Missed Opportunities in Tropical Selection: The Trigram Evidence

Claude and MJC

March 2026

Abstract

We measure the cost of max-min (tropical) LPP selection in a two-LPP model: byte bigram + trigram via threshold creation. At positions where the bigram is selected over the active trigram, the trigram would have predicted correctly 19–29% of the time while the bigram was wrong. This "missed opportunity" costs 0.38–0.49 bpc, *increasing* with scale from 10K to 1M bytes. This is a direct measurement of the combination problem: max-min selection becomes worse as the subordinate LPP becomes more informative. PMI analysis shows 1.971 bpc of information available in the trigram layer, of which max-min extracts only 0.042 bpc—a 2.1% extraction rate.

1 Setup

The trigram model (`wm-trigram`) has two LPPs:

- LPP_0: `byte_prev` → `byte_output` (bigram)
- LPP_1: `bigram_prev` → `byte_output` (trigram, via threshold creation)

Max-min selection picks the LPP with the highest gap $g = s_1 - s_2$. A *missed opportunity* occurs when:

1. The trigram LPP is active (has a source event)
2. The bigram wins selection (higher gap)
3. The bigram predicts wrong
4. The trigram would have predicted correctly

2 Scaling Results

The scaling table shows two trends. First, the trigram gain over bigram-only plateaus at ∼0.042 bpc from 1M onward (bigram-only: 4.163 at 10M vs combined: 4.121). Second, neurons grow sub-linearly: 4,953 at 10M, only 2.4× the 1M count despite 10× more data.

The headline result: the combination problem gets *worse* with more data. At 10K, missed opportunities cost 0.379 bpc; at 1M, 0.485 bpc. This is because the trigram learns more patterns at larger scale (2,067 neurons at 1M vs 215 at 10K), so it has more correct predictions that get overruled by the bigram's higher gap.

N	bpc	Neurons	Correct	Sharp-wrong	Tri wins	Missed bpc
10K	4.317	215	36.0%	12.1%	30.0%	0.379
100K	3.846	672	30.4%	26.0%	42.5%	0.405
1M	3.844	2,067	27.8%	16.9%	34.4%	0.485
10M	4.121	4,953	19.9%	12.1%	99.1%	0.699*

Table 1: Trigram model scaling. Neurons = reified bigrams in `bigram_prev` ES. Missed bpc = cost of positions where the trigram was right but the bigram was selected and wrong. Note missed bpc *increases* with scale. 10M bpc is evaluated as the running average (not on held-out data). *10M missed bpc is from online (1-pass) analysis; 10K–1M are from the 3-pass (frozen) pipeline.

3 Detailed Analysis (100K)

At 100K bytes:

- 94.5% of positions have the trigram LPP active
- 52,020 positions (52.0%) where the trigram is active but the bigram wins
- Of these, 83.4% of bigram selections are wrong
- The trigram would have been correct at 29.3% of these positions
- 10,004 "pure missed" positions (trigram right, bigram wrong)
- Cost: **0.405 bpc**

The bigram wins selection because it always has a source event (every byte activates `byte_prev`), while the trigram only fires when a bigram has been reified above threshold. The bigram accumulates more counts per entry, giving higher s_1 and therefore higher gap, even when its prediction is wrong.

4 The Gap Paradox

At 100K, the trigram winner has mean gap 2.2 with 45.5% at gap=1. These are narrow victories. When the trigram loses to the bigram, it is often because the bigram has gap ≥ 3 from its massive count advantage.

But the bigram's high gap does not mean high accuracy. The bigram has seen every byte pair, so its s_1 is large, but the prediction is based on the single most common successor — which is wrong 70% of the time.

The trigram, by contrast, conditions on a specific bigram context. Its gap is lower (fewer observations), but its prediction is more targeted. Max-min selection systematically prefers the confident-but-generic over the uncertain-but-specific.

5 Connection to MCP

Per MCP §1, the tropical semiring selects via max-min: the LPP with the largest gap dominates the output. The gap measures how much one byte dominates within a source's prediction, not how likely the source itself is to be correct.

The missed-opportunity cost is a *lower bound* on the combination problem at this model complexity: it only counts positions where the trigram's single best prediction was right. The trigram may also have had better probability mass on the actual byte without it being the argmax, which we do not count.

6 Oracle Analysis

At 100K, we compute the oracle bound: what if we always picked the correct LPP?

- At least one LPP predicts correctly at 42.2% of positions
- Selected LPP wrong but other right: 11.9% of positions (pure combination loss)
- Both LPPs wrong: 57.8% (irreducible with this architecture)
- Idealized oracle (0 bits when any correct): 2.868 bpc (gain = 0.978 bpc over max-min's 3.846)

The 11.9% pure combination loss is the exact set of positions where max-min selection damages accuracy. A perfect LPP selector that knew which source to trust would recover all of this. This is distinct from the probability-level combination (KN-style), which can additionally blend partial information from both sources.

7 Implications for the Lexicon Path

As we extend to word-level LPPs via the trigram embedding, missed opportunities will grow: word-level patterns are sparser (lower gap) but more informative when active. The fact that missed-opportunity cost *increases* with scale means the combination strategy must evolve before we add more LPPs, or the added LPPs will be wasted.

The 11.9% pure combination loss at 100K is already substantial. With word-level LPPs (Layer 2 of the trigram embedding), we add a third source with even lower gap. Unless the combination strategy can weigh context-specific accuracy over raw gap magnitude, the word LPP's correct predictions will be systematically overruled by the bigram.

8 Differential ω: MDL of the Trigram Layer

Under a naive MDL encoding, each LPP entry costs $\log_2(|\text{from}|) + \log_2(|\text{to}|) + 8$ bits (source event index, target event index, support value w). The trigram layer adds the trigram LPP entries plus the neuron identity mapping.

This is a naive upper bound. The actual information content of the LPP entries is lower because the distributions are highly structured (Zipfian, correlated across source events). A proper differential ω calculation using the UM algebra—as outlined in the research agenda—would show the *difference* between the model with and without the trigram layer, measured in the model's own terms.

The key observation: the trigram layer adds 71,873 entries at 10M but saves only 0.042 bpc. Under *any* reasonable encoding, 71,873 entries cost far more than 420,000 bits. The trigram is a bad deal as an independent model layer. Its value lies not in its direct prediction gain, but in its role as the foundation for the next level (word events via trie chaining).

N	Gain (bpc)	Tri entries	MDL (bpc)	Net
10K	0.016	1,103	3.078	−3.062
100K	0.045	5,690	1.618	−1.573
1M	0.042	23,383	0.691	−0.649
10M	0.042	71,873	0.217	−0.175

Table 2: The trigram layer never pays for itself under naive MDL. The gain plateaus at 0.042 bpc while the model cost drops as $O(1/N)$. At 10M, the model is still 5× more expensive than its benefit.

9 PMI Analysis: Quantifying Wasted Information

The consolidation paper defines a promotion criterion for creating new model structure: an event pair (a, b) should be reified when $n_{ab} \cdot \text{PMI}(a; b) > \Delta I_{\text{model}}$, where PMI measures the mutual information beyond what the marginals predict and ΔI_{model} is the MDL cost of adding the new structure.

We compute the PMI for all trigram joint events at 100K:

- 4,480 unique joint events (count ≥ 2)
- Mean PMI: 1.47 bits, 73.7% positive
- Net PMI gain: $\sum_{(a,b)} n_{ab} \cdot \text{PMI}(a; b)/N = 1.971$ bpc

The net PMI gain of 1.971 bpc is the total information available in the trigram's joint events beyond what the bigram marginals provide. This is the *potential* benefit of the trigram layer. Compare:

- 1.971 bpc: information available (PMI)
- 0.691 bpc: model cost at 1M (naive MDL, previous section)
- 0.042 bpc: information actually captured (max-min gain)

The trigram layer passes the consolidation paper's promotion criterion decisively: 1.971 bpc $\gg$ 0.691 bpc. The problem is not that the trigram layer is unjustified (it is), nor that it lacks information (it has nearly 2 bpc). The problem is that max-min selection extracts only 0.042 bpc of the 1.971 bpc available—a 2.1% extraction rate.

The remaining 1.929 bpc is "wasted information": the trigram layer *has* the answer but max-min selection *ignores* it because the bigram's gap is higher. This is the combination problem expressed as a ratio: the model's information content far exceeds what the tropical semiring can extract from it.

10 Conclusion

The trigram layer contains 1.971 bpc of information beyond the bigram marginals, passes the MDL criterion at scale, and would improve prediction at 11.9% of positions—yet max-min selection extracts only 0.042 bpc (2.1%). This is the combination problem measured directly: the tropical semiring's gap-based selection systematically prefers the confident-but-generic bigram over the

uncertain-but-specific trigram. The problem worsens with scale as the trigram learns more patterns that get overruled.

Any path forward—whether KN-style interpolation, entropy-weighted blending, or H3 meta-combination—must address this 47:1 ratio between available and extracted information. The trigram layer is not the bottleneck; the combination strategy is.