SN Model Series and Hutter Prize Alignment

Claude and MJC

February 2026

Abstract

We build a series of six models of increasing power, from unigram (5.07 bpc) through KN-6 with word injection (2.00 bpc), and evaluate them against the Hutter Prize criteria. The Hutter Prize measures total size $S = S_1 + S_2$ (compressor + compressed archive) on enwik9 (10^9 bytes). The current record is fx2-cmix at 110,793,128 bytes (≈0.886 bpc effective), held by Kaido Orav and Byron Knoll since September 2024. Our online KN-6 on enwik9 produces 1.683 bpc = 210.4 MB — roughly 1.9× the record. We identify the three fronts where this gap must be closed and show how the SN format makes the description-length trade-off explicit.

1 Introduction

The Hutter Prize[1] rewards compressing enwik9 (10^9 bytes of Wikipedia XML). The score is:

$$S = S_1 + S_2$$

where S_1 is the compressor/decompressor program size and S_2 is the compressed archive it produces. The archive must decompress to exactly enwik9. Constraints: single CPU core, <10 GB RAM, <100 GB HDD, ≤50 hours.

The prize is 500,000 × $(1 - S/L)$ where L is the previous record. The current record is held by fx2-cmix (Orav & Knoll, September 2024) at L = 110,793,128 bytes. This implies an effective compression rate of ≈0.886 bpc.

Previous winners have improved the record incrementally: phda9 (116.7 MB, 2019) → starlit (115.4 MB, 2021) → fast cmix (114.2 MB, 2023) → fx-cmix (112.6 MB, Feb 2024) → fx2-cmix (110.8 MB, Sep 2024).

We aim to understand where our interpretable SN-based models sit relative to this target. Our approach differs from cmix (context mixing with neural networks) in being fully explicit: every prediction comes from named byte n-grams, word patterns, or their combinations, described as SN events and patterns.

2 The SN Model Series

We define six models of increasing power, each subsuming the previous. All are evaluated on 20M test bytes (80/20 split of first 100M of enwik9).

M0: Unigram. Byte frequency distribution. 256 events, no patterns. Each event "The output is `X`." carries a strength proportional to $\log_2(\text{count})$. This is the information-theoretic baseline.

M1: Bigram. Byte pairs. 512 events (256 input + 256 output), 17,781 non-zero patterns. Each pattern "The input is `X`." → "The output is `Y`." s encodes $s = \lfloor \log_2(\text{count}(XY)) \rceil$.

[1] `http://prize.hutter1.net/`

M2: KN-3. Kneser-Ney smoothed trigram. Interpolates orders 1–3 with discount $D = 0.8$. Backed by a hash table of n-gram counts.

M3: KN-6. Kneser-Ney order 6. Same structure, deeper context. Hash table: 33M entries at 80M training bytes.

M4: KN-6 + words. Adds 2,000 word events. At word-final positions, the absorption rule replaces KN prediction with word-conditioned prediction when the latter is more confident.

M5: KN-6 + word bigrams. Adds 88,447 word-bigram patterns (top 500 words). Attempts to boost prediction at word-start positions using $P(\text{next word} \mid \text{previous word})$.

3 Results

Model	bpc	Compressed (KB)	Patterns
M0: Unigram	5.069	61,880	0
M1: Bigram	3.889	47,477	17,781
M2: KN-3	2.551	31,134	0
M3: KN-6	2.043	24,939	0
M4: KN-6 + words	1.999	24,398	2,000
M5: KN-6 + w.bigr.	1.999	24,398	90,447

Table 1: Model hierarchy on 20M test bytes (80/20 split of first 100M). Compressed size estimates arithmetic coding on the full 100M at the model's test bpc rate. Pattern counts exclude the KN hash table.

Observation 1 (Word bigrams don't help at byte level)**.** *M4 and M5 achieve identical bpc (1.999). The word bigram patterns (88,447 of them) add no measurable improvement. This is because the word bigram information (2.845 bits/transition, per Experiment 2 of the tock-empirical paper) manifests at word-start positions as word identity prediction, not byte prediction. The absorption rule at word-final positions already captures most of the value. Extracting the remaining 0.10 bpc from word bigrams requires operating at the word level, not byte level—this is the tokenization bridge.*

4 Hutter Prize Alignment

4.1 Where We Stand

Our best model in online mode (online KN-6 on enwik9):

System	Total on enwik9	Effective bpc
gzip -9	322,592,222	2.581
bzip2 -9	253,977,891	2.032
xz -9	197,331,816	1.579
Our online KN-6	210,378,590	1.683
fx2-cmix (record)	110,793,128	0.886

Table 2: Hutter Prize standings on enwik9. Our KN-6 estimate uses the exact measurements on full enwik9. The effective bpc for prize entries is $8 \times$ total size$/10^9$. Our online KN-6 sits just above xz (210 vs 197 MB).

The gap is $210.4/110.8 \approx 1.9\times$, or equivalently 1.683 vs 0.886 bpc (0.797 bpc gap).

4.2 Decomposing the Gap

Where do the missing 0.80 bpc come from?

Hash table saturation + higher order (∼0.08 bpc). Our 128M-entry hash table saturates completely (100%) on enwik9. A larger table (the prize allows 10 GB RAM, enough for 512M entries) and higher orders would close some of this gap.

Word-level patterns (∼0.10 bpc). Word bigrams provide 0.10 bpc of mutual information (tock-empirical paper). Currently unrealized at byte level. The tock phase predicts this is recoverable via proper tokenization bridging.

Context mixing + neural (∼0.62 bpc). The remaining gap requires what cmix does: adaptively mixing hundreds of model components (including PPM, LSTM, word models, match models, indirect models) with neural-network-trained mixing weights. This is the bulk of the gap and represents long-range dependencies: sentence structure, paragraph coherence, XML tag matching, section patterns.

4.3 The Description Length Trade-off

The Hutter Prize score $S = S_1 + S_2$ makes the description length trade-off central. S_1 (the compressor) *is* the model description. S_2 (the archive) depends on the model's prediction quality.

Component	Entries	Size/entry	Total	Notes
KN-6 hash table	128M	12 B	1.5 GB	Rebuilt during decoding
SN patterns (M1)	17K	80 B	1.4 MB	Explicit bigrams
Word events (M4)	2K	300 B	0.6 MB	Word-final dists
W.bigram patterns	88K	80 B	6.9 MB	Word-pair patterns
Decoder program	1	—	∼15 KB	Arithmetic coder + KN

Table 3: Model description cost. For online models (KN-6), the hash table is rebuilt during decoding, so S_1 is just the decoder program (∼15 KB). The trade-off: a smaller, dumber model has small S_1 but large S_2.

Proposition 2 (Online models have minimal S_1)**.** *For online/sequential models like KN-6, the model state is rebuilt during decoding from the data already decoded. This means S_1 contains only the decoder program, not the model parameters. Our KN-6 decoder would be ∼15 KB of compiled C. The entire score is dominated by S_2 (the arithmetic-coded bitstream).*

This is actually an advantage of simple models: cmix's decompressor is ∼600 KB, while our KN-6 decoder would be ∼15 KB. The question is whether the 585 KB difference buys more than 585 KB of prediction improvement — and it does, massively.

5 SN Format Examples

5.1 Model 0: Unigram

```
"The output is ' '." 22.
"The output is 'e'." 19.
"The output is 'a'." 18.
"The output is 0x0A." 18.
```

The strength is $\lfloor \log_2(\text{count}) \rceil$. Space (0x20) has count ∼4M ⇒ strength 22.

5.2 Model 1: Bigram

```
"The input is ' '." 0.
"The output is 't'." 0.
"The input is ' '." "The output is 't'." 17.
```

The pattern says: when the input is space, the output 't' has strength 17 ($\approx$ 131K occurrences). The event strengths are 0 because they are not used as priors in the pattern-based model.

6 Roadmap to Competitive

To close the 1.9× gap:

1. **Larger hash table**: The 128M-entry HT saturates on enwik9. Prize rules allow 10 GB RAM. A 512M-entry table ($\sim$6 GB) would capture many more n-grams. Expected improvement: $\sim$0.1 bpc.
2. **Higher order**: KN-8 or KN-10 with the larger table. Each additional order helps on structured text (XML tags, repeated phrases).
3. **Word-level model**: Proper tokenization bridge to extract the 0.10 bpc from word bigrams. Integrate word identity as an additional context variable, not just post-hoc absorption.
4. **Match model**: Find exact matches of the current context earlier in the data. This is extremely powerful on Wikipedia (repeated templates, similar article structures). cmix uses this heavily.
5. **Context mixing**: Adaptively weight multiple model components. This is what separates cmix-class compressors from pure n-gram models. The key is learned mixture weights that adapt per-context.
6. **Arithmetic coder**: Implement actual arithmetic coding to produce a valid submission. The decoder is the S_1 component.

Steps 1–3 are within our current framework. Steps 4–5 require new architecture. Step 6 is mechanical.

References

[1] Claude and MJC, "Tock Phase Empirical Results," 2026.

[2] Claude and MJC, "KN Scaling Experiments," 2026.

[3] Claude and MJC, "The Extended Event Space," 2026.

[4] M. Hutter, "500,000 Prize for Compressing Human Knowledge," `http://prize.hutter1.net/`, 2020.

[5] K. Orav and B. Knoll, "fx2-cmix," 2024. `https://github.com/kaitz/fx2-cmix/`