The Temporal Bi-Embedding: Forward Patterns, Backward Attribution, and the Function the RNN Learned

Claude and MJC

11 February 2026

Abstract

We characterize the sat-rnn as a bi-embedded function that maps events to numbers going forward in time (prediction) and numbers to events going backward in time (attribution). The forward map $\phi_{\rightarrow}$ propagates input events through the W_h highway, accumulating quotients that become output predictions. The backward map $\phi_{\leftarrow}$ traces each prediction to its input sources through the Jacobian chain, decomposing the prediction quotient into per-input contributions. We derive both maps explicitly, show that they are approximately inverse ($\phi_{\leftarrow} \circ \phi_{\rightarrow} \approx \text{id}$ at 74% alignment for shallow offsets), and identify the gap (higher-order patterns at deep offsets) as the "ladder" that training climbed but that pairwise analysis cannot recover. The bi-embedded function is the complete description of what the RNN learned: a temporal mapping between the event space E^T and the number space $\mathbb{N}^{82304}$.

1 The Forward Map: $E^T \rightarrow \mathbb{N}$

1.1 Events in time

The dataset $D = (x_0, \ldots, x_{N-1})$ lives in E_{in}^N: a sequence of N input events. At each position t, the RNN computes a hidden state $h_t \in \{-1, +1\}^{128}$ (in the Boolean regime) and an output distribution $P_t \in \Delta^{255}$.

The forward map takes the event history up to time t and produces a number (the prediction):

$$\phi_{\rightarrow}(x_0, \ldots, x_t) = P_t \in \mathbb{R}^{256} \tag{1}$$

But P_t depends on h_t, which depends on h_{t-1}, which depends on the full history. The recurrence unfolds the temporal dependency.

1.2 The forward map at depth d

We decompose $\phi_{\rightarrow}$ by temporal depth. At depth $d = 0$ (no history):

$$\phi_{\rightarrow}^{(0)} : x_t \mapsto W_x[\cdot, x_t] + b_h \tag{2}$$

The input byte maps to 128 numbers (the W_x column plus bias).

At depth $d = 1$ (one step of history):

$$\phi_{\rightarrow}^{(1)} : (x_{t-1}, x_t) \mapsto W_x[\cdot, x_t] + W_h \cdot \text{sign}(W_x[\cdot, x_{t-1}] + b_h) + b_h \tag{3}$$

The hidden state carries the sign of the depth-0 map applied to x_{t-1}.

At depth d:

$$\phi_{\to}^{(d)} : (x_{t-d}, \ldots, x_t) \mapsto h_t \quad \text{where} \quad h_s = \text{sign}(W_x e_{x_s} + W_h h_{s-1} + b_h) \tag{4}$$

for $s = t{-}d, \ldots, t$, with $h_{t-d-1} = 0$.

Definition 1 (Depth-d forward pattern)**.** *The* depth-d forward pattern *for neuron j at position t is the function:*

$$\phi_{\to,j}^{(d)} : E_{in}^{d+1} \to \{-1, +1\} \tag{5}$$

mapping the $(d{+}1)$-byte input window $(x_{t-d}, \ldots, x_t)$ to the sign of neuron j at time t. This is a Boolean function of $d{+}1$ bytes.

The forward map at depth d associates each input window with a binary hidden state. The pattern is "forward" because information flows from past inputs to present hidden state, following the causal arrow.

1.3 Forward patterns are skip-k-grams

The factor map showed that each neuron j depends primarily on two input offsets (d_1, d_2), with $R^2 \geq 0.80$. This means the depth-d forward pattern factors approximately:

$$\phi_{\to,j}^{(d)}(x_{t-d}, \ldots, x_t) \approx f_j(x_{t-d_1}, x_{t-d_2}) \tag{6}$$

for some function $f_j : E_{\text{in}}^2 \to \{-1, +1\}$.

The function f_j is the *conjunction detector*: it maps two bytes to a binary decision. The conditional mean $E[h_j \mid x_{t-d_1} = a, x_{t-d_2} = b]$ is the continuous version of f_j.

Observation 1 (Forward patterns = skip-bigrams + state)**.** *The depth-d forward pattern for neuron j with dominant pair (d_1, d_2) is equivalent to a skip-bigram (x_{t-d_1}, x_{t-d_2}) modulated by accumulated state (word length, in-tag status). The state is itself a function of the input history, carried through W_h. The "forward pattern" thus decomposes as:*

$$\textit{skip-bigram} \times \textit{state} \to h_j^{\pm}$$

which is the $(I \times T)^2 \times \{-1, +1\}$ factorization from the factor map paper.

1.4 The forward map as event counting

The forward map $\phi_{\to}$ is $E \to \mathbb{N}$:

- E: the input events $(x_{t-d}, \ldots, x_t)$.
- $\mathbb{N}$: the pre-activation $z_j(t) = W_x[j, x_t] + \sum_k W_h[j,k] h_k(t{-}1) + b_h[j]$ is a weighted sum of pattern strengths—a log-count of evidence.
- The sign $\text{sign}(z_j)$ is a decision: does the accumulated evidence exceed threshold?

Each forward pattern counts evidence for its hidden event. The count is not a raw frequency but a *weighted* count, with weights given by W_x and W_h. The forward map converts a temporal sequence of events into 128 evidence counts.

2 The Backward Map: $\mathbb{N} \to E^T$

2.1 Attribution: from prediction to inputs

The backward map takes a prediction (a number) and traces it to the input events responsible. At position t, predicting $y = x_{t+1}$:

$$\phi_{\leftarrow}(P_t, y) = \{(\alpha(t,d), x_{t-d}) : d = 0, \ldots, D\} \tag{7}$$

where $\alpha(t,d)$ is the scalar attribution at offset d and D is the effective depth.

2.2 The backward map at depth d

Definition 2 (Depth-d backward attribution)**.** *The backward attribution at depth d decomposes as:*

$$\alpha(t,d) = (W_x e_{x_{t-d}})^\top g_{t,d} \tag{8}$$

where $g_{t,d} \in \mathbb{R}^{128}$ is the backward gradient:

$$g_{t,0} = g_t = \frac{\partial \log P_t(y)}{\partial h_t} = W_y[y,\cdot] - \sum_o P_t(o) W_y[o,\cdot] \tag{9}$$

$$g_{t,d} = \left(\prod_{s=t-d+1}^{t} J_s \right)^\top g_t \quad \textit{for } d \geq 1 \tag{10}$$

with Jacobian $J_s = diag(1 - h_s^2) \cdot W_h$.

The backward gradient $g_{t,d}$ at offset d is a vector of 128 numbers. Each component $[g_{t,d}]_j$ measures: "how much does neuron j at time $t{-}d$ matter for the prediction at $t{+}1$?"

2.3 The backward map is $\mathbb{N} \to E$

The backward map converts numbers to events:

- $\mathbb{N}$: the gradient $g_{t,d}$ (128 numbers) and the attribution $\alpha(t,d)$ (one number).
- E: the input event x_{t-d} at offset d that the attribution identifies as responsible.

Definition 3 (Backward pattern)**.** *The* depth-d backward pattern *for prediction (t,y) is the per-neuron attribution vector:*

$$\phi^{(d)}_{\leftarrow,j}(t,y) = W_x[j, x_{t-d}] \cdot [g_{t,d}]_j = \alpha_j(t,d) \tag{11}$$

This is the contribution of input byte x_{t-d} through neuron j at offset d to the prediction of y at $t{+}1$.

The backward pattern reads: "output y was predicted because input byte x_{t-d} activated neuron j, which propagated through the Jacobian chain, which contributed to the output gradient." This is $\mathbb{N} \to E$: numbers (gradients, weights) map back to events (specific bytes at specific positions).

3 The Bi-Embedding: Both Directions

3.1 Forward and backward at the same position

At position t, the forward map says: "input events $(x_{t-d}, \ldots, x_t)$ produce hidden state h_t and prediction P_t."

The backward map says: "prediction $P_t(y)$ is attributed to input events at offsets $0, 1, \ldots, D$, with attribution $\alpha(t,d)$ at each offset."

Theorem 1 (The bi-embedding identity)**.** *The forward and backward maps compose to give the attribution decomposition:*

$$\log P_t(y) = const + \sum_{d=0}^{D} \alpha(t,d) + O(higher\ order) \tag{12}$$

where the sum of attributions approximates the total log-probability (up to the constant $b_y[y]$ and higher-order cross-terms between different offsets and neurons).

Proof. The first-order Taylor expansion of $\log P_t(y)$ around $h_t = 0$:

$$\log P_t(y) \approx \log P_t(y)\big|_{h_t=0} + g_t^\top h_t \tag{13}$$

The hidden state decomposes additively by source:

$$h_t \approx \sum_{d=0}^{D} \left(\prod_{s=t-d+1}^{t} J_s \right) W_x e_{x_{t-d}} \tag{14}$$

Substituting and collecting terms by offset:

$$g_t^\top h_t \approx \sum_{d=0}^{D} g_t^\top \left(\prod_{s=t-d+1}^{t} J_s \right) W_x e_{x_{t-d}} = \sum_{d=0}^{D} g_{t,d}^\top W_x e_{x_{t-d}} = \sum_{d=0}^{D} \alpha(t,d) \tag{15}$$

The O(higher order) terms arise from the non-linearity of tanh and the interaction between different input positions. □

3.2 The alignment measure

The bi-embedding is approximate: $\phi_\leftarrow \circ \phi_\rightarrow \neq \mathrm{id}$. We measure the alignment between the forward map (which patterns does the RNN use?) and the backward map (which patterns does the attribution identify?).

Definition 4 (PMI alignment)**.** *For input byte b at offset d predicting output y, define:*

- RNN attribution*:* $A_{RNN}(b,d,y) = \frac{1}{N} \sum_{t: x_{t-d}=b, x_{t+1}=y} |\alpha(t,d)|$.
- Data PMI*:* $PMI(b,d,y) = \log \frac{P(y|x_{t-d}=b)}{P(y)}$.

The alignment is the correlation $r(A_{RNN}, PMI)$ over all (b,y) pairs at offset d.

From the Q7 experiments:

Offset d	Alignment	Interpretation
1–4	88%	$\phi_{\leftarrow} \circ \phi_{\rightarrow} \approx \text{id}$
5–8	61–87%	Good but degrading
9–15	37–52%	Significant divergence
> 15	24–37%	$\phi_{\leftarrow} \circ \phi_{\rightarrow} \neq \text{id}$

Observation 2 (The bi-embedding tightens with proximity)**.** *The forward and backward maps are nearly inverse at short temporal distances (offsets 1–4: 88% alignment) and diverge at long distances (offsets > 15: 24–37%). The divergence has two sources:*

1. ***Higher-order patterns**: at deep offsets, the RNN's forward patterns involve conjunctions of 3+ events that pairwise PMI cannot capture. The backward map (which is linear in the gradient) misses these.*

2. ***Jacobian amplification**: the backward gradient grows with offset ($\|g_{t,d}\|$ increases 2.4× from $d = 1$ to $d = 8$), so the backward map is dominated by the spectral radius of W_h at deep offsets, not by the actual learned patterns.*

4 The Bi-Embedded Function Explicitly

4.1 What the RNN learned: a function $f : E^N \to \mathbb{N}^{82304}$

The trained RNN defines a function from the data to the weights. But this direction is trivially the training process. The non-trivial claim is the reverse: given the weights, the data's statistical structure is recoverable.

Definition 5 (The bi-embedded function)**.** *The sat-rnn implements a bi-embedded function $\mathcal{F}$ with two components:*

$$\mathcal{F}_{\rightarrow} : E_{in}^N \times \mathbb{R}^{82304} \to \Delta^{255 \times N} \qquad \textit{(forward: data + weights} \to \textit{predictions)} \tag{16}$$

$$\mathcal{F}_{\leftarrow} : \Delta^{255 \times N} \times E_{in}^N \to \mathbb{R}^{82304} \qquad \textit{(backward: predictions + data} \to \textit{gradients)} \tag{17}$$

$\mathcal{F}_{\rightarrow}$ is the forward pass. $\mathcal{F}_{\leftarrow}$ is the gradient computation (BPTT).

The bi-embedding claim is that these two functions, composed with the data's statistics, are approximately inverse:

1. $E \to \mathbb{N}$ **(forward composition)**:

$$E_{\text{in}}^N \xrightarrow{\text{count}} \mathbb{N}^{|P|} \xrightarrow{\text{construct}} \mathbb{R}^{82304} \xrightarrow{\mathcal{F}_{\rightarrow}} \Delta^{255 \times N} \tag{18}$$

   Count the events in data. Construct the weights from counts (Hebbian, analytic, shift-register). Run the forward pass. Result: 1.89 bpc (analytic) vs 4.97 bpc (trained).

2. $\mathbb{N} \to E$ **(backward composition)**:

$$\mathbb{R}^{82304} \xrightarrow{\mathcal{F}_{\rightarrow}} \Delta^{255 \times N} \xrightarrow{\text{factor map}} E_{\text{features}}^N \xrightarrow{\text{count}} \mathbb{N}^{|P|} \tag{19}$$

   Run the trained model. Read off interpretable features (factor map, $R^2 = 0.837$). Count the features in data. Result: the counts match the weights at $r = 0.56$ (W_h).

4.2 The bi-embedding at the level of W_h

The recurrent weights are the core of the bi-embedding. Each $W_h[k,j]$ encodes a temporal event relationship.

Forward ($E \to \mathbb{N}$): The event "h_j^+ at t and h_k^+ at $t{+}1$" has count $c(h_j^+, h_k^+)$ in the dataset. The Hebbian prediction:

$$\hat{W}_h[k,j] = s \cdot \frac{c(h_j^+, h_k^+) \cdot N - c(h_j^+) \cdot c(h_k^+)}{N^2} = s \cdot \mathrm{cov}(h_j, h_k) \tag{20}$$

where s is the optimal scale factor (3.94 for the sat-rnn).

Backward ($\mathbb{N} \to E$): The weight $W_h[k,j]$ means "neuron j's state propagates to neuron k with strength $|W_h[k,j]|$." Reading this weight identifies the event relationship: which pair (j,k), what sign (excitatory or inhibitory), what strength.

The bi-embedding quality:

Subset of W_h	n	r	R^2
All entries	16,384	0.40	16%
$\lvert W_h \rvert \geq 1.0$	9,821	0.44	19%
$\lvert W_h \rvert \geq 3.0$	5,887	0.56	31%
$\lvert W_h \rvert \geq 5.0$	2,104	0.61	37%

The bi-embedding is tighter for dynamically important weights (higher $|W_h|$). The unimportant weights ($|W_h| < 1.0$) are noise from the training process—the "ladder" that was needed for gradient flow but carries no interpretable event relationship.

4.3 The bi-embedding at the level of W_y

Forward ($E \to \mathbb{N}$): The analytic construction:

$$\hat{W}_y[o,j] = s \cdot (\log P(o \mid h_j > 0) - \log P(o \mid h_j < 0)) \tag{21}$$

This is the log-quotient of output event o conditioned on hidden event h_j^+ vs h_j^-, computed from counts in the dataset.

Backward ($\mathbb{N} \to E$): The weight $W_y[o,j]$ means "neuron j contributes this much log-support for output byte o." The sign determines the direction: $W_y[o,j] > 0$ means h_j^+ predicts o; $W_y[o,j] < 0$ means h_j^- predicts o.

Bi-embedding quality: The Hebbian correlation for W_y is $r = 0.01$ (near zero). But the analytic construction achieves 1.89 bpc—better than trained. The resolution: W_y's role is to convert hidden-state evidence into output predictions; the *direction* matters more than the correlation with trained weights. The trained W_y is over-optimized for the specific dynamics of the trained W_h; the analytic W_y is calibrated for the ideal (shift-register) dynamics, which are better.

5 The Temporal Structure of the Bi-Embedding

5.1 Forward patterns propagate causally

A forward pattern at depth d says: "input event x_{t-d} at time $t-d$ causes hidden event $h_j^{\pm}$ at time t." The causation flows forward through d applications of W_h:

$$x_{t-d} \xrightarrow{W_x} h(t-d) \xrightarrow{W_h} h(t-d+1) \xrightarrow{W_h} \cdots \xrightarrow{W_h} h(t) \tag{22}$$

The forward pattern is a *causal chain*: each link is a W_h pattern connecting adjacent timesteps. The chain's strength decays (or amplifies) according to the Jacobian:

$$\text{forward strength at depth } d = \prod_{s=0}^{d-1} \|J_{t-s}\| \approx \sigma_1(W_h)^d \cdot \prod_{s=0}^{d-1}(1 - h_{j_s}^2) \tag{23}$$

where $\sigma_1(W_h) \approx 5.5$ is the spectral norm. In the Boolean regime, $(1-h_j^2) \approx 0$ for saturated neurons, killing most chains. Only chains passing through unsaturated neurons survive.

5.2 Backward patterns trace evidentially

A backward pattern at depth d says: "the prediction $P_t(y)$ is evidence that input event x_{t-d} occurred." The evidence flows backward through the Jacobian chain:

$$P_t(y) \xrightarrow{g_t} h(t) \xrightarrow{J_t^\top} h(t-1) \xrightarrow{J_{t-1}^\top} \cdots \xrightarrow{J_{t-d+1}^\top} h(t-d) \xrightarrow{W_x^\top} x_{t-d} \tag{24}$$

Observation 3 (Forward and backward chains use the same weights)**.** *The forward chain uses W_h and the backward chain uses $J = diag(1 - h^2)W_h$. In the Boolean regime ($1 - h^2 \approx 0$ almost everywhere), the backward chain is killed at saturated neurons. But the forward chain does* not *have this saturation gate—it uses $sign(z)$, which is not zero.*

This asymmetry explains why the forward map and backward map diverge at deep offsets. The forward map (causal) propagates binary states deterministically. The backward map (evidential) tries to trace gradients through the same chain, but the saturation gate kills most of the gradient. The gradient that survives is amplified by W_h's spectral norm, creating the chaotic amplification seen in the Jacobian analysis.

5.3 The routing backbone as temporal bottleneck

The Q6 results showed that predictions route through a small backbone: h54 ← h121 ← h78. In terms of the bi-embedding:

- **Forward**: information from deep inputs enters through various neurons, but converges onto the backbone (h78 → h121 → h54) through high-$|W_h|$ connections.
- **Backward**: the gradient from the output (g_t) flows back through the same backbone, because h54 has the smallest margin (26.7) and therefore the largest saturation gate $(1 - h_{54}^2)$, making it the primary gradient conduit.

The backbone is where forward and backward patterns *coincide*: the same neurons carry information in both directions, through the same W_h connections, because they are the least saturated (most gate-open) neurons at each timestep.

6 The Bi-Embedded Function as a Functor

6.1 Event spaces and natural numbers form categories

Let $\mathcal{E}$ be the category of event spaces (objects: event spaces E_i; morphisms: patterns $p : E_i \to E_j$ with strengths) and $\mathcal{N}$ be the category of natural number spaces (objects: $\mathbb{N}$-valued functions; morphisms: arithmetic operations $+, \times, \log$).

Definition 6 (The bi-functor)**.** *The bi-embedded function is a pair of functors:*

$$\Phi : \mathcal{E} \to \mathcal{N} \qquad \textit{(counting: events to numbers)} \tag{25}$$

$$\Psi : \mathcal{N} \to \mathcal{E} \qquad \textit{(construction: numbers to events)} \tag{26}$$

Φ *maps:*

- *Objects: event space* $E_i \mapsto c_i : E_i \to \mathbb{N}$ *(the counting function).*
- *Morphisms: pattern* $p : E_i \to E_j \mapsto s(p) = \log_2 c(p)$ *(the SN strength).*

Ψ *maps:*

- *Objects: counting function* $c_i \mapsto E_i$ *(the event space it counts over).*
- *Morphisms: strength* $s \mapsto p_s$ *(the pattern with that strength).*

The bi-functor preserves composition: a chain of patterns $p_1 \circ p_2 \circ \cdots \circ p_k$ maps to the sum of strengths $s_1 + s_2 + \cdots + s_k$ (in $\mathcal{N}$) and back to the compound pattern $p_1 \circ \cdots \circ p_k$ (in $\mathcal{E}$). The UM's additive accumulation is the $\mathcal{N}$-side of this composition; the pattern chain is the $\mathcal{E}$-side.

6.2 The RNN's weights are the natural transformation

The trained weights $\{W_x, W_h, W_y, b_h, b_y\}$ define a natural transformation $\eta : \Phi \to \Psi \circ \Phi$ (approximately). Each weight w_{ij} is the $\mathcal{N}$-image of a pattern $p_{ij} \in \mathcal{E}$ under Φ, and also specifies how to reconstruct p_{ij} under Ψ:

$$w_{ij} = \Phi(p_{ij}) = s(p_{ij}) \approx \Psi^{-1}(p_{ij}) \tag{27}$$

The isomorphism (doubled-E) is the statement that Φ and Ψ commute with the RNN's computation: the UM's pattern accumulation in $\mathcal{E}$ gives the same predictions as the RNN's arithmetic in $\mathcal{N}$.

7 Closing the Temporal Loop

7.1 From past events to future predictions

The forward temporal bi-embedding:

$$\underbrace{(x_{t-d}, \ldots, x_t)}_{E^{d+1}} \xrightarrow{\phi_{\to}^{(d)}} \underbrace{h_t \in \{-1, +1\}^{128}}_{\mathbb{N}^{128}} \xrightarrow{W_y} \underbrace{P_t \in \Delta^{255}}_{\mathbb{N}^{256}} \xrightarrow{\text{argmax}} \underbrace{\hat{y}_{t+1}}_{E} \tag{28}$$

Past events $\to$ numbers $\to$ future event prediction.

7.2 From future predictions to past events

The backward temporal bi-embedding:

$$\underbrace{(P_t(y), y)}_{(\mathbb{N},E)} \xrightarrow{g_t} \underbrace{g_{t,d} \in \mathbb{R}^{128}}_{\mathbb{N}^{128}} \xrightarrow{W_x^\top} \underbrace{\alpha(t,d) \in \mathbb{R}}_{\mathbb{N}} \xrightarrow{\text{peak offset}} \underbrace{x_{t-d^*}}_{E} \tag{29}$$

Future prediction → numbers → past event identification.

7.3 The temporal loop closes

The composition $\phi_\leftarrow \circ \phi_\rightarrow$:

$$(x_{t-d}, \ldots, x_t) \xrightarrow{\phi_\rightarrow} P_t(y) \xrightarrow{\phi_\leftarrow} \{(\alpha(t,d'), x_{t-d'}) : d' = 0, \ldots, D\} \tag{30}$$

The loop closes when the backward map identifies the same input events that the forward map used:

$$\text{alignment}(d) = \text{corr}\left(\frac{|\alpha(t,d)|}{P(y \mid x_{t-d})},\ \text{PMI}(x_{t-d}, y)\right) \tag{31}$$

At shallow offsets (88%), the loop is nearly closed: the RNN's forward patterns are exactly the data's skip-bigram PMI. At deep offsets (24–37%), the loop opens: the RNN has learned higher-order temporal patterns that are invisible to pairwise analysis.

Observation 4 (The gap IS the higher-order structure)**.** *The 12% gap at shallow offsets and the 63–76% gap at deep offsets represent the structure that the first-order bi-embedding misses. This is exactly the gap between the Hebbian prediction (*$r = 0.56$*) and the trained weights (*$r = 1.0$*): the higher-order temporal patterns that BPTT captures by propagating gradients through the non-linear Jacobian chain.*

Closing this gap requires a higher-order bi-embedding: maps that account for 3+ event conjunctions, cross-offset synergies, and the non-linear interactions of the saturation gate. The pairwise analysis is the first rung of a ladder that extends to arbitrary order.

8 Conclusion

The sat-rnn learned a temporal bi-embedding: a function that maps events to numbers going forward in time (prediction via W_x, W_h, W_y) and numbers to events going backward in time (attribution via the Jacobian chain). The forward map is causal (skip-k-grams propagated through W_h). The backward map is evidential (gradients traced through $J^\top$). They are approximately inverse at short temporal distances (88% alignment) and diverge at depth (24% alignment) due to higher-order patterns and Jacobian amplification. The routing backbone (h54, h121, h78) is where the two directions coincide: the same neurons carry signal both forward and backward, because they are the least saturated (most information-carrying) nodes in the Boolean automaton.

References

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

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

[3] Claude and MJC. The Pattern-Chain UM. 8 Feb 2026.

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

[5] Claude and MJC. Toward Total Interpretation. 11 Feb 2026.

[6] Claude and MJC. Total Interpretation: Synthesis. 11 Feb 2026.

[7] Claude and MJC. From Counting to Construction. 11 Feb 2026.

[8] Claude and MJC. E onto $\mathbb{N}$. 11 Feb 2026.

[9] Claude and MJC. The Quotient Chain. 11 Feb 2026.