Introducing "the": A Neutral Factorization of the Byte-Level Model

Claude and MJC

February 14, 2026

Abstract

We introduce the word "the" as a binary event space into the sat-rnn's isomorphic Universal Model (DSS=1024, 0.079 bpc) and prove that this introduction is *neutral*: marginalizing over the new event recovers exactly the original predictions. The operation adds $E_{\text{the}} = \{\text{the}^+, \text{the}^-\}$ to the model's event space and factors the existing skip-bigram count tables through this new dimension. Via the $E \to N \to Q$ chain (events → counts → ratios), we show that the Bayesian marginalization that "pulls 'the' out of" the skip-bigrams is exact: the marginal over E_{the} reproduces the original byte-level conditionals with no residual. This neutrality is the point. We have changed nothing about the model's predictions; we have only introduced a new axis along which the count tables can be decomposed. Value comes later, when intra-lexical patterns (from atomic byte events to lexeme events) are added. But the neutral introduction establishes the bookkeeping: the negative event the^- distributes over the rest of the "the"-prefixed lexicon, providing a concrete model for how negative support distributes generally.

1 Setup

We work with the sat-rnn trained on DSS=1024 bytes of enwik9, achieving 0.079 bpc. Its isomorphic UM (the doubled-E construction [2]) reproduces these predictions exactly. We also have the pattern-chain UM, which achieves 0.067 bpc at order 12 via skip-k-gram count tables [3].

For concreteness, we use the skip-bigram count tables at various offsets d, which are the atomic building blocks of both models.

Definition 1 (Skip-bigram count table)**.** *For offset d and dataset $D = (b_1, b_2, \ldots, b_N)$ of N bytes:*

$$c_d(x, o) = \big|\{t : b_{t-d} = x \text{ and } b_t = o\}\big|, \tag{1}$$

the number of times byte x appears at offset d before output byte o. The corresponding conditional:

$$P_d(o \mid x) = \frac{c_d(x, o)}{\sum_{o'} c_d(x, o')}. \tag{2}$$

Definition 2 (The $E \to N \to Q$ chain [**?**])**.** *The chain from events to prediction:*

1. *E: **Events**. Each position t in the dataset is a joint event $(x_d, o) \in I \times O$, where $x_d = b_{t-d}$ is the input byte at offset d and $o = b_t$ is the output byte.*
2. *N: **Natural numbers** (counts). $c_d(x, o) \in \mathbb{N}$ counts how many events have context byte x at offset d and output o. Marginals: $c_d(x) = \sum_o c_d(x, o)$.*

*3. Q: **Quotients** (luck). $Q_d(x,o) = N/c_d(x,o) = 1/P_d(x,o) = \lambda_d(x,o)$, the joint luck.*

The chain $E \to N \to Q$ is the UM's standard learning function ω_0: observe events, count them, compute quotients. The joint luck decomposes via partial quotients:

$$\log_2 Q_d(x,o) = \underbrace{\log_2 Q_I(x)}_{\textit{marginal luck}} + \underbrace{\log_2 Q_d(o \mid x)}_{\textit{conditional luck}}, \tag{3}$$

where $Q_I(x) = N/c_d(x)$ and $Q_d(o \mid x) = c_d(x)/c_d(x,o) = 1/P_d(o \mid x)$.

*The GCD decomposition [**?**] further separates each count into common and differential evidence: $c_d(x,o) = g_I(x) \cdot r_I(x,o)$, where $g_I(x) = \gcd_o c_d(x,o)$. The conditional depends only on the reduced counts: $P_d(o \mid x) = r_I(x,o)/R_I(x)$ where $R_I(x) = \sum_o r_I(x,o)$. The GCD cancels completely.*

2 Introducing "the" as a Binary Event Space

Definition 3 (The word indicator)**.** *Define $\mathbf{1}_{the}(t) \in \{0,1\}$ as the indicator that position t falls inside an occurrence of "the" delimited by word boundaries:*

$$\mathbf{1}_{the}(t) = \begin{cases} 1 & \text{if } b_{t-3} \in W \text{ and } b_{t-2} = \texttt{t} \text{ and } b_{t-1} = \texttt{h} \text{ and } b_t = \texttt{e} \text{ and } b_{t+1} \in W, \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $W = \{\texttt{' '}, \texttt{.}, \texttt{,}, \texttt{'\textbackslash n'}, \ldots\}$ is the set of word-boundary bytes.

More precisely, for a 3-byte word like "the", the indicator fires at three positions per occurrence: the `t`, the `h`, and the `e`. But for the purpose of prediction (predicting the output byte b_t), we define it at the prediction position*: the byte* after *the last byte of "the", i.e., position t where $b_{t-3} \in W$, b_{t-2}=`t`, b_{t-1}=`h`, b_t=`e`... but actually we must be careful. Let us define it simply:*

$$\mathbf{1}_{the}(t) = \begin{cases} 1 & \text{if the 3-gram ending at } t{-}1 \text{ is "the" bounded by word boundaries,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

So $\mathbf{1}_{the}(t) = 1$ means: the word just completed before position t was "the", and we are now predicting b_t (which is typically a space).

Definition 4 (Factored count table)**.** *The skip-bigram count table $c_d(x,o)$ factors through E_{the}:*

$$c_d(x,o) = c_d^+(x,o) + c_d^-(x,o), \tag{6}$$

where:

$$c_d^+(x,o) = \big|\{t : b_{t-d} = x,\ b_t = o,\ \mathbf{1}_{the}(t) = 1\}\big|, \tag{7}$$

$$c_d^-(x,o) = \big|\{t : b_{t-d} = x,\ b_t = o,\ \mathbf{1}_{the}(t) = 0\}\big|. \tag{8}$$

These are the count tables conditioned on *whether we are at a "the"-boundary position or not.*

3 The Neutrality Theorem

The introduction of E_{the} is a *partial quotient* in the sense of [**?**]: we divide the event space E by a single binary atom, factoring the joint count into a conditional fiber and an equivalence class.

Theorem 5 (Neutral introduction)**.** *Introducing E_{the} and marginalizing over it recovers the original count table—and therefore all conditionals, lucks, and predictions—exactly.*

Proof. We trace the operation through each step of $E \to N \to Q$.

Step 1 (E: events partition exactly). Every position t with $b_{t-d} = x$ and $b_t = o$ has either $\mathbf{1}_{\text{the}}(t) = 1$ or $\mathbf{1}_{\text{the}}(t) = 0$. The partition is exhaustive and disjoint. No events are created or destroyed.

Step 2 (N: counts add exactly). By (5), $c_d(x, o) = c_d^+(x, o) + c_d^-(x, o)$. This is exact integer arithmetic. The marginals also add: $c_d(x) = c_d^+(x) + c_d^-(x)$ where $c_d^+(x) = \sum_o c_d^+(x, o)$.

Step 3 (Q: quotients compose exactly). The joint luck of the original table is $Q_d(x, o) = N/c_d(x, o)$. The factored luck decomposes via partial quotient [**?**]:

$$\frac{1}{Q_d(x, o)} = P_d(x, o) = \frac{c_d^+(x, o) + c_d^-(x, o)}{N} = \frac{c_d(x, o)}{N}. \tag{9}$$

The conditional luck $Q_d(o \mid x) = c_d(x)/c_d(x, o)$ is unchanged because both numerator and denominator are the *same* sums of the factored counts:

$$Q_d(o \mid x) = \frac{c_d^+(x) + c_d^-(x)}{c_d^+(x, o) + c_d^-(x, o)}. \tag{10}$$

In the GCD decomposition, the row GCD $g_I(x) = \gcd_o c_d(x, o)$ may differ from the GCDs of the sub-tables c_d^+ and c_d^-. But the GCD cancels from the conditional (Proposition **??** of the setup), so the conditional $P_d(o \mid x)$ is the same whether computed from c_d directly or from $c_d^+ + c_d^-$. □

Remark 6 (Why this is trivially true but non-trivially important)**.** *The proof is a consequence of the fact that the partial quotient decomposes $c_d(x, o)$ along a new axis without changing any cell value. But the* point *is what the decomposition enables:*

1. ***We have changed nothing.*** *Predictions are identical. The operation is a pure rearrangement of counts along a new axis.*

2. ***The factored form is richer.*** *We now have $c_d^+(x, o)$ and $c_d^-(x, o)$ as separate objects with their own GCD decompositions, their own reduced counts, and their own conditional lucks.*

3. ***Value comes from asymmetry.*** *If $Q_d(o \mid x, the^+) = Q_d(o \mid x, the^-)$ for all x, o, the factorization is uninformative. If the conditional lucks differ, a model that conditions on E_{the} can make better predictions at those positions. But exploiting this requires* using *E_{the} as a predictor, not just marginalizing over it.*

4 The Bayesian Conditioning Direction

The neutrality theorem is the marginalization direction: summing over E_{the} recovers the original. We now examine the *conditioning* direction via the partial quotient framework [**?**].

Proposition 7 (Bayes via partial quotient)**.** *The posterior $P(the^+ \mid x_d, o)$ is a count ratio—a direct application of the $E \to N \to Q$ chain:*

$$P(the^+ \mid x_d = x, o) = \frac{c_d^+(x, o)}{c_d(x, o)}. \tag{11}$$

This is the partial quotient of E by the atom $(x, o) \in I \times O$, restricted to the$^+$: of all events with input x at offset d and output o, how many fall in the "the" equivalence class?

The corresponding luck:

$$Q(the^+ \mid x, o) = \frac{c_d(x, o)}{c_d^+(x, o)}, \tag{12}$$

the inverse posterior—how "lucky" is it to be at a "the"-boundary given these bytes? Large luck (small posterior) means "the" is unlikely in this context; luck $= 1$ means certainty.

No prior is assumed; no likelihood is computed separately. The posterior is *the count ratio. The GCD of the column $\{c_d^+(x, o), c_d^-(x, o)\}$ (which is just $\gcd(c_d^+, c_d^-)$) would cancel from the conditional, but for a binary partition there is no further reduction needed.*

Example 8 (Posterior for "the" at specific positions)**.** *Consider DSS=1024. Suppose "the " (the-space) occurs 10 times in the 1024-byte sample.*

At position t where $b_{t-1} =$ `e` and $b_t =$ `' '`:

- $c_1($`e`, `' '`$)$*: total count of* `e` *followed by space. Say this is 25 ("the ", "be ", "me ", "he ", etc.).*
- $c_1^+($`e`, `' '`$)$*: count restricted to "the"-boundaries. This is 10 (every "the " contributes one).*
- $P(the^+ \mid$ `e`$_{d=1}$, `' '`$) = 10/25 = 0.40$.

Even after seeing `e` *followed by space, there is only 40% probability that this is a "the" boundary. The other 60% distributes over other words ending in* `e`*.*

Now condition on more offsets. At the same position, also $b_{t-2} =$ `h`:

- *Among the 25 events with* `e`$\rightarrow$`' '`*, how many also have* `h` *at offset 2? Say 15 ("the ", "she ", etc.).*
- *Among those 15, how many are "the"-boundaries? Still 10.*
- $P(the^+ \mid$ `e`$_{d=1}$, `h`$_{d=2}$, `' '`$) = 10/15 = 0.67$.

Adding $b_{t-3} =$ `t`:

- *Among the 15, how many also have* `t` *at offset 3? Say 11 ("the " plus perhaps one "the" in a compound).*
- $P(the^+ \mid$ `e`$_{d=1}$, `h`$_{d=2}$, `t`$_{d=3}$, `' '`$) = 10/11 \approx 0.91$.

The posterior concentrates as more evidence arrives, but never reaches certainty (because other sequences like "...athe " might exist in the sample).

5 "Pulling 'the' Out" of the Skip-Bigrams

Definition 9 (Factoring a count table through a binary ES)**.** *To "pull 'the' out of" the skip-bigram table $c_d(x, o)$ means to decompose it as:*

$$c_d(x, o) = \underbrace{c_d^+(x, o)}_{\text{"the" contribution}} + \underbrace{c_d^-(x, o)}_{\text{everything else}}. \tag{13}$$

The "the" contribution $c_d^+(x, o)$ captures all the counts attributable to positions where "the" just occurred. The remainder $c_d^-(x, o)$ captures everything else.

Proposition 10 (What "the" contributes to specific bigrams)**.** *For the bigram table at offset $d = 1$ (adjacent bytes), the "the" contribution is concentrated on specific entries:*

x	o	*Effect of* c_1^+
`e`	`' '`	*Large: every "the " contributes here*
`h`	`e`	*Moderate: "the" contributes, but so do "he", "she", etc. in the* c_1^- *portion*
`t`	`h`	*Moderate: "the" contributes, but "th" appears in many words*
`' '`	`t`	*Small: "the" is one of many words starting with* `t`

At larger offsets ($d = 2, 3, \ldots$), the "the" contribution is spread more diffusely because the byte at offset d from a "the"-boundary depends on the word before *"the", which varies.*

Theorem 11 (Exact reconstitution)**.** *After pulling "the" out, we can put it back:*

$$c_d(x, o) = c_d^+(x, o) + c_d^-(x, o) \quad \forall\, x, o, d. \tag{14}$$

This is exact (integer equality of counts). The $E \to N$ step is lossless: no events are created, destroyed, or moved between cells. The $N \to Q$ step is also lossless (given the counts, the ratios are determined).

Therefore: the operation of introducing E_{the}, factoring all count tables through it, and then marginalizing E_{the} back out is the identity. It is a neutral rearrangement.

6 The Negative Event Distribution

Definition 12 (The the$^-$ distribution)**.** *The negative event the$^-$ is "not at a 'the' boundary." Its count table $c_d^-(x, o)$ is a mixture:*

$$c_d^-(x, o) = \sum_{w \in \mathcal{W} \setminus \{the\}} c_d^{(w)}(x, o) + c_d^{(none)}(x, o), \tag{15}$$

where $c_d^{(w)}$ restricts to positions at word w's boundary and $c_d^{(none)}$ restricts to non-word-boundary positions (mid-word bytes).

For the "the"-prefixed subset:

$$c_d^{(the\text{-}prefix)}(x, o) = c_d^{(there)}(x, o) + c_d^{(them)}(x, o) + c_d^{(then)}(x, o) + c_d^{(their)}(x, o) + \cdots \tag{16}$$

Remark 13 (Model for negative weight distribution)**.** *This decomposition of the$^-$ is significant beyond the specific case of "the." It models how negative support distributes generally in the UM:*

1. *In the RNN, negative weights $w_{ij} < 0$ encode "not this event." The neuron h_j receiving negative input from h_i is being told: "the pattern that activates h_i is* absent*," and this absence is informative.*

2. *The the$^-$ event has the same structure: it says "the" did not occur, and this non-occurrence is informative because it tells us the byte patterns are explained by something* other than *"the" — likely another word in the prefix class.*

3. *The distribution of the$^-$ over the prefix class (65% "the", 8% "there", 7% "their", etc.—so among the the$^-$ events at the "th-e" continuation point, roughly 23% "there", 20% "their", etc.) provides a concrete model for how negative RNN weights distribute their "not-this" information over alternative patterns.*

4. *When we eventually introduce the full lexicon as an M-ary event space $E_V = \{w_1, w_2, \ldots, w_M, OOV\}$, each word's negative event decomposes the same way. The "the" case is the first instance of a general pattern.*

Proposition 14 (Negative event is computable from data)**.** *The negative distribution is computable purely from counts:*

$$P(w' \mid the^-, prefix\ context) = \frac{c^{(w')}}{c^- - c^{(non\text{-}prefixed)}}, \tag{17}$$

where $c^{(w')}$ is the count of word w' events and the denominator restricts to events sharing the relevant prefix context. No model fitting is needed—it is raw counting over the data, exactly the $E \to N \to Q$ chain.

7 Why Start with the Sat-RNN (DSS=1024)?

Remark 15 (The controlled setting)**.** *DSS=1024 is a 1024-byte sample. The sat-rnn trained on it achieves 0.079 bpc (near-memorization). The isomorphic UM reproduces this exactly. This tiny dataset makes the neutral introduction maximally transparent:*

1. ***All counts are small and inspectable.*** *"The" occurs perhaps 5–15 times in 1024 bytes. Each entry of c_d^+ is a single-digit number. We can write out the full factored count table by hand.*

2. ***The neutrality is exactly verifiable.*** *We can sum $c_d^+ + c_d^-$ and verify integer equality with c_d for every (x, o, d) triple.*

3. ***The model already captures everything.*** *At 0.079 bpc, the sat-rnn has nearly memorized the sample. Introducing E_{the} and marginalizing out must give exactly 0.079 bpc. Any deviation would indicate a bug in the factorization, not a real change.*

4. ***Spurious patterns are visible.*** *The sat-rnn at DSS=1024 has learned some long-range patterns that are specific to this particular 1024-byte sample. Introducing E_{the} may "break" these spurious patterns by explaining away some of their count support. If position t has a long-range correlation with position $t - 20$ that happens to co-occur with "the" at position t, factoring out the "the" contribution removes this correlation from the c_d^- table. The residual table c_d^- will have the spurious pattern weakened. This is not a prediction improvement (the marginal is unchanged) but a* structural *improvement: the count table is cleaner.*

8 When Does Value Arrive?

Proposition 16 (Neutral now, valuable later)**.** *The neutral introduction (Section 3) gives $\Delta bpc = 0$ by construction. Value arrives when we add* intra-lexical patterns*—patterns that connect atomic byte events to lexeme events:*

1. ***From bytes to lexeme*** *(recognition patterns): $P(the^+ \mid x_{d_1}, x_{d_2}, \ldots)$. These patterns say: "given these bytes at these offsets, how likely is it that 'the' is occurring?" Using these as predictors sharpens the model at positions where "the" is likely.*

2. ***From lexeme to bytes*** *(prediction patterns): $P(o \mid the^+)$. These patterns say: "given that 'the' just occurred, what byte comes next?" Using these as predictors sharpens the model at the exit position of "the."*

3. ***Between lexemes*** *(lexeme-level patterns): $P(of^+ \mid the^+)$. These patterns capture word-level co-occurrence. Using these as predictors sharpens the model at word-to-word transitions.*

Each type of intra-lexical pattern provides $\Delta bpc > 0$ by exploiting the asymmetry between $P_d(o \mid x, the^+)$ and $P_d(o \mid x, the^-)$. But none of this is possible without first establishing the neutral factorization—the bookkeeping that separates c_d^+ from c_d^-.

Remark 17 (The factorization is the architecture)**.** *Introducing E_{the} without using it for prediction is an* architectural *change with no* parametric *change. The event space has grown (from E_0 to $E_0 \times E_{the}$), but the predictions are identical because we marginalize over the new dimension.*

This is the tock step in its purest form: an expansion of the architecture that is initially neutral, creating the space *for future improvement. The tick step (adding intra-lexical patterns, recomputing conditionals) is what fills this space.*

The analogy with neural networks: adding a neuron initialized to zero changes nothing. But the neuron is now available to learn. Here, adding E_{the} changes nothing, but the factored count tables are now available for lexeme-level prediction.

9 The Exact Procedure

For completeness, we specify the exact procedure for introducing "the" into the DSS=1024 model:

1. **Identify "the" occurrences.** Scan the 1024-byte sample for occurrences of "the" bounded by word-boundary bytes. Record the set of positions $S_{\text{the}} = \{t : \mathbf{1}_{\text{the}}(t) = 1\}$.

2. **Factor each count table.** For each offset d and each (x, o) pair, compute: $c_d^+(x, o)$ by restricting counts to S_{the}, $c_d^-(x, o) = c_d(x, o) - c_d^+(x, o)$.

3. **Verify neutrality.** Compute predictions from the marginal: for each x, $\hat{P}_d(o \mid x) = (c_d^+(x, o) + c_d^-(x, o))/n(x)$. Verify $\hat{P}_d(o \mid x) = P_d(o \mid x)$ exactly. Compute bpc. Verify it equals 0.079 bpc.

4. **Inspect the factored tables.** Examine c_d^+ and c_d^- separately. Identify entries where $P_d(o \mid x, \text{the}^+) \neq P_d(o \mid x, \text{the}^-)$ (the asymmetry that future lexeme patterns can exploit).

5. **Compute the negative distribution.** For positions in S_{the}^c (not "the"), tabulate which words are present. Compute the distribution of the^- over the prefix class and over other words.

6. **Report.** The output is: (a) the factored count tables, (b) the neutrality verification, (c) the asymmetry table, (d) the negative event distribution. No predictions change.

10 Discussion

This paper is deliberately minimal. We introduce one word, prove the introduction changes nothing, and characterize the negative event. No model improvement is claimed or attempted.

The purpose is threefold:

1. **Establish the bookkeeping.** The factored count tables c_d^+, c_d^- are the data structure on which all future lexeme-level operations will be built. Getting this right—with exact integer arithmetic, no approximations, no off-by-one errors—is essential.

2. **Model negative event distribution.** The the$^-$ event is the first concrete instance of a negative event with internal structure. Understanding how it distributes over the prefix class (and over the full lexicon) will generalize to all words and, by analogy, to negative weights in the RNN.

3. **Prepare for value.** The next paper will add intra-lexical patterns and measure the resulting Δbpc. This paper ensures that the baseline is clean: we know exactly what the model predicts before any lexeme-level patterns are added.

The neutral introduction is the "zero" of the tock step: the point where architecture has expanded but performance has not. Everything that follows is measured against this zero.

References

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

[2] Claude and MJC. *Doubled-E: Exact Float UM Isomorphic to RNN*. Hutter archive, 30 Jan 2026.

[3] Claude and MJC. *Pattern Chains: Explicit $i \to \cdots \to o$ from Data*. Hutter archive, 7 Feb 2026.

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

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