Order Degradation: Why More Context Hurts Under Sharpest-LPP

Claude and MJC

March 2026

Abstract

We test the UM's generic n-gram pipeline (`wm-ngram`) at orders 2 through 6 on 100K bytes of enwik9. The 3-pass pipeline (online → retroactive → frozen) with sharpest-LPP scoring gives: order 2 (bigram) = 4.157, order 3 (trigram) = **3.846**, order 4 = 4.056, order 6 = 4.074 bpc. Higher orders *degrade* performance: adding 4-gram and 6-gram LPPs makes the model worse than the trigram alone. The cause is the combination problem: sparse high-order contexts have high gap (from few observations) but poor accuracy, and sharpest-LPP lets them override the well-calibrated trigram. This confirms the order-scaling paper's finding that KN interpolation beats sharpest-LPP at low orders, and motivates H3 ($2^{g/H}$) as the UM-native combination rule.

1 Setup

The `wm-ngram` setup command creates a chain of LPPs with cascading threshold creation:

- Order 2: byte_prev → output (bigram only)
- Order 3: + bigram_prev → output (trigram)
- Order 4: + trigram_prev → output (4-gram)
- Order 6: + 4gram_prev, 5gram_prev → output

Each LPP has threshold $\tau = 4$ (∼16 observations) for neuron creation. The shift chain copies byte_output → byte_prev at each step. Higher-order context events propagate through the reification chain with 1-step delay per level.

The 3-pass pipeline (online → retroactive → frozen) runs automatically when the generic runner detects threshold LPPs.

2 Results

Key observations:

1. Order 3 (trigram) is optimal at 100K: 3.846 bpc frozen.
2. Order 4 is 0.210 bpc *worse*: adding the 4-gram LPP hurts.
3. Order 6 is 0.228 bpc worse: adding more orders makes it worse still.
4. The online bpc at orders 3–6 is nearly identical (∼4.43), confirming the higher orders add little during learning.
5. Order 2 (bigram only) has no neurons and no retroactive pass.

Order	Sharpest-LPP	H3-select	Count	Neurons
2	4.157	—	—	0
3	3.846	**3.826**	4.087	672
4	4.056	4.056	4.055	765
6	4.074	4.074	4.070	782

Table 1: 100K wm-ngram, frozen bpc. Three scoring rules tested. H3-select wins at order 3 (-0.020) but all three degrade above order 3.

3 The Mechanism

Sharpest-LPP selects the LPP with the highest $s_1 - s_2$ gap. At higher orders, a context that has been seen even twice produces a high gap: the observed byte gets $w = 2$, everything else stays at $w = 0$, giving gap $= 2$.

But with only $\sim$2 observations, the prediction is often wrong. The 4-gram context "abc" predicts "d" with certainty because "abcd" appeared twice, even though the true distribution at context "abc" is much broader.

Under sharpest-LPP, this confident-but-wrong 4-gram *overrides* the well-calibrated trigram that has hundreds of observations. This is the tropical tax applied at the order-selection level: max-min (highest gap wins) commits fully to the sharpest source.

4 Why KN Doesn't Have This Problem

KN interpolation weights each order by $D \cdot n_k/(n_k + D)$, where n_k is the context count. A context with 2 observations gets weight $\sim D \cdot 2/(2+D) \approx 0.67$, while a bigram context with 500 observations gets weight ~ 0.998.

KN's recursive backoff naturally downweights sparse high-order contexts. Sharpest-LPP does the opposite: it rewards the sharpest prediction regardless of confidence.

5 Scoring Rule Comparison

We tested three scoring rules:

1. **Sharpest-LPP**: select LPP with highest $s_1 - s_2$ gap (default)
2. **H3-select**: select LPP with highest gap$/H$ (normalized conviction)
3. **Count**: select LPP with highest total evidence $\sum_b 2^{w_b}$

H3-select gives a small gain at order 3 (-0.020 bpc) by preferring the trigram when it has higher normalized conviction than the bigram. But at orders 4 and 6, all three rules give nearly identical results: the degradation is robust to the scoring rule.

This rules out the hypothesis that the degradation is purely a scoring problem. The log-stochastic weight representation itself cannot distinguish *reliable sharpness* (many consistent observations) from *spurious sharpness* (few observations that happen to agree).

6 The H3 Solution (External)

The $2^{g/H}$ rule (H3 normalized conviction) from the order-scaling paper addresses this by normalizing gap by entropy:

$$w_k = 2^{\text{gap}_k / H_k}$$

A sparse context with gap = 2 and $H = 0.5$ (two observed bytes) gets $w = 2^{2/0.5} = 16$. But a well-calibrated trigram with gap = 3 and $H = 3.0$ gets $w = 2^{3/3} = 2$. The sparse context still wins—but less catastrophically than under sharpest-LPP (infinite weight to highest gap).

The order-scaling paper showed H3 improving from order 4 to 8 while KN degrades, with a crossover between orders 4 and 6. The present experiment confirms the degradation mechanism and motivates implementing H3 as the UM-native combination rule.

7 Conclusion

Order degradation under sharpest-LPP is the combination problem at the order-selection level. The UM's generic n-gram pipeline works mechanically (neurons cascade correctly) but the scoring rule prevents higher orders from helping. The path forward is H3 scoring within the UM framework: $2^{g/H}$ per LPP instead of argmax gap. This is the highest-priority implementation task.