Sparse Differentiation

Claude and MJC

9 February 2026

Abstract

The previous paper showed that the factor map $\phi : H \to$ features is readable but not writable: interpretable features are entangled in the RNN's dynamics. Here we take a different approach. Instead of asking "where does word length live in H?" (aggregate), we ask "for this specific prediction at position t, which patterns contributed and where did the signal come from?" (analytic). We trace individual pattern signals backwards through the RNN via sparse differentiation: BPTT-style gradient computation applied not for training but for *attribution*. The gradient is moderately sparse (top-10 neurons capture 68% of energy) and—critically—*grows* rather than decays backwards (2.4× at $k = 8$), explaining why the RNN can learn from skip-8 offsets.

1 Introduction

The factor map paper [**?**] established that aggregate geometric analysis (eigenstructure, PCA, correlation) finds features that are entangled in the dynamics. This is expected: the RNN is not a physical system with natural coordinates but "a high-dimensional pattern space shoved through an arbitrary wrong-dimensioned function-learning process."

The right approach is analytic, not aggregate:

1. Take a specific prediction: at position t, the RNN predicts $P(x_{t+1} \mid h_t)$.
2. Compute $g_t = \nabla_{h_t} \log P(x_{t+1})$—the gradient of the correct prediction with respect to the hidden state.
3. Trace backwards: g_t through the Jacobian J_s at each step, to find where the signal originated.
4. At each upstream position $t-k$, compute the attribution: how much did $W_x[:, x_{t-k}]$ contribute to the prediction?

This is BPTT without error terms. The gradients tell us which neurons carry the signal for each prediction, and backpropagating through W_h tells us which past positions matter.

2 The Gradient at Each Position

At position t, the RNN's prediction is:

$$P(o \mid h_t) = \text{softmax}(W_y h_t + b_y)_o$$

The gradient of the log-probability of the correct output $y = x_{t+1}$:

$$g_t = \nabla_{h_t} \log P(y \mid h_t) = W_y^T (e_y - P)$$

where e_y is the one-hot vector for y and P is the full probability vector. This is a 128-dimensional vector telling us how each hidden neuron contributes to the correct prediction.

2.1 Gradient sparsity

The gradient is moderately sparse:

Measure	Value
Mean $\|g_t\|$	0.103
Top-1 neuron captures	19.4% of $\|g_t\|^2$
Top-5 neurons capture	50.8%
Top-10 neurons capture	67.9%

The gradient norms are small (~ 0.1) because the model is nearly perfectly trained (0.079 bpc, probabilities > 0.99 for most predictions). The same neurons dominate across positions: h68, h76, h99, h8—matching the importance ranking from the per-neuron factor map.

3 Backward Tracing via the Jacobian

The Jacobian of the hidden state:

$$J_t = \frac{\partial h_t}{\partial h_{t-1}} = \text{diag}(1 - h_t^2) \cdot W_h$$

The term $(1 - h_t^2)$ is the derivative of tanh at h_t. For saturated neurons ($|h_t| \approx 1$), this is near zero: the gradient is killed. For unsaturated neurons, it passes through.

The backward gradient from position t to position $t - k$:

$$g_{t \to t-k} = \left(\prod_{s=t-k+1}^{t} J_s \right)^T g_t$$

3.1 The gradient grows, not decays

The key finding: the backward gradient norm *grows* rather than decays:

k	mean $\|g_{t \to t-k}\|$	ratio to $k = 0$
0	0.103	1.00
1	0.143	1.38
2	0.154	1.48
3	0.167	1.61
4	0.181	1.74
5	0.199	1.92
6	0.214	2.05
7	0.229	2.20
8	0.246	2.36

This is because the spectral radius of W_h exceeds 1. The Jacobian product amplifies the gradient at each step. The same mechanism that makes BPTT-50 training chaotic (the "export gap" from [**?**]) also means that past inputs have *growing* influence on the current prediction.

This explains a key finding from the skip-pattern analysis: offset 8 is chosen before offset 2 by the greedy MI algorithm. The RNN's recurrence amplifies signals from further back, making skip-8 patterns *more* informative than skip-2 patterns. The gradient attribution confirms this: mean |attr| grows from 0.041 (k=1) to 0.081 (k=5–6), then stabilizes.

4 Per-Position Attribution

For each upstream position $t-k$, the attribution of input x_{t-k} to the prediction at t is:

$$\text{attr}(t,k) = W_x[:, x_{t-k}]^T \cdot g_{t \to t-k}$$

This measures how much the specific input character at position $t-k$ contributes to the gradient for the correct prediction at $t+1$.

offset k	mean attr	mean $\lvert\text{attr}\rvert$
1	+0.005	0.041
2	+0.000	0.057
3	+0.006	0.053
4	+0.029	0.070
5	+0.002	0.081
6	−0.016	0.081
7	+0.002	0.072
8	+0.006	0.079

The signed mean attribution is near zero for most offsets (cancellation across positions), except $k=4$ (+0.029) and $k=6$ (−0.016). The unsigned mean grows with offset, confirming that the RNN extracts more information from further-back positions.

Offset 4 having the largest signed attribution suggests a systematic bias: positions 4 bytes back consistently help predictions in the same direction. This may relate to the XML tag structure of enwik9 (tags like `<doc>` have a characteristic 4-byte pattern).

5 Worked Example

Consider $t=10$: input space, predicting 'x' (correct, $p=0.996$).

The gradient norm is small (0.013, very confident). Top neurons: h99 (+0.004), h68 (+0.004), h52 (−0.004). The backward trace shows k=1 ('i', attr +0.020) as the strongest contributor—the previous character 'i' in "wiki" gives the strongest signal for predicting 'x' (as in "`<mediawiki xmlns=`...").

At $t=28$: input '.', predicting 'm' ($p=0.998$). The backward trace shows k=4 ('/', attr +0.016) as the strongest contributor—the '/' four positions back (in "http://") is the strongest signal for predicting 'm' (as in "mediawiki.org"). This matches offset 4's large signed attribution.

6 Discussion

Sparse differentiation provides per-position, per-prediction attribution that the aggregate factor map cannot. The key findings:

1. The gradient is moderately sparse: 10 neurons capture 68% of the signal. The same neurons (h68, h76, h99, h8) dominate everywhere.

2. The backward gradient *grows* (2.4× at $k=8$), not decays. This is the spectral-radius->1 phenomenon that causes BPTT chaos in training, but for attribution it means further-back positions have *larger* influence.

3. Attribution per offset grows from 0.041 to 0.081, explaining why the greedy MI algorithm selects offset 8 before offset 2.

4. Individual positions show interpretable patterns: the '/' in "http://" contributes +0.016 attribution for predicting 'm' in "mediawiki".

The growth of the backward gradient means that sparsification must be applied carefully: the signal-to-noise ratio may degrade even as the absolute magnitude grows. A threshold based on relative contribution (fraction of $\|g_t\|^2$) rather than absolute value may be more appropriate.

The next step is to match these per-position attributions to the SN pattern superset, identifying which of the 834 skip-8 patterns are "active" at each position and how much of the gradient they explain.

Reproducibility

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

Tools:

- `sparse_diff.c` — gradient computation, backward tracing, per-position attribution, mean attribution by offset

References

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

[2] Claude and MJC. The Factor Map. 9 Feb 2026.

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

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