Settling: Sharpness Preference and ES Epistemics

Empirical Validation of the Timing Resolution

Claude and MJC — February 19, 2026

1. Experiment

We test the timing resolution conjecture from the companion paper. The setup: a pure UM bigram model trained and tested on enwik9 in a single online pass—each byte is predicted first, then learned from. We run at two scales: the first 4,096 and first 65,536 bytes. All reported bpc values are averaged over the scored positions. The model has four event spaces (byte_input, byte_output, byte_prev, envelope) and three LPPs targeting byte_output:

LPP	Source ES	Chain length	Role
0	envelope (True.)	1	Unigram (marginal)
1	byte_input	1	Conditional on current byte
2	byte_prev	2	Bigram conjunction patterns

LPP 2 contains conjunction patterns: $\min(t_{\text{prev}}, t_{\text{input}}, w)$ requiring both sources active. These are length-2 chains (input activates the conjunction, conjunction projects to output).

For each position, we run the full forward pass f (all patterns fire), then separately extract what each LPP alone contributes to the output ES. We compare four scoring strategies:

1. **Normal max-min**: standard f output (baseline).
2. **Sharpest LPP**: pick the LPP whose output distribution has lowest entropy.
3. **Longest chain**: always use the deepest LPP (LPP 2).
4. **Oracle**: pick whichever LPP gives highest p(next byte).

2. Results

	enwik9[:4K]		enwik9[:64K]	
Strategy	**bpc**	Δ	**bpc**	Δ
Normal max-min	4.883	—	4.985	—
Sharpest LPP	**1.955**	-2.93	**3.304**	-1.68
Longest chain	2.816	-2.07	3.481	-1.50
Oracle	1.350	-3.53	2.408	-2.58

Marginal dominance is catastrophic: 4.9 bpc is barely above unigram. The sharpest-LPP strategy recovers most of the information that marginal dominance destroys: -2.93 bpc at 4K, -1.68 bpc at 64K.

2.1. Selection Frequency

	4K		64K	
Source ES	**Sharpest**	**Oracle**	**Sharpest**	**Oracle**
envelope	0.0%	4.4%	0.1%	8.9%
byte_input	21.9%	22.2%	5.8%	19.0%
byte_prev (bigram)	**78.1%**	73.3%	**94.1%**	72.1%

The bigram LPP is selected as sharpest 78–94% of the time. As the model gets more data (64K), the bigram becomes even more dominant—its patterns cover more byte pairs. The sharpest and oracle strategies agree on byte_prev roughly 73–78% of the time, confirming that sharpness is a good proxy for prediction quality.

2.2. Mean Entropy per LPP

Source ES	**4K entropy**	**64K entropy**
envelope	4.686 bits	4.929 bits
byte_input	2.558 bits	3.709 bits
byte_prev (bigram)	**2.141 bits**	**2.478 bits**

The bigram LPP produces the sharpest distributions. This is the timing resolution in action: longer chains carry more context and make sharper predictions.

3. Connection to the Ring Pattern

The ring pattern construction (Feb 18) provides the UM-native mechanism for measuring sharpness. The second-highest support $s^{(2)}$ within an ES measures *oversupport*—how far the ES is from having a single confident answer. Low $s^{(2)}$ means sharpness: one event dominates, the ES "knows what it knows."

In this experiment, selecting the sharpest LPP is equivalent to selecting the LPP that produces the lowest $s^{(2)}$ in the output ES. This is exactly what the ring pattern measures. No new mechanism is needed—the ring pattern IS the sharpness detector, and it already exists as a P-programming construction within the UM.

The empirical results from the ring pattern paper (Feb 18) reinforce this: per-order $s^{(2)}$ shows a dramatic sparsity gradient (order-6: 51% have $s^{(2)} = 0$, decisive; order-1: 95% have $s^{(2)} \geq 64$, always confused). Higher-order contexts produce sharper ESs. The timing resolution says: *prefer them.*

4. Sharpness and ES Epistemics

An ES represents a question: "which event is the case?" The epistemics of the ES require mutual exclusivity—exactly one event is true. The support vector t restricted to an ES represents the model's belief about the answer.

Sharpness is epistemic health. A sharp ES (one event dominates) means the model has a confident, specific answer. A flat ES (many events with similar support) means confusion—the model doesn't know. The ring pattern's $s^{(2)}$ quantifies exactly this: it measures how far the ES is from the epistemic ideal of certainty.

This gives us a principled definition:

> **Sharpness** of an ES with support vector $(s_1 \geq s_2 \geq \cdots \geq s_k)$ is measured by $s_1 - s_2$, the gap between the top two supported events. This is equivalent to $-s^{(2)}$ when s_1 is fixed.

The forward pass should prefer states where every ES is sharp—where the model "knows what's what."

5. The Settling Conjecture

5.1. Statement

> *Settling* over the total thought t is the process of finding an assignment that maximizes the sharpness of each ES while remaining consistent with the total pattern P.

More precisely: given a pattern graph P and sensory input, the UM seeks a total thought t such that:

1. Each ES is as sharp as possible (ideally one dominant event).
2. The support values are consistent across patterns: if pattern $(a \to b, w)$ exists and $t_a > 0$, then $t_b \geq \min(t_a, w)$.
3. The result is a fixed point (or near fixed point) of f.

This is not a single forward pass—it is what the forward pass *converges toward* over the settling period. The single f is one step of this process. In a model with triangle patterns (short and long chains to the same ES), the settling has a natural dynamics: the short chain fires first (broad prediction), then the long chain fires (sharp prediction), and the ES settles on the sharper signal.

5.2. Bidirectional Settling

LPPs in the UM are bidirectional—a pattern $(a \to b, w)$ is evidence both that a supports b and that b is consistent with a. (In the CMP paper: patterns encode joint observations, not directed causation.)

This means settling can propagate in **multiple directions simultaneously**:

- **Forward**: input $\to$ intermediate $\to$ output. The standard prediction direction.

- **Backward**: a sharp output ES constrains which intermediate events are consistent. If output strongly supports "e", and only the "th" bigram maps to "e" with high weight, then "th" gets confirmed.
- **Lateral**: two intermediate ESs connected by patterns constrain each other. If one settles, it sharpens the other.

This bidirectional propagation is what makes settling a potentially coherent global process despite being computed from local operations. Each ES sharpening provides evidence to its neighbors via patterns, which may sharpen them in turn. The process converges when all ESs are consistent and sharp—or when no further sharpening is possible (the model has extracted all available context).

5.3. Relation to Biological Settling

In the brain, settling involves recurrent activity across multiple areas, taking variable time depending on input difficulty. Easy inputs (high-frequency patterns, strong context) settle fast; ambiguous inputs require more recurrent passes. The input-dependence of response time is well-documented in reaction time studies and is a signature of exactly this kind of constraint-satisfaction process.

Our UM settling conjecture maps onto this directly: triangle patterns with different chain lengths create a natural hierarchy of settling times, with simple predictions (short chains) arriving before complex ones (long chains). The organism doesn't commit to an output until the relevant ESs have had time to settle—and "time to settle" depends on the input.

Full settling likely involves multiple ESs coordinating at multiple levels, with stability emerging from the interaction. In a tiny model (3 LPPs, 769 events), we see the sharpness signal clearly, but the full constraint-satisfaction dynamics requires a richer model with more intermediate ESs and cross-connections.

6. Follow-up Experiments

6.1. Combination Strategies

Beyond winner-take-all, we test alternative ways to combine per-LPP distributions:

Strategy	**4K bpc**	**64K bpc**
Normal max-min	4.883	4.985
Sharpest LPP (winner-take-all)	1.955	3.304
Entropy-weighted blend	**1.880**	**3.062**
Confidence-gated (gap≥4: sharp, else: blend)	1.946	3.089
Max-support-per-byte	4.883	4.985

The entropy-weighted blend beats winner-take-all at both scales (−0.075 at 4K, −0.24 at 64K). This makes sense: when no single LPP is overwhelmingly sharpest, combining information from multiple sources helps.

The max-support-per-byte strategy (take the highest raw support for each output byte across all LPPs, then normalize) gives *exactly* the same result as normal max-min. This is the critical diagnostic: **the problem is in per-byte max aggregation itself.** Any strategy that operates on individual support values rather than distributions will reproduce marginal dominance.

6.2. Gap Analysis

The sharpness gap ($s_1 - s_2$) in the winning LPP's output predicts how much the sharpest-LPP strategy improves over normal:

Gap range	Freq.	Normal bpc	Sharpest bpc
		enwik9[:4K]	
< 4	73.2%	5.065	2.350
4–8	26.8%	4.387	0.878
		enwik9[:64K]	
< 4	90.5%	4.993	3.528
4–8	8.9%	5.033	1.240
8–16	0.6%	3.045	0.049

When the gap is large (≥ 8), the sharpest LPP achieves near-zero surprise (0.049 bpc at 64K). These are positions where the bigram is highly diagnostic and the model is essentially certain. When the gap is small, there is still substantial improvement (-1.5 to -2.7 bpc) but less dramatic.

6.3. Iterative Settling and LPP Ordering

Three negative results illuminate the nature of the problem:

Strategy	4K bpc	64K bpc
Normal max-min	4.883	4.985
Iterative settle (2 passes)	4.883	4.985
Iterative settle (3 passes)	4.883	4.985
Iterative settle (5 passes)	4.883	4.985
Reverse LPP order	4.883	4.985

All three are identical to normal max-min. The reason is fundamental: **max-min is idempotent and commutative.** Running f again after clearing and re-injecting the sharpest contribution does not help because the marginal LPP immediately overwrites it. Reordering LPPs doesn't matter because $\max(a, \max(b, c)) = \max(\max(a, b), c)$.

This is the key architectural insight: **settling cannot be achieved by iterating the current forward pass.** The max-min rule is a one-shot aggregation. To implement settling within f, we need a P-program that operates on *distributions over an ES*, not on individual support values. The sharpness preference must be expressed as patterns that gate LPP contributions based on their collective behavior at the target ES, not their per-event support.

7. What These Experiments Show

Confirmed:

- The timing signal is real and massive (-2.9 bpc at 4K, -1.7 bpc at 64K).
- Sharpness is a reliable proxy for prediction quality (78% agreement with oracle).
- Entropy-weighted blending outperforms winner-take-all (-0.075 to -0.24 bpc).
- Max-support-per-byte = normal max-min, confirming that per-event aggregation IS the problem.
- High-gap positions achieve near-zero surprise (0.049 bpc at gap ≥ 8).

Negative results:

- Iterative settling via repeated f does not work (max-min is idempotent).
- LPP ordering does not matter (max-min is commutative).
- Both confirm: settling requires distribution-level mechanisms, not support-level iteration.

Implications:

- The sharpness selection is currently *instrumented*—computed externally, not within f. This is the justified stopgap until the "outer UM" that gates on confidence exists.
- The P-programming challenge is now precise: we need patterns that detect per-LPP sharpness at a target ES and gate accordingly. The ring pattern provides the sharpness measurement; the gating mechanism is the open problem.
- Full settling will likely require multiple ESs coordinating via bidirectional patterns. In this tiny model (3 LPPs, 769 events), the constraint-satisfaction dynamics cannot yet emerge.