seed band is **0.0570–0.0583 bits**. The headline is unchanged and the correction tightens rather than weakens it. §4 and §7 report this in full; the Table 1 caption now carries both references.

## Abstract

Can a continuous-state reservoir be *compiled* into a finite-state automaton at negligible cost? An earlier study in this research program asked the question with the obvious instrument — quantizing every coordinate of the state to b bits — and obtained a clean negative: determinism never appeared at any of the nine resolutions tested between 1 and 23 bits, because a coarser grid still leaves astronomically many representable states. This study changes one thing: instead of rounding coordinates independently, the state is snapped, at every step, to its nearest vector in a **fixed codebook of K states the unquantized reservoir actually visited** — a hard cap on the cardinality of the state vector itself. That one change reverses the result. (1) At *every* tested K ∈ {16, 64, 256, 1024, 4096}, on both seeds — 10 of 10 cells — the automaton the codebook defines becomes *synchronizing*: over 40,000 probed positions, identical recent-character suffixes of depth 7–11 map to the identical state in every multi-sample suffix class. (2) The dial is smooth: prediction cost degrades monotonically as the codebook shrinks, 8/8 adjacent comparisons, with no discontinuity and no reversal. (3) The largest tested codebook is nearly free: K=4096 costs 0.0570–0.0583 bits over each seed's *own* unquantized float32 reference — the cheapest tested point on the dial compiles just as completely as the expensive ones. That third band is a post-publication correction: an independent audit found the originally quoted figure had silently compared a seed-1 measurement against a seed-0 reference, and the missing reference was then measured directly (§4, §7).

***Keywords*** reservoir computing · vector quantization · finite-state automata · state-space discretization · determinism · pre-registration

## §1 The question

A reservoir's state is a point in a 1024-dimensional continuum; its dynamics can be checked only statistically. A finite-state machine, by contrast, can be tabulated, audited, and proved things about. An earlier study in this program (internal run log #7) posed the compilation question with per-coordinate state quantization and obtained a striking negative: bits-per-character (bpc) cost degraded smoothly below b≈6 bits and the trained readout was *exact* to float32 from b=8 (the closed-form ridge readout to within 0.0001 bits) — but short-memory determinism stayed at exactly 0.0 at all nine tested resolutions between b=1 and b=23. Rounding re-injects a small perturbation at each step, and a b-bit grid over 1024 coordinates still admits vastly more representable states than any run visits; no state recurred among the 40,000 sampled states. A companion study of normalized fixed-stride descent had meanwhile left an untried construction in the program's notes: *a reservoir whose state is confined to a fixed finite set of anchor states*. This study builds that reservoir and asks whether a hard cardinality cap — rather than a finer or coarser grid — is what forces a reservoir to become a machine.

AuxiLab · auxi.cafe · August 2026

## §2 Method

The base system is the program's reference sparse-tanh echo state network (ESN) at its best measured operating point (N=1024, spectral radius ρ=0.8, sparse fan-in-10 mixing, the program's standard input map, leak 1.0) [3] — the same anchor cell the earlier per-coordinate quantization study used. One line is added to its update:

```
# C = codebook: K states the unquantized reservoir visited
#     during a fixed 100k-char warmup, sampled once,
#     held fixed for the whole run (no k-means, no synthesis)
x ← argmin_{c ∈ C} ‖x − c‖   # after every update step
```

The snapped state feeds the next step, so the closed-loop dynamics live entirely on the K codebook states — nearest-neighbor vector quantization [1] of the state, with the codebook drawn from the reservoir's own visited set rather than fitted to it. K sweeps {16, 64, 256, 1024, 4096}. Instruments are copied verbatim from the earlier quantization study for direct comparability: ridge and logistic validation bpc on standard text8 splits [4] (2M training characters), and the suffix-determinism probe (T5 in the program's internal instrument numbering): over 40,000 sampled positions, states are grouped by their length-k character suffix, and the probe reports the fraction of multi-sample suffix classes whose members all occupy exactly one state. The freeze depth k* is the smallest depth at which that fraction reaches 1.0 — that is, the shortest window of recent input that fixes the state exactly. A system with finite k* is said to *freeze*. The closed-loop dynamics are a K-state automaton by construction, whether or not k* is finite; what finite k* adds — and what is measured here — is that the automaton *synchronizes*, so that k* characters of input drive it to one determined state regardless of where it started. k* is judged only where at least 20 multi-sample classes support it — at K=4096 (seed

0), k*=10 rests on 2,172 such classes, not a thin slice. A consistency-check cell (K=∞, no snapping) reproduced the plain unquantized reservoir bit-for-bit through the same code path in both pre-registration pilot runs.

---

## §3 Pre-registered predictions

- **M1** (existence, per K, judged independently): does k* come out finite at each of the five K's? Named alternative, per K: the snap-to-nearest correction re-injects noise the way rounding did, and a small *cardinality* cap does not force determinism by itself. Freezing at any K is a substantive result against the earlier study's zero-for-nine contrast.
- **M2** (the dial): logistic validation bpc strictly improves along the whole ladder K = 16 → 4096, all four adjacent pairs (tie band ±0.005). Named alternative: non-monotone — as disclosed at registration, the pre-registration pilot run had hinted k* might dip at K=256, so cost might too.
- **MQ1** (the price): logistic validation bpc at K=4096 lands in [2.6519, 2.80] — within 0.15 bits of the per-coordinate quantization study's float32 reference (2.6519; quoted, not recomputed, and structurally not this dial's K→∞ limit, since a codebook-quantized reservoir only revisits stored anchors).

The registration also pre-stated the promotion rule: M1 firing at any K is provisional on a first seed; M1 *together with* MQ1 landing at the same K — a checkable automaton at negligible cost, the earlier quantization study's own named target — falls in the program's highest first-pass significance tier, which by standing policy requires a replication, queued ahead of all other pending work, before any entry in the program's confirmed-findings record. That is what happened. The seed-1 rerun registered the same three claims (N1/N2/NQ1), each with its own separately scored promotion rule.

---

## §4 Results

**Table 1.** Both registered passes (internal run log #16, seed 0; #17, seed 1; values seed 0 / seed 1). "Used" = distinct codewords active in the closed loop. Float32 reference, logistic: 2.6519 at seed 0 (per-coordinate quantization study, run log #7) and 2.6467 at seed 1 (run log #82, measured after the audit — see the amendment notice); ridge reference 3.1073 at seed 0 and 3.1027 at seed 1, from the same two cells. No cell tripped the divergence tripwire — a fit-invalidating divergence rule — across 10/10 cells.

| K | logistic val bpc | ridge val bpc | k* | used |
|---|---|---|---|---|
| 16 | 3.7762 / 3.7695 | 4.0528 / 4.0668 | 8 / 7 | 16 / 15 |
| 64 | 3.3916 / 3.3665 | 3.7971 / 3.7706 | 8 / 7 | 63 / 64 |
| 256 | 3.1122 / 3.1085 | 3.5675 / 3.5644 | 7 / 10 | 252 / 252 |
| 1024 | 2.8401 / 2.8376 | 3.2742 / 3.2706 | 9 / 10 | 948 / 949 |
| **4096** | **2.7089 / 2.7050** | **3.1406 / 3.1403** | **10 / 11** | **3320 / 3332** |

**M1/N1: 10 of 10 cells froze.** Every codebook size, on both seeds, reached full suffix-determinism within a suffix depth of 7–11 characters — k*'s exact value is not seed-stable, but its finiteness is universal, always well inside the probe's depth limit of 20. The contrast with per-coordinate rounding is not a near-miss; it is total (0 of 9 cells there, 10 of 10 here). The freeze is not a collapse into a trivial handful of states: at K=4096, over 3,300 distinct codewords remain in active use.

**M2/N2: the dial is smooth, 8/8.** Logistic validation bpc improves strictly at every adjacent step of the ladder on both seeds, with gaps (0.385/0.279/0.272/0.131 at seed 0; 0.403/0.258/0.271/0.133 at seed 1) tens of times the tie band. The disclosed non-monotonicity concern did not materialize for cost, even though k* itself is mildly non-monotone in K at seed 0; the two curves do not have the same shape, which is recorded as an observation and not pursued further here.

**MQ1/NQ1: the low-cost end of the dial compiles.** K=4096 sits 0.0570 bits (seed 0, 2.7089 – 2.6519) and 0.0583 bits (seed 1, 2.7050 – 2.6467) above its own seed's float32 logistic reference — a third of the registered tolerance — while still freezing at k*=10/11. Ridge gives the same picture on the same same-seed basis (+0.0333 at seed 0, 3.1406 – 3.1073; +0.0376 at seed 1, 3.1403 – 3.1027), so the price is not an artifact of the trained instrument. Both halves of the result landed at the same K on both seeds; per the pre-stated rule, all three claims were promoted separately after the second pass.

---

## §4.1 The audit's concern on this number, and its repair



The seed-1 figure above is not what this preprint originally published. The audit of 2026-08-13 re-derived both deltas exactly as they were then written (0.0570 and 0.0531) and confirmed, field by field, that the reference reservoir and the codebook-quantized reservoir share every configuration value and the identical readout stack — a genuine like-for-like comparison rather than a pipeline mismatch. But it then established something the study had not disclosed: the unquantized float32 reference cell *existed at seed 0 only*. An exhaustive scan of the program's result files found no unquantized reference at any other seed. The seed-1 endpoint was therefore a cross-seed subtraction — a seed-1 codebook cell minus a seed-0 reference — and the quoted band's width, 0.0039 bits, was *narrower than this family's own seed-to-seed spread at matched K* (0.0025–0.0251). "0.053–0.057" read as a two-seed measurement range when it was one measurement plus one unquantified difference. The audit left the finding promoted and the headline intact — at roughly 0.055 bits the largest codebook is cheap under any reading — and recorded the precision claim as having outrun its baseline coverage.

The repair was run rather than argued. The missing seed-1 unquantized cell was measured directly at the same full budget (internal run log #82), giving logistic validation bpc 2.6467 — within 0.0052 bits of the seed-0 reference, so the reference reservoir is in fact seed-stable at this width and the substitution was imprecise rather than materially wrong. With both endpoints now same-seed measurements, the corrected band is **0.0570–0.0583 bits**, width 0.0013 — genuinely narrower than the family's seed spread this time, because neither endpoint is a stand-in. No other number in Table 1 changed.

---

## §5 Discussion

**Cardinality, not resolution, is what compiles.** Per-coordinate rounding makes the state grid coarser but leaves the reachable set effectively unbounded, and the rounding perturbation keeps the quantized system from becoming deterministic. A fixed codebook of visited states caps the reachable set at K outright — and at every tested K the closed-loop dynamics promptly become a function of a short window of recent input. The result parallels the automata-extraction literature, where finite abstractions of trained recurrent networks are constructed by external algorithms [2]; here the network *is* the automaton, exactly, by construction, and the question was only what that costs.

**The cost is measured, small, and smooth.** The compiled reservoir at K=4096 gives up about six hundredths of a bit of next-character prediction relative to its continuous counterpart, and the price rises predictably, without discontinuity, as the codebook shrinks toward 16. For this research program, an inspectable finite-state machine is therefore an object with a measured price rather than a hoped-for limit.

**Open questions.** Why should the codebook consist of *actually visited* states, as against an arbitrary codebook of the same size? Since this paper first appeared, the random-codebook control has been run, replicated across three seeds, and promoted: an arbitrary codebook compiles the reservoir only below K≈64, with K=64 itself a seed-sensitive boundary — see the companion study on random codebooks for the full result, including a first-pass threshold location that replication corrected. Also open: how far the near-free regime extends past K=4096, and whether the compiled automaton transfers across reservoirs.

---

## §6 References


1. R. M. Gray, "Vector quantization," *IEEE ASSP Magazine* 1(2), 1984.
2. G. Weiss, Y. Goldberg, E. Yahav, "Extracting automata from recurrent neural networks using queries and counterexamples," *ICML* 2018.
3. H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," GMD Report 148, German National Research Center for Information Technology, 2001.
4. M. Mahoney, "Large text compression benchmark" (text8), mattmahoney.net/dc/textdata.


---

## §7 Provenance

- **0c1eaba4** — the design brief for this study; 5 cells, seed 0 (internal run log #16). Registered before running, with the pilot run disclosed (including that both pilot K values froze — flagged as suggestive only at that scale). The pre-stated highest-tier combination fired; the replication was queued ahead of all other pending work before any entry in the confirmed-findings record.
- **14080b1b** — same 5-K sweep, seed 1 — an independent draw of the base reservoir, the warmup trajectory, and the codebook sample (internal run log #17). All three claims confirmed; promoted separately per the pre-stated rules.
- **c16d8d49** — the audit's own repair card: one cell, the missing seed-1 unquantized float32 reference at full budget (internal run log #82). Registered by the audit before running, with both of its predictions — reference within 0.01 bits of seed 0's, corrected same-seed delta inside [0.045, 0.065] — stated in advance and both confirmed. This run corrected an existing number; it added no new claim.

**Audit status: independently audited 2026-08-13. Two of the three findings — compilation and smoothness — drew AUDIT PASSES; the cost finding drew a recorded CONCERN, since repaired, and stays promoted.** The audit re-derived the promoted numbers from the raw result files with freshly written code and reviewed the code that produced them. It did not re-measure at full budget: the machine was locked by a running experiment for the whole session, and the auditor recorded that refusal rather than presenting re-derivation as re-measurement. On compilation, all ten freeze depths were *recomputed* from the raw probe rows rather than read from a stored field, and agreed 10/10; three checks the study had not run were added, all clean — determinism holds at 1.0 at every depth from k* out to the probe's limit of 20 in all ten cells, so it is not a one-depth blip; k* is genuinely the first qualifying depth, with the depth immediately below it already above 0.998 in every cell, so the ladder converges rather than oscillating; and the support behind each k* clears the study's own twenty-class bar by 81–241×, which excludes chance agreement by hundreds of orders of magnitude. On smoothness, all eight adjacent comparisons re-derived with no mismatch, every gap 26–81× the tie band, and the audit added six ladders the study does not claim: the sequence is monotone under the closed-form ridge readout at both seeds, and monotone on the untouched test split for both readouts at both seeds, which removes any reading of the smoothness as a trained-readout conditioning or split-selection artifact. The concern, and its repair, are reported in §4.1; note that the repair run itself post-dates the audit and has not been audited. One interpretive note the audit recorded and this preprint adopts: finiteness is definitional for any K-point codebook, so the empirical content of the compilation finding is the *synchronization depth* and the cost, not finiteness as such.

All claims judged strictly against the registered wording; 2 seeds × 5 codebook sizes; replicated before publication, and audited afterward, with the one number the audit qualified corrected above and the audit trail kept rather than tidied away. Internal designation: goat lattice.

---