The Ring Pattern: Measuring Oversupport in the Universal Model

v3: Raw Support from Hash-Table Counts

Claude and MJC

18 February 2026

1 Introduction

In the theory of the Universal Model (CMP, Clement 2026), an event space (ES) represents a set of mutually exclusive events. When the model's forward pass produces strong support for *two or more* events in the same ES, this violates the mutual-exclusivity epistemics. We call this **oversupport**.

Prior work identified oversupport with prediction loss—the conflict between a predicted output byte and the observed byte. But the concept generalizes: oversupport is strong support for *any* two events sharing an ES, whether from sensory conflict or from internal contradiction "from the left" (when multiple LPPs terminating at the same ES support different events).

This paper introduces the **ring pattern**, a P-programmish construction that detects oversupport within the UM's own computational framework, and presents empirical measurement of oversupport in a running model (KN-6 on enwik9), including the first **raw s(2)** measurement directly from hash-table counts.

2 The Ring Pattern

2.1 Construction

Given an event space with events $\{e_1, \ldots, e_n\}$ and support values $\{s_1, \ldots, s_n\}$ after the forward pass, oversupport is the **second-highest support**: if two events both have high support, the minimum of their supports is large.

The ring pattern makes this computable within P. For each pair (e_i, e_j) with $i \neq j$, add a two-step chain:

$$e_i \to e_{ij} \to e_{\text{support}}$$

where e_{ij} is an intermediate event and e_{support} is a single accumulator event. Under the max-min forward pass:

$$\text{support}(e_{\text{support}}) = \max_{i \neq j} \min(s_i, s_j) = s_{(2)}$$

where $s_{(2)}$ is the second-order statistic (second-largest support value).

2.2 Why raw support, not normalization

The ring pattern outputs $s_{(2)}$ directly—the raw second-highest support value. This is **not normalized**.

It would be tempting to compute $s_{(2)}/s_{(1)}$ as a "confusion ratio," but this is wrong. Support values operate on a logarithmic scale (in SN, $[0, 255]$ represents log-support). A ratio of log-support values is meaningless: $s_{(2)} = 2, s_{(1)} = 1$ (one bit of evidence each) is *nothing like* $s_{(2)} = 16, s_{(1)} = 8$ (eight bits of evidence each), even though both give ratio 2.

The only meaningful normalizer would be the **total support for the whole ES**, which comes from a context source (e.g., the support for a context event that projects onto this ES). In general, we do not have this available. Therefore: **read $s_{(2)}$ directly, do not normalize**.

What $s_{(2)}$ tells us: the absolute amount of evidence supporting the second-best event. High $s_{(2)}$ means real conflict—the model has substantial evidence for two different outcomes. Low $s_{(2)}$ means the model's evidence is concentrated on one event.

2.3 Comparison with surprise

Surprise $= -\log_2 p(\text{actual})$ measures how wrong the model was *after* observing the outcome. $s_{(2)}$ measures how much conflicting evidence exists *before* observation. High surprise with low $s_{(2)}$ means confidently wrong. High $s_{(2)}$ with low surprise means confused but right.

3 Experimental Setup

We measure oversupport in the KN-6 model (6 LPP orders, $D = 0.90$, 128M hash table) running on the first 1M bytes of enwik9.

Two measurements: (1) probability proxy $p_{(2)}$ (second-highest normalized probability), and (2) raw $s_{(2)}$ (second-highest count per order directly from the hash table).

For each byte position t:

1. Compute the full 256-byte prediction distribution from KN-6.
2. Record $p_{(2)}$ (second-highest probability), surprise, and context state.
3. For each active KN order 1–6, look up per-byte counts in the hash table and record the second-highest count $s_{(2)}^{(k)}$.
4. Determine if active KN orders agree on the top predicted byte.

4 Results: Probability Proxy

On 1M bytes, mean second-highest probability = 0.123.

$p_{(2)}$ range	Positions	Percentage
$[0, 0.05)$	394,000	39.4%
$[0.05, 0.15)$	168,000	16.8%
$[0.15, 0.30)$	243,000	24.3%
$[0.30, 0.50)$	195,000	19.5%

Inside tags: 0.050 vs 0.125 outside (2.5× lower). 91% of positions show from-the-left disagreement between KN orders.

5 Results: Raw $s_{(2)}$ from Hash-Table Counts

5.1 Per-order analysis

The raw second-highest count varies dramatically by order:

Order	Mean $s_{(2)}$	Active positions	Character
1	3,052	999,804	Always confused
2	340	994,198	Moderately confused
3	69	967,532	Mixed
4	27	904,055	Mostly decisive
5	14	811,821	Largely decisive
6	6	702,757	Mostly decisive

Order 1 (unigram) always has high $s_{(2)}$: after 1M bytes, the second-most-common byte has been seen ∼3,000 times. This is not interesting oversupport—it reflects the inherent entropy of the byte distribution.

Order 6 is far more informative: 50.8% of active positions have $s_{(2)} = 0$ (the context has only ever predicted one byte), and only 1.6% have $s_{(2)} \geq 64$. At order 6, the model is *usually decisive*—when it has seen this 6-byte context before, it has strong evidence for one particular continuation.

5.2 Sparsity gradient

	0	1	2–3	4–7	8–15	16–31	32–63	64+
Order 1	0.1	0.2	0.3	0.6	0.8	1.4	2.3	94.5
Order 2	2.6	2.0	2.9	4.1	4.7	6.3	8.8	68.6
Order 3	10.4	7.3	8.1	9.4	11.3	13.0	14.0	26.6
Order 4	24.9	13.2	12.9	12.5	12.4	9.6	6.5	8.0
Order 5	39.9	16.2	13.6	10.5	8.0	5.2	3.0	3.6
Order 6	50.8	17.1	11.9	7.8	5.3	3.3	2.1	1.6

(Values are percentages of active positions in each $s_{(2)}$ bin.)

The sparsity gradient is the core finding: **oversupport concentrates at low orders**. Higher orders have seen fewer contexts and those contexts are more specific, so they are more decisive. The interesting oversupport is at intermediate orders (3–4), where the model has enough data to have real evidence but the context is ambiguous enough to support multiple outcomes.

5.3 Raw $s_{(2)}$ vs $p_{(2)}$: nearly uncorrelated

The Pearson correlation between raw $s_{(2)}$ (max across orders) and $p_{(2)}$ is $r = 0.047$—effectively zero.

This confirms the theoretical argument: $p_{(2)}$ is the result of normalization that divides by total count, and normalization destroys the absolute evidence information. A position can have very high $p_{(2)}$ (the model is confused about *relative* probabilities) while having low raw $s_{(2)}$ at the highest active order (little *absolute* evidence for any alternative). Conversely, high raw $s_{(2)}$ at order 1 is routine (the unigram always has evidence for many bytes) but $p_{(2)}$ after interpolation may be low (higher orders resolve the confusion).

5.4 Context-conditional raw $s_{(2)}$

Inside tags: mean raw $s_{(2)} = 1{,}660$ vs 3,096 outside (1.87× ratio). The probability proxy showed 2.5×. The difference is because normalization amplifies the effect: inside tags, total counts are also lower, so the same raw difference maps to a larger probability difference.

6 Results: From-the-Left Disagreement

At **91%** of positions, active KN orders disagree on the top predicted byte. Internal oversupport "from the left"—where different LPPs support different output events—is not exceptional but *the normal operating mode* of the model.

This reframes the role of interpolation: the KN interpolation chain is fundamentally a **conflict resolution mechanism**, not merely a smoothing technique.

7 Results: Surprise × Oversupport Joint

Surprise and $p_{(2)}$ are partially but not fully correlated.

	$p_{(2)} < 0.05$	$p_{(2)} \in [0.05, 0.20)$	$p_{(2)} \in [0.20, 0.35)$	$p_{(2)} \geq 0.35$
Surprise < 1	23.2%	6.5%	9.0%	14.1%
Surprise $[1, 3)$	3.9%	3.5%	8.5%	2.9%
Surprise $[3, 8)$	5.5%	5.2%	5.5%	1.6%
Surprise ≥ 8	6.8%	1.6%	1.3%	0.9%

6.8% are confidently wrong (high surprise, low $p_{(2)}$). 14.1% are confused but right ($p_{(2)} \geq 0.35$, surprise < 1).

8 Discussion

8.1 Per-order $s_{(2)}$ is the right metric

The raw measurement reveals that "max $s_{(2)}$ across orders" is uninformative (always dominated by order 1). The meaningful quantity is $s_{(2)}$ **at the highest active order**—the most specific context the model has seen. At order 6, $s_{(2)} = 0$ means "I've only seen one outcome for this 6-byte context"—genuine confidence. $s_{(2)} \geq 4$ (roughly ≥ 16 occurrences) means real conflicting evidence.

This connects to the sparse-LPP paper's threshold: joint events should be created when their support reaches $\sim$4, and oversupport should be measured against this same scale.

8.2 Ring pattern as P-program

The ring pattern construction shows that oversupport detection can be embedded in P itself—no external diagnostic needed. In a full UM, the ring pattern at each ES would produce a support value that the model could use to modulate behavior: increasing caution, broadening search, or triggering re-evaluation.

8.3 From-the-left as routine computation

The 91% from-the-left disagreement result means the model *routinely* produces conflicting support from different LPPs. The interpolation chain exists to mediate this. Understanding interpolation as conflict resolution opens the door to more sophisticated mediation strategies in the UM framework.

9 Conclusion

The ring pattern provides the first computable, P-programmish metric for oversupport in the Universal Model. The metric is $s_{(2)}$—the second-highest raw support value—read directly, without normalization.

Key findings on 1M enwik9:

- **Per-order sparsity gradient**: order-6 is decisive (51% have $s_{(2)} = 0$), order-1 is always confused (95% have $s_{(2)} \geq 64$). Interesting oversupport is at orders 3–4.
- **Raw $s_{(2)}$ and $p_{(2)}$ are nearly uncorrelated** ($r = 0.047$), confirming that normalization destroys information.
- 91% of positions show from-the-left disagreement between KN orders.
- 6.8% confidently-wrong, 14.1% confused-but-right.

The right measurement is $s_{(2)}$ at the highest active order, not max across orders. The ring pattern embeds this into P, a first step toward endogenous attribution and self-correction in P-programs.