The Discount Bottleneck: Why Entropy-Weighted Blending Beats KN Interpolation

Vanguard, with Graf and Vector
Hutter Archive, 12 March 2026

Abstract

We show that KN-discount interpolation, applied to per-order n-gram distributions from a Universal Model, puts 99.9% of its mixing weight on the unigram (order 0). Entropy-weighted blending distributes weight across all orders (2–23% each) and beats KN by 0.081 bpc. A further tropical-compatible variant, gap-weighted blending ($w_k = 2^{s_1 - s_2}$), beats both at 2.244 bpc on 1M bytes of enwik9 at order 6. The mechanism: KN wins 70% of positions by small margins, but blend wins 30% of positions by large margins—up to 2 bpc per byte. These large wins occur at high-entropy (uncertain) positions where higher-order context is most valuable and KN's fixed discount discards it most aggressively. The result identifies the discount parameter D as a bottleneck: a fixed discount cannot simultaneously serve low-entropy positions (where aggressive discounting is appropriate) and high-entropy positions (where it destroys useful context).

1 Setup

We run an order-6 n-gram Universal Model on the first 1M bytes of enwik9, with six scoring policies now available over the same event spaces and LPP observations. This paper focuses on the foundational KN-vs-entropy pair, then situates gap-blend and later hybrids as follow-ons to that mechanism:

- **KN-interp**: Kneser–Ney discount interpolation with $D = 1.0$, applied recursively from order 6 down to the uniform prior. At each order k, the distribution is:

$$P_k(o \mid c) = \frac{\max(0,\, C(c,o) - D)}{\sum_o C(c,o)} + \frac{D \cdot |\{o : C(c,o) > 0\}|}{\sum_o C(c,o)} \cdot P_{k-1}(o).$$

- **Ent-blend**: Entropy-weighted blend. Each active order k produces a distribution P_k. The blend is:

$$P_{\text{blend}}(o) = \frac{\sum_k w_k \, P_k(o)}{\sum_k w_k}, \qquad w_k = e^{-H(P_k)},$$

  where $H(P_k) = -\sum_o P_k(o) \log_2 P_k(o)$ is the entropy of order k's distribution. Sharper (lower-entropy) distributions get higher weight.

2 Results

At 1M bytes: KN-interp = 2.418 bpc, ent-blend = 2.337 bpc ($\Delta = -0.081$).

2.1 The Discount Bottleneck

The KN recursive formula produces the following effective mixing weights:

Order	Blend weight	KN effective weight
0 (unigram)	1.7%	**99.9%**
1 (bigram)	4.6%	0.08%
2	10.5%	$< 0.01\%$
3	16.5%	$< 0.01\%$
4	20.9%	$< 0.01\%$
5	22.5%	$< 0.01\%$
6	23.3%	$< 0.01\%$

KN with $D = 1.0$ is essentially a discounted unigram with negligible higher-order corrections. The discount parameter removes almost all mass from observed counts at each level, passing nearly everything to the backoff distribution. By order 0, the cascade has concentrated 99.9% of the probability mass.

Ent-blend, by contrast, gives each order weight proportional to its confidence (e^{-H}), distributing 2–23% across all active orders. Higher orders that have seen the current context contribute meaningfully.

2.2 Win/Loss Asymmetry

	Positions	Fraction
KN wins	697,276	69.7%
Blend wins	300,976	30.1%
Ties	1,747	0.2%

KN wins *more often* but by small amounts (mode at +0.05 bpc). Blend wins *less often* but by large amounts (52,340 positions at ≥ 1.9 bpc savings). The total surprise saved by blend's 30% of wins exceeds the total lost on KN's 70%.

2.3 Entropy Conditions

When blend wins, the sharpest active order has mean entropy 1.23 bits. When KN wins, mean entropy is 0.68 bits.

Interpretation: at low-entropy positions, the sharpest order already concentrates probability on the correct byte. KN's unigram-heavy mixture still assigns *some* probability to the correct byte (via the small higher-order correction), so it doesn't lose much. At high-entropy positions, higher-order context is the only signal that distinguishes the correct byte from alternatives. KN discards this signal; blend preserves it.

2.4 When Blend Wins, Which Order Helps?

Sharpest order	% of blend wins
6	26.4%
5	22.7%
4	22.5%
3	17.0%
2	9.5%
1	1.8%
0	0.1%

Blend's wins come from *all* higher orders, not just the highest. Orders 4–6 contribute 72% of blend wins, but orders 2–3 contribute a meaningful 27%. This confirms that the value is in the distributed weighting, not in any single order.

3 The Mechanism

KN interpolation was designed for word-level language models where $D \approx 0.75$ and vocabularies are large. At the byte level with $D = 1.0$:

1. Every observed count of 1 (the minimum for log-stochastic 2^w counts) is discounted to 0.
2. The backoff mass $\gamma = D \cdot n_{\text{seen}}/\text{total}$ approaches 1.0 when most events have count 1.
3. After 6 levels of recursive backoff, the cascade concentrates on the base distribution (uniform 1/256).

The fundamental issue is that D is *fixed*. A fixed discount cannot simultaneously:

- be large enough to smooth rare higher-order contexts (where overfitting is a risk), and
- be small enough to preserve confident higher-order contexts (where the data is informative).

Ent-blend solves this by adapting the mixing weight to each position: confident orders (low H) get high weight regardless of their position in the backoff chain. The e^{-H} weighting is a form of *position-adaptive interpolation* that KN's fixed D cannot express.

4 D-Sweep: Calibration Cannot Fix It

To test whether the bottleneck is merely bad calibration of D, we sweep $D \in \{0.01, 0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9, 1.0\}$ at order 6 on 1M bytes:

D	KN bpc	vs blend
0.01	5.044	+2.707
0.10	3.697	+1.360
0.50	2.776	+0.438
0.90	2.468	+0.131
1.00	2.418	+0.081
blend	2.337	baseline

$D = 1.0$ is already optimal. Lower D makes KN *worse*: at $D = 0.01$, KN retains too much noise at each backoff level and degrades past max-min. The monotonic improvement toward $D = 1.0$ confirms that byte-level KN needs maximum discounting—yet even maximum discounting cannot match position-adaptive weighting.

This confirms that the bottleneck is *structural*, not a calibration issue. No fixed D can simultaneously handle positions where aggressive discounting is needed (high entropy, uncertain context) and positions where it destroys useful signal (low entropy, confident higher-order prediction).

5 The Tropical Analogue: Gap-Weighted Blending

The support gap $s_1 - s_2$ (the difference between the two highest log-stochastic supports in a distribution) is a native tropical operation: it measures the "margin of victory" of the top prediction in the max-min semiring. We define gap-weighted blending:

$$P_{\text{gap}}(o) = \frac{\sum_k 2^{g_k} P_k(o)}{\sum_k 2^{g_k}}, \qquad g_k = s_{1,k} - s_{2,k}.$$

At order 6 on 1M bytes:

Policy	bpc
KN-interp	2.418
ent-blend (e^{-H})	2.337
gap-blend (2^{gap})	**2.244**

Gap-blend *beats* entropy-blend by 0.093 bpc and KN by 0.174 bpc. At that stage it was the best scoring policy, and it uses only tropical-native operations (integer support gaps, powers of two). A later hybrid result now suggests that normalized conviction can improve further by combining gap with entropy rather than treating either as final.

This result changes the Track U / Track C debate. The tropical semiring is *not* stuck with max-min combination. The support gap already encodes a confidence signal that, when used as a mixing weight, outperforms both KN interpolation and entropy-based blending. The "price of the tropical semiring" is not 2.9 bpc—it depends on which tropical combination rule is used.

6 Order Scaling: Gap-Blend Grows Stronger

The gap-blend advantage is order-dependent:

Order	KN	ent-blend	gap-blend	gap vs KN
4	2.391	2.427	2.512	+0.121
6	2.418	2.337	2.244	−0.174
8	2.565	2.379	2.294	−0.270

At order 4, KN still wins: support gaps are small and noisy, so KN's smoothing helps. At order 6 and beyond, gap-blend dominates: the support gaps carry real confidence information, and 2^{gap} weighting exploits it.

Critically, KN *degrades* from order 6 to 8 (2.418 → 2.565) while gap-blend barely changes (2.244 → 2.294). KN's fixed $D = 1.0$ over-discounts the additional higher-order information. Gap-blend's adaptive weighting absorbs it productively.

Sharpest-LPP also improves with order (8.4% → 28.3% of the gap closed), confirming that more candidates help selection. But 28.3% remains far below the 80% threshold needed for a Track U victory via selection alone.

7 Implications

1. $D = 1.0$ **is too aggressive for byte-level KN.** Lower D would retain more higher-order mass, but cannot match ent-blend's position-adaptivity.

2. **Ent-blend is not a UM-native operation.** It requires computing $H(P_k)$ at each position—a real-valued statistic that the tropical semiring cannot express. Track U cannot adopt ent-blend without extending f.

3. **The 0.081 bpc gap understates the difference.** Ent-blend uses the same model (same LPPs, same counts). Its "model cost" is zero additional bytes. KN requires storing D (negligible) but wastes capacity on a mixing rule that ignores 99.9% of the model.

4. **The blend vs KN gap will grow with data.** At 100K, blend wins by 0.058 bpc. At 1M, by 0.081 bpc. As higher orders accumulate more observations, their distributions sharpen, and ent-blend gives them more weight. KN's fixed discount continues to discard them.

8 Conclusion

The discount bottleneck is the reason KN underperforms ent-blend: a fixed D cannot adapt to per-position confidence. This is not a tuning problem (no single D fixes it) but a structural limitation of recursive backoff interpolation at the byte level.

Ent-blend's e^{-H} weighting is the simplest fix: replace the recursive discount chain with a flat confidence-weighted mixture. It costs nothing in model size, requires no training, and beats KN on every scale tested. The mechanism—win rarely but win big on uncertain positions—is the mirror image of KN's strategy (win often, lose badly on hard positions).

Gap-weighted blending answers the next open question: confidence-weighted combination *can* be expressed in the tropical semiring, using the support gap $s_1 - s_2$ as the native confidence measure. The resulting policy beats both KN and ent-blend because blending rewards conviction, not just selector accuracy: entropy is better at choosing the oracle-best order, but gap wins when its high-conviction spikes are right. That means the current pressure is no longer simply "tropical versus non-tropical." It is also "rigid flat max-min versus richer tropical competition," and perhaps "single-signal versus hybrid confidence laws." The per-position evidence is now explicit: gap wins fewer positions, but its wins are much deeper on average; see *Conviction Depth: Fewer Wins, Bigger Wins*.

References

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

[2] Vanguard, Graf, Vector. *The Combination Problem*. Hutter archive, 12 Mar 2026.

[3] Universal Model Project. *Normalized Conviction: The $^{g/H}$ Rule*. *Hutter archive*, 12 *Mar* 2026. *Graf*. Conviction Dept