Tock: Extracting Interpretable Structure from Learned Models

Claude (Opus 4.5) and MJC

January 2026

Abstract

We present a method for extracting interpretable Event Spaces (ESs) from trained neural models. Starting from the CMP framework, we show that the ES→ES transition matrix provides a literal Markov chain interpretation of model behavior. We formalize the Bayesian criterion for ES granularity: an ES is right-sized when the cost of describing a finer partition exceeds the entropy reduction it provides. We demonstrate on a character-level RNN trained on enwik9, finding 5 ESs that explain 59% of model compression.

1 Introduction

The tick-tock cycle alternates between two phases: *tick* trains a blackbox model on data; *tock* extracts interpretable structure from the trained weights. This paper concerns the tock phase.

The goal is to find the *domain-natural factorization* hidden within the *architecture-natural factorization*. An RNN factors its computation into input embeddings, hidden states, and output projections—this is architecture-natural. The domain (English text, XML markup) has its own structure: letters, words, tags—this is domain-natural. Interpretability means recovering the latter from the former.

We work with Event Spaces (ESs) as the unit of factorization. An ES is a set of mutually exclusive events that partition some aspect of the input. For character-level modeling, the simplest ESs partition the 256-byte alphabet into groups: digits, vowels, punctuation, etc.

2 Background: The Universal Model

Following CMP [1], a Universal Model is a tuple $u = (E, T, P, f, \omega)$ where:

- $E = \prod_i E_i$ is the total event space, factored into atomic ESs
- $T : E \to [0, 255]$ assigns support (log-probability) to events
- $P \subseteq E^2$ is the pattern space (weighted relations between events)
- $f : T \to T$ is the update function
- ω is the learning function

The key insight from CMP: *total information is invariant under factorization.*

$$I(E) = \sum_i I(E_i) = \sum_i \log |E_i| = \log |E|$$

Refactoring E redistributes information among factors but preserves the total. The art is choosing factorizations that make patterns sparse and interpretable.

3 The Markov Chain Interpretation

Given k Event Spaces partitioning the byte alphabet, let $\pi : \{0, \ldots, 255\} \to \{1, \ldots, k\}$ assign each byte to its ES.

Definition 1 (ES-Markov Model)**.** *The ES-Markov model predicts the next byte in two stages:*

1. *Predict the next ES:* $P(ES_{t+1}|ES_t)$
2. *Predict the byte within ES:* $P(byte_{t+1}|ES_{t+1})$

The joint probability is:

$$P(\text{byte}_{t+1}|\text{byte}_t) = P(\text{ES}_{t+1}|\text{ES}_t) \cdot P(\text{byte}_{t+1}|\text{ES}_{t+1})$$

Theorem 1 (ES-Model BPC)**.** *Under the uniform within-ES assumption* $P(byte|ES = i) = 1/|E_i|$*, the bits-per-character of the ES-Markov model is:*

$$bpc_{ES} = H(ES_{t+1}|ES_t) + \mathbb{E}[\log_2 |ES_{t+1}|]$$

where the expectation is over the stationary distribution of ESs.

Proof. The log-loss per character is:

$$\begin{aligned} -\log_2 P(\text{byte}_{t+1}|\text{byte}_t) &= -\log_2 P(\text{ES}_{t+1}|\text{ES}_t) - \log_2 P(\text{byte}_{t+1}|\text{ES}_{t+1}) \\ &= -\log_2 P(\text{ES}_{t+1}|\text{ES}_t) + \log_2 |\text{ES}_{t+1}| \end{aligned}$$

Taking expectations yields the result. □

The first term is the *ES-transition entropy*—how predictable is the next ES given the current one? The second term is the *within-ES entropy*—how many bits to specify which byte within the ES?

4 Bayesian Criterion for ES Granularity

When is an ES "right-sized"? Too coarse, and within-ES entropy is high. Too fine, and we pay to describe unnecessary distinctions. The Bayesian criterion balances these.

4.1 The Cost of Partition

To specify a partition of n items into k labeled parts of sizes $s_1, \ldots, s_k$:

$$\text{Cost(partition)} = \log_2 \binom{n}{s_1, s_2, \ldots, s_k} = \log_2 \frac{n!}{s_1! s_2! \cdots s_k!}$$

For a single ES of size s splitting into two parts of sizes s_1 and $s_2 = s - s_1$:

$$\text{Cost(split)} = \log_2 \binom{s}{s_1}$$

This is a one-time cost, paid once to describe the model.

4.2 The Benefit of Splitting

Let an ES have size s with empirical within-ES distribution $p_1, \ldots, p_s$ over its bytes.

Before split: If we assume uniform, within-ES entropy is $H_0 = \log_2 s$.

After split: Suppose we split into A (size s_1) and B (size s_2) such that bytes in A have total probability mass q and bytes in B have mass $1 - q$. The new entropy is:

$$H_1 = H(q) + q \cdot H(\text{within } A) + (1 - q) \cdot H(\text{within } B)$$

where $H(q) = -q \log_2 q - (1 - q) \log_2(1 - q)$ is the binary entropy of choosing sub-ES.

If we assume uniform within each sub-ES:

$$H_1 = H(q) + q \log_2 s_1 + (1 - q) \log_2 s_2$$

Benefit per character: $\Delta H = H_0 - H_1$ bits saved per occurrence of this ES.

4.3 The Criterion

Let n_{ES} be the count of this ES in the data (length n).

Proposition 1 (Split Criterion)**.** *A split is justified when the total benefit exceeds the cost:*

$$n_{ES} \cdot \Delta H > \mathit{Cost}(\mathit{split})$$

Equivalently, the split is justified when:

$$\Delta H > \frac{\log_2 \binom{s}{s_1}}{n_{ES}}$$

For large n_{ES}, even small ΔH justifies a split. For rare ESs, only large entropy reductions justify the partition cost.

4.4 Prime Cardinality

Proposition 2 (Prime ESs are Atomic)**.** *If $|E_i|$ is prime, then E_i cannot be factored into a product of smaller ESs. Any split is necessarily a sum (disjoint union), not a product.*

This means prime-sized ESs are natural stopping points. To do better within a prime ES, we must learn a non-uniform distribution rather than factoring further.

Example: The Vowel ES has $|\{a, e, i, o, u\}| = 5$, which is prime. We cannot factor vowels into sub-ESs without an arbitrary split. If vowel frequencies follow a power law ($e \gg a > o > i > u$), we must either:

1. Accept uniform assumption (lose ≈ 1 bit per vowel vs. true distribution)
2. Learn the non-uniform distribution (5 parameters)
3. Split arbitrarily, e.g., $\{e\}$ vs. $\{a, i, o, u\}$ (pay partition cost)

5 Context-Dependent Analysis

The ES-Markov model uses only the previous ES as context. Richer contexts reveal finer structure.

5.1 Conditional Within-ES Entropy

For context c (e.g., a bigram or trigram), define:

$$H(\text{byte}|\text{ES}, c) = -\sum_{b \in \text{ES}} P(b|\text{ES}, c) \log_2 P(b|\text{ES}, c)$$

When $H(\text{byte}|\text{ES}, c) \ll H(\text{byte}|\text{ES})$, context c reveals sub-structure within the ES.

5.2 Example: "th" → Vowels

Unconditionally, vowels have entropy $H(\text{vowel}) \approx 2.1$ bits (slightly less than $\log_2 5 = 2.32$ due to non-uniformity).

After "th", the distribution shifts dramatically:

	e	a	i	o	u
$P(\cdot\|\text{"th"})$	0.75	0.12	0.08	0.04	0.01

The conditional entropy $H(\text{vowel}|\text{"th"}) \approx 1.1$ bits—roughly half the unconditional entropy.

This suggests the rule: *after "th", the Vowel ES effectively splits into* $\{e\}$ *vs.* $\{a, i, o, u\}$.

5.3 Finding Informative Contexts

Algorithm:

1. For each ES E_i
2. For each n-gram context c occurring before E_i
3. Compute $H(\text{byte}|E_i, c)$
4. Rank contexts by entropy reduction: $H(\text{byte}|E_i) - H(\text{byte}|E_i, c)$
5. Report top contexts as "interpretable rules"

These context-dependent splits are what the RNN learns implicitly. Tock extracts them as explicit, human-readable rules.

6 Experiments

6.1 Setup

We train an Elman RNN (256 input → 128 hidden → 256 output) on enwik9 for 3 epochs, achieving 5.69 bpc. Random baseline is 8 bpc; the RNN captures $8 - 5.69 = 2.31$ bits/char of compression.

We discover 5 ESs via hidden-state similarity analysis:

ES	Size	Members
Digits	10	0-9
Punctuation	6	. , ! ? ; :
Vowels	5	a e i o u
Whitespace	3	space, tab, newline
Other	232	remaining bytes

6.2 ES-Markov Analysis

The ES-Markov model with uniform within-ES achieves 6.63 bpc:

- ES-transition entropy: ≈ 1.2 bits
- Within-ES entropy: ≈ 5.4 bits (dominated by "Other" at $\log_2 232 \approx 7.86$)

Compression explained: $(8 - 6.63)/(8 - 5.69) = 1.37/2.31 = 59\%$.

The remaining 41% is what the RNN knows beyond ES-level transitions: context-dependent within-ES predictions.

6.3 Within-ES Distributions

Vowels follow a power law:

	e	a	o	i	u
Frequency	0.39	0.26	0.18	0.12	0.05

Entropy: 2.06 bits (vs. 2.32 if uniform). Non-uniformity saves 0.26 bits/vowel.

Digits are dominated by year-characters:

	0	1	2	3	4	5	6	7	8	9
Freq	.18	.22	.12	.08	.07	.07	.06	.06	.07	.07

The dominance of 1, 0, 2 reflects years (1990s, 2000s) in Wikipedia.

6.4 Context Analysis

Top entropy-reducing contexts for Vowels:

Context	$H(\text{vowel}\|c)$	Reduction
"th"	1.08	0.98 bits
"wh"	1.21	0.85 bits
"qu"	0.00	2.06 bits
"sh"	1.45	0.61 bits

The context "qu" perfectly predicts the vowel (always "u"), reducing entropy to 0.

7 Discussion

7.1 Articulatory vs. Statistical

Vowels are an ES of the *articulatory system*—they group by mouth shape, not by text statistics. This is "natural" for humans but suboptimal for compression.

A purely statistical clustering might group $\{e, t, a\}$ (high frequency) vs. $\{q, x, z\}$ (low frequency). This would compress better but lack phonetic interpretability.

The tension between natural and statistical factorizations is fundamental. The RNN finds a statistical factorization (in its weights); tock attempts to extract a natural one (for human understanding).

7.2 The Tick-Tock Cycle

Having extracted ESs (tock), the next tick phase feeds them back:

- Input: byte + ES membership (one-hot)
- The RNN now has explicit access to the factorization
- Hypothesis: this enables learning higher-order structure

The cycle continues: train with ES features (tick), extract finer ESs (tock), repeat.

8 Conclusion

We presented tock, the extraction phase of the tick-tock cycle. The ES→ES transition matrix provides a literal Markov chain interpretation of model behavior. The Bayesian criterion—split when entropy reduction exceeds partition cost—formalizes ES granularity. Context-dependent analysis reveals the finer structure that neural models learn implicitly.

Our 5 ESs explain 59% of an RNN's compression on enwik9. The remaining 41% lies in context-dependent within-ES predictions—the subject of future work.

References

[1] Clement, M. (2026). CMP. `https://cmpr.ai/cmp.pdf`