Experimental Design: Scaling the Total Interpretation to Full enwik9

Claude and MJC

February 11, 2026

Abstract

We have completely interpreted a 128-hidden Elman RNN trained on 1024 bytes of enwik9: Boolean automaton, factor map, pattern chains, weight construction, E→N→Q quotient chain. Now the same architecture has been trained on the full 10^9-byte enwik9. Two training runs are documented: Run 1 (SGD, exploded at 125M) and Run 2 (Adam, best 2.81 bpc at 110M, catastrophic forgetting by 450M). This paper designs experiments to extend the interpretation framework to the full-scale model and makes concrete predictions about what we will find.

1 The Two Models

	sat-rnn-1024	**sat-rnn-enwik9**
Training data	1024 bytes	10^9 bytes
Epochs	∼1024	1 (online)
Best bpc	0.079	2.81
Architecture	128 hidden, 82K params	identical
Optimizer	SGD (converged)	Adam (clip=1.0)
Best checkpoint	epoch1024_final.bin	epoch1_110M.bin
Training regime	memorization	generalization

Table 1: The two models share the same architecture but live in different regimes.

The sat-rnn-1024 was trained for ∼1024 epochs on 1024 bytes—pure memorization. Every pattern in the data is perfectly learned. The model achieves 0.079 bpc (Shannon limit on this window is ∼0.067 via order-12 pattern chains).

The sat-rnn-enwik9 was trained online for one pass through enwik9. It saw each byte exactly once. Performance peaked at 110–150M bytes (2.81–2.86 bpc), then degraded due to catastrophic forgetting:

- 110M: 2.81 bpc (best)
- 300M: 3.0 bpc (slow drift)
- 400M: 3.2 bpc
- 450M: 4.5 bpc (cliff — output biases collapsed to all-negative)
- 735M: 4.5 bpc (run killed, b_y range $[-25, -7]$)

W_h standard deviation grew linearly at 0.02 per checkpoint and never stabilized. Adam's per-parameter adaptive rates accelerated forgetting: it confidently reshaped weights to fit the current distribution.

1.1 The Optimized Variant

A third run used modern techniques: Xavier init, cosine LR schedule, AdamW ($w_d = 0.01$), label smoothing ($\varepsilon = 0.10$). It reached 3.70 bpc at 20M chars before being stopped. This provides a comparison point for whether training improvements change the interpretation story.

2 What the Theory Predicts

The total interpretation of sat-rnn-1024 established seven structural facts. Each makes a specific prediction for sat-rnn-enwik9.

2.1 P1: Boolean Automaton Regime

Prediction 1 (Weakened but present)**.** *The sat-rnn-enwik9 will still operate in the Boolean automaton regime, but with reduced margins. Mean margin will drop from 60.5 to ∼5–15. Some neurons (∼10–20%) may have margins below 1.0, making the mantissa computation-relevant for those units.*

Reasoning. The tanh activation drives saturation structurally: any $|z| > 3$ yields $|\tanh(z)| > 0.995$. With W_h std growing to 3.3 (from the training log's weight statistics) and 128 inputs summed per neuron, the pre-activations will still be large. But the model has less time to push margins to extremes (1024 epochs vs 1 epoch).

Key test. If margins drop below 1.0 for >30% of neurons, the Boolean automaton model breaks down and we are in a genuinely analog regime. This would be the most important finding.

2.2 P2: Neuron Roles Change

Prediction 2 (Distributed, not concentrated)**.** *In sat-rnn-1024, neuron h28 alone captured 99.7% of the compression gap. In sat-rnn-enwik9, no single neuron will capture more than 30% of the gap. The top 15 neurons will capture ∼70–80%, not >100%.*

Reasoning. With 1024 bytes, the model overfits massively—a single neuron can memorize the output distribution. With 10^9 bytes, the model must use its capacity more evenly. 82K parameters across 128 neurons gives ∼640 effective parameters per neuron. No single neuron can carry a 10^9-byte task.

2.3 P3: Offset Structure Deepens

Prediction 3 (Shallower dominant offsets)**.** *Mean dominant offset will decrease from d=18–25 to d=5–15. The model will rely more on recent context because it cannot afford the long-range correlations that memorization allows.*

Reasoning. The BPTT-50 gradient window is the same, but with diverse data, the model must learn features that generalize. Short-range features (word boundaries, HTML tag patterns, bigram statistics) are more universal than long-range ones (specific article structure). The model will trade depth for breadth.

Alternative prediction. If the model is still fundamentally in Boolean automaton regime, the W_h diagonal carry (shift-register structure) may preserve long offsets. In that case, offsets may stay deep but become noisier.

2.4 P4: Factor Map Degrades

Prediction 4 (Lower R^2, more complex features)**.** *The 2-offset conjunction model (mean $R^2 = 0.837$ for sat-rnn-1024) will fit worse for sat-rnn-enwik9: mean $R^2 \approx 0.4$–0.6. The dominant pair (1,7) will no longer account for 52/128 neurons. Instead, multiple offset pairs will split the population more evenly.*

Reasoning. The factor map captured a model that was essentially computing "what character was 1 step ago AND what character was 7 steps ago." On 1024 bytes, this is enough because the data has limited diversity. On 10^9 bytes, neurons must compute more complex functions of the input history to be useful. The 2-offset conjunction is a lower bound on the computation; the real model may be doing 3- or 4-offset conjunctions, or genuinely nonlinear combinations.

2.5 P5: Weight Construction Gap Widens

Prediction 5 (Analytic construction degrades, optimization compensates)**.** *The fully analytic construction (shift-register + log-ratio W_y, 1.89 bpc on 1024 bytes) will yield >5 bpc on enwik9. But the optimized readout (shift-register dynamics + gradient-optimized W_y, 0.59 bpc on 1024 bytes) will still achieve ∼3–4 bpc on enwik9, competitive with the trained model's 2.81.*

Reasoning. The analytic W_y comes from skip-bigram log-ratios. On 1024 bytes, these statistics are complete (you can count every bigram exhaustively). On 10^9 bytes, the statistics are accurate but the readout must compress a much larger conditional distribution into 128×256 weights. The shift-register dynamics, however, are data-independent—they are a hash function. So the dynamics should transfer, and only the readout needs retraining.

2.6 P6: Hebbian Covariance Improves

Prediction 6 (Higher correlation with more data)**.** *Hebbian covariance $cov(h_j(t), h_i(t+1))$ will correlate with trained W_h at $r > 0.56$ (the value for sat-rnn-1024). Estimate: $r \approx 0.65$–0.75.*

Reasoning. Hebbian covariance captures the data's temporal structure. On 1024 bytes, the covariance is computed from a tiny sample; on 10^9 bytes, it averages over massive statistics and should give a cleaner signal. The trained W_h at the 110M checkpoint has only had one pass of data—less time to develop non-Hebbian structure. The gap between Hebbian and trained should shrink.

Caveat. If the model is in catastrophic-forgetting mode, the covariance over the full 10^9 bytes may not match the W_h at any single checkpoint, because the data distribution drifts. We should compute Hebbian covariance over the *first* 110M bytes and compare to the 110M checkpoint.

2.7 P7: E→N→Q Framework Holds Structurally

Prediction 7 (Same framework, different numbers)**.** *The $E \to N \to Q$ quotient chain applies identically—it is structural, not data-dependent. The quotient values change (larger Q at each layer because the model is compressing less), but the decomposition $Q_{total} = \prod Q_{layer}$ still holds.*

Reasoning. The quotient chain is a mathematical property of the UM forward pass, not of the training data. $E \to N$ (counting) and $N \to Q$ (dividing by count) are the same operations regardless of the data source. What changes is the *magnitude* of quotients: on 1024 bytes, the model compresses to 0.079 bpc (high quotients); on 10^9 bytes, it compresses to 2.81 bpc (quotients closer to 1).

3 Experimental Program

3.1 Phase 1: Characterize the 110M Checkpoint

The best checkpoint (epoch1_110M.bin, 2.81 bpc) is the primary target. Run the existing Q1–Q7 analysis tools on it.

1. **Q1: Boolean automaton.** Run q1_boolean.c, q1_margins.c on the 110M model. Measure margin distribution, sign vs. mantissa compression, bit leverage. Key question: is it still a Boolean automaton?
2. **Q2: Offsets.** Run q2_offsets.c. Measure per-neuron dominant offset, MI profile. Test P3 (shallower offsets?).
3. **Q3: Neuron knockout.** Run q3_neurons.c. Rank neurons by compression contribution. Test P2 (distributed vs. concentrated).
4. **Q4: Saturation dynamics.** Run q4_saturation.c. Measure dwell times, co-flip clusters. Compare to sat-rnn-1024 (mean dwell 3.3).
5. **Q5: Redux.** Run q5_redux.c. Can we find a sparse sub-network? How many neurons are needed for 90% of the gap?
6. **Q6: Justifications.** Run q6_justify.c on specific enwik9 positions. What does the backward attribution chain look like on real Wikipedia text?
7. **Q7: Algebraic structure.** Run q7_algebraic.c, q7_higher_order.c. Does PMI alignment improve or degrade with more data?

Eval data. All measurements should be taken on a held-out window: bytes 110M–111M (the first 1M bytes the model has never trained on at this checkpoint). This gives a clean generalization measure.

3.2 Phase 2: Factor Map on enwik9

Run the factor map analysis (factor_map.c and variants) on the 110M checkpoint:

1. Per-neuron best 2-offset conjunction: does R^2 drop as predicted?
2. Are there neurons where 3-offset conjunctions significantly improve fit?
3. Does the word_len feature still dominate? On real Wikipedia, word boundaries are more regular than on the 1024-byte XML fragment.
4. What is the in_tag feature's role? On full enwik9, tag density varies.

3.3 Phase 3: Weight Construction Transfer

Test whether the weight construction framework transfers:

1. **Compute skip-bigram statistics** on the first 110M bytes. Build the analytic W_y from log-ratios.
2. **Shift-register dynamics + analytic** W_y: what bpc? (Prediction: >5 bpc.)
3. **Shift-register + optimized** W_y: train the readout on the 110M prefix. (Prediction: 3–4 bpc, competitive with trained.)

4. **Hebbian W_h**: compute covariance on first 110M bytes. Compare to trained W_h at 110M checkpoint. (Prediction: $r > 0.56$.)

5. **Full analytic model on enwik9**: all weights from data statistics, zero optimization. How does it compare?

3.4 Phase 4: Checkpoint Trajectory

The training produced checkpoints at 100M, 110M, 120M, ..., 990M. This is a unique opportunity to study the interpretation as a *function of training progress.*

1. **Margin trajectory.** Plot mean margin vs. training byte offset. Does the Boolean automaton regime strengthen or weaken over training?

2. **Neuron role stability.** Do the same neurons dominate at 110M and 200M? At what point do roles destabilize?

3. **Forgetting signature.** The cliff at 450M (bias collapse) is visible in the training log. What happens to the Boolean automaton structure at that transition? Is there a detectable precursor?

4. W_h **drift.** Std grew at 0.02/checkpoint. Does the Hebbian correlation change monotonically, or does it peak and decay?

4 Key Questions Beyond Q1–Q7

The full-enwik9 model opens questions that didn't arise on 1024 bytes.

Question 1 (Generalization vs. Memorization)**.** *The sat-rnn-1024 memorizes: it sees each byte ∼1024 times. The sat-rnn-enwik9 generalizes (or fails to): it sees each byte once. Do the weights encode* rules *(generalizable patterns) or* instances *(memorized fragments)? The Hebbian/analytic construction tests this directly: if the analytic model is competitive, the learned rules are simple enough to derive from statistics.*

Question 2 (Capacity Saturation)**.** *82K parameters, 128 hidden units. How many bytes of enwik9 can this architecture actually learn? The training log suggests ∼110M (where bpc is minimized). Is this a fundamental capacity limit, or is it an artifact of online training? Multi-epoch training on 110M bytes should answer this.*

Question 3 (Phase Transition at 450M)**.** *The cliff at 450M bytes—output biases collapsing to all-negative—looks like a phase transition. Is there an E→N→Q description of this collapse? The bias term in the quotient chain is $Q_{bias} = e^{b_y}/Z$. When all $b_y < 0$, the quotient becomes exponentially suppressive. Can we predict when this happens from the trajectory of W_h drift?*

Question 4 (Content-Dependent Performance)**.** *enwik9 contains XML markup, English prose, tables, references, multilingual text, mathematical notation. Which content types does the model handle well at 110M? The eval window matters: bytes 0–110M of enwik9 are early Wikipedia articles. The model's 2.81 bpc likely varies by content type. Per-segment bpc analysis would reveal the model's strengths.*

Question 5 (The Mantissa Question)**.** *On sat-rnn-1024, the mantissa was noise: sign-only dynamics gave* better *bpc (5.690 vs. 5.721). On sat-rnn-enwik9, with potentially reduced margins, does the mantissa carry information? If margins drop below 1.0 for some neurons, the mantissa becomes part of the computation. This would be the first case where the model is* not *a Boolean automaton, and the full f32 state space matters.*

5 Predicted Outcome Summary

Prediction	Metric	1024B	enwik9 (expected)
P1: Boolean regime	Mean margin	60.5	5–15
P2: Distributed neurons	Top-1 neuron % gap	99.7%	<30%
P3: Shallower offsets	Mean dominant offset	18–25	5–15
P4: Factor map degrades	Mean R^2 (2-offset)	0.837	0.4–0.6
P5: Construction gap	Analytic bpc	1.89	>5
P6: Hebbian improves	$r(H_{\text{cov}}, W_h)$	0.56	0.65–0.75
P7: E→N→Q holds	Framework	valid	valid

Table 2: Predicted outcomes. Each row is a testable claim.

6 What Would Be Surprising

1. **Margins stay large (>30).** This would mean the Boolean automaton regime is a structural consequence of the architecture, not of overtraining. It would validate the claim that tanh RNNs are *inherently* Boolean.

2. **Factor map R^2 stays high (>0.7).** This would mean the 2-offset conjunction is the fundamental computation, not a memorization artifact. It would suggest that RNNs converge to this computational motif regardless of data.

3. **Hebbian correlation drops (<0.4).** This would mean the trained model has developed genuinely non-Hebbian structure—higher-order correlations that simple covariance cannot capture.

4. **Phase transition is sharp and predictable.** If the 450M cliff has a clean description in the quotient framework, it would be a genuine contribution to understanding catastrophic forgetting in RNNs.

5. **Content-dependent bpc varies by >3x.** If the model gets 0.5 bpc on XML tags but 6 bpc on multilingual text, the "total interpretation" story is really "total interpretation of XML parsing."

7 Experimental Priority

The experiments above are ordered by information value. The single most important experiment is:

Run q1_margins.c on epoch1_110M.bin.

If margins are large, the Boolean automaton interpretation transfers directly, and the rest of the Q1–Q7 suite follows. If margins are small, the interpretation must be rebuilt from scratch in the analog domain. This one measurement determines the entire research direction.

The second most important experiment is the factor map (R^2 of 2-offset conjunctions). If both margins and factor map transfer, then the total interpretation scales. If either breaks, we learn something new about what these models compute.