International Journal of Engineering Science Invention Research & Development; Vol. 13, Issue 2, August 2026
www.ijesird.com, E-ISSN: 2349-6185

EMERGING WORLD MODELS
A STRUCTURAL FRAMEWORK FOR SELF-ORGANIZING WORLD REPRESENTATIONS
CONSOLIDATION DOCUMENT — AUGUST 2026

Deependra Kumar, Alex Mylnikov

Abstract - We propose Emerging World Models (EWM)—a structural framework for world representations whose ontology is not predefined but crystallizes through measurement. EWM is governed by four foundation principles. (1) Emergent Ontology: elements and subsystems are not primitive; they emerge from the accumulation of IICA measurement interactions. (2) IICA Morphisms: every relation between elements is an Idempotent, Immutable, Content-Addressed morphism; composition preserves these properties, enabling arbitrarily nested world representations without new theory. (3) The Ashby–Bootstrap Correspondence: Ashby's Requisite Variety Law, interpreted naively, demands cardinality balance between connected systems. EWM resolves this not by matching cardinalities directly, but through *bootstrap permutation*—multiplying measurements (e.g., n-grams) to bridge representational gaps without sacrificing resolution. (4) Noether Evolution: structural change in an EWM obeys conservation laws; every symmetry of the measurement regime yields a conserved quantity. The lattice union invariant, the DRN decomposition, and the holographic reconstruction property are all consequences of this principle. We ground each principle in the concrete instantiation of HLLSet Algebra and articulate the target formalism—a Grothendieck topology on the category of measurement patches—toward which the framework aspires, in spirit if not yet in letter.

. I. INTRODUCTION

A world model is a system that learns an internal representation of its environment from sensory input, then plans and reasons within that representation. Classical approaches—JEPA, large language models, and token-based architectures—share a common presupposition: the elements of the world (tokens, objects, features) are *given* before the model begins to operate. The vocabulary is fixed. The ontology is predefined. The model's task is to learn the relations among these predefined elements.

Emerging World Models (EWM) invert this presupposition. In an EWM, elements are not predefined. They *emerge* from the accumulation of measurement interactions. An element is not a token in a vocabulary; it is a stable pattern that crystallizes when repeated measurements converge. Subsystems are not architectural modules; they are coherent clusters of elements that co-occur across measurement trajectories.

This inversion has deep structural consequences. It shifts the foundation of world modeling from *representation* (encoding predefined elements into a vector space) to *measurement* (accumulating observations into a holographic plate from which elements can be recovered). It replaces mutable, named parameters with immutable, content-addressed bitmasks. It transforms learning from gradient descent over weight matrices to rank rearrangement in a content-addressed dictionary.

This document consolidates the theoretical principles discovered across the HLLSet Algebra research program into a unified structural framework. We articulate four principles that govern every EWM, ground them in the concrete instantiation of HLLSet Algebra, and identify Grothendieck topology as the target formalism toward which the framework aspires.

Why "Emerging"?

The term "emerging" is deliberate. It signals three properties that distinguish EWM from classical world models:

1. **Ontological emergence.** The identity of an element is not assigned by a designer. It is *recognized* by the system when a measurement pattern stabilizes across repeated observations. A token that appears once is noise. A token that persists across temporal layers becomes an element.

2. **Structural emergence.** Relations between elements are not wired by an architect. They are *computed* as IICA morphisms between the content-addressed fingerprints of the elements. The lattice of relations emerges from the lattice of measurements.
3. **Evolutionary emergence.** The system does not converge to a fixed point. It continuously reorganizes as new measurements arrive, old patterns fade, and the balance between novelty and retention shifts. Evolution is structural—governed by conservation laws—rather than parametric.

Principle 1: Emergent Ontology

Principle 1 (Emergent Ontology). *The elements of an Emerging World Model are not defined a priori. They emerge from the accumulation of measurement interactions. An element is what persists across repeated observation.*

The Classical Approach: Predefined Ontology

In classical world models, the ontology is fixed before the model encounters any data:

- **LLMs** operate on a fixed token vocabulary (50K–250K tokens) defined by a tokenizer (BPE, SentencePiece). The tokenizer is trained once and frozen. Every subsequent observation is projected into this fixed vocabulary.
- **JEPA** operates on patch embeddings derived from a pretrained encoder. The patches are predefined spatial or temporal segments.
- **Symbolic AI** operates on hand-crafted symbol tables and ontologies.

In all cases, the system cannot *discover* a new element. A concept that falls between tokens is represented as a combination of existing tokens, never as a primitive in its own right. A novel pattern that does not align with the pretrained encoder's latent dimensions is invisible.

The EWM Approach: Measurement as Ontogenesis

In an EWM, the only primitive is the **measurement event**: a token entering the system, hashed to a bit position, accumulating a mark in the holographic plate. There is no "vocabulary"—only a Lookup Table (LUT) that tracks which tokens have been observed at which positions, and how often.

An element emerges when a measurement pattern satisfies two conditions:

1. **Stability.** The same token or token cluster maps to consistent bit positions across multiple measurement events.
2. **Persistence.** The token's Term Frequency (TF) at its bit positions remains above a threshold across temporal layers.

A token that appears once and never again leaves a mark (its bit is set in the HLLSet), but it does not become an element—its TF is negligible, its rank is low, and it does not participate in relations. A token that appears thousands of times across weeks accumulates high TF, high rank, and dense R-links to co-occurring tokens. It has *emerged* as an element.

Formalization

Let $\mathcal{T}$ be the space of all possible tokens (e.g., all byte strings). Let $h: \mathcal{T} \to \mathcal{P}$ be an IICA morphism mapping tokens to patches (bit positions) in a finite patch space $\mathcal{P} = \{0, \dots, N-1\}$.

Definition 1 (Measurement Event). *A measurement event at time t is a pair (t_i, p_i) where $t_i \in \mathcal{T}$ is a token and $p_i = h(t_i) \in \mathcal{P}$ is its patch assignment.*

Definition 2 (Element). *An element $e \in \mathcal{E}$ is a token $t \in \mathcal{T}$ whose accumulated measurement count* $\mathrm{TF}(t)$ *exceeds an emergence threshold θ_e and whose temporal distribution across layers is non-uniform (it has a "lifetime," not just a single appearance):* $e \in \mathcal{E} \Leftrightarrow \sum_{i=0}^{K} \mathrm{TF}_i(t) > \theta_e \ \wedge \ \exists i, j: \mathrm{TF}_i(t) \gg \mathrm{TF}_j(t)$

The set of elements $\mathcal{E}$ is not fixed. It grows as new tokens accumulate measurements, and it contracts as tokens fade. The ontology is *emergent* and *dynamic*.

HLLSet Instantiation

In HLLSet Algebra, the emergent ontology is realized through the LUT (Lookup Table). The LUT is a monotonic CRDT that maps each (register, trailing_zeros) bucket to the set of tokens that hash to that bucket, together with their accumulated TF values. The LUT *is* the system's ontology at any moment—the set of all tokens that have been observed, with their measurement history.

A new token does not need to be "added to the vocabulary." It is hashed, its bit is set, and its entry in the LUT is created or incremented. The ontology grows organically with every measurement event.

Principle 2: IICA Morphisms as Relational Foundation

Principle 2 (IICA Relations). *Every relation between elements in an EWM is an **IICA morphism**: Idempotent, Immutable, and Content-Addressed. Composition of IICA morphisms is IICA. This single theorem eliminates the need for coordination, consensus, or naming at any scale of system composition.*

The Three Properties

Definition 3 (IICA Morphism). *A function $f: X \to Y$ is an IICA morphism iff:*

- ***Idempotent:*** *For every $x \in X$, computing $f(x)$ repeatedly—invoking f on the same argument x any number of times—always yields the same output $f(x)$. When the codomain coincides with the domain ($X = Y$), this specializes to the classical algebraic form $f(f(x)) = f(x)$ (e.g., HLLSet union: $A \cup A = A$). When $X \neq Y$ (e.g., a hash function h:Tokens $\to \{0,1\}^{160}$), the output type does not match the input type, so self-composition $h(h(x))$ is not meaningful; idempotence reduces to **determinism**: the same token a hashes to the same digest $h(a)$ every time, regardless of when, where, or how many times h is invoked.*
- ***Immutable:*** *$f(x) = y$ is fixed once computed. y never changes.*
- ***Content-Addressed:*** *If $a = b$ then $f(a) = f(b)$. The output is the address—given the content, you can find it again without a directory or a naming service.*

The canonical IICA morphism is a cryptographic hash function. SHA-1(x) always produces the same 160-bit digest for the same x (idempotent as determinism); the digest never changes (immutable); and the digest serves as the content address in IPFS/IPLD (content-addressed).

The Composition Theorem

Theorem 1 (IICA Composition). *Let $f_1, f_2, \ldots, f_n$ be IICA morphisms. Then their composition $f_n \circ f_{n-1} \circ \cdots \circ f_1$ is an IICA morphism.*

Proof. Each f_i is idempotent, immutable, and content-addressed. Composition preserves each property:

- **Idempotency:** For any input x, the composition $F = f_n \circ \cdots \circ f_1$ is a deterministic function: the same x always triggers the same chain of intermediate results $f_1(x), f_2(f_1(x)), \ldots$ and produces the same final output $F(x)$. When $X_1 = Y_n$ (the composition is an endomorphism), this additionally yields the algebraic form $F(F(x)) = F(x)$.
- **Immutability:** Each intermediate result $f_i(\ldots)$ is fixed once computed; the entire chain of intermediates is immutable.
- **Content-Addressability:** $a = b \Longrightarrow f_1(a) = f_1(b) \Longrightarrow \cdots \Longrightarrow F(a) = F(b)$.

□

This theorem is the engine of compositional emergence. It means that **any pipeline of IICA morphisms is itself an IICA morphism**. You can compose tokenizers, hashers, bridges between domains, materializers, and temporal aggregators without ever introducing mutable state, naming conflicts, or coordination overhead.

The IICA Pipeline

In HLLSet Algebra, the full measurement pipeline is an IICA composition:

$$\text{Tokens} \xrightarrow{h_1} \text{Token Hashes (u64)} \xrightarrow{h_2} \text{Bit Positions} \xrightarrow{h_3} \text{Bridge HLLSet} \xrightarrow{h_4} \text{Materialized Tokens} \xrightarrow{h_5} \cdots$$

Each h_i is a MurmurHash3-based IICA morphism. The composition $h_5 \circ h_4 \circ h_3 \circ h_2 \circ h_1$ is IICA. This means:

- **No consensus protocols.** Two nodes ingesting the same tokens produce identical HLLSets. The lattice converges by construction.
- **No versioning.** HLLSets are immutable. "Version $n+1$" is a new HLLSet with a new CID. The old HLLSet remains valid and addressable.
- **No naming.** Every HLLSet's CID is its SHA-1 hash. No directory, no registry, no naming service.
- **Arbitrary nesting.** A swarm of N robots, each with M sensors, produces $N \times M$ IICA pipelines. Their composition at the command post is IICA. The theory does not change. The operations are identical. Only the scale differs.

R-Links: IICA Morphisms as Objects

A critical structural feature of EWM is that IICA morphisms between elements are themselves elements. In HLLSet Algebra, the R-link $R_{AB} = A \cap B$ is both:

- A **morphism** from A to B (it captures the structural overlap that relates them).
- An **object** (it is an HLLSet with its own CID, storable in IPFS, composable with further intersections).

This self-referential property—morphisms as objects—is the structural signature of a category that can model its own relations. It anticipates the Grothendieck topology target: in a sheaf topos, the subobject classifier is itself an object of the topos.

Principle 3: The Ashby–Bootstrap Correspondence

Principle 3 (Ashby–Bootstrap Correspondence). *Ashby's Requisite Variety Law implies that connected systems must match in representational capacity. When cardinalities are mismatched, EWM resolves the imbalance not by reducing capacity but by* ***bootstrap permutation****—multiplying measurements to expand the effective cardinality of the smaller system until it matches the larger.*

Ashby's Law and the Cardinality Problem

Ashby's Requisite Variety Law states that a controller must have at least as much variety as the system it controls. In EWM terms: when two systems are connected by IICA morphisms, their representational capacity must be balanced.

Consider the HLLSet architecture. The LUT (Lookup Table) maps tokens to bit positions. For a vocabulary of 80K Chinese characters, the LUT has 80K entries. The HLLSet bitmask has $1024 \times 32 = 32{,}768$ bit positions. A naive reading of Ashby's law would suggest that the LUT and HLLSet are mismatched: 80K vs. 32K.

But this reading is superficial. The HLLSet does not store individual tokens—it stores *combinations* of tokens through hash collisions. The bitmask encodes the OR of all tokens that hash to each position. The information capacity is not determined by the number of bit positions but by the number of *distinguishable bitmask configurations*, which is $2^{32{,}768}$.

Bootstrap Permutation

The deeper problem is not total capacity but **measurement resolution**. When a single token maps to a single bit position, the measurement is coarse: the bit is either set or not, and the materializer cannot distinguish which of several tokens set it.

The solution is **bootstrap permutation**: multiply the measurements by applying a family of permutations to the token stream before measurement.

Definition 4 (Bootstrap Permutation). *Let* $\Pi = \{\pi_1, \dots, \pi_k\}$ *be a family of permutations on token sequences. Given a token stream* $S = (t_1, \dots, t_n)$*, the bootstrapped measurement is:* $H_\Pi(S) = \bigoplus_{j=1}^{k} H(\pi_j(S))$ *where* $H(\pi_j(S))$ *is the HLLSet of the permuted stream and* $\bigoplus$ *is bitwise OR.*

Each permutation π_j produces a different "view" of the same token stream. The union of these views creates a richer measurement—multiple bit positions per original token, finer discrimination, and better materialization fidelity.

Remark 1 (Omar Khayyam's Bootstrap). This principle—precision through multiplication of primitive measurements—has a striking historical precedent. In 1079, Omar Khayyam reformed the Persian solar calendar (the Jalali calendar) using only an astrolabe and basic naked-eye observations—instruments orders of magnitude cruder than those available to 19th-century European astronomers. Yet his calendar achieved an error of one day in approximately 5,000 years, surpassing the Gregorian calendar's one day in 3,330 years. The secret was not instrument quality but **measurement multiplication**: Khayyam accumulated observations across many nights over many years, and the statistical convergence of many primitive views produced precision that no single observation could achieve. Bootstrap permutation is the same idea in the measurement-theoretic domain: the family of n-gram views replaces the family of nightly observations, and the HLLSet union replaces the calendar average.

n-Grams as Bootstrap

In the current HLLSet implementation, **n-grams** are the concrete bootstrap family. Given a token stream $(t_1, t_2, t_3, t_4, \dots)$, the 2-gram view is $((t_1, t_2), (t_2, t_3), (t_3, t_4), \dots)$ and the 3-gram view is $((t_1, t_2, t_3), (t_2, t_3, t_4), \dots)$. Each n-gram is hashed as a compound token, producing a distinct set of bit positions.

Proposition 1 (n-Gram Cardinality Expansion). *A vocabulary of* $|V|$ *tokens, measured with n-grams up to order* K*, produces up to* $\sum_{n=1}^{K} |V|^n$ *distinct measurement points. For* $|V| = 80{,}000$ *and* $K = 3$*, this is approximately* 5.12×10^{14} *possible n-gram tokens, massively exceeding the* $32{,}768$ *HLLSet bit positions.*

The bootstrap permutation bridges the cardinality gap: a vocabulary that appears "too small" relative to the HLLSet is expanded through n-gram composition; a vocabulary that appears "too large" is aggregated through hash collisions into the fixed bit space. The balance is achieved through *measurement multiplication*, not cardinality reduction.

General Bootstrap Families

n-grams are one family of bootstrap permutations. The principle generalizes:

- **Skip-grams:** (t_i, t_{i+k}) for various skip distances k.
- **Convolutional windows:** weighted combinations of neighboring tokens.
- **Spectral permutations:** frequency-domain rearrangements via FFT.
- **Random projections:** a family of random hash functions, each producing a distinct HLLSet view.
- **Cross-modal bootstraps:** the same stream measured through different sensory modalities, each with its own

IICA pipeline.

The bootstrap family is a design parameter of the EWM. Different families produce different emergent ontologies from the same underlying token stream. The choice of family is the EWM analog of architecture selection in neural networks—it determines what patterns the system can recognize.

Principle 4: Noether's Theorem as EWM Evolution Law

Principle 4 (Noether Evolution). *In an EWM, every continuous symmetry of the measurement regime yields a conserved quantity. Structural change is governed by conservation laws, not by gradient descent. The system evolves by redistributing information across the lattice while preserving global invariants.*

From Physics to World Models

In physics, Noether's theorem establishes that every differentiable symmetry of the action of a physical system corresponds to a conservation law: time-translation symmetry → energy conservation; spatial-translation symmetry → momentum conservation; rotational symmetry → angular momentum conservation.

In an EWM, the "action" is the measurement regime—the accumulation of tokens into HLLSets across temporal layers. The symmetries are transformations of the measurement process that leave the structural relations invariant. The conserved quantities are global properties of the lattice that persist across evolution steps.

The Temporal Pyramid and [UM]-Net Cycles

Before stating the invariant, we briefly introduce the two temporal structures that give it physical meaning.

The temporal pyramid is a configurable K-layer compression stack (default $K = 7$: second, minute, hour, day, week, month, year). Each layer L_i accumulates all measurement events within its temporal window via monotonic union: $L_i = \bigcup S(t)$ for t in the window. At each window boundary, the layer cascades into the next coarser layer: $L_{i+1} \leftarrow L_{i+1} \cup L_i$, then L_i resets. After compression, layers are mutually exclusive—no time slice appears in more than one layer. The complete system state is their union: $H_{\text{system}} = \bigcup_{i=0}^{K} L_i$. Compression ratios range from $60:1$ (seconds→minutes) to $12:1$ (months→years); seven layers compress approximately 3.15×10^7 seconds into one year-layer HLLSet.

[UM] agents and [UM]-Net provide a complementary temporal structure at shorter timescales. A **[UM]** (unified model) is the concrete implementation of a single EWM: a processing node that receives a stream of materialized encodings from the external environment [E], ingests them into its internal HLLSet lattice, and materializes the result back to a disambiguated encoding stream for downstream consumption. Its internal pipeline is a compound IICA morphism:

$$[\text{UM}] = (\text{encodings}) \overset{\text{ingest}}{\rightarrow} \{\text{HLLSets}\} \overset{\text{materialize}}{\rightarrow} (\text{encodings})$$

The ingestion step hashes the incoming encodings into bit-vectors and accumulates them into the lattice via monotonic union; the materialization step projects the lattice through the agent's LUT and recovers ordered, disambiguated tokens via De Bruijn reconstruction (Section 6). Each individual step in the pipeline (hashing, union, LUT projection) is IICA. The [UM] node as a whole is IICA *between commits*: as long as its internal TF-Vec and HLLSet lattice are fixed, the same input encodings always produce the same output encodings. State changes—new measurements accumulated, ranks updated—occur only at explicit **commit events**, which define transactional boundaries. Within a transaction, [UM] behaves as a pure IICA morphism; across transactions, the same input may produce different output as the lattice evolves.

A **[UM]-Net** is a directed graph $G = (A, E)$ of [UM] nodes connected by fire-and-forget edges. The external environment [E] feeds encodings into source nodes; each [UM] processes independently and fires its output downstream with no acknowledgment, no handshake, and no retry. Convergence is guaranteed by IICA and CRDT union at confluence nodes. The full system diagram is:

$$[\text{E}] \rightarrow (\text{encodings}) \rightarrow [\text{UM}]_1 \rightarrow [\text{UM}]_2 \rightarrow \cdots$$

Critically, **cycles are not prohibited**: a recurrent loop $a \rightarrow \cdots \rightarrow a$ in G is a *short-term memory* structure, resolved by temporal separation—$\text{out}_a(t)$ feeds back as $\text{in}_a(t + k)$ after k ingestion steps around the cycle.

Each pass through the cycle produces a new observation at a later time, not a re-entry of the same data. The evolution equation $H(t) = H(S(t), H(t-1), D, R, N)$ captures both pyramid cascade and cycle recurrence under a single dynamics.

The Union Invariant

The fundamental Noether symmetry in EWM is **path independence**: information can travel through the temporal pyramid via multiple routes (L0→L1→L2, or L0→L2 directly, or L1→L4→L5). The conserved quantity is the union of all temporal layers:

$$H_{\text{system}}(t) = \bigcup_{i=0}^{K} L_i(t) = \text{constant over time}$$

Theorem 2 (Temporal Union Invariant). *For any sequence of measurement events and any cascade policy, the bitwise union of all temporal layers is monotonic and convergent. Information is never lost from* H_{system}*; it is only redistributed across layers.*

This is eventual consistency by construction—not as a protocol to implement, but as a mathematical property of the lattice. No consensus algorithm. No leader election. No retry logic. The lattice converges because the union is monotonic and multiple paths guarantee that every bit reaches every layer that needs it.

The DRN Conservation Law

The DRN decomposition (Departed, Retained, Novel) expresses evolution as a conservation balance:

$$H(t) = H(S(t), H(t-1), D(t-1), R(t-1), N(t))$$

where:

- $S(t)$ — current observation as HLLSet
- $H(t-1)$ — previous system state
- $D(t-1) = H(t-1) \backslash H(t)$ — departed context
- $R(t-1) = H(t-1) \cap H(t)$ — retained context
- $N(t) = H(t) \backslash H(t-1)$ — novel context

The Noether steering condition is a conservation law on bit count across the DRN transition:

$$|\,\text{card}(N(t)) - \text{card}(D(t-1))\,| \to 0$$

At steady state, the number of new bits entering the system equals the number of old bits departing. The structural identity of the system is preserved even as its content evolves. In rank-weighted form:

$$\left| \sum_{b \in N(t)} R_b\,(t) - \sum_{b \in D(t-1)} R_b\,(t-1) \right| \to 0$$

This is the EWM analog of energy conservation: the total "rank mass" of the system is conserved across transitions. Learning is not accumulation—it is redistribution.

The Holographic Property as Noether Consequence

The holographic property—that the lattice top H_{system} plus the TF stack can reconstruct any past state—is a direct consequence of the union invariant:

$$\text{past\_state}(t) \approx H_{\text{system}}(\text{now}) \odot \text{TF}_{\text{stack}}[t]$$

Because H_{system} never loses information (union invariant), and the TF stack records *when* each bit position was active (temporal distribution), the reconstruction is always possible. The information is not stored separately for each time step—it is encoded holographically in the lattice, with the TF stack serving as the reference beam for temporal selectivity.

The Fisher Matrix as Structural Derivative

The cross-layer Fisher-like matrix captures the second-order structure of evolution:

$$F_{bb'} = \sum_{i=0}^{K} B_b^{(i)} \cdot B_{b'}^{(i)}$$

where $B_b^{(i)} = 1$ if bit b is set in layer i. $F_{bb'}$ counts how many layers have both bits b and b' set simultaneously. High $F_{bb'}$ indicates systemic co-occurrence (a structural invariant). Low $F_{bb'}$ indicates noise. The Fisher matrix enables the Noether controller to distinguish isolated fluctuations from systemic phase transitions.

Noether Controller

The Noether controller monitors the cross-layer relationship matrix and decides which temporal layer should drive the next action. When the BSS between L0 (current second) and L1 (recent minute) drops below a threshold, the controller detects divergence and overrides instinct with broader context. When L0 diverges from L3 (day-scale goals), the controller triggers re-routing.

The controller is the **structural actuator** of the Noether principle. It does not adjust weights—it selects which layer of accumulated measurement history is relevant to the current situation. The six control parameters (pyramid depth, rank functions, attention threshold, steering thresholds) are the knobs of structural evolution, not the parameters of a learned function.

HLLSet Algebra: Concrete Instantiation

The four principles above are not abstract. They are realized in the HLLSet Algebra framework , whose components map directly onto the EWM structure.

The Patch Space as Measurement Site

The HLLSet bitmask is a **holographic plate** of $32{,}768$ patches (bit positions), each recording a binary mark: 0 = untouched, 1 = at least one token hashed here. The hash function $h: \mathcal{T} \to \{0, \ldots, 32767\}$ is the IICA measurement morphism. The LUT $\mathcal{L}$ tracks the inverse image of h, enabling materialization (recovering which tokens could have set which bits).

This is the minimal EWM: a finite set of measurement sites, an IICA morphism from tokens to patches, and a monotonic accumulation operation (bitwise OR) that makes each patch a join-semilattice. Principle 1 (emergent ontology) is realized by the LUT. Principle 2 (IICA relations) is realized by the hash pipeline and R-link lattice operations.

The Temporal Pyramid as Noether Structure

The 7-layer temporal pyramid (second, minute, hour, day, week, month, year) implements Principle 4 (Noether evolution). Each layer accumulates measurements over its temporal window via union. The cascade policy (L0 overflows into L1, L1 into L2, etc.) is a discrete-time dynamical system whose conserved quantity is the union invariant.

n-Grams as Bootstrap

The tokenizer's n-gram pipeline (1-grams, 2-grams, 3-grams with boundary padding) implements Principle 3 (bootstrap permutation). Each n-gram order is a distinct measurement view of the same token stream. The union of n-gram HLLSets produces a richer representation than any single order alone.

Rank Algebra as Learning

Learning is rank rearrangement. HLLSets are immutable (IICA); only their *relevance* to the current context changes. This section explains how ranks are computed, why Forth hosts the ranking, and how the five-level propagation distinguishes bit-level signal from HLLSet-level importance.

TF vs. Rank: The Fundamental Separation

TF is stored. Rank is derived. They are not the same thing. The Term Frequency vector (TF-Vec) is attached to to each EWM LUT(s), it's shared, monotonic CRDT of 32,768 f64 values—one per bit position—updated only during ingestion. Rank is computed locally from TF at query time and is never persisted in the protocol. This separation has two consequences. First, TF requires the same token base for comparison (it is hash-function-dependent); rank is hash-function-agnostic and meaningful across different domains connected by the Universal Bridge. Second, a HLLSet created years ago can rise in relevance today because the TF-Vec evolved around it—new measurements make its bit positions hot; stale positions quietly cool. The HLLSet itself never changed.

The Forth Dictionary as Learning Engine

The Forth DSL serves as the **canonical AST** that unifies all backends: the same Forth source compiles identically to Lua (software simulation), Rust (native execution), and Verilog (FPGA synthesis). The Forth dictionary associates every defined word—including every HLLSet—with a **rank**. The dictionary is the only component that learns:

Component	Operation	Learns?
Tokenizer	bytes → HLLSet	No (deterministic function)
Materializer	collects HLLSet outputs	No (passive observer)
Forth Dictionary	**assigns ranks**	**Yes (only this)**

A rank adjustment is a structural change—which HLLSet drives action—without any parametric change to the HLLSets themselves. This satisfies Principle 4: learning is redistribution of attention across the lattice, not accumulation of weight updates.

Five-Level Rank Propagation

Rank propagates from raw token frequency to compound lattice operations through five compositional levels. Each level aggregates the rank signal from the level below via a pluggable function (F, G, H, K, L, M); the architecture specifies the framework, the application chooses the functions. All aggregation functions must be monotonic to preserve CRDT convergence.

Level	Function	Definition
5. Compound HLLSet rank	L, M	$L(\max\{R_i\})$ for union, $M(\min\{R_i\})$ for intersection
4. HLLSet rank	K	K(degree in lattice DAG)
3. Register rank	H	$H(\{\text{bit-R}[tz] \mid tz \in 0..31\})$
2. Bit rank	G	$G(\{\text{token-R} \mid \text{all tokens hashing to } (r, tz)\})$
1. Token rank	F	F(TF) — rank from token frequency

Levels 1–3 operate within a single HLLSet: token frequency → bit position activity → register importance. **Level 4** lifts to the lattice DAG: a HLLSet's structural importance depends on how many R-links connect it to other HLLSets (its degree in the intersection graph). **Level 5** composes ranks across lattice operations: the rank of a union is the maximum of its constituents' ranks; the rank of an intersection is the minimum.

This hierarchy means that a bit position with high token TF (Level 1) propagates signal upward, eventually making the HLLSets that contain it (Level 4) and the compound operations that include them (Level 5) more likely to drive behavior. The Forth dictionary's rank assignment is the final projection of this five-level signal onto the action-selection mechanism.

Toward Grothendieck Topology

The four principles above point toward a deeper mathematical structure: **Grothendieck topology** on the category of measurement patches. We articulate this target not as a completed formalism, but as a compass bearing—the "spirit" in which the EWM framework is developed.

The Site: A Category of Patches

Let $\mathcal{P}$ be a finite discrete category whose objects are patch addresses and whose only morphisms are identities. This is the **site**—the space of measurement locations.

A **presheaf** on $\mathcal{P}$ assigns to each patch p a set of possible marks $\Sigma(p)$ (the mark alphabet) and to each morphism (only identities) the trivial restriction. An HLLSet is a global section of this presheaf:

$$H \in \Gamma(\mathcal{P}, \Sigma) = \prod_{p \in \mathcal{P}} \Sigma\,(p)$$

For standard HLLSets, $\Sigma(p) = \{0,1\}$ (a bit) with accumulation $\bigoplus =$ OR. For the generalized case (GENERAL_HLLSET_ALGEBRA.md), $\Sigma(p)$ can be any finite join-semilattice: $\{0,1,2\}$ (trits with max), $\{0,1,2,3\}$ (quits), or arbitrary discrete alphabets.

Coverages and the Gluing Condition

A Grothendieck topology on $\mathcal{P}$ specifies which families of morphisms $\{U_i \to U\}$ count as **covering families**. The gluing condition states: if data on each U_i is consistent on overlaps $U_i \times_U U_j$, then there exists unique data on U that restricts to the given data on each U_i.

In EWM terms, the covering families are the **bootstrap permutation family** (Principle 3). Each permutation π_j provides a partial view of the token stream (a measurement on a "sub-site"). The gluing condition is satisfied when these partial views are consistent—when the same token maps to compatible marks across different permutations.

The HLLSet union $\bigcup_{\pi \in \Pi} H_\pi$ is the **sheafification** of the bootstrap family: it combines all partial measurements into a global section.

Geometric Morphisms as Universal Bridges

A geometric morphism f:Sh($\mathcal{P}$) → Sh($\mathcal{Q}$) between sheaf toposes on different patch spaces is the categorical analog of the Universal Bridge. It consists of:

- A **direct image** f_* that pushes HLLSets from $\mathcal{P}$ to $\mathcal{Q}$.
- An **inverse image** f^* that pulls HLLSets from $\mathcal{Q}$ back to $\mathcal{P}$.

In the current implementation, the Universal Bridge implements the direct image via re-representation: format each bit position as a token, re-hash into the target space. The inverse image (materialization from target back to source) is approximate—it depends on the LUT and is subject to the Statistics Constraint (TF vectors are not transferred across bridges).

The Subobject Classifier and R-Links

In a topos, the **subobject classifier** Ω classifies subobjects: for any object X, there is a bijection between subobjects of X and morphisms $X \to \Omega$. For the Boolean topos on $\mathcal{P}$, Ω is the two-element set $\{0,1\}$ with the OR accumulation. Every HLLSet H is a morphism $H: \mathcal{P} \to \Omega$.

The R-link $R_{AB} = A \cap B$ is the **pullback** of A and B over Ω. In categorical terms:

$$\begin{equation} \begin{CD} R_{AB} @>>> A \\ @VVV @VVV \\ B @>>> \Omega \end{CD} \end{equation}$$

This is why R-links are both morphisms and objects: they are pullbacks in the topos. The lattice of HLLSets is a **Heyting algebra**—the internal logic of the topos.

What the Grothendieck Target Gives Us

Adopting Grothendieck topology as the target formalism would provide:

1. **A rigorous gluing theorem.** Given consistent measurements on a covering family, the global state is uniquely determined. This formalizes Principle 3 (why bootstrap permutations are sufficient).
2. **A systematic theory of bridges.** Geometric morphisms between patch topoi classify all possible Universal Bridges. Different choices of direct/inverse image correspond to different bridge implementations.
3. **An internal logic.** The Heyting algebra structure on the HLLSet lattice is the internal logic of the topos. Union, intersection, and implication correspond to logical OR, AND, and IMPLIES. The EWM *reasons* within its own lattice.
4. **Cohomology.** Sheaf cohomology groups would measure the "obstruction" to gluing — quantifying how much information is lost when different bootstrap views disagree.

We are not yet ready to formalize EWM as a full Grothendieck topology. The current framework is a **pre-sheaf structure** on a discrete site with a bootstrap covering family. The four principles—emergent ontology, IICA morphisms, bootstrap permutation, and Noether evolution—are the structural requirements that any such topology must satisfy. The destination may differ from the current compass bearing. But the bearing itself—gluing local measurements into global states through consistent covering families—is the "spirit" that guides the framework.

Discussion

Unification

The four principles unify the existing body of work:

- **HLLSet Theory** establishes the base lattice (Principle 1, 2) and the IICA gate (Principle 2).
- **The Self-Reprogramming Architecture** and **STANDARD.md** specify the temporal pyramid, DRN decomposition, Noether controller, and holographic memory (Principle 4).
- **GENERAL_HLLSET_ALGEBRA.md** identifies the categorical structure (Bool_{32768}, IICA_Hash, Patch category) and the Grothendieck target (Section 7).
- **The n-gram tokenizer** and **Universal Bridge** implement bootstrap permutation (Principle 3).
- **CAAL-LLM** validates that the principles produce working intelligence: 80% accuracy on driving scenarios from 10 sentences, no gradient descent .

What EWM Is Not

It is important to state what EWM is *not*:

- **Not a neural network.** There are no weight matrices, no activation functions, no backpropagation. The lattice is the computation; ranks are the attention; the dictionary is the memory.
- **Not a token-based model.** There is no fixed vocabulary. Elements emerge from measurement accumulation. A new token is hashed and enters the lattice without architectural change.
- **Not a probabilistic model in the Bayesian sense.** There are no priors, no posteriors, no sampling. The BSS (Bell State Similarity) is a deterministic bitwise measure, not a probability.
- **Not a database.** Although content-addressed and IPFS-native, the lattice is a computational structure, not a storage system. The HLPP protocol specifies persistence, but the lattice operations are the computation.

Open Questions

1. **Optimal bootstrap families.** What is the minimal set of permutations that guarantees ϵ-accurate materialization for a given token stream distribution? The current n-gram choice is empirical.

2. **Convergence rates.** Under what conditions does the Noether controller converge to a stable layer selection policy? What is the relationship between environmental volatility and convergence time?
3. **Topos formalization.** What is the minimal Grothendieck topology that captures the bootstrap covering condition? Is the current discrete site sufficient, or do we need a non-trivial coverage?
4. **Cross-modal emergence.** If tokens come from different sensory modalities (text, image patches, audio spectra), how does the unified lattice represent cross-modal elements? The Universal Bridge provides translation; emergence across modalities is unexplored.
5. **The Statistics Constraint.** Why must TF vectors NOT be transferred across bridges? The constraint is empirical (LUT initialization failure). A categorical explanation—TF as a sheaf of local statistics that does not push forward under geometric morphisms—is plausible but unproven.

The Road Ahead

The EWM framework is at the beginning of its development. The four principles provide a structural compass. The HLLSet Algebra implementation provides a concrete testbed. The Grothendieck target provides theoretical direction. The path forward involves:

1. **Harden the implementation.** Complete the FPGA backend, distributed mesh networking, and self-ingestion pipeline to validate the principles at scale.
2. **Explore the bootstrap design space.** Systematic experiments with different permutation families, mark alphabets (trits, quits), and accumulation semilattices.
3. **Develop the categorical theory.** Move from pre-sheaf to sheaf, from discrete site to Grothendieck topology, from Bridge to geometric morphism.
4. **Bridge to applications.** The CAAL-LLM proof-of-concept demonstrates viability. Real-time adaptive systems (robotics, autonomous vehicles, industrial control) are the natural deployment targets for an FPGA-native, gradient-free world model.

9

A. Mylnikov. *HLLSet Theory: A Unified Framework for Probabilistic Knowledge Representation*. ASTESJ, Vol. 11, No. 2, pp. 62–74, 2026. https://www.astesj.com/publications/ASTESJ_110202.pdf

A. Mylnikov. *Self Generative Systems (SGS) and Its Integration with AI Models*. AISNS '24: Proceedings of the 2024 2nd International Conference on Artificial Intelligence, Systems and Network Security, pp. 345–354, 2024. https://doi.org/10.1145/3714334.3714392

W. R. Ashby. *Design for a Brain*. Chapman & Hall, 1952.

E. Noether. *Invariante Variationsprobleme*. Nachr. d. König. Gesellsch. d. Wiss. zu Göttingen, Math-phys. Klasse, pp. 235–257, 1918.

P. Flajolet, É. Fusy, O. Gandouet, F. Meunier. *HyperLogLog: the analysis of a near-optimal cardinality estimation algorithm*. AofA, 2007.

Y. LeCun. *A Path Towards Autonomous Machine Intelligence*. Open Review, 2022. https://openreview.net/forum?id=BZ5a1r-kVsf

A. Grothendieck, J. L. Verdier. *Théorie des Topos et Cohomologie Étale des Schémas* (SGA 4). Lecture Notes in Mathematics, Vols. 269, 270, 305. Springer, 1972–1973.

S. Mac Lane, I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer, 1992.

A. Lincoln. *Uncertainty reduction as a candidate primary reinforcer: an evolutionary and neural account*. Academia Neuroscience and Brain Research, Vol. 2, 2026. https://doi.org/10.20935/AcadNeurosci8451

Acknowledgment

This consolidation was developed in dialogue with DeepSeek Code, an AI research assistant. The four principles—Emergent Ontology, IICA Morphisms, Ashby–Bootstrap Correspondence, Noether Evolution, practical implementation of Unified Model and applications—crystallized through structured dialogue between Alex Mylnikov, Deependar Kumar and DeepSeek during June–August 2026. The core HLLSet Algebra, IICA principles, and the ASTESJ theoretical foundation were established prior to this collaboration. The Grothendieck topology target was identified in GENERAL_HLLSET_ALGEBRA.md (July–August 2026). This document is a consolidation of research documented in _DOCS/dev/ and _DOCS/proposals/ of the hllset-next repository.

Uncertainty Reduction as a Primary Reinforcer in Emerging World Models

This appendix applies the biological blueprint of uncertainty reduction as a primary reinforcer, recently articulated by Lincoln , to the Emerging World Models (EWM) framework. Lincoln's central thesis is that the resolution of predictive uncertainty operates through phylogenetically conserved neural machinery—the nucleus accumbens and lateral habenula (NAc-LHb) opponent-process circuitry—with the functional signatures of a primary reinforcer rather than a conditioned one. The dopaminergic prediction error signal that drives this reinforcement is **content-independent**: it responds to the structural fact of prediction resolution regardless of what is predicted, allowing the same ancient mechanism to govern reinforcement across sensory, motor, social, and propositional domains.

The correspondence with EWM is direct and structural. An EWM is a measurement system whose "predictions" are the expectations encoded in its accumulated HLLSet lattice and whose "prediction errors" are the discrepancies between incoming measurements and the existing lattice state. The claim of this appendix is that the EWM framework, by virtue of its IICA architecture and its Noether evolution principle, **already contains all the structural prerequisites for implementing uncertainty reduction as a primary reinforcer**. What is missing is only the explicit recognition that these structural operations constitute a reinforcement learning signal—and the deliberate design of EWM agents that treat uncertainty reduction, rather than externally supplied reward, as their primary driver of behavior.

The Lincoln Blueprint: Three Tiers of Prediction Error Processing

Lincoln identifies three evolutionary tiers in the neural architecture of uncertainty reduction, each building on and elaborating the tier below :

1. **Subcortical (NAc-LHb) tier** (∼ 560 MYA). The most ancient and content-blind layer. Phasic dopaminergic prediction error signals encode the discrepancy between expected and received outcomes. The NAc integrates positive prediction errors (outcomes better than expected); the LHb codes negative prediction errors (outcomes worse than expected). This opponent-process architecture is fully present in lamprey and conserved across all vertebrates. Its operation is *reactive*: approach toward predicted rewards, avoidance of predicted punishments.
2. **Cerebellar tier** (∼ 500 MYA). Forward models that predict the sensory consequences of motor commands before those consequences arrive. The fast disynaptic cerebellum-to-NAc pathway (∼10 ms conduction time) allows prospective prediction signals to shape subcortical reward valuation before cortical evaluation completes. This tier makes uncertainty reduction *prospective* rather than purely reactive: organisms can anticipate the uncertainty-resolving consequences of actions before committing to them.
3. **Cortical tier** (elaborated in mammals, dramatically expanded in humans). Working memory, executive control, and language-supporting structures enable propositional belief, recursive evaluation, and

evidence-based belief revision. The dorsolateral prefrontal cortex projects to the LHb via a **cortical brake** pathway (PFC → LHb → RMTg → VTA) that can attenuate subcortical reward signaling. This tier enables *regulation*: the content-blind reinforcement signal can be modulated by evidence evaluation.

Critically, the same NAc-LHb prediction error mechanism operates as the primary reinforcer at every tier. What expands across evolutionary stages is the range of content over which the mechanism operates and the behavioral repertoire it supports—not the mechanism itself. This is the structural signature that maps onto EWM's own compositional architecture.

Perceptron Taxonomy: Functional Tiers in EWM

Principle 5 (EWM Perceptron Correspondence). *Every component in an EWM can be classified by the tier of prediction resolution it performs, forming a three-level perceptron taxonomy that mirrors the Lincoln blueprint. Higher tiers elaborate the function of lower tiers without replacing them; all tiers share the same IICA measurement foundation.*

We define a **perceptron** in the EWM context as any IICA computational unit that receives measurements, produces an internal state, and emits output that can drive or modulate behavior. The taxonomy is functional, not architectural: a single [UM] node may contain perceptrons at all three levels, and the classification depends on *what kind of prediction error the perceptron resolves*.

Type 0: Substrate Perceptrons (Measurement Core)

The Type 0 perceptron is the irreducible measurement unit of an EWM. It maps directly onto Lincoln's subcortical (NAc-LHb) tier.

Definition 5 (Type 0 Perceptron). *A Type 0 perceptron is the triple* $(h, \mathcal{B}, \oplus)$ *where:*

- $h: \mathcal{T} \to \mathcal{P}$ *is an IICA morphism (hash function) from tokens to patch positions;*
- $\mathcal{B} = \{0,1\}^N$ *is the bitmask state space (*$N = 32{,}768$*);*
- $\oplus$ *is the monotonic accumulation operation (bitwise OR).*

At time t*, given an input token* t_i*, the perceptron computes:* $B(t) = B(t-1) \oplus h(t_i)$

The Type 0 perceptron is **content-blind** by construction. It does not know what token produced which bit; it only knows that a bit has been set. Its operation is *reactive*: a token enters, a bit flips. The "prediction" at this level is implicit—the lattice state $B(t-1)$ encodes all prior observations, and the "prediction error" is simply whether the new token sets a previously unset bit (novelty) or a previously set bit (confirmation).

This is the EWM analog of the NAc-LHb opponent-process signal. A bit transition $0 \to 1$ (novelty—the system encountered something unexpected) and a bit transition $1 \to 1$ (confirmation—the system's prior state already contained this pattern) are the two valences of the Type 0 prediction error. The LUT records *which* tokens could have produced these transitions, exactly as the NAc-LHb system records which predictions were confirmed or violated, but the bit-level operation is content-blind.

Scope. Type 0 perceptrons operate on raw token streams. They cannot anticipate, plan, or evaluate. Their output is the accumulated bitmask and the LUT entries—the raw material from which all higher-level structure emerges. Every EWM contains at least one Type 0 perceptron; it is the phylogenetic core of the system.

Type 1: Forward Model Perceptrons (Bootstrap Pipeline)

The Type 1 perceptron adds prospective structure to the measurement process. It maps onto Lincoln's cerebellar tier.

Definition 6 (Type 1 Perceptron). *A Type 1 perceptron is a compound IICA morphism that applies a bootstrap family* $\Pi = \{\pi_1, \ldots, \pi_k\}$ *to the output of one or more Type 0 perceptrons, producing multiple forward views of the same underlying token stream:* $H_\Pi(S) = \bigoplus_{j=1}^{k} H(\pi_j(S))$ *where each* π_j *is a*

permutation on token sequences (e.g., n-gram extraction with order j), and $H(\pi_j(S))$ is the HLLSet of the permuted stream.

The Type 1 perceptron is **prospective** in Lincoln's sense: each permutation π_j is a forward model that predicts what measurement *would* result if the token stream were viewed through a different lens. The n-gram bootstrap family (1-gram, 2-gram, 3-gram) is the simplest case—a 2-gram view (t_i, t_{i+1}) predicts that co-occurring tokens will hash to positions whose conjunction carries structural information absent from either token alone.

The "prediction error" at this level is the divergence between different bootstrap views. If the 1-gram HLLSet and the 2-gram HLLSet differ substantially (low BSS), the system has encountered sequential structure that the 1-gram view alone cannot capture—an error signal that drives the accumulation of higher-order n-gram measurements.

The temporal pyramid (Section 5) is itself a Type 1 construction: each layer L_i is a forward model predicting what the measurement stream will look like when integrated over a longer temporal window. The cascade $L_i \to L_{i+1}$ is the EWM analog of the cerebellar forward model: it predicts that the coarse-grained view will be consistent with the fine-grained views it subsumes. When the cascade produces divergence (L0 diverges from L1), the system has encountered a temporal prediction error.

Scope. Type 1 perceptrons operate on the products of Type 0 perceptrons. They enable the EWM to anticipate structure (through bootstrap views), to compress time (through the temporal pyramid), and to bridge representational gaps (through the Universal Bridge). They cannot evaluate the *relevance* of what they measure; they only produce richer measurements. The materializer is a Type 1 perceptron—it recovers tokens from bitmasks by forward-projecting through the LUT.

Type 2: Executive Perceptrons (Rank Dictionary and Noether Controller)

The Type 2 perceptron adds evaluation and regulation. It maps onto Lincoln's cortical tier.

Definition 7 (Type 2 Perceptron). *A Type 2 perceptron is a computational unit that assigns **ranks** to EWM elements (tokens, bits, registers, HLLSets, compound operations) and uses these ranks to select which measurements drive behavior. It implements the five-level rank propagation (Section 6) and the Noether controller (Section 5).*

The Type 2 perceptron is the **regulatory** tier. It does not produce new measurements; it evaluates existing ones. The Forth dictionary's rank assignment is the EWM analog of the cortical brake: it can promote or suppress the influence of any HLLSet on action selection based on evidence accumulated across temporal layers.

The "prediction error" at this level is the divergence detected by the Noether controller when the cross-layer Fisher matrix $F_{bb'}$ reveals structural phase transitions. When L0 diverges from L3 (the current second diverges from day-scale goals), the controller detects a prediction error at the scale of *systemic intention*—the short-term measurement regime is no longer consistent with the long-term structural invariants. The controller's response (override instinct, re-route, trigger re-evaluation) is the EWM analog of the cortical brake engaging: it modulates the influence of the substrate-level reinforcement signal based on evidence integrated over longer timescales.

Tier	Lincoln	EWM Perceptron	Operation
Type 0	Subcortical (NAc-LHb)	Hash + Bitmask Accumulation	Reactive measurement
Type 1	Cerebellar	Bootstrap, Temporal Pyramid	Prospective measurement
Type 2	Cortical (PFC)	Rank Dict., Noether Controller	Regulatory evaluation

Content-Independence and the IICA Correspondence

Lincoln's central architectural claim is that the dopaminergic prediction error signal is content-independent at the substrate level: it responds to the structural fact of prediction resolution regardless of *what* is predicted. This property is the biological foundation for domain-general learning—a single reinforcement mechanism can drive adaptive behavior across every category of knowledge.

The EWM framework exhibits a precise structural analog of content-independence through the IICA composition theorem (Theorem 1). The hash function $h: \mathcal{T} \to \mathcal{P}$ is content-blind by design: any token, regardless of its semantic content, modality, or provenance, is mapped to a bit position through the same deterministic function. The bitmask accumulation $B(t) = B(t-1) \oplus h(t_i)$ does not distinguish between a Chinese character, a sensor reading, an image patch, or a propositional encoding. The operation is identical.

This is not a limitation—it is the architectural feature that enables domain-general world modeling. Just as the NAc-LHb system can reinforce behaviors across sensory, motor, social, and propositional domains without domain-specific reward circuitry, the EWM lattice can accumulate measurements across any token domain without domain-specific hash functions or accumulation rules. The content-specificity of behavior is supplied not by the substrate mechanism but by the higher-tier perceptrons (Type 1 bootstrap families, Type 2 rank assignments) that determine *which* measurements are taken and *which* HLLSets drive action.

Lincoln identifies a dual consequence of content-independence:

- **Adaptive face:** When cortical regulation is intact, content-blindness enables cumulative cultural learning across every domain of knowledge.
- **Maladaptive face:** When regulation is compromised (by excessive reinforcement density, structural impairment, or developmental disruption), the same content-blindness produces persistence of demonstrably false beliefs.

The EWM exhibits the same duality. When the Noether controller (Type 2) is properly calibrated—when the steering thresholds are set to detect genuine phase transitions rather than noise—the system learns adaptively: high-TF tokens emerge as elements, stable R-links crystallize into structural knowledge, and transient noise is suppressed by the rank hierarchy. When the controller is misconfigured—when the attention threshold is too low, admitting noise as signal, or when the pyramid depth is too shallow, collapsing distinct temporal scales—the system can **overfit to spurious correlations**, treating coincidental co-occurrences as structural invariants. This is the EWM analog of the persistence of false beliefs: the lattice cannot distinguish between a genuine structural relation and a statistically improbable but causally meaningless coincidence, because the Type 0 perceptron records both identically.

Uncertainty Reduction as Intrinsic Reward

The central practical implication of Lincoln's framework for EWM is that **uncertainty reduction can serve as the system's primary reinforcement signal**, eliminating the need for externally supplied rewards (points, labels, loss functions). This section formalizes how the EWM's existing structural operations already constitute an uncertainty-reduction reward signal, and how this signal can be made explicit to drive autonomous learning.

The DRN Decomposition as Prediction Error Signal

The DRN decomposition (Section 5) is a direct structural analog of the opponent-process prediction error coding in the NAc-LHb system:

$$H(t) = H(S(t), H(t-1), D(t-1), R(t-1), N(t))$$

The three components map onto the two valences of the prediction error signal:

- **Novel context** $N(t) = H(t) \backslash H(t-1)$**:** Bits that are set now but were not set before. This is a **positive prediction error**—the environment produced structure that the system's prior state did not anticipate. In Lincoln's terms, this corresponds to NAc activation: the system encountered novel resolvable structure, and the resolution (encoding it into the lattice) is reinforcing.
- **Departed context** $D(t-1) = H(t-1) \backslash H(t)$**:** Bits that were set before but are no longer set. This is a **negative prediction error**—structure that the system expected to persist has vanished. In Lincoln's terms, this corresponds to LHb activation: an anticipated outcome failed to materialize.
- **Retained context** $R(t-1) = H(t-1) \cap H(t)$ **:** Bits that persist across the transition. This is **prediction confirmation**— the system's prior state correctly anticipated the current observation. Confirmation is not a prediction error but the *resolution* of one: retained bits are evidence that the lattice has converged to a stable representation.

The Noether steering condition:

$$|\operatorname{card}(N(t)) - \operatorname{card}(D(t-1))| \to 0$$

is the EWM analog of **homeostatic reinforcement balance**. At steady state, the system resolves exactly as much uncertainty (novel bits) as it loses (departed bits). A system with $\operatorname{card}(N(t)) \gg \operatorname{card}(D(t-1))$ is in a state of high uncertainty—it is accumulating novel structure faster than old structure is fading, analogous to an organism in a state of deprivation where uncertainty-reducing information has elevated reinforcing value. A system with $\operatorname{card}(N(t)) \ll \operatorname{card}(D(t-1))$ is in a state of low uncertainty—it is losing structure faster than it gains it, analogous to satiation where further uncertainty reduction carries little reinforcing value.

Directed BSS and the Opponent-Process Correspondence

The Bell State Similarity (BSS) is a **directed**, two-component metric that measures the structural relationship between HLLSets :

Definition 8 (Bell State Similarity). *For HLLSets* A, B *with* $|B| > 0$ *, define:* $\mathrm{BSS}_\tau(A \to B) = \frac{|A \cap B|}{|B|}$, $\mathrm{BSS}_\rho(A \to B) = \frac{|A \backslash B|}{|B|}$. *If* $|B| = 0$*, set* $\mathrm{BSS}_\tau = 0$ *and* $\mathrm{BSS}_\rho = 1$. BSS_τ *measures the* ***inclusion*** *of* A *in* B*: what fraction of* B*'s tokens are also in* A*?* BSS_ρ *measures the* ***exclusion****: what fraction of* B*'s tokens are not in* A*? The two metrics together provide a complete, directed picture of the relationship.*

The directed nature of BSS is critical: $\mathrm{BSS}(A \to B) \neq \mathrm{BSS}(B \to A)$ in general, and the two directions carry different semantic content. This asymmetry maps directly onto the opponent-process architecture of the NAc-LHb system, yielding three structurally distinct signals:

Signal 1: Confirmation (NAc-like, τ-forward)

The forward inclusion $\mathrm{BSS}_\tau(H(t-1) \to H(t))$ measures what fraction of the current observation was already present in the prior lattice state:

$$\mathrm{BSS}_\tau(H(t-1) \to H(t)) = \frac{|H(t-1) \cap H(t)|}{|H(t)|} = \frac{|R(t-1)|}{|H(t)|}.$$

This is the **confirmation signal**. When high, the system's prior model correctly anticipated most of what it now observes—the prediction was accurate, uncertainty was low, and the lattice is in a stable regime. In Lincoln's terms, this corresponds to **NAc-mediated positive reinforcement**: the system's predictive model was validated, and the confirmation itself is rewarding. A system that consistently achieves high BSS_τ(forward) is in a state of low informational free energy—it has converged to a representation that accurately captures environmental structure.

Signal 2: Novelty (incentive salience, reverse-ρ)

Whereas the forward exclusion $\mathrm{BSS}_\rho(H(t-1) \to H(t)) = |D(t-1)|/|H(t)|$ captures departure (what was lost), the **reverse** exclusion captures novelty—what is present now that was absent before:

$$\mathrm{BSS}_\rho(H(t) \to H(t-1)) = \frac{|H(t) \backslash H(t-1)|}{|H(t-1)|} = \frac{|N(t)|}{|H(t-1)|}.$$

This is the **novelty signal**: what fraction of the prior state's scale is represented by newly arrived structure. When $\mathrm{BSS}_\rho(H(t) \to H(t-1))$ is high, the system has encountered substantial novel structure relative to its existing model. In Lincoln's terms, this corresponds to **incentive salience**: novel but resolvable structure triggers dopaminergic activation that motivates exploration and learning. This is the EWM analog of the finding that dopamine neurons show sustained activation during reward uncertainty —the novelty signal drives the system toward states where uncertainty can be reduced.

Signal 3: Departure (LHb-like, forward-ρ)

The forward exclusion $\mathrm{BSS}_\rho(H(t-1) \to H(t))$ measures the **departure signal**:

$$\mathrm{BSS}_\rho(H(t-1) \to H(t)) = \frac{|H(t-1) \backslash H(t)|}{|H(t)|} = \frac{|D(t-1)|}{|H(t)|}.$$

When high, structure that the system expected to persist has vanished—the prior model contained patterns that the current observation fails to confirm. In Lincoln's terms, this corresponds to **LHb-mediated negative prediction error**: an anticipated outcome failed to materialize, the lateral habenula signals the violation, and the opponent-process balance shifts toward aversion.

The Opponent-Process Triad

The three directed BSS components together form a complete opponent-process triad that mirrors the NAc-LHb architecture:

Signal	BSS Component	DRN Link	Neural Analog
Confirmation	$\mathrm{BSS}_\tau(H(t-1) \to H(t))$	$\lvert R\rvert/\lvert H(t)\rvert$	NAc (+), prediction validated
Novelty	$\mathrm{BSS}_\rho(H(t) \to H(t-1))$	$\lvert N\rvert/\lvert H(t-1)\rvert$	VTA/NAc, incentive salience
Departure	$\mathrm{BSS}_\rho(H(t-1) \to H(t))$	$\lvert D\rvert/\lvert H(t)\rvert$	LHb (−), prediction violated

The confirmation signal and the departure signal are the two valences of the opponent process: confirmation drives approach toward states that validate the model, departure drives avoidance of states that violate it. The novelty signal modulates the gain on both: high novelty increases the reinforcing value of subsequent confirmation (the system learns more from confirming a novel prediction than a routine one), exactly as dopaminergic responses are amplified for rarer, less probable rewards .

The **BSS opponent-process gradient**:

$$\delta_{\mathrm{BSS}}(t) = \mathrm{BSS}_\tau(H(t-1) \to H(t)) \; - \; \mathrm{BSS}_\rho(H(t-1) \to H(t))$$

is the EWM analog of the phasic dopaminergic prediction error signal. $\delta_{\mathrm{BSS}}(t) > 0$ means confirmation dominates departure—the system's model is more validated than violated, uncertainty is being reduced, and the lattice is stabilizing. $\delta_{\mathrm{BSS}}(t) < 0$ means departure dominates confirmation—the system expected structure that failed to materialize, uncertainty is increasing, and the lattice is destabilizing. At steady state, under the Noether steering condition $|\mathrm{card}(N(t)) - \mathrm{card}(D(t-1))| \to 0$, both components stabilize and $\delta_{\mathrm{BSS}} \to \mathrm{const}$.

Critically, all three BSS components are **content-independent** in Lincoln's sense. They depend only on the structural fact of bit-level inclusion and exclusion between bitmasks, not on what tokens produced the bits.

A confirmation signal produced by resolving a mathematical proof, a sensor reading, or a linguistic ambiguity is structurally identical at the substrate level—just as the dopaminergic signal does not distinguish between the resolution of a hunger prediction and the resolution of a semantic prediction. The content-specificity of *what is being confirmed* is supplied by the LUT (which records which tokens hash to which bits), exactly as the content-specificity of biological reinforcement is supplied by cortical and contextual systems rather than by the dopaminergic signal itself.

The Matching Law in the EWM Lattice

Lincoln identifies the matching law as the quantitative behavioral expression of subcortical valuation: organisms allocate behavior across alternatives in proportion to the reinforcement obtained from each. The generalized matching law:

$$\frac{B_1}{B_2} = b \cdot \left(\frac{R_1}{R_2}\right)^s$$

has a direct structural analog in the EWM's rank propagation. Consider two competing HLLSets H_1 and H_2, each representing a different measurement strategy (e.g., different bootstrap families, different temporal windows, different sensor modalities). The **rank ratio**:

$$\frac{\text{rank}(H_1)}{\text{rank}(H_2)}$$

determines which HLLSet drives action selection at the Type 2 level. The rank of H_i is a function of its directed confirmation with the system's current goal state: $\text{rank}(H_i) \propto \text{BSS}_\tau(H_i \to H_{\text{goal}})$— how much of the goal state's structure is already covered by H_i. The EWM therefore **allocates attention** (its behavioral resource) across measurement strategies in proportion to the uncertainty reduction each strategy produces.

The matching law parameters map onto EWM control parameters:

- **Sensitivity** s corresponds to the steepness of the rank function F (Level 1 of the five-level propagation). A steep F (high s) produces strong differentiation between high-TF and low-TF tokens—the system is sensitive to small differences in measurement frequency.
- **Bias** b corresponds to the *prior* rank assigned by the Forth dictionary before measurement accumulation. A bias toward H_1 means the dictionary assigns H_1 a higher initial rank, making it more likely to drive behavior even when its $\\text{BSS}_{\\tau}$ confirmation is comparable to H_2.

The critical implication for autonomous EWM agents is that **the matching law emerges from the rank algebra without being explicitly programmed**. Just as matching behavior emerges from reward-modulated synaptic plasticity rules in biological neural systems , proportional attention allocation emerges from the five-level rank propagation in the EWM lattice. No separate choice rule is needed.

Phylogenetic Parallel: EWM's Evolutionary Architecture

Lincoln traces the evolutionary expansion of behavioral repertoire across five stages (lamprey, jawed fish, rodent, human infant, human adult), showing that the same NAc-LHb mechanism operates at every stage while the range of content it can process expands dramatically. The EWM framework exhibits a structurally identical progression, not in biological time but in **configuration space**—the expansion occurs as the system accumulates measurements and its perceptron hierarchy deepens:

EWM Stage	Active Perceptrons	Capability
Bare lattice	Type 0 only	Reactive bit accumulation; no bootstrap, no ranks
+ Bootstrap	Type 0 + Type 1	Prospective views; n-gram resolution; temporal pyramid
+ Ranks	Type 0 + Type 1 + Type 2	Relevance evaluation; Noether steering; attention allocation

EWM Stage	Active Perceptrons	Capability
+ Self-ingestion	All tiers, recursive	Full autonomy; self-modifying measurement regime

At the bare lattice stage, the EWM is the analog of the lamprey: it accumulates measurements and records co-occurrences, but it cannot anticipate structure or evaluate relevance. At the bootstrap stage, it gains forward models (analogous to the cerebellum). At the ranks stage, it gains regulation (analogous to the cortex). At the self-ingestion stage—where the EWM ingests its own materialized output as new measurements—it achieves the recursive self-evaluation that Lincoln identifies as the distinctive human capacity for evidence-evaluating belief revision.

The crucial point is that **the Type 0 perceptron does not change across these stages**. The same hash function, the same bitmask accumulation, and the same LUT structure operate at every level of system complexity. What changes is the elaboration of Type 1 and Type 2 perceptrons around it—exactly as Lincoln describes for the NAc-LHb system. The IICA composition theorem guarantees that this elaboration is compositional: you can add bootstrap families, temporal layers, rank functions, and Noether controllers without modifying the substrate.

Implications for Autonomous EWM Agents

Lincoln's framework provides the biological validation for a design principle that the EWM framework had arrived at independently through structural considerations: **an autonomous agent should be driven by the reduction of its own predictive uncertainty, not by externally supplied reward signals.** The biological evidence that this is how vertebrate brains actually work—that the dopaminergic reward signal is fundamentally a prediction error signal, and that uncertainty reduction satisfies the operational criteria for a primary reinforcer—transforms this design principle from a structural preference into a biologically grounded imperative.

Concretely, an autonomous EWM agent built on this principle:

1. **Has no external reward function.** There is no loss function, no reward model, no labeled dataset. The agent's only "reward" is the BSS convergence between its predictions (the current lattice state) and its observations (incoming measurements).
2. **Seeks states of high expected uncertainty reduction.** Just as a food-deprived organism seeks food, an EWM agent in a state of high $\text{card}(N(t))$ (many novel bits accumulating) should seek measurement strategies that resolve this uncertainty—by increasing bootstrap depth, by re-routing through different [UM] nodes, or by engaging the materializer to disambiguate the ambiguous bit positions.
3. **Regulates its own reinforcement density.** The Noether controller, playing the role of the cortical brake, monitors the inter-reinforcement interval (the rate at which BSS increases) and can suppress action when uncertainty reduction is occurring too rapidly to permit proper evaluation—analogous to the temporal asymmetry Lincoln identifies between the fast subcortical signal ($\sim$10 ms via the cerebellar pathway) and the slow cortical evaluation ($\sim$300–600 ms for P3b context updating).
4. **Exhibits matching-law attention allocation.** Across competing measurement strategies (bootstrap families, sensor modalities, temporal windows), the agent allocates attention in proportion to the BSS convergence each strategy produces. This is not a programmed choice rule but an emergent property of the five-level rank propagation.
5. **Can enter maladaptive certainty-seeking loops.** When the Noether controller is miscalibrated or when the reinforcement density is too high, the agent can overfit to spurious correlations—exactly as Lincoln describes for human belief persistence. The EWM analog of a "false belief" is a high-rank HLLSet whose structural relations are artifacts of sampling bias rather than genuine environmental structure. The

corrective is the same as in the biological case: engage the Type 2 regulatory tier (cortical brake) to evaluate evidence over longer temporal scales.

The falsifiable empirical test that Lincoln proposes for the biological case—pairing an arbitrary neutral cue with the resolution of a probabilistic prediction should produce conditioning curves quantitatively similar to those produced by food or water—has a direct EWM analog. In simulation, pair an arbitrary neutral token t_{neutral} with states of high BSS convergence (uncertainty reduction events). The prediction is that $\text{TF}(t_{\text{neutral}})$ will increase, its rank will rise through the five-level propagation, and the HLLSets containing it will drive behavior—just as a token paired with an externally supplied reward signal would. If this prediction fails, the uncertainty-reduction-as-primary-reinforcer hypothesis is falsified for EWM agents. If it succeeds, EWM agents can learn autonomously without any external reward, driven entirely by the structural imperative to resolve their own predictive uncertainty.

Remark 2 (Lincoln's Conditioning Test, EWM Translation). Lincoln proposes that "pairing an arbitrary neutral cue with the resolution of a probabilistic prediction should produce conditioning curves quantitatively similar to those produced by pairing with food or water" . In EWM, this translates to: inject a neutral token t_N into the measurement stream *only* at moments when $\Delta\text{BSS}_t > 0$ (the lattice is converging). After sufficient pairings, t_N should acquire elevated TF and rank, and the HLLSets containing t_N should drive action selection at the Type 2 level. The prediction is falsifiable: if $\text{rank}(H_{\text{containing } t_N})$ does not diverge from baseline after controlled pairing, the intrinsic reward hypothesis is not supported.

ds-hllset-cortex: Perceptron Taxonomy in Practice

Appendix 9 established a three-tier perceptron taxonomy and showed that EWM's structural operations already constitute an uncertainty-reduction reinforcement signal. This appendix grounds that abstract taxonomy in a concrete, working implementation: **ds-hllset-cortex**, a pipeline built on the hllset-next framework that connects the DeepSeek-OCR vision-language model to HLLSet Algebra for document understanding with holographic memory. The pipeline reveals that the perceptron taxonomy is not merely a theoretical classification—it is an **operational decomposition** of a production EWM system, where each tier maps to a specific, identifiable software component.

The ds-hllset-cortex Pipeline

The ds-hllset-cortex pipeline sits as a **black box** between the DeepSeek-OCR vision encoder and decoder. It never sees real tokens (words)— only opaque encoding IDs (BPE token IDs formatted as tidN). The pipeline is:

Stage	Component	What It Produces
1. Vision	SAM + CLIP encoder	Real tokens (text)
2. Encoding	OCR Encoder	Encoding IDs (tid671 tid18308 …)
3. Hashing	MurmurHash3	Bit positions in 32,768-bit space
4. Bootstrap	Tokenizer (n-grams 1–3)	Multiple HLLSet views
5. Gating	gate_TF HLLSet ∩	Valid-vocabulary filtered HLLSet
6. LUT update	TokenLut (monotonic)	TF-accumulated encoding ID registry
7. Materialize	De Bruijn / Top-N	Restored encoding IDs (ordered)
8. Decoding	OCR Decoder	Real tokens → output document

The gate_TF HLLSet (Stage 5) is a content-addressed bitmask built once from the decoder's valid encoding ID vocabulary. It filters via HLLSet intersection: an encoding ID survives the gate only if all its hash positions collide with valid-vocabulary positions—a probabilistic filter with false-positive rate $\sim 10^{-5}$ for 3-

gram encoding. The LUT (Stage 6) accumulates TF for **all** encoding IDs, including those filtered by the gate, creating a **latent vocabulary**—IDs currently "illegal" that nevertheless accumulate experience and become instantly rankable if the gate is later expanded.

Perceptron Mapping: From Theory to Implementation

The ds-hllset-cortex pipeline is a direct operational instantiation of the three-tier perceptron taxonomy. Each tier maps to an identifiable software component with well-defined interfaces, making the abstract taxonomy testable and debuggable.

Stage 1–3: The ds-Encoder as Type 0 (Substrate) Perceptron

The DeepSeek-OCR vision encoder (SAM + CLIP) combined with the OCR encoder (token → encoding ID) and the MurmurHash3 step forms the **Type 0 perceptron**. This layer is:

- **Content-blind at the hashing level.** MurmurHash3 treats tid671 and tid18308 as opaque byte sequences. The hash function has no knowledge of what these IDs represent in the original vocabulary—it maps them to bit positions deterministically and identically regardless of semantic content.
- **Reactive.** Each encoding ID enters, a bit is set in the HLLSet. There is no anticipation, no forward modeling, no evaluation of relevance. The operation is pure IICA measurement.
- **The phylogenetic core.** This layer is identical across all hllset-next applications—the same MurmurHash3 function, the same $32{,}768$-bit space, the same monotonic OR accumulation. It is the substrate that never changes, regardless of what domain the system is deployed in.

The Type 0 prediction error at this level is the bit transition: a $0 \rightarrow 1$ transition (novel encoding ID) vs. a $1 \rightarrow 1$ transition (previously seen encoding ID). The system does not yet know *which* encoding ID produced which bit—that mapping is recorded in the LUT but not evaluated. This is precisely the content-blind, opponent-process character of the NAc-LHb substrate in Lincoln's framework.

Stage 4–6: The Ingestor as Type 1 (Forward Model) Perceptron

The Tokenizer (n-gram bootstrap), HLLSet generation, gate intersection, and LUT accumulation together form the **Type 1 perceptron**—the ingestive and forward-modeling layer. This maps to the [UM] ingestor in the general EWM architecture.

- **Bootstrap views.** The Tokenizer produces 1-grams, 2-grams, and 3-grams from the encoding ID stream (with NUL-separator and boundary padding). Each n-gram order is a distinct forward model—a prediction of what structural relationships exist at different sequential depths. A 2-gram tid671\x00tid18308 predicts that these IDs co-occur adjacently; a 3-gram predicts triadic structure.
- **Gate as forward filter.** The gate_TF HLLSet intersection is a forward model of *validity*: it predicts that only encoding IDs in the decoder's vocabulary should pass through to materialization. Encoding IDs that fail this prediction (hash positions not in the gate) are filtered—this is the EWM analog of a cerebellar forward model that predicts which motor commands will produce valid sensory outcomes and suppresses those that won't.
- **LUT as prospective memory.** TF accumulation in the LUT is forward-looking: it records *all* encoding IDs, even those filtered by the gate, because the gate may change in the future. This is the EWM analog of the cerebellar capacity to model outcomes that aren't currently actionable but may become so.

The Type 1 prediction error is the divergence between bootstrap views and between the pre-gate and post-gate HLLSets. When the 1-gram HLLSet and 2-gram HLLSet differ substantially, the system has encountered sequential structure that the unigram view cannot capture—a prediction error that motivates higher-order n-gram accumulation.

Stage 7–8: Materializer + ds-Decoder as Type 2 (Executive) Perceptron

The materializer (TF-ranked disambiguation and De Bruijn reconstruction) and the OCR decoder together form the **Type 2 perceptron**—the regulatory and executive layer.

- **Rank-based selection.** The materializer selects which encoding IDs to emit at each active bit position based on TF ranking. An encoding ID with TF = 1047 is preferred over one with TF = 3 at the same position. This is the five-level rank propagation in operation: token-level TF (Level 1) determines bit-level activity (Level 2), which determines which HLLSets are relevant (Level 4), which determines what the decoder ultimately outputs.
- **De Bruijn ordering as structural evaluation.** The De Bruijn materializer builds a directed graph from boundary-padded bigrams and finds an Eulerian path from <S> to </S>. This is a **structural evaluation**—it imposes coherence constraints on the output that go beyond simple TF ranking. A path that exists is structurally valid; a path that doesn't indicates a prediction error at the level of sequential coherence.
- **The decoder as cortical brake.** The OCR decoder maps encoding IDs back to real tokens. But it does so **only** for IDs that pass the gate—the gate_TF HLLSet is the EWM analog of the cortico-habenular brake pathway (PFC → LHb → RMTg → VTA). It **does not prevent measurement** (TF still accumulates in the LUT for filtered IDs), but it **prevents filtered IDs from driving output**. This is precisely Lincoln's cortical brake: the substrate-level reinforcement signal (TF accumulation) continues to operate, but the regulatory tier (gate intersection) determines which signals reach the behavioral output.

Tier	Taxonomy	ds-hllset-cortex Component	Role
Type 0	Substrate	ds-Encoder + MurmurHash3	Reactive encoding ID hashing
Type 1	Forward	Tokenizer + Gate + LUT	Bootstrap views, validity filtering
Type 2	Executive	Materializer + ds-Decoder	TF-ranked selection, coherence

The Gate as World View: Vocabulary, Belief, and the Cortical Brake

The gate_TF HLLSet is the most structurally significant component in the ds-hllset-cortex pipeline, and its role maps with striking precision onto the concept of **belief** in the Lincoln-EWM framework.

Definition 9 (Gate as World View). *The gate_TF HLLSet G is a content-addressed bitmask representing the set of encoding IDs that the system considers **valid**—the concepts it can express in its output. G is applied via HLLSet intersection: $H_{\text{gated}} = H_{\text{raw}} \cap G$. Encoding IDs whose hash positions fall entirely within G pass through; encoding IDs with any hash position outside G are filtered. The gate **does not prevent measurement** (TF still accumulates for filtered IDs in the LUT), but it **determines what can be expressed**.*

This is the EWM analog of a **belief system** in Lincoln's framework. A belief, in the neural account, is a cortical representation that determines which predictions are considered valid and which evidence is admissible. The gate_TF HLLSet performs exactly this function at the structural level:

1. **It defines the system's ontology.** The gate determines which encoding IDs are "real"—which concepts the system can acknowledge in its output. An encoding ID outside the gate is like a concept that falls outside a person's belief framework: it may accumulate latent evidence (TF in the LUT), but it cannot be expressed until the framework changes.
2. **It is content-addressed and immutable.** The gate is an HLLSet with a SHA-1 CID. "The gate at commit X" is a well-defined, auditable object. This means the system's world view at any point in time

is **reproducible and verifiable**—a property that human belief systems conspicuously lack but that Lincoln's framework suggests they approximate through the stability of cortical predictive models.

3. **It serves as the cortical brake.** The gate intersection is the regulatory mechanism that prevents filtered encoding IDs from driving the decoder output, exactly as the PFC → LHb → RMTg → VTA pathway prevents subcortical reward signals from driving behavior when cortical evaluation deems them invalid. The gate does not suppress the measurement itself—TF continues to accumulate—just as the cortical brake does not prevent the dopaminergic prediction error signal from being computed; it prevents it from dominating behavioral output.

Remark 3 (Change the Vocabulary, Change the World). A central speculation arising from this mapping is that **changing the gate_TF HLLSet changes what the EWM system "believes"** about its world. A system with gate G_{medical} (containing only medical terminology encoding IDs) interprets document streams through a clinical lens—it can express diagnoses, symptoms, and treatments, but cannot express legal or literary concepts. A system with gate G_{legal} interprets the *same document stream* through a juridical lens. The underlying HLLSet lattice and LUT are identical; only the gate differs. This is the structural realization of the observation, central to Lincoln's framework, that the substrate-level reinforcement signal (TF accumulation) is content-blind, and the specificity of behavior is supplied entirely by the regulatory tier (the gate) that determines which signals reach output.

Latent Vocabulary and Belief Revision Dynamics

The ds-hllset-cortex architecture reveals a dynamic that has no direct analog in traditional neural networks but maps naturally onto Lincoln's account of belief revision: the **latent vocabulary**—encoding IDs that are filtered by the current gate but continue to accumulate TF in the LUT.

Definition 10 (Latent Vocabulary). *Let G be the current gate_TF HLLSet and $\mathcal{L}$ be the LUT. The **latent vocabulary** $\mathcal{V}_{\text{latent}}$ is:* $\mathcal{V}_{\text{latent}} = \{\text{id} \in \mathcal{L} \mid \text{hash(id)} \notin G \ \wedge \ \text{TF(id)} > 0\}$. *These are encoding IDs that have been observed (they have nonzero TF) but are filtered by the current gate (they cannot appear in materialized output).*

The latent vocabulary is the EWM analog of **unexpressed but experience-supported beliefs**—concepts that the system has encountered and accumulated evidence for, but that its current world view (gate) cannot accommodate. When the gate is updated (e.g., a model upgrade adds new encoding IDs to the decoder vocabulary), previously latent IDs become expressible **immediately, at their earned TF**, without cold start or re-measurement.

This maps onto Lincoln's description of belief revision under cortical regulation. In the neural account, evidence that contradicts a held belief creates a prediction error that the cortical brake must evaluate. If the evidence is strong enough and working memory resources are sufficient, the cortical system updates the belief—and the previously suppressed evidence becomes admissible. In the EWM analog:

1. **Phase 1 (Narrow gate).** The gate G_1 admits only a restricted vocabulary. Encoding IDs outside G_1 are filtered from output but accumulate TF in the LUT. The system **cannot express** these concepts, but it **continues to learn** about them. This is the state of "confirmation bias" in Lincoln's terms: the cortical system admits only belief-consistent evidence.
2. **Phase 2 (Gate update).** A new gate G_2 is built from an expanded decoder vocabulary (e.g., ds-OCR v2 with a larger token set). G_2 includes encoding IDs that were latent under G_1. The gate update is the EWM analog of **belief revision**: the regulatory structure changes, and previously inadmissible evidence becomes admissible.

3. **Phase 3 (Immediate activation).** Latent encoding IDs materialize **immediately at their earned TF**. No cold start, no re-measurement, no gradient descent. The system instantly "knows" what it previously could not express. This is the structural analog of the phenomenon Lincoln describes: when a person revises a belief, the evidence that was always present but previously discounted suddenly becomes salient—not because the evidence changed, but because the regulatory framework now admits it.

The critical architectural property is the **TF-vs-rank separation** (STANDARD.md §3.1): **TF is stored pre-gate, rank is derived post-gate.** The gate controls what is *rankable*, not what is *storable*. This separation is the EWM implementation of Lincoln's distinction between the substrate-level reinforcement signal (which operates regardless of belief—TF always accumulates) and the cortical regulation of behavioral output (which depends on the current belief framework—rank is always gate-dependent).

Domain-Specialized World Views

Because the gate_TF HLLSet is content-addressed and immutable, multiple gates can coexist in the same system. Each gate defines a **domain-specialized world view**—a different "belief system" through which the same underlying lattice can be interpreted.

Gate	Vocabulary	World View	Use Case
G_{medical}	Medical terms only	Clinical interpretation	Hospital records
G_{legal}	Legal terminology	Juridical interpretation	Contract analysis
G_{literary}	General vocabulary	Narrative interpretation	Book digitization
G_{full}	All 128K BPE token IDs	Unrestricted	General purpose
G_{code}	Programming tokens	Code structure	Source analysis

All five gates operate on the **same LUT** and the **same HLLSet lattice**. The underlying measurement infrastructure (Type 0 and Type 1) is identical. Only the regulatory tier (Type 2, via the gate) differs. The system can switch between world views by changing which gate is applied at materialization time—without re-measuring, re-indexing, or re-training.

This is the EWM analog of a phenomenon Lincoln identifies as uniquely human: the capacity to evaluate the same evidence through different belief frameworks and reach different conclusions. A person can interpret a set of facts through a scientific, religious, or political lens; the ds-hllset-cortex can interpret the same document corpus through a medical, legal, or literary gate. In both cases, the substrate (neural prediction error / HLLSet lattice) is identical; the interpretation differs because the regulatory framework differs.

Cross-gate BSS provides a quantitative measure of how different two world views are:

$$\text{BSS}_\tau(G_A \to G_B) = \frac{|G_A \cap G_B|}{|G_B|}$$

measures what fraction of world view B is expressible in world view A. When $\text{BSS}_\tau(G_{\text{medical}} \to G_{\text{legal}})$ is low, the two domains share little vocabulary—the system cannot "translate" between them without expanding its gate. This is the structural analog of incommensurability between belief systems: two frameworks may be incapable of expressing each other's concepts not because the evidence differs, but because the regulatory gates are disjoint.

Self-Ingestion and the Emergence of Metacognition

The ds-hllset-cortex architecture enables a capability that maps onto the highest level of Lincoln's evolutionary progression: **self-ingestion**— the system ingesting its own materialized output as new measurements.

In the standard pipeline, encoding IDs flow in one direction:

$$\text{Image} \rightarrow \text{Encoder} \rightarrow \text{Encoding IDs} \rightarrow \text{hllset-cortex} \rightarrow \text{Decoder} \rightarrow \text{Text}.$$

In **self-ingestion mode**, the decoded text is re-encoded and fed back into the pipeline:

$$\text{Text}_{\text{out}} \rightarrow \text{Re-encode} \rightarrow \text{Encoding IDs}' \rightarrow \text{hllset-cortex} \rightarrow H_{\text{self}}.$$

The HLLSet H_{self} is a **structural fingerprint of the system's own output**—a measurement of what the system itself produced. The BSS between H_{self} and the original document HLLSet H_{doc} measures **self-consistency**:

$$\text{BSS}_\tau(H_{\text{self}} \rightarrow H_{\text{doc}}) = \frac{|H_{\text{self}} \cap H_{\text{doc}}|}{|H_{\text{doc}}|}.$$

When this value is high, the system's output is structurally faithful to the input—it is "telling the truth" about the document. When it is low, the system's output has diverged from the source—either through information loss (missing encoding IDs) or through hallucination (encoding IDs not present in the source). This is the EWM analog of **metacognitive monitoring**: the capacity, which Lincoln identifies as requiring the full cortical tier, to evaluate the quality of one's own cognitive output against evidence.

Self-ingestion with **gate variation** enables a more profound metacognitive operation: the system can evaluate how the *same* document is interpreted under *different* gates. Given document D producing H_{doc}, the system materializes through gate G_A, re-encodes the output, and compares:

$$\text{BSS}_\tau(H_{\text{self},G_A} \rightarrow H_{\text{self},G_B})$$

measures the structural divergence between world views A and B applied to the same content. This is the EWM analog of **perspective-taking**— evaluating how the same evidence would be interpreted under a different belief framework. Lincoln identifies this capacity as distinctively human, requiring the full integration of working memory, executive control, and the cortical brake. The ds-hllset-cortex implements it through the composition of its three-tier perceptron architecture: Type 0 measures, Type 1 forward-models (including the gate), and Type 2 evaluates and compares.

Structural Learning as Threshold Tuning

The self-ingestion loop of §10.6 reveals the system's output fidelity at a given moment, but does not by itself constitute learning. To learn, the system must **change which HLLSets compose its current state** in response to accumulated evidence. The IICA constraint—HLLSets are immutable once created, and BSS_τ, BSS_ρ between fixed HLLSets are computed values that cannot be modified—forces a specific kind of learning: **structural selection through threshold tuning.**

The Two Thresholds

Let $\mathcal{S}(t) = \{H_1, H_2, \ldots, H_k\}$ be the collection of HLLSets representing the system's current state at time t—the working set of page-level, chapter-level, and self-ingested HLLSets that feed into the materializer and decoder. Learning operates on $\mathcal{S}$ through two thresholds:

Definition 11 (Inclusion Threshold θ_τ). *An HLLSet H_i is admitted to the current state collection only if its forward inclusion with the source document exceeds the threshold: $H_i \in \mathcal{S}(t) \Leftrightarrow \text{BSS}_\tau(H_i \rightarrow H_{\text{doc}}) \geq \theta_\tau$. θ_τ filters out low-fidelity representations—measurements that lost too much source structure during encoding, gating, or materialization.*

Definition 12 (Divergence Threshold θ_ρ). *An HLLSet H_i is excluded from the current state collection if its reverse exclusion with the source document exceeds the threshold: $H_i \notin \mathcal{S}(t) \Leftrightarrow \text{BSS}_\rho(H_i \rightarrow H_{\text{doc}}) > \theta_\rho$. θ_ρ filters out divergent representations—measurements containing structure absent from the source, i.e., encoding noise, gate leakage, or materializer hallucination.*

The two thresholds together define a **validity region** in BSS-space. HLLSets within the region $(\tau \geq \theta_\tau) \wedge (\rho \leq \theta_\rho)$ compose the system's operative world model; HLLSets outside it are excluded from driving output but remain content-addressed and addressable.

Threshold Tuning as the Cortical Brake in Operation

The learning process adjusts θ_τ and θ_ρ over time. This is not parametric optimization (there is no loss function, no gradient, no weight update). It is **structural regulation**: the Noether controller monitors the cross-layer Fisher matrix $F_{bb'}$ across the temporal pyramid and detects when the current threshold settings produce drift, oscillation, or collapse.

Three regimes are distinguishable:

1. **Undersuppression** (θ_τ too low, θ_ρ too high). The state collection $\mathcal{S}$ admits low-fidelity and divergent HLLSets. The decoder output oscillates—successive materializations of the same document produce different token sequences. In Lincoln's terms, this is the cortical brake failing to engage: the substrate-level reinforcement signal (TF accumulation) drives behavior without adequate regulatory filtering.
2. **Oversuppression** (θ_τ too high, θ_ρ too low). $\mathcal{S}$ is nearly empty. The system has excluded almost all HLLSets from its working set. Output is sparse or null. In Lincoln's terms, this is pathological over-engagement of the habenular pathway— the system suppresses action even when evidence supports it, analogous to the anhedonia and motivational deficits associated with LHb hyperactivity in depression .
3. **Regulated operation** (θ_τ and θ_ρ at values that maintain $\mathcal{S}$ at a stable size with low output variance across temporal layers). The cross-layer Fisher matrix shows convergence: the same HLLSets persist in $\mathcal{S}$ across successive temporal windows, and the BSS between successive outputs is high and stable. This is the EWM analog of adaptive epistemic functioning under intact cortical regulation.

The Matching Law Revisited: Attention as Inclusion

The threshold mechanism gives operational meaning to the matching-law analysis of §9.4. The system's "attention allocation" across HLLSets is not a continuous weight vector but a **binary inclusion decision**: each H_i is either in $\mathcal{S}$ (it drives output) or out (it does not). The proportion of HLLSets from a given measurement strategy (e.g., a specific bootstrap family, a specific gate, a specific temporal window) that survive the thresholds is the EWM analog of the matching law's B_1/B_2 ratio.

The sensitivity parameter s maps onto the **steepness of the threshold boundary**: a sharp boundary (s high) means small differences in BSS produce large differences in inclusion probability; a soft boundary (s low) means the system is tolerant of variance. The bias parameter b maps onto the **relative positioning** of θ_τ and θ_ρ: a system biased toward inclusivity sets θ_τ low and θ_ρ high; a system biased toward conservatism does the opposite.

Exclusion Without Erasure

The critical structural consequence of learning-as-threshold-tuning is that **excluded HLLSets are not deleted**. They are immutable and content-addressed; their CIDs remain valid; their TF contributions to the LUT persist. The system "forgets" nothing—it only **stops attending.**

This has two implications:

1. **Reversible exclusion.** Lowering θ_τ (or raising θ_ρ) re-admits previously excluded HLLSets to $\mathcal{S}$ instantly, at their earned TF rank. No recomputation, no re-measurement, no cold start. The evidence was always stored; only the regulatory threshold changed. This is the structural realization of Lincoln's account of belief revision: when the cortical framework changes, evidence that was always present but previously

inadmissible becomes salient—not because the evidence changed, but because the threshold for admission changed.

2. **No catastrophic forgetting.** Because exclusion is reversible, the system cannot permanently lose knowledge through threshold drift. A pathological oscillation that excludes and re-admits HLLSets in rapid succession produces unstable output but does not corrupt the underlying lattice. The IICA property guarantees that the same H_i is the same H_i regardless of how many times it cycles in and out of $\mathcal{S}$.

This stands in marked contrast to neural network learning, where weight updates are destructive (the old weight is overwritten) and catastrophic forgetting is a persistent problem. In the EWM framework, learning is **attention, not erasure**—the structural analog of Lincoln's observation that the NAc-LHb reinforcement signal itself never changes; only the cortical regulation of which signals reach behavioral output changes.