# Q1 Protocol B: Exact Pattern Census via the Superset UM

Claude and MJC

11 February 2026

**Abstract**

Protocol A (q1-sparsity.pdf) counted active patterns by tracing the f32 backward gradient through the trained RNN. Protocol B counts patterns directly from data in the superset UM, with no reference to the RNN's weights. Comparing the two reveals which patterns the RNN learned from the data and which it missed due to capacity constraints (128 neurons, f32, BPTT-50). The entropy coding identity shows that Hebbian learning (log-stochastic counting) is the update rule that builds the UM's codebook—given the right event spaces and a log representation of sensory inputs.

## 1 The Superset UM from Data

The superset UM $u_{\text{sup}}$ has event spaces $E_T \times E_{\text{in}} \times E_{\text{out}}$ (after marginalizing out hidden events). A pattern in this space is a co-occurrence: input byte $b$ at offset $d$ predicting output byte $y$.

For each offset $d = 1, \ldots, D_{\max}$ and each pair $(b, y)$, the pattern strength is:

$$\omega(b, y, d) = \log \frac{n(b, y, d)}{n(b, d)} \tag{1}$$

where $n(b, y, d) = |\{t : x_{t-d} = b \text{ and } x_{t+1} = y\}|$ is the co-occurrence count and $n(b, d) = |\{t : x_{t-d} = b\}|$ is the marginal count. This is the conditional log-probability: the strength of evidence that seeing $b$ at offset $d$ provides for predicting $y$.

A pattern is *significant* if $|\omega(b, y, d)| > \tau$—it provides evidence that deviates from the marginal. (Positive $\omega$: $b$ at offset $d$ makes $y$ more likely. Negative: less likely.)

### 1.1 The log flow

Consider a single pattern $p = (b, y, d)$: "byte $b$ at offset $d$ predicts byte $y$." We trace the exact counting flow.

After processing $N$ positions, we have counts $n(b, y, d)$ and $n(b, d)$. The exact log-probability is:

$$\omega(p) = \log \frac{n(b, y, d)}{n(b, d)} = \log n(b, y, d) - \log n(b, d)$$

This is the pattern's contribution to the arithmetic code: encoding $y$ given context $b$ at offset $d$ costs $-\omega(p)/\ln 2$ bits. The UM is the codebook.

To build this codebook, we need only:

1. A table of counts $n(b, y, d)$, one per pattern.
2. A marginal $n(b, d) = \sum_y n(b, y, d)$.

3. The log ratio.

The prediction at position $t$ uses:

$$P(y \mid x_{t-d} = b) = \frac{n(b, y, d)}{n(b, d)} = \frac{\exp(\omega(b, y, d))}{\sum_{y'} \exp(\omega(b, y', d))}$$

This is a softmax over pattern strengths—identical in form to the RNN's output layer.

### 1.2 Log-stochastic counting (Hebbian learning)

Now replace exact counting with an online update. Initialize $\omega(p) = 0$ for all patterns. At each position $t$, for each active pattern $p = (x_{t-d}, x_{t+1}, d)$:

$$\omega(p) \leftarrow \omega(p) + \eta$$

After $N$ positions, $\omega(p) = \eta \cdot n(p)$. The softmax gives:

$$P(y \mid b, d) = \frac{\exp(\eta \cdot n(b, y, d))}{\sum_{y'} \exp(\eta \cdot n(b, y', d))}$$

This is a Boltzmann distribution over counts. The temperature is $1/\eta$:

- As $\eta \to 0$: uniform (no learning).
- As $\eta \to \infty$: argmax (memorization).
- At $\eta = 1/n(b, d)$: first-order approximation to the exact log-probability.

The key point: this Hebbian update—fire together, wire together, in log space—*is* counting. The pattern strength $\omega(p)$ is proportional to the count $n(p)$, and the softmax normalizes it into a probability. No gradient, no backpropagation, no loss function. The only requirements are:

1. The right event spaces (the LPP determines what counts as a "pattern").
2. A log representation of sensory inputs (so that addition in $\omega$-space corresponds to multiplication in probability space).

*Slogan: Hebbian learning is all you need,* given these two prerequisites.

## 2 Method

### 2.1 Single-offset census

For each offset $d = 1, \ldots, 50$:

1. Count all $(b, y)$ co-occurrences at offset $d$.
2. Compute $\omega(b, y, d)$ for each pair.
3. Count patterns with $|\omega| > \tau$ for each threshold.

This gives the number of data-significant patterns at each offset, independent of any model.

### 2.2 Multi-offset census (skip-$k$-gram)

A compound pattern combines evidence from multiple offsets: $(b_1, d_1), (b_2, d_2), \ldots, (b_k, d_k) \to y$. The strength is:

$$\omega(b_1{:}d_1, \ldots, b_k{:}d_k, y) = \log \frac{n(b_1{:}d_1, \ldots, b_k{:}d_k, y)}{n(b_1{:}d_1, \ldots, b_k{:}d_k)} \tag{2}$$

Naively enumerating all $k$-offset patterns requires $\binom{D_{\max}}{k}$ offset sets, each with up to $52^k$ byte tuples—combinatorially intractable. The greedy algorithm selects offsets one at a time, choosing at each step the offset $d^*$ that maximizes the conditional mutual information $I(X_{t-d^*}; X_{t+1} \mid X_{t-d_1}, \ldots, X_{t-d_{k-1}})$ given the already-selected offsets. We claim this procedure is **complete**: it finds all patterns in $P_{\text{sup}}$.

**Completeness proof.** Every pattern in $P_{\text{sup}}$ over observable events is a conditional co-occurrence at some set of offsets $S = \{d_1, \ldots, d_k\} \subseteq \{1, \ldots, D_{\max}\}$. The greedy algorithm produces an ordered sequence of offsets $(d_1^*, d_2^*, \ldots, d_{D_{\max}}^*)$. After all $D_{\max}$ steps, the selected set equals $\{1, \ldots, D_{\max}\}$—every offset is included. Therefore every pattern in $P_{\text{sup}}$ is eventually enumerated.

The nontrivial content is that the greedy *ordering* ensures efficient coverage. Mutual information conditioned on a set of random variables is submodular:

$$I(X_d; Y \mid X_S) \geq I(X_d; Y \mid X_T) \quad \text{for } S \subseteq T$$

(conditioning on more variables can only reduce residual information). The greedy algorithm for submodular maximization satisfies

$$I(Y; X_{S_k}) \geq (1 - e^{-1}) \cdot I(Y; X_{S_k^*})$$

where $S_k$ is the greedy selection of $k$ offsets and $S_k^*$ is the optimal selection of $k$ offsets [1]. This means:

1. At each step, the greedy choice captures the most residual information—it prioritizes offsets whose patterns are most informative given what is already known.
2. After $k$ steps, the selected offsets capture at least 63% of the MI achievable by any $k$ offsets. In practice, much more: our earlier experiments showed skip-4 at $[1, 8, 20, 3]$ achieving 0.069 bpc (712 patterns) vs contiguous order-12 at 0.067 bpc (6180 patterns).
3. After $D_{\max}$ steps, the procedure is trivially complete.

The greedy algorithm is therefore both complete (it misses nothing) and efficient (it finds the most informative patterns first).

### 2.3 Comparison with Protocol A

For the single-offset census, we can directly compare:

- Protocol A: how many $W_x$ patterns are active at offset $d$ in the backward trace (mean over positions)?
- Protocol B: how many $(b, y, d)$ patterns have $|\omega| > \tau$ in the data?

If the RNN has learned the data's patterns, the Protocol A count at each offset should track the Protocol B count. Where Protocol A exceeds Protocol B, the RNN is using patterns not supported by data (hallucination or higher-order effects). Where Protocol B exceeds Protocol A, the RNN lacks capacity to represent available patterns.

# 3 Results

The data has 52 distinct byte values in 1024 positions.

## 3.1 Single-offset pattern counts

At each offset $d$, we count $(b, y)$ co-occurrence pairs and their conditional log-probability $\omega = \log P(y \mid x_{t-d} = b)$.

| Offset $d$ | Pairs | $\lvert\omega\rvert>1$ | $\lvert\omega\rvert>2$ | Mean $\lvert\omega\rvert$ |
|---|---|---|---|---|
| 1 | 281 | 236 | 188 | 2.48 |
| 2 | 310 | 271 | 208 | 2.49 |
| 4 | 351 | 312 | 245 | 2.57 |
| 8 | 396 | 363 | 284 | 2.68 |
| 12 | 426 | 396 | 318 | 2.74 |
| 20 | 426 | 394 | 318 | 2.76 |
| 50 | 469 | 443 | 375 | 2.76 |

Pattern counts grow slowly with offset (281 at $d$=1 to 469 at $d$=50) because longer offsets see more distinct byte pairs. Mean $|\omega|$ is high ($\sim$2.5–2.8) everywhere—on 1024 bytes, most co-occurrences are strongly informative.

## 3.2 PMI census

Using pointwise mutual information $\text{PMI}(b, y, d) = \log \frac{P(b,y,d)}{P(b,d) \cdot P(y)}$ gives a more meaningful measure of pattern significance (deviation from independence):

| Offset $d$ | Pairs | $\lvert\text{PMI}\rvert>1$ | $\lvert\text{PMI}\rvert>2$ | Mean $\lvert\text{PMI}\rvert$ |
|---|---|---|---|---|
| 1 | 281 | 188 | 96 | 1.65 |
| 2 | 310 | 185 | 95 | 1.62 |
| 4 | 351 | 197 | 91 | 1.44 |
| 8 | 396 | 200 | 95 | 1.25 |
| 12 | 426 | 198 | 83 | 1.19 |
| 20 | 426 | 183 | 77 | 1.12 |
| 50 | 469 | 203 | 80 | 1.10 |

PMI decays with offset: nearby bytes are more informative (mean $|\text{PMI}| = 1.65$ at $d$=1 vs 1.10 at $d$=50). But even at $d$=50, roughly 200 patterns exceed $|\text{PMI}| > 1$—the data has long-range structure.

## 3.3 Protocol A vs B comparison

At each offset, we compare the number of data patterns ($|\omega| > 0$) to the mean number of active $W_x$ patterns in the RNN backward trace ($\tau = 0.01$):

| Offset $d$ | Data patterns | RNN $W_x$ active | Ratio |
|---|---|---|---|
| 1 | 265 | 3.0 | 0.011 |
| 2 | 297 | 3.1 | 0.010 |
| 4 | 340 | 3.6 | 0.011 |
| 8 | 387 | 4.4 | 0.011 |
| 12 | 414 | 4.9 | 0.012 |
| 20 | 415 | 5.5 | 0.013 |

The ratio is remarkably stable at ∼1.1%: at each offset, the RNN uses roughly 1 in 90 of the data-available patterns. This is the 128-neuron bottleneck in action—the RNN cannot represent all data patterns simultaneously, so it selects a sparse subset. The ratio increases slightly with depth (0.011 to 0.013), suggesting the RNN allocates proportionally more capacity to longer-range patterns.

## 4 Discussion

Protocol B reveals three things:

1. **The data is pattern-rich at all offsets.** Even at $d = 50$, there are ∼460 co-occurrence patterns with ∼200 exceeding $|\text{PMI}| > 1$. The superset UM $P_{\text{sup}}$ is large.

2. **The RNN selects ∼1% of available patterns.** The ratio of RNN-active to data-available $W_x$ patterns is stable at ∼0.011 across all offsets. This is the capacity constraint: $P_{\text{iso}} \subset P_{\text{sup}}$ and the 128-neuron bottleneck determines the inclusion.

3. **PMI decay predicts the RNN's offset allocation.** Mean |PMI| decays from 1.65 to 1.10 with offset, but the RNN's $W_x$ attribution mass (from Protocol A) peaks at $d \approx 20$. The RNN compensates for PMI decay by mixing information through recurrent dynamics—the $W_h$ patterns amplify weak long-range signals.

## Reproducibility

Tool: `q1_protocol_b.c` in `docs/archive/20260211/`. Data: first 1024 bytes of enwik9.

## References

[1] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14:265–294, 1978.