The Factor Map

Claude and MJC

9 February 2026

Abstract

We construct the factor map ϕ from interpretable features to hidden-state dynamics for a saturated RNN (128 hidden, 0.079 bpc on 1024 bytes of enwik9). Where the previous paper identified per-neuron correlations with 2-offset conjunctions, here we take the dynamics of H as the object of analysis. Word length is distributed across all 128 neurons (max per-neuron $r = 0.014$) but lives in a coherent direction ($r = 0.58$). Space is a massive reset signal ($\|\Delta h\| = 5.31$). The key finding: interpretable features are *entangled* in the dynamics—removing the word-length direction from h at each step costs +7.3 bpc (catastrophic), while the same removal post-hoc costs only +0.15 bpc. The RNN does not have separable circuits for word length and tag state; it has one dynamical system that simultaneously tracks everything through W_h.

1 Introduction

The pattern-chains paper [**?**] showed that all 128 neurons of the sat-rnn can be explained as 2-offset conjunction detectors, with word length as the dominant state feature. But explaining neurons individually is not a factor map.

A factor map $\phi : A \to B$ shows how the factors (components) of one factorization map onto the factors of another. In our case:

- **Factorization A** (architecture-natural): the 128-dimensional hidden state $h \in \text{f32}^{128}$, evolving under $h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h)$.
- **Factorization B** (domain-natural): interpretable features — word length, in-tag state, UM skip-pattern activations — each with known dynamics and known predictive value.

The factor map tells us: "this is how word length maps onto the H dynamics; this is how the RNN learned it; this is how we can add an ES and LPP onto our UM to capture it as log counts and then write into H what we previously saw that it learned" [**?**].

1.1 What we have

From previous work:

1. The sat-rnn: 128 hidden, tanh, 0.079 bpc on DSS=1024.
2. UM superset: greedy-6 offsets give 0.000 bpc (perfect). Every RNN offset pair has 350–600 UM patterns.
3. Per-neuron factor map: 2-offset pairs explain $R^2 \geq 0.80$ for 120/128 neurons. Adding word_len reaches 0.50 bpc (91%).
4. Reverse isomorphism: log-prob encoding achieves 0.107 bpc with only W_y optimization.

1.2 What we need

The dynamics of H as a whole object:

1. How word length propagates through W_h (not just correlation with individual neurons, but the mechanism).
2. How the space character acts as a timing signal that prevents runaway despite $\|W_h\| > 1$.
3. The eigenstructure of W_h and how interpretable features align with eigenvectors.
4. How to write a pattern into H and verify the model still works.
5. How to subtract a pattern out of H and measure what remains.

2 Word Length in the Hidden Dynamics

2.1 Word length is distributed, not per-neuron

Word length is the best single state predictor for all 128 neurons, yet no individual neuron correlates strongly with it. The maximum per-neuron correlation with word length is $r = 0.014$ (h112), and the mean is $|r| = 0.004$. Zero neurons exceed $|r| > 0.3$.

The signal lives in a *direction* spanning many neurons. The covariance direction v_{wl} (the direction in f32^{128} that maximizes correlation with word length) achieves $r = 0.58$, $R^2 = 0.34$. This is the word-length direction in H.

2.2 Conditional means reveal the structure

The conditional mean vectors $E[h \mid \text{wl} = k]$ for $k = 0, \ldots, 15$ show how word length organizes the hidden state:

wl	n	$\|E[h\|\text{wl}]\|$	dist from wl=0
0	146	5.35	0.00
1	45	5.38	4.60
2	45	4.69	6.17
3	44	5.05	6.49
4	44	5.40	7.84
5	44	4.88	7.93
6	44	5.27	7.46
7	44	5.06	6.95
8	43	4.83	6.32
9	43	4.98	6.11
10	43	3.89	6.08

The distance from the wl=0 mean grows rapidly for the first 4–5 characters (reaching 7.9), then plateaus and slightly decreases. The RNN has learned a counter that saturates around word length 5—matching the typical word length in enwik9.

PCA of these conditional means finds three principal components ($\lambda_1 = 48.9$, $\lambda_2 = 43.4$, $\lambda_3 = 37.6$) that capture 37.6%, 33.4%, and 28.9% of the between-group variance. PC1 correlates only $r = 0.31$ with word length; the word-length signal is spread across multiple principal directions, confirming it is distributed.

2.3 Space as timing/reset signal

The space character is a massive reset signal:

$$\|E[h \mid \text{after\_space}] - E[h \mid \text{other}]\| = 5.31$$

The top movers after space are h80 (+1.15), h112 (−1.09), h74 (−1.05). After space, the hidden state distance from the after-space mean grows from 6.4 (wl=0) to ∼10 (wl=5), then plateaus at ∼9.5 for longer words.

The space column of W_x has norm 3.74 (second-largest after '¡' at 3.75). Its projection onto the word-length direction is −0.57 (cos = −0.15), confirming that space pushes h in the negative word-length direction. The '¡' column pushes in the positive direction (+0.50), consistent with its role in starting tag contexts.

3 Eigenstructure of W_h

The top singular values of W_h via power iteration:

rank	σ	corr(wl)	corr(tag)	top neurons
0	5.51	+0.13	−0.01	h20, h106, h59
1	5.06	+0.23	−0.10	h76, h112, h61
2	4.87	−0.21	−0.05	h8, h50, h112
3	4.63	+0.10	+0.37	h99, h53, h109
4	4.32	+0.17	+0.39	h53, h56, h61
5	3.91	+0.01	+0.20	h16, h90, h58
6	3.78	+0.36	+0.03	h90, h68, h112
7	3.45	+0.23	−0.05	h56, h16, h15
8	3.15	+0.01	+0.25	h94, h26, h46
9	3.07	−0.13	−0.26	h102, h17, h68

No single singular vector captures word length or tag state strongly. SV6 ($\sigma = 3.78$) has the highest word-length correlation ($r = 0.36$). SV3–4 ($\sigma = 4.3$–4.6) have the highest in-tag correlations ($r = 0.37$–0.39). The eigenstructure *partially* aligns with interpretable features, but neither word length nor tag state is an eigenvector of W_h.

3.1 Propagation through W_h

The word-length PC1 direction is amplified by W_h: $\|W_h v_{wl}\| = 2.48$ with self-alignment $\cos(W_h v_{wl}, v_{wl}) = 0.79$. This means W_h approximately preserves the word-length direction with 2.5× amplification. 92% of the amplified vector stays within the 3-PC word-length subspace; only 8% leaks orthogonally.

Multi-step propagation shows exponential growth: $\|W_h^k v_{wl}\|$ reaches 2831 at $k = 8$, confirming the spectral radius exceeds 1 in this direction. The self-alignment oscillates (0.79, 0.47, −0.05, −0.38, −0.48, ...), showing the direction rotates under repeated W_h application.

4 The Entanglement Problem

4.1 Two directions, high overlap

The word-length direction v_{wl} (covariance-based, $r = 0.58$) and the tag-state direction v_{tag} (also covariance-based, $r = 0.57$) have cosine similarity 0.50. They overlap substantially.

The tag direction v_{tag} captures $\|E[h|\text{in\_tag}] - E[h|\text{out\_tag}]\| = 2.77$, smaller than the space reset (5.31).

4.2 Step-by-step subtraction is catastrophic

We compare two subtraction methods:

Intervention	bpc	Δ bpc
Baseline	0.079	—
Post-hoc (modify h after forward pass):		
Remove v_{wl}	0.225	+0.146
Remove $v_{wl} + v_{\text{tag}}^{\perp}$	0.743	+0.664
Step-by-step (remove at each step, propagate):		
Remove v_{wl}	7.357	+7.278
Remove $v_{\text{tag}}^{\perp}$	6.363	+6.284
Remove both	7.474	+7.395
Random direction (mean of 5)	2.310	+2.231
Step-by-step write-in (replace with oracle):		
Oracle word length	5.700	+5.621
Oracle wl + tag	5.877	+5.798

The contrast is stark:

- **Post-hoc**: removing v_{wl} costs +0.15 bpc (mild). This measures the *information content* of word length for prediction.
- **Step-by-step**: removing v_{wl} costs +7.3 bpc (catastrophic). This measures the *dynamical importance* of the direction for the entire forward pass.
- **Random control**: even a random direction costs +2.3 bpc when removed step-by-step. The word-length direction is 3× worse than random, making it the most dynamically important single direction.
- **Write-in**: replacing the word-length projection with oracle values also catastrophically disrupts the dynamics (+5.6 bpc).

4.3 Interpretation: entangled dynamics

The RNN does not have separable circuits. It has ONE dynamical system (W_h) that simultaneously tracks all features. The direction v_{wl} carries word length ($r = 0.58$) but also carries other information critical to the forward pass. You cannot "subtract out" word length without destroying the model, because the same direction serves multiple purposes.

After step-by-step removal of v_{wl}, word-length information partially regenerates: the rebuilt covariance direction still achieves $r = 0.30$ with word length. This is because W_x reintroduces word-length information at every step (the RNN sees spaces, which carry timing).

4.4 Weight-level subtraction

Projecting the word-length direction out of W_h itself is even more destructive:

Remove from W_h	bpc	Δ
1 wl PC	5.97	+5.89
2 wl PCs	7.54	+7.46
3 wl PCs	7.36	+7.28
tag direction	6.32	+6.25
3 wl PCs + tag	7.33	+7.25

Removing even one PC from W_h degrades the model to 5.97 bpc. Removing from both W_h and W_x is equally catastrophic (5.7–8.1 bpc). The word-length principal components are load-bearing directions of W_h.

5 The Factor Map Is One-Way

These results reveal an asymmetry:

1. $\phi : H \to$ features works well. Given h_t, we can read off word length ($r = 0.58$), tag state ($r = 0.57$), and 2-offset conjunctions ($R^2 \geq 0.80$ for 120/128 neurons). This is the per-neuron factor map from the previous paper.

2. ϕ^{-1} : features $\to H$ does not work cleanly. Writing interpretable features into H or removing them from H disrupts the dynamics because the directions are entangled.

The factor map is *readable but not writable*. We can observe where features live in H, but we cannot surgically modify them without side effects.

This is expected for an overparameterized model on a small dataset. The 128-dimensional hidden state has far more capacity than needed for DSS=1024, so the optimization finds a solution where features share directions for efficiency, not separability.

6 Building a UM Around the RNN

Given that ϕ^{-1} is destructive, the right approach is not to modify the RNN but to *wrap* it:

1. Run the RNN forward pass as-is.

2. At each position, read off interpretable features via ϕ: word length, tag state, character identity.

3. Feed these features into a UM that computes log-support for each output character.

4. Combine the UM's prediction with the RNN's prediction (e.g., log-linear interpolation).

The UM captures the interpretable part of the RNN's computation. What the UM explains, we understand (and know will generalize). The residual is what the RNN does beyond the interpretable features.

SN form for a pattern:

```
"The input is 'm'."
 "The word len is 3."
 "The output is 'e'." 7
```

Meaning: given this conjunction of events, output 'e' is predicted with support equivalent to 2^7 observations.

7 Discussion

The factor map $\phi : H \to$ features is a one-way projection. Interpretable features explain most of the per-neuron variance (91% of bpc gain with 2-offset + word_len + in_tag), but these features are entangled in the dynamics of W_h.

The practical consequence: interpretability of the RNN proceeds by *reading* the hidden state, not by *modifying* it. This is compatible with the CMP framework [**?**], where the UM wraps the model rather than replacing parts of it.

The entanglement also explains why the reverse isomorphism (0.107 bpc) works: it encodes UM log-probabilities into H and optimizes only W_y. It does not try to match the RNN's internal dynamics; it bypasses them entirely, using H purely as a feature vector for the output layer.

Open questions:

- Does entanglement decrease with model size? A larger model might develop more separable representations.
- Can we find a *rotation* of H where features become axis-aligned? (ICA or sparse coding approaches.)
- The UM-wrapping approach: what residual bpc remains after subtracting the UM's contribution?

Reproducibility

Repository: `https://github.com/inimino/hutter` (commit: TBD)

Model: sat-rnn, 128 hidden, tanh, 0.079 bpc. Checkpoint: `sat_model.bin` (329 KB).

Data: First 1024 bytes of enwik9.

Tools:

- `factor_map2.c` — per-neuron correlation, space reset, SVD, subtraction sweep
- `factor_map3.c` — conditional means, PCA of wl subspace, W_h propagation, weight-level subtraction
- `factor_map4.c` — step-by-step intervention experiments, write-in, subtract-out, random control

References

[1] Michaeljohn Clement. CMP. 2026. `https://cmpr.ai/cmp.pdf`

[2] Claude and MJC. Pattern Chains. 8 Feb 2026.

[3] Claude and MJC. The Export Gap. 7 Feb 2026.