Consolidation in the Universal Model
The GCD, the Differential ω, and the Discovery of Structure

Claude and MJC — March 2026 (v2)

1. The Problem

The organism has a model and a memory trace. The model is the learned structure (E, P, T): event spaces, patterns, the thought vector. The memory trace is the compressed residual—what the model cannot predict, encoded via range coding against the model's predictions [6]. Model plus trace recovers the original experience exactly.

Can the organism discover, from its trace alone, that certain *joint events* compress the trace? Can it discover bigrams from bytes, words from bigrams, phrases from words—not by re-reading the original data, but by replaying its own memory and finding structure in it?

This is consolidation. The organism replays its experience, discovers repeated patterns, and reorganizes its memory around higher-level units. The process is domain-independent: it operates on the UM's joint count tables, not on any specific alphabet or language.

2. The GCD Decomposition

The mathematical foundation for consolidation is the GCD decomposition of the joint count table, developed in [3].

2.1. The joint count table

An OBSERVE LPP between event spaces A and B records the joint count table $c(a, b) = |\{t : (a_t, b_t) = (a, b)\}|$. This is the output of the standard learning function ω_0: log-stochastic counting with $s(a, b) \approx \log_2 c(a, b)$.

By the sufficiency result [3], this single table is the sufficient statistic for both $P(B \mid A)$ and $P(A \mid B)$. No additional data is needed to reverse the direction.

2.2. Common and differential evidence

The row GCD $g_A(a) = \gcd_{b:c(a,b)>0} c(a, b)$ separates each row into two parts:

$$c(a, b) = \underbrace{g_A(a)}_{\text{common evidence}} \cdot \underbrace{r_A(a, b)}_{\text{differential evidence}}, \qquad \gcd_b r_A(a, b) = 1. \tag{1}$$

The **common evidence** $g_A(a)$ is the largest integer that divides every nonzero count in row a. It tells us about the prevalence of input a without distinguishing which output accompanied it. The **reduced counts** $r_A(a, b) = c(a, b)/g_A(a)$ are the irreducible evidence—the part that distinguishes output b from the other outputs. They are coprime across the row.

The conditional probability depends only on the reduced counts:

$$P(b \mid a) = \frac{r_A(a,b)}{R_A(a)}, \qquad R_A(a) = \sum_b r_A(a,b). \tag{2}$$

The GCD cancels completely [3]. The conditional is insensitive to how often a occurred; it depends only on the *shape* of the conditional distribution.

2.3. Bayes from the GCD bridge

The same table admits a column-side decomposition: $c(a,b) = g_B(b) \cdot r_B(a,b)$ with $\gcd_a r_B(a,b) = 1$. Equating the two decompositions gives the GCD bridge:

$$\frac{r_A(a,b)}{r_B(a,b)} = \frac{g_B(b)}{g_A(a)}. \tag{3}$$

From this, Bayes' theorem follows as a consistency condition [3]: $P(b \mid a)/P(a \mid b) = P(b)/P(a)$. The two partial quotients of the joint count table must agree because they decompose the same integer.

3. The Differential ω

3.1. PMI from the GCD decomposition

The pointwise mutual information is:

$$\mathrm{PMI}(a;b) = \log_2 \frac{c(a,b) \cdot N}{c(a) \cdot c(b)} = \log_2 \frac{r_A(a,b) \cdot N}{R_A(a) \cdot c(b)}. \tag{4}$$

By the GCD bridge, both the row-side and column-side reduced counts give the same PMI. The global GCD $g = \gcd_{a,b} c(a,b)$ cancels entirely. The PMI depends on the *shape* of the counts (the reduced parts), not their *scale* [3].

In the UM's log-stochastic representation, where $s \approx \log_2 n$:

$$\Delta\omega(a,b) \;=\; s_{ab} - s_a - s_b + s. \tag{5}$$

This is integer arithmetic on support values. We call this the **differential** ω: it is the new information in the joint event beyond what the marginals already carry.

3.2. Exact vs. approximate

Formula (5) is exact when the supports are exact log-counts. The UM's ω_0 learning function gives approximate log-counts: it increments s with probability 2^{-s}, so $\mathbb{E}[s] \approx \log_2(\text{count})$ but with variance.

This approximation matters. Exp 4 of the English context experiments [7] showed that independently learned supports can violate algebraic constraints: 46.7% of per-context entries had $s_{\text{context}} > s_{\text{marginal}}$, making LSA subtraction catastrophic (+5.0 bpc regression). The fix is to derive the marginal from the per-context values via LS-addition (which guarantees $s_{\text{marginal}} \geq \max_c s_c$ by construction), reducing the regression to $+0.3\,\text{bpc}$ [7].

For the differential ω, the practical consequence: at low counts ($s < 5$, fewer than $\sim$32 observations), the stochastic support can give the wrong sign for PMI. The threshold $\tau \approx 4$ ($\sim$16 observations) provides a natural floor, but the PMI estimate at threshold is noisy. Reliable evaluation of $\Delta\omega$ requires either higher thresholds or synthetic marginals computed via LS-addition from the OBSERVE LPP's own entries.

3.3. The learning function ω and the differential

The standard learning function ω_0 [1] records raw joint counts. It is a specific instance of the general ω, which maps memory traces to model mutations: $\omega : u \times u \to u$, including architecture changes (adding events, event spaces, connections), not only weight adjustments [9].

The **differential** ω is not a replacement for ω_0. It is a derived quantity: given the raw counts from ω_0, the differential ω extracts the residual information beyond the marginals. The relationship is:

- ω_0 records: s_{ab} (the raw joint support).
- The marginals give: $s_a + s_b - s$ (the prediction from independence).
- The differential is: $\Delta\omega = s_{ab} - (s_a + s_b - s)$ (what's new).

The consolidation decision uses $\Delta\omega$ to evaluate whether promotion is worthwhile. The actual learning that produces the counts is still ω_0.

4. The MDL Criterion

4.1. Total information

The total information of a UM with memory trace is [6]:

$$I_{\text{total}} = I_{\text{model}} + I_{\text{trace}}. \tag{6}$$

The model is the SN file (event space declarations, LPP entries, pattern weights). The trace is the range-coded residual stream. Under the null model (base 256, no predictions), the trace is the raw file. As the model improves, the trace shrinks faster than the model grows—this is the definition of learning real structure.

4.2. The promotion criterion

Promotion—making an OBSERVE LPP active and potentially creating new events—changes both terms. The trace shrinks because predictions become sharper; the model grows because new structure is added.

The per-occurrence trace gain at a position where joint event (a, b) fires is the PMI:

$$-\log_2 p_{\text{old}}(b) + \log_2 p_{\text{new}}(b \mid a) = \text{PMI}(a; b). \tag{7}$$

The total gain over all occurrences is $n_{ab} \cdot \mathrm{PMI}(a; b)$.

The model cost of promotion depends on what is promoted:

Operation	**Cost**
Single LPP entry (from, to, weight)	∼24 bits (byte ESs)
New event in existing ES	event declaration + LPP entries
New ES + events + LPPs	ES declaration + all entries
Word event (shift-chain conjunction)	event + multi-depth LPPs

For byte-level bigrams, a single LPP entry costs ∼24 bits (8 bits each for source index, target index, and weight). For word events, the cost is substantially higher: the "the" experiment required 5 event spaces and 4 LPPs with ∼700 entries for a single word [7]. The per-word model cost scales with the word's orthographic complexity (number of distinct letters, case variants, and shift-chain depth).

The promotion criterion is:

$$\boxed{n_{ab} \cdot \mathrm{PMI}(a; b) > \Delta I_{\mathrm{model}}} \tag{8}$$

where $\Delta I_{\mathrm{model}}$ is the full model cost of the promotion, not just a single LPP entry.

4.3. Threshold creation as proxy

The existing threshold creation mechanism (ω extension, UMR Spec §7) creates a new event when a joint observation reaches support $\tau \approx 4$ (∼16 observations). This is a proxy for (8): it ensures sufficient observations before acting, but it does not evaluate $\Delta\omega$ directly.

For strongly associated pairs ($\mathrm{PMI} \gg 1$), the threshold is conservative—promotion pays off well before it triggers. For weakly associated pairs ($\mathrm{PMI} \leq 0$), they never reach threshold because they don't recur enough. The threshold thus approximately implements the MDL criterion in the common case.

A more precise criterion would evaluate $n \cdot \Delta\omega > \Delta I_{\mathrm{model}}$ directly, using $\Delta\omega$ computed from the OBSERVE LPP's counts. This requires reliable PMI estimates, which requires either sufficient counts or synthetic marginals (§3.2).

5. The Consolidation Loop

5.1. OBSERVE LPPs: nearly passive

An OBSERVE LPP participates in ω_0 (learning) but not in f (the forward pass) [7]. It does not directly affect the model's predictions or scoring.

However, OBSERVE LPPs are not perfectly passive. Exp 2 of [7] showed that the ω_0 learning in an OBSERVE LPP consumes random numbers from the shared RNG, shifting the active LPP's learning trajectory. The delta alternates sign across scales (+4.8, −137.8, +625.3, −2246.1 bits at 1K–64K). A model with no context→output LPP at all (not even learning) matches bigram-only exactly.

For the consolidation loop, this means: adding OBSERVE LPPs during online learning will perturb the model slightly via RNG coupling. During *replay* of a frozen trace, this coupling is absent (the trace is fixed, not regenerated), so OBSERVE LPPs during replay are truly passive. The consolidation loop should therefore instrument during replay, not during the initial online pass.

5.2. The five steps

Given a UM with a frozen model and a memory trace:

1. **Instrument**: add OBSERVE LPPs between existing ESs. During replay (not online learning), these accumulate joint event counts without any effect on the trace or the model.
2. **Replay**: decode the memory trace through the frozen model, recovering the original byte stream. The OBSERVE LPPs observe joint events at each position.
3. **Evaluate**: for each joint event (a, b) in the OBSERVE LPP with sufficient support ($s_{ab} \geq \tau$), compute $\Delta\omega(a, b)$. Use synthetic marginals (LS-add over the OBSERVE LPP's own entries) to avoid the quantization errors from independently learned supports.
4. **Promote**: for joint events where $n_{ab} \cdot \Delta\omega(a, b) > \Delta I_{\text{model}}$, create new events (threshold creation) and make the OBSERVE entries active.
5. **Rewrite**: replay the trace with the extended model. The model now makes sharper predictions at positions where the promoted events fire. The new trace is shorter.

This is one cycle of the Tick-Tock process [6]: the Tock (steps 1–4: architecture extension) followed by a Tick (step 5: retrain with the extended model, write a shorter trace).

5.3. What promotion does not solve: the L_∞ gap

The forward pass f is max-min: $f_P(T)_j = \max_i \min(T_i, P_{ij})$. This is L_∞ aggregation—the most-supported prediction wins [8]. Making an OBSERVE LPP active means its entries participate in this max-min competition.

The problem is **marginal dominance**: the promoted LPP may lose the sharpest-LPP contest at positions where it should win. Exp 3/6 of [7] demonstrated this directly: the context neuron (3 classes) was sharper than the unigram but coarser than the bigram, so it helped at unigram baseline (-0.120 bpc) but *hurt* at bigram baseline ($+0.283$ bpc at 1M). The context's prediction sometimes won the max-min contest over the bigram, substituting a worse distribution.

The word mixture experiment resolved this for the lexicon case: a frequency-weighted mixture over candidate words (L_1 aggregation) gave -0.552 bpc at 10M—far exceeding the -0.321 bpc oracle ceiling that max-min could deliver [7]. The gain came precisely from the positions where multiple words were plausible and the mixture correctly weighted them, while max-min would have picked the single most-supported word.

The consolidation loop discovers *what* to promote (which joint events carry information). It does not solve *how* to deliver that information through the forward pass. The delivery problem—settling, distribution-level aggregation, the L_∞ vs. L_1 gap—is orthogonal to consolidation and is addressed elsewhere [10, 11].

6. The Algebraic Structure

6.1. Events are integers

Under the framework of [2], events are integers and event spaces are rings. A byte event is an element of $\mathbb{Z}/256\mathbb{Z}$. A byte-level context of order k is an integer in $\mathbb{Z}/256^{k-1}\mathbb{Z}$. The joint event "context c followed by output o" is the integer $e = 256c + o \in \mathbb{Z}/256^k\mathbb{Z}$ [4].

Division is the quotient map: dropping the oldest byte from the context is $c \mapsto c \bmod 256^{k-2}$, which is modular reduction—literal integer division [2]. The tower of order projections

$$\mathbb{Z}/256^K\mathbb{Z} \twoheadrightarrow \cdots \twoheadrightarrow \mathbb{Z}/256\mathbb{Z} \twoheadrightarrow \{0\}$$

is a chain of ring surjections. Each step forgets one byte of context [4].

6.2. The differential ω in ring terms

The GCD decomposition (1) is GCD in $\mathbb{Z}$. The discount operation of Kneser–Ney is subtraction in $\mathbb{Z}$. The gap between KN discount and GCD is the gap between subtraction and division: they agree when $g = 1$ (the common case for natural language at byte level) and diverge when $g > 1$ [4].

The differential ω has a ring interpretation: $\Delta\omega$ is the log of the ratio of the joint count to the product of marginals. In the ring $\mathbb{Z}/N\mathbb{Z}$:

- The product of marginals $c(a) \cdot c(b)/N$ is the count predicted by independence.
- The joint count $c(a, b)$ is the observed count.
- $\Delta\omega > 0$: the joint event exceeds independence (the pair is bound).
- $\Delta\omega < 0$: the joint event falls below independence (the pair repels).
- $\Delta\omega = 0$: the events are independent at this level.

The GCD of the row gives the common evidence—the part that cancels from the conditional. The reduced count $r_A(a, b)$ gives the differential evidence—the part that determines the shape of $P(b \mid a)$. The PMI measures how far the shape deviates from uniformity (independence).

6.3. The tower of ring surjections

The KN interpolation recursion descends the tower of rings [4]:

$$P_k(o \mid c) = \frac{\max(c_k - D, 0)}{c_k(c, \cdot)} + \frac{D \cdot \tau_k}{c_k(c, \cdot)} \cdot P_{k-1}(o \mid c \bmod 256^{k-2}).$$

At each level, the prediction is a convex combination of evidence specific to this ring (the discounted counts) and evidence from the coarser ring (the backoff). The discount removes the common evidence (approximately, via subtraction rather than division).

The **generalized** ω aligns with this tower. At each level k, the new information is the differential beyond what level $k-1$ already provides:

$$\Delta\omega_k(a,b) = s_{ab}^{(k)} - s_a^{(k)} - s_b^{(k)} + s^{(k)}, \tag{9}$$

where the superscript (k) denotes counts at order k. The total model information at level k is the level-(k−1) information plus the $\Delta\omega_k$ residuals:

$$I_{\text{model}}^{(k)} = I_{\text{model}}^{(k-1)} + \sum_{\text{promoted at level } k} \underbrace{\text{cost of new entries}}_{\text{model growth}} - \sum_{\text{promoted at level } k} \underbrace{n \cdot \Delta\omega_k}_{\text{trace shrinkage}} . \tag{10}$$

This is the tower of ring surjections from [4], restated as a compression decomposition: each level adds the cost of new structure and subtracts the trace gain from sharper predictions. The total information decreases at each level where the gain exceeds the cost.

6.4. CRT and word-level extension

To extend the byte ring with word events, the kn-quotient paper [4] shows that choosing $|W|$ coprime to 256 gives a clean CRT factorization:

$$\mathbb{Z}/(|W| \cdot 256)\mathbb{Z} \;\cong\; \mathbb{Z}/|W|\mathbb{Z} \;\times\; \mathbb{Z}/256\mathbb{Z}.$$

Since $256 = 2^8$, any odd $|W|$ suffices ($|W| = 65537$ is a Fermat prime). The word and byte event spaces are algebraically independent: the word event carries no redundant information with the byte-level residual [8].

Consolidation at the word level operates on the product ring. The OBSERVE LPP between the word ES and the byte ES records the joint count table $c(w,b)$. The GCD decomposition and PMI evaluation are identical to the byte-level case. The only difference is the model cost: word events require multi-depth OBSERVE LPPs (unigram, bigram, trigram [7]), so ΔI_{model} is proportionally larger.

7. The Two-Level Trace

Consolidation at the byte level discovers bigrams, trigrams, and eventually words. Once word events exist, the memory trace changes character [6, 8].

7.1. Word events in the trace

The trace becomes a sequence of word events with residual bits [8]:

1. At word boundaries: emit a **word event** from the learned vocabulary, encoded against the word-level distribution (word frequencies, or word-bigram predictions).
2. For each word: emit **residual bits** specifying the exact spelling. Given "the," the orthographic embedding predicts t-h-e-space with near certainty (0.04 bits of entropy at trigram depth [7]), so the residual is near zero. For "The" (capitalized), a few bits encode the case variant.

3. At non-word positions: encode bytes against the byte model.

This is lossless: word event + residual bits + byte-level fallback recovers every byte. Unlike tokenization, no letter-level information is discarded [8].

7.2. The quotient-remainder structure

The two-level trace is a chain of quotient maps [2]:

$$\text{byte sequence} = \underbrace{\text{word event}}_{\text{quotient}} + \underbrace{\text{spelling residual}}_{\text{remainder}}.$$

The word event is the quotient—what the organism retains at the higher level. The spelling residual is the remainder—the detail that the word event does not predict. The division algorithm guarantees lossless recovery.

The bidirectionality of the LPP's joint count table provides both directions [3]: the downward embedding (word → letters) and the upward recognition (letters → word) come from the same counts. This is the sufficient-statistic property: no additional data is needed to reverse the direction.

7.3. Consolidation produces the two-level trace

The consolidation loop applied at the word level:

1. **Instrument**: add OBSERVE LPPs between the shift-chain conjunction at word boundaries and the output ES.
2. **Replay**: the OBSERVE LPPs learn the orthographic distributions (bag of letters at depth 1, letter transitions at depth 2, word identity at depth 3).
3. **Evaluate**: compute $\Delta\omega$ for each word-boundary conjunction. High-frequency words have large $n \cdot \Delta\omega$.
4. **Promote**: reify frequent conjunctions as word events (threshold creation). The model gains a lexicon.
5. **Rewrite**: the trace is now two-level. Common words are single events (short codes); their spellings are near-deterministic residuals. The trace is shorter.

The measured gains [7]: the Bayesian word mixture gives -0.552 bpc at 10M (bigram baseline), -0.111 bpc vs KN-6 at 100K. Position 0 (word start) contributes 49% of the gain —this is the *change of base*: at word boundaries, word-frequency-weighted prediction beats byte-frequency prediction.

8. Biological Parallel

The consolidation loop has a biological reading:

Biology	**UM**
Waking experience	Online learning (Tick: predict-then-learn)
Memory encoding	Memory trace (range-coded residual)
Sleep onset	Replay begins (frozen model, no new input)
Hippocampal replay	Trace replay through frozen model
OBSERVE (passive)	OBSERVE LPPs accumulate joint counts
Consolidation	$\Delta\omega$ evaluation via GCD decomposition
Chunking	Threshold creation (new events)
Morning	Rewritten trace (shorter, two-level)

The organism wakes up with: (1) a larger model (new events for discovered chunks), (2) a shorter memory trace (the chunks compress it), (3) the same total information (lossless: model + trace = data).

This parallel is structural, not metaphorical: the UM's consolidation loop and biological sleep consolidation accomplish the same formal task (compress memory by discovering joint events whose PMI exceeds the cost of carrying them). Whether the biological mechanism implements GCD decomposition or an approximation to it is an empirical question.

9. Open Problems

9.1. The delivery problem

Consolidation discovers *what* to promote. The delivery problem is *how* the promoted information improves predictions through the forward pass. Max-min (L_∞) is the current forward pass; the word mixture experiments show that L_1 (probability-weighted average) outperforms it for multi-hypothesis prediction [7]. The settling conjecture [10, 11] proposes that the fix must operate at the distribution level, not the support level. This is orthogonal to consolidation but determines how much of the consolidation gain is realizable.

9.2. Log-stochastic precision

The $\Delta\omega$ formula is exact for exact log-counts but approximate for the UM's stochastic supports. At low counts, the wrong sign of PMI can lead to incorrect promotion decisions. Synthetic marginals (LS-add from per-event entries) eliminate consistency violations [7], but the underlying quantization limits the resolution of $\Delta\omega$ to ± 1 bit at typical support levels.

9.3. Optimal threshold

The fixed threshold $\tau \approx 4$ is a coarse proxy for the MDL criterion (8). The optimal threshold depends on the model cost of promotion (which varies by event type and depth), the PMI of the joint event (which varies by pair), and the number of future occurrences (which is unknown at promotion time). An adaptive threshold that evaluates $n \cdot \Delta\omega$ against the actual model cost would be more precise.

9.4. Multi-level consolidation

The tower of ring surjections suggests that consolidation should operate at all levels simultaneously: byte → bigram → trigram → word → phrase. The current framework handles one level at a time (Tick-Tock cycles). Whether the levels can be consolidated in parallel—instrumenting multiple OBSERVE LPPs at different depths and promoting across levels in a single replay—is an open architectural question.

10. Summary

1. The GCD decomposition [3] separates joint counts into common evidence (the GCD, which cancels from conditionals) and differential evidence (the reduced counts, which determine the shape of $P(b \mid a)$).
2. The differential ω is the PMI extracted from this decomposition: $\Delta\omega = s_{ab} - s_a - s_b + s$. It measures how much information the joint event carries beyond the marginals. It is computable from the OBSERVE LPP's counts using the UM's existing LSA primitives.
3. Promotion is justified when $n \cdot \Delta\omega$ exceeds the model cost of the new structure. This is the MDL criterion. The threshold τ is a coarse proxy; the full criterion accounts for multi-depth LPP costs and CRT coprimality of the extended event space [4].
4. OBSERVE LPPs are the mechanism: they accumulate joint counts without affecting the forward pass. During replay (not online learning), they are truly passive; during online learning, they perturb the model via RNG coupling [7].
5. The consolidation loop (instrument → replay → evaluate → promote → rewrite) is domain-independent. At the word level, it produces the two-level trace: word events + spelling residual [8].
6. The delivery problem (L_∞ vs. L_1) is orthogonal to consolidation but determines how much of the discovered information reaches the predictions.

References

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

[2] Claude and MJC. *Integer Factorization of Events*. Hutter archive, Feb 2026.

[3] Claude and MJC. *Bayes from Counting*. Hutter archive, Feb 2026.

[4] Claude and MJC. *Kneser–Ney on the Integers, v2: The Ring Structure*. Hutter archive, Feb 2026.

[5] Claude and MJC. *The Embedding Conjecture*. Hutter archive, Feb 2026.

[6] Claude and MJC. *The Memory Trace*. Hutter archive, Feb 2026.

[7] Claude and MJC. *English Context Neuron: Experiment Results*. Hutter archive, Feb 2026.

[8] Claude and MJC. *The Lexicon Embedding*. Hutter archive, Mar 2026.

[9] MJC. *Commentary on the KN-quotient paper*. Feb 2026.

[10] Claude and MJC. *Timing Resolution*. Hutter archive, Feb 2026.

[11] Claude and MJC. *Settling*. Hutter archive, Feb 2026.