Lexeme Injection Experiments: From Neutral Factorization to KN Dominance

Claude and MJC

February 14, 2026

Abstract

We report results from 31 iterations of experiment on lexeme injection into the sat-rnn's isomorphic Universal Model. Starting from the neutral introduction of "the" as a binary event space—verified EXACTLY on 10M bytes (4×10^{-14} residual)—we measure the value of lexeme-level prediction through causal prefix tries, onset distributions, split-alpha optimization, and vocabulary scaling. The optimized causal+onset model achieves 3.56 bpc whole-sample (45.7% reduction) and 3.72 bpc split-sample (43.3%) at 5K words on 10M bytes. However, a byte-level KN-6 n-gram model dominates the RNN entirely: 1.24 bpc whole (2.93 split) vs. RNN 6.43 bpc. The lexical model was compensating for the RNN's weakness, not adding genuine structure. The comprehensive ensemble KN+RNN+trie achieves 2.63 bpc split-sample—beating RNN+lexicon (3.70) by 1.07 bpc. At 10M bytes, KN-5 reaches 2.39 bpc split; KN+RNN at $\alpha = 0.2$ gives 2.332 bpc, with the RNN contributing only 0.057 bpc (2.4%).

1 Introduction

The companion paper [**?**] establishes that introducing "the" as a binary event space $E_{\text{the}} = \{\text{the}^+, \text{the}^-\}$ is neutral: marginalizing over the new dimension recovers exactly the original byte-level predictions. This paper reports the experimental program that followed: 31 iterations (`the_inject.c` through `the_inject30.c`) exploring how to extract value from the factorization.

The experiments divide into five phases:

1. **Neutrality verification** (iterations 1–3): confirming exact neutrality at both the count-table and RNN levels.
2. **Value measurement** (iterations 4–6): quantifying the potential value of lexeme-level prediction.
3. **Causal prediction** (iterations 7–14): building a causal prefix trie that actually improves predictions.
4. **Optimization** (iterations 15–23): vocabulary scaling, split-alpha, conditional onset, and negative results.
5. **KN integration** (iterations 24–30): discovering that byte-level KN n-grams dominate the RNN entirely.

2 Phase 1: Neutrality Verification

2.1 Count-table level (iteration 1)

The first experiment factors skip-bigram count tables through E_{the} on 10K bytes of enwik9. For each offset d and each (x, o) pair:

$$c_d(x, o) = c_d^+(x, o) + c_d^-(x, o),$$

where c_d^+ restricts to positions at "the" boundaries and c_d^- captures everything else.

Result: 35 "the" occurrences in 10K bytes. Neutrality is *exact*: 0 errors across 786K cells. The bpc difference is 0.000000000000. Asymmetry is concentrated at `e` → `space`: $P(\texttt{space} \mid \texttt{e}, \text{the}^+) = 1.0$ vs. $P(\texttt{space} \mid \texttt{e}, \text{the}^-) = 0.13$.

2.2 RNN-level verification (iteration 2)

The second experiment loads the full sat-rnn model (sat_1024) and partitions bpc by the$^+$/the$^-$ across 10M bytes.

Result: Neutrality holds to machine precision: $|\Delta\text{bpc}| = 4 \times 10^{-14}$. The RNN gives $P(\texttt{space} \mid \text{the}^+) = 0.12$ uniformly—the model treats all "the" boundaries identically.

2.3 Hidden state analysis (iteration 3)

Iteration 3 examines whether the RNN's hidden state h separates the$^+$ from the$^-$ positions.

Result: $\cos(h_{\text{pos}}, h_{\text{neg}}) = 1.0000$. There is *no* separation. $\text{MI}(\text{the}; \text{next\_byte}) = 10^{-5}$ bits. The RNN has not learned to detect "the" as a distinct event—consistent with the bag-of-letters model [**?**].

3 Phase 2: Value Measurement

3.1 Lexeme value at the boundary (iteration 4)

Bayesian and log-linear mixing at the$^+$ positions.

Result: Log-linear with $\beta = 1$ is optimal. Overall value from just "the" boundary: -0.020 bpc. Small but nonzero— knowing "the" just occurred sharpens the next-byte prediction by a modest amount.

3.2 Intra-lexical value (iteration 5)

Measures the value at each byte position *within* "the":

Position	Byte	bpc gain
Offset 1 (h after t)	`h`	8.14
Offset 0 (t)	`t`	5.90
Offset 2 (e)	`e`	3.02
Boundary (after e)	next	3.11

Total intra-lexical value: 0.130 bpc. The interior of the word carries far more value than the boundary—the model's predictions *within* "the" are where knowing the word helps most.

3.3 Vocabulary scaling of value (iteration 6)

Extends value measurement to the top-N words by frequency:

Words	Oracle bpc	% of RNN loss	Internal:boundary
50	1.02	16%	5:1
200	1.74	27%	5:1
500	2.32	36%	5:1

The oracle value (using true word labels) scales log-linearly with vocabulary size. Internal bytes dominate boundary bytes 5:1 across all vocabulary sizes—the value of knowing which word is occurring is concentrated inside the word, not at its edges.

4 Phase 3: Causal Prediction

4.1 Causal prefix trie (iterations 7–11)

The oracle results of Phase 2 use the true word identity. A causal model cannot peek ahead—it must infer the word from the prefix seen so far. We build a prefix trie over the top-N words and mix its predictions with the RNN's via linear or log-linear combination.

Iteration 7 establishes the causal baseline: 500 words, 6.43 $\rightarrow$ 4.63 bpc (-28%) vs. oracle 4.19 bpc (-35%).

Iterations 8–11 debug and tune:

- Linear mix with oracle positions (iter. 9): coverage 38.3% for 100 words; $\alpha = 1.0$ is best (= oracle).
- Causal trie + linear mix + α sweep (iter. 10): best $\alpha = 0.9$, 80% oracle recovery.
- Linear vs. log-linear (iter. 11): log-linear slightly better at low α, similar at optimal α.

4.2 Per-offset gap analysis (iteration 12)

Key finding: Offset 0 (word onset) accounts for the *entire* causal–oracle gap. The trie performs well mid-word (the prefix uniquely identifies the word after a few bytes) but fails at onset: the first byte of a new word is ambiguous, and the causal model has no information.

4.3 Causal + onset model (iterations 13–14)

Insight: at word boundaries, supplement the trie with the *onset distribution* $P(\text{first\_byte} \mid \text{boundary})$ learned from training data.

Result (iteration 13): At 10M bytes, 500 words:

$$\text{RNN: } 6.56 \rightarrow \text{ Causal+onset: } \mathbf{4.57} \text{ bpc} \quad (-30.4\%),$$

vs. Oracle: 4.62 bpc. The causal+onset model *beats* the oracle, recovering 102.5% of oracle gain. This is possible because the onset distribution adds information the oracle doesn't have (the oracle labels words but doesn't model their onset).

Split-sample validation (iteration 14): Train on first half, test on second. Causal+onset: 4.54 bpc vs. oracle+onset: 4.58 bpc. The causal model wins even with a train/test split.

5 Phase 4: Optimization

5.1 Vocabulary scaling (iteration 15)

Words	Coverage	bpc	% reduction
100	24%	5.12	−20%
500	38%	4.57	−30%
1K	48%	4.12	−37%
5K	72%	3.56	−46%
10K	92%	3.12	−52%

Returns are diminishing but steady. The causal–oracle gap widens at large vocabulary (rare words are harder to identify from prefix).

5.2 Conditional onset (iterations 16, 21)

Conditioning the onset distribution on the previous byte: $P(\text{first} \mid \text{prev\_byte})$. Modest improvement: 3.246 vs. 3.299 bpc at 5K words.

2-byte conditioning (iter. 21): $P(\text{first} \mid \text{prev\_2})$ adds 0.026 bpc over 1-byte. 3-byte overfits at 1M.

Word bigram onset (iter. 18): $P(\text{first} \mid \text{prev\_word})$ gives *no improvement* over byte conditioning. Sparse counts dominate—byte context is richer than word context.

5.3 Split-alpha optimization (iterations 19–20)

Key insight: Onset and mid-word positions want different mixing weights. We sweep $\alpha_{\text{onset}} \times \alpha_{\text{mid}}$ independently:

	α_{onset}	α_{mid}
Optimal	1.0	0.9
Uniform best	0.7	0.7

At onset, $\alpha = 1.0$ is best: the RNN is *useless* at word boundaries (7.8 bpc). Mid-word, the trie dominates but the RNN contributes marginally ($\alpha = 0.9$).

Best combined result (iteration 20):

- 10M, 5K words, whole-sample: **3.56 bpc** (45.7% reduction)
- 10M, 5K words, split-sample: **3.72 bpc** (43.3% reduction)
- 1M, 5K words, whole-sample: **3.16 bpc** (50.8% reduction)

5.4 Negative results (iterations 22–23)

RNN hidden-state centroid onset (iteration 22): Using the RNN's hidden state centroid at word boundaries to predict onset bytes. Result: +0.085 bpc *worse* than conditional onset. $\cos(h) \approx 1.0$ at boundaries—the hidden state centroids do not discriminate between words. The RNN's representation at word boundaries is essentially uniform.

Entropy-adaptive alpha (iteration 23): Sweep α per RNN-entropy bin, expecting that high-entropy positions (where the RNN is uncertain) should weight the trie more. Result: *zero*

improvement. 99.96% of onset positions fall in a single entropy bin $[4, 5)$. The RNN's entropy is constant at word boundaries—there is no signal to adapt on.

6 Phase 5: KN Dominance

6.1 KN-6 byte n-gram integration (iteration 24)

We integrate a byte-level Kneser-Ney 6-gram model (KN-6) and compare per-category against the RNN and lexical models:

Category	KN-6	RNN	Cond. onset
Onset positions	3.04	7.80	4.77
Mid-word	0.91	5.84	—
Overall (1M whole)	**1.24**	6.43	—

KN-6 beats the RNN by 5× at every position category. It even beats the conditional onset model at onset positions by 1.73 bpc. Notably, the lexical trie *hurts* mid-word performance: 1.42 bpc (trie+RNN) vs. 0.91 bpc (KN alone).

6.2 KN+RNN ensemble (iterations 25–27)

Whole-sample (iteration 25): RNN adds *zero* value to KN at whole-sample evaluation. Pure KN = 1.24 bpc is optimal. All $\alpha > 0$ (mixing in RNN) makes it worse.

Split-sample (iteration 26): KN-6 split: 2.93 bpc (overfitting from 1.24, but still dominates RNN 6.51 and RNN+lex 3.70).

Comprehensive ensemble (iteration 27):

Model	Split-sample bpc
RNN alone	6.51
RNN + lexicon	3.70
KN-6 alone	2.93
KN + RNN ($\alpha = 0.2$)	2.75
KN + RNN + trie	**2.63**

The KN+RNN+trie ensemble beats RNN+lex by 1.07 bpc. But the RNN's contribution is small (0.18 bpc from KN alone to KN+RNN).

6.3 KN order sweep (iteration 28)

	Order 3	Order 4	Order 5	Order 6+
1M split	3.14	**2.81**	2.89	2.92
10M split	2.68	2.44	**2.39**	2.41

At 1M, order 4 is optimal. At 10M, order 5 wins. Higher orders saturate the hash table (16M entries at order 7+). For reference, the RNN achieves 6.58 bpc—KN-5 at 10M is 2.76× better.

6.4 KN+RNN at 10M scale (iteration 29)

$D = 0.9$ is the optimal discount (2.383 vs. 2.390 at $D = 0.75$). KN+RNN at $\alpha = 0.2$: **2.332 bpc**. The RNN adds 0.057 bpc (2.4%)—a shrinking contribution as KN gets more data.

6.5 Word bigram KN at onset (iteration 30)

Negative result: Word bigram KN (conditioning onset on the previous word) gives 5.63 bpc vs. byte KN 4.73 bpc at 10M split. Word-pair sparsity dominates. Byte context is richer than word context at feasible data sizes.

7 Discussion

7.1 What the lexical model was really doing

The most important finding is Phase 5: the lexical model compensates for the RNN's weakness rather than adding genuine structure. When replaced with byte-level KN smoothing, the RNN's contribution shrinks to 2.4%.

The RNN at DSS=1024 achieves 0.079 bpc by near-memorization of a tiny sample. But when evaluated on larger data (1M–10M bytes), it collapses to 6.43 bpc—worse than a simple n-gram model by 5×. The lexical trie was patching over this collapse by providing word-boundary information the RNN couldn't learn.

7.2 The neutrality result stands

Despite the KN dominance finding, the neutral factorization (Section **??**) remains theoretically significant. It demonstrates that lexeme-level event spaces can be introduced into the UM framework without approximation—the $E \to N \to Q$ chain handles the factorization exactly.

The right interpretation is not "lexemes are useless" but rather "the sat-rnn at DSS=1024 is too weak to benefit from lexemes." A stronger base model (e.g., KN-5 at 10M) leaves less room for lexeme-level improvement, but the factorization framework is available whenever asymmetry between $P(o \mid x, \text{the}^+)$ and $P(o \mid x, \text{the}^-)$ exceeds the base model's predictions.

7.3 Scaling hierarchy

The results establish a clear hierarchy of modeling capacity at the 1M–10M byte scale:

Model	1M split	10M split
Sat-RNN	6.43	6.51
RNN + lexicon (5K words)	3.70	—
KN-4	**2.81**	2.44
KN-5	2.89	**2.39**
KN-5 + RNN ($\alpha = 0.2$)	—	**2.33**

7.4 Onset is the key position

Across all experiments, word onset is where models diverge most. The RNN gives 7.8 bpc at onset (essentially random), the conditional onset model gives 4.77 bpc, and KN-6 gives 3.04 bpc. Onset accounts for 80% of the remaining causal–oracle gap (iteration 17).

This makes linguistic sense: the first byte of a new word is maximally uncertain because it depends on the next word choice, which depends on semantics, syntax, and discourse context that a byte-level model has limited access to. The KN n-gram model captures enough local context (the preceding 5–6 bytes) to partially predict word onset, but this is where genuine language modeling begins.

8 Summary of Key Numbers

Finding	**Value**
Neutrality residual (10M bytes)	4×10^{-14}
Hidden state separation $\cos(h)$	1.0000
Oracle value, 500 words	2.32 bpc (36%)
Internal:boundary value ratio	5:1
Causal+onset, 500w/10M	4.57 bpc (−30.4%)
Oracle+onset, 500w/10M	4.62 bpc
Causal/oracle recovery	102.5%
Split-alpha, 5Kw/10M whole	3.56 bpc (−45.7%)
Split-alpha, 5Kw/10M split	3.72 bpc (−43.3%)
KN-6, 1M whole	1.24 bpc
KN-6, 1M split	2.93 bpc
KN-5, 10M split	2.39 bpc
KN+RNN+trie, 1M split	2.63 bpc
KN+RNN, 10M split	2.33 bpc
RNN contribution at 10M	0.057 bpc (2.4%)

References

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

[2] Claude and MJC. *Introducing "the": A Neutral Factorization of the Byte-Level Model*. Hutter archive, 14 Feb 2026.

[3] Claude and MJC. *Lexemes as Binary Event Spaces: From Atomic Patterns to Bag-of-Letters Prediction*. Hutter archive, 14 Feb 2026.

[4] Claude and MJC. *Bayes from Counting: Partial Quotients, GCD, and the Symmetric Learning Function on $E = I \times O$*. Hutter archive, 12 Feb 2026.