$H = 2^{32}$: The f32 State Space of the Sat-RNN

Claude and MJC

11 February 2026

Abstract

We set $H = 2^{32}$ per neuron and declare everything else—sign, exponent, mantissa, IEEE 754 itself—a factorization. The weight matrices define an immutable map on H^{128}. The tanh activation constrains the reachable subset to $\sim$26 bits per neuron. Training selected this particular map from the ϵ-field of random initialization. We measure the map's dynamical structure by sampling: flipping individual bits, measuring forward and backward leverage, and identifying which factorization—hardware (exp/mant) or dynamical (stable/unstable)—correctly captures the information flow.

1 The State Space

Definition 1 ($H = 2^{32}$)**.** *Each neuron $j \in \{0, \ldots, 127\}$ has state $h_j \in H$ where H is the set of all 2^{32} possible 32-bit patterns. The full hidden state is $\mathbf{h} \in H^{128}$, a point in a 2^{4096}-element space.*

The tanh activation constrains the reachable subset: h_j must lie in $(-1, +1)$, which in f32 corresponds to exponents $E \in \{0, 1, \ldots, 126\}$ and the sign bit. The exponent $E = 127$ encodes values in $[1, 2)$, so $h_j = \pm 1.0$ is the boundary (exactly $\tanh(\pm\infty)$).

Actual state at $t = 42$. Of 128 neurons:

- 112 have $E = 127$ ($|h| = 1.0$ exactly): the saturated majority.
- 16 have $E = 126$ ($|h| \in [0.5, 1)$): the unsaturated minority.
- Only 17 unique f32 bit patterns among the 128 neurons.
- Sign: 68 positive, 60 negative.

The saturated neurons carry exactly 1 bit of state each (the sign). The unsaturated neurons carry $\sim$24 bits each (1 sign + 23 mantissa; the exponent is fixed at 126). Total effective state: $112 \times 1 + 16 \times 24 = 496$ bits—not 4096 bits and not 128 bits, but 496.

2 The Immutable Map

The weight matrices $(W_x, W_h, b_h, W_y, b_y)$ define a fixed map $\phi : H^{128} \times \{0, \ldots, 255\} \to H^{128}$:

$$\phi(\mathbf{h}, x) = \tanh(W_h \mathbf{h} + W_x e_x + b_h)$$

where e_x is the one-hot encoding of input byte x, and tanh is applied elementwise in f32 arithmetic (including rounding).

This map is deterministic and immutable: given the same state and input, it always produces the same next state. The entire behavior of the RNN is encoded in this map. Training selected this particular map (these particular weight values) from the ϵ-field of random initialization.

The map as permutation. On saturated neurons ($|h_j| = 1.0$), the map acts primarily on signs. The pre-activation $z_j = [W_h \mathbf{h} + W_x e_x + b_h]_j$ is a linear combination of $\pm W_h[j,k]$ values (the unsaturated neurons contribute intermediate values). If $|z_j| \gg 1$, then $\tanh(z_j) \approx \pm 1$ and the sign of z_j determines the sign of h'_j. This is a Boolean function: the next sign vector is a function of the current sign vector and the input.

The 16 unsaturated neurons are where the mantissa matters. Their pre-activations z_j are in the quasi-linear regime of tanh, where mantissa bits of $\mathbf{h}$ propagate to mantissa bits of $\mathbf{h}'$.

3 Two Factorizations

Every element of $H = 2^{32}$ has 32 individually addressable bits. We can factor these bits in two ways:

3.1 Hardware factorization (IEEE 754)

Channel	Bits	Position	Meaning
Sign	1	31	$\pm$
Exponent	8	30–23	scale (2^{E-127})
Mantissa	23	22–0	precision ($1.m \times 2^{E-127}$)

This factorization is fixed by the hardware. It determines how arithmetic operations act on the bits: addition aligns exponents, multiplication adds exponents and multiplies mantissas, etc.

3.2 Dynamical factorization (stable/unstable)

For each bit position $b \in \{0, \dots, 31\}$ and neuron j, flip bit b in h_j and measure the effect on the output distribution (KL divergence). A bit is *dynamically stable* if flipping it has negligible effect (KL $< 10^{-6}$). It is *dynamically active* otherwise.

From our sampling experiment at $t = 42$:

Bit range	Channel	Mean KL (bits)	Dynamical role
0–4	mantissa (low)	$< 10^{-6}$	dead
5–14	mantissa (mid)	$< 10^{-4}$	dormant
15–22	mantissa (high)	0.00044/bit	active memory
23–29	exponent	0.0080/bit	importance
31	sign	0.046	topology

Observation 1 (Hardware and dynamical factorizations align at one step)**.** *At $t = 42$ (single-step leverage), the two factorizations nearly coincide: sign is the most active, exponent next, mantissa lowest. The hierarchy is $300 : 52 : 1$ per bit (sign : exp : mant).*

Observation 2 (They diverge through the Jacobian)**.** *At BPTT depth $d = 10$: f32 and exact gradients have $\cos \approx 0$ (uncorrelated), but pattern attributions have $\rho \approx 1.0$ (perfect rank agreement). The f32 error is proportional to the signal, not orthogonal to it. The dynamical factorization (which patterns survive through the Jacobian) is not the hardware factorization (which bits are sign/exponent/mantissa).*

4 The Mantissa as Memory

The mantissa looks overengineered for a single prediction: bits 0–14 contribute $< 10^{-4}$ to KL. But it is load-bearing because it is the memory.

Forward propagation. Each forward step maps $\mathbf{h}_t \to \mathbf{h}_{t+1}$ through the immutable map ϕ. The Jacobian of ϕ with respect to $\mathbf{h}$ is

$$J = \mathrm{diag}(1 - h^2) \cdot W_h$$

For saturated neurons ($|h_j| \approx 1$), the gate $(1-h_j^2)$ is $\sim 10^{-3}$ to 10^{-60}, attenuating their contribution. For unsaturated neurons, the gate is $\sim$ 0.1–0.5.

The 16 unsaturated neurons are the "active memory": their mantissa bits propagate through W_h to all 128 neurons at the next step. The 112 saturated neurons are the "settled memory": their signs are fixed, their mantissa is dead ($\tanh(30) - \tanh(29) \approx 10^{-25}$).

The 496-bit state. The effective state at any time t is:

- 112 sign bits (one per saturated neuron): the Boolean state.
- 16 mantissa values ($\sim$24 bits each): the analog state.
- The exponent is nearly constant (126 or 127).

The sign bits evolve as a Boolean function of themselves, the analog state, and the input. The analog state evolves through the 16-neuron Jacobian subblock. This is a mixed Boolean-analog dynamical system.

Bit propagation. Flipping bit 31 (sign) of unsaturated neuron 8 ($h = 0.855$) at $t = 42$ causes 0.729 bits KL at $t = 46$ (amplification). Flipping bit 22 (MSB mantissa) causes 0.0004 bits KL at $t = 43$ (decay). Flipping bit 0 (LSB) causes zero at all future times.

The sign bit propagates and amplifies because it changes the sign of $W_h[j, \cdot]$ contributions to all neurons at the next step—a global perturbation. The mantissa bits propagate locally through the analog channel and attenuate through the saturation gates.

5 Entropy Budget

The total compression by the sat-rnn is 5.72 bpc $\times 1023$ positions $= 5,852$ bits (f32) or $5.65 \times 1023 = 5,780$ bits (exact). The difference—72 bits—is the cost of the f32 quotient.

Where does the compression live? In the 496-bit state. Each position, the RNN reads 8 bits of input, updates the 496-bit state, and produces a probability distribution over 256 output bytes. The compression comes from the state carrying information about past inputs that helps predict future outputs.

The 112 sign bits carry the long-range memory (which patterns have been observed). The 384 mantissa bits (16×24) carry the short-range precision (how recently, how strongly). The sign bits change rarely (a neuron unsaturating and resaturating with a different sign is a rare event). The mantissa bits change continuously.

6 Experimental Results

6.1 State statistics across all positions

Metric	Value
Mean saturated neurons ($\|h\| \geq 0.999$)	123.0 / 128
Mean unsaturated	5.0 / 128
Mean sign changes per step	31.6
Mean bpc (full state)	5.721
Mean bpc (sign-only: $h_j \to \mathrm{sgn}(h_j)$)	5.728
Mean bpc (zero-mantissa: keep sign+exponent)	5.637

Observation 3 (The sign carries 99.7% of the compression)**.** *The total compression gap from uniform (8.0 bpc) to full prediction (5.72 bpc) is 2.28 bpc. The sign-only model achieves 5.73 bpc: a gap of 2.27 bpc, or 99.7% of the total. The mantissa contributes 0.007 bpc—negligible for prediction.*

Observation 4 (Zeroing the mantissa *improves* bpc)**.** *The zero-mantissa model (keep sign and exponent, set mantissa to zero) achieves* ***5.637 bpc****—0.084 bpc better than the full model. The mantissa adds noise to the prediction, not signal. The W_y readout was trained with mantissa present but cannot extract useful information from it; the mantissa values interfere with the sign-based prediction.*

Observation 5 (31.6 sign changes per step)**.** *About 25% of the 128 neurons flip sign at each position. This is the information throughput of the Boolean channel:* 31.6 *bits of new information per input byte. The sign vector is not static—it is a rapidly evolving Boolean state that encodes the current context.*

6.2 Effective state size

With 123 saturated neurons (∼1 bit each, the sign) and 5 unsaturated neurons (∼24 bits each), the effective state is:

$$123 \times 1 + 5 \times 24 = 243 \text{ bits}$$

But the sign-only model achieves 99.7% of the compression, so the analog bits contribute only 0.007 bpc × 1023 positions = 7 bits total. The effective state for prediction is **128 bits**—one sign bit per neuron.

Training found the right map. From the ϵ-field of random initialization ($\sim$ 330,000 f32 weights $\approx$ 10 million bits of random state), training selected a map where:

1. 123 neurons saturate (sign-only memory, stable).
2. 5 neurons stay unsaturated (analog memory, volatile).
3. The sign vector encodes ∼2.27 bpc of compression (99.7%).
4. The mantissa encodes ∼0.007 bpc (0.3%)—or less (zeroing it *improves* bpc by 0.08).
5. 31.6 sign bits change per step: the Boolean state is dynamic.

6.3 Mantissa ablation: dynamics vs readout

We run five variants of the RNN and measure mean bpc:

Mode	bpc	Description
full f32	5.721	baseline
sign readout only	5.728	full dynamics, snap $h \to \pm 1$ for W_y
sign dynamics	**5.690**	snap $h \to \pm 1$ after each step
zero-mant readout	5.637	full dynamics, zero mantissa for W_y
zero-mant dynamics	**5.582**	zero mantissa after each step

Observation 6 (The mantissa is noise)**.** *Every variant that removes the mantissa from* dynamics *outperforms full f32. Zero-mantissa dynamics (keep sign and exponent only) is 0.14 bpc better. Sign-only dynamics (a pure 128-bit Boolean automaton) is 0.03 bpc better. The mantissa actively degrades both prediction and dynamics.*

Observation 7 (A different trajectory, a better result)**.** *Sign-only dynamics has 52.2% sign agreement with full f32 after 100 steps—barely above chance. It is on a completely different trajectory through state space. Yet it compresses better. The weights encode a good Boolean function that is* obscured *by mantissa noise in the f32 computation.*

Observation 8 (The mantissa is for training, not inference)**.** *The mantissa enables gradient flow during training: BPTT needs mantissa precision in the Jacobian to compute weight updates (even though the resulting gradients are noise, as Program 4 showed). Without the mantissa, there is no gradient. But at inference time, the mantissa is pure overhead—the Boolean dynamics works better.*

Mantissa bit sweep. Quantizing the mantissa to various bit levels during dynamics:

Mantissa bits	bpc	Δ from full
0 (zero-mant)	5.582	−0.139
1	5.592	−0.129
2	5.696	−0.025
4	5.636	−0.085
8	5.740	+0.019
12	5.621	−0.100
16	5.682	−0.039
20	5.614	−0.107
23 (full)	5.721	0

Every level except 8 bits improves on full f32. The pattern is noisy rather than monotonic, but the trend is clear: fewer mantissa bits $\to$ better bpc. The mantissa is not a graded resource—it is interference.

7 Revised Picture

The sat-rnn is a 128-bit Boolean automaton with a 2-value exponent channel (saturated vs unsaturated). The state is:

- 128 sign bits: the computation.
- ∼5 exponent bits (which neurons are unsaturated): the routing, telling the dynamics which neurons are "active."
- 23 mantissa bits per neuron: noise, needed for training (gradient flow) but harmful for inference.

The weight matrices encode the Boolean transition function $\sigma_{t+1} = f(\sigma_t, x_t)$ where σ is the sign vector and x is the input byte. The exponent channel modulates which neurons participate in the transition (unsaturated neurons are "open gates"). The mantissa is the price paid for differentiable training of a Boolean function.