Shunyaya Behavioral Mathematics (SBM)

Deterministic Structural Emergence & Operator Classification Framework

---

Version: Public Open Standard (v1.8)
Release Date: February 27, 2026
Status: Public Open Standard (Deterministic Structural Emergence Classification Layer)

Caution: Research, structural observability, and deterministic behavioral classification only. Not a predictive engine, optimization system, randomness tester, cryptographic validator, AI performance evaluator, or safety-critical monitoring layer.

Scope Notice

Shunyaya Behavioral Mathematics (SBM) defines a **deterministic structural classification framework** for discrete transition operators and symbolic transition streams.

SBM operates as a structural observability layer alongside arithmetic, algorithmic, cryptographic, stochastic, or AI-generated systems.

SBM:

• Does not modify operator definitions
• Does not alter numerical magnitude
• Does not redefine randomness
• Does not inject probabilistic inference
• Does not perform entropy estimation
• Does not optimize performance
• Does not simulate system physics
• Does not evaluate correctness of outputs

SBM performs only one function:

It measures structural emergence topology under a fixed finite horizon.

SBM converts any deterministic transition generator into a structural operator through finite signature extraction.

Conforming implementations shall publish:

• Declared operator definition
• Horizon parameter `H`

• Domain size `N`
• Alphabet growth function `alpha(N,H)`
• Emergence indices
• Closure-front metrics
• Structural growth exponent `gamma(H)`
• Gap statistics
• Deterministic replay artifacts (CSV outputs + SHA-256 manifest)
• Exact reproduction commands

Sufficient documentation must be provided to enable independent deterministic replay.

Replay determinism is the conformance standard.

---

0. Abstract

Shunyaya Behavioral Mathematics (SBM) is a deterministic structural classification framework for transition operators.

Given a deterministic operator that produces a transition stream, SBM extracts finite binary signatures of fixed length `H` and evaluates:

- Alphabet growth `alpha(N,H)`
- Emergence density `E_N`
- Structural growth exponent `gamma(H)`
- Closure-front position `last_emergence_n`
- Gap statistics between novelty events

Core structural quantities:

- `alpha(N,H)` = number of distinct signatures observed up to N
- `gamma(H,N) = log2(alpha_N) / H`
- `E_N = emergence_count / N`
- `R_alpha(H,N) = alpha_N / 2^H`

SBM introduces:

• Finite signature extraction discipline
• Deterministic operator fingerprinting
• Closure-front topology classification
• Cross-domain operator taxonomy
• Structural fracture detection (Phase C)
• Replay-verifiable evidence discipline

SBM does not measure randomness.
SBM does not compute Shannon entropy.
SBM does not estimate probability distributions.

Instead, SBM answers a structural question:

How does novelty fill the finite signature space under deterministic evolution?

This establishes a new deterministic axis for:

• Arithmetic operators
• Feedback generators
• Cryptographic transforms
• PRNG families
• AI transition streams
• Structural regime shifts

SBM preserves classical magnitude exactly.

SBM operates purely at the level of structural signature topology.

TABLE OF CONTENTS

---

1. Motivation

Classical mathematics primarily studies:

• Magnitude
• Equality
• Limits
• Algebraic structure
• Topological structure

These frameworks analyze objects as static entities or continuous forms.

However, when **deterministic transition operators** are repeatedly applied to mathematical objects, a distinct structural phenomenon appears:

Deterministic systems do not generate unbounded behavioral diversity.
They compress into finite structural vocabularies.

Under repeated iteration, objects exhibit:

- Stabilization
- Recurrence
- Behavioral clustering
- Finite structural alphabets
- Delayed novelty events
- Bounded emergence growth

This reveals a new mathematical dimension:

The study of **structural behavioral compression** under deterministic transition.

Shunyaya Behavioral Mathematics (SBM) formalizes this dimension.

SBM does not alter magnitude.
SBM does not redefine algebra.
SBM introduces a **structural observability layer** over deterministic evolution.

---

2. Core Definitions

2.1 Deterministic Transition Operator

Let `T` be a **deterministic operator** acting on a domain `X`.

```
T : X -> X
```

Repeated application generates a transition stream:

```
x_0 = x
x_1 = T(x_0)
x_2 = T(x_1)
...
x_k = T^k(x_0)
```

This defines a **deterministic orbit** under `T`.

No randomness is introduced.
The evolution is fully reproducible.

2.2 Behavioral Signature Extraction

Define a deterministic signature extraction function:

```
S(n, H)
```

Where:

• `n` is an index in the transition stream
• `H` is a **finite horizon**
• `S(n, H)` encodes the structural transition pattern over the next `H` steps

Formally, if a binary extraction rule `b(n)` is defined over transitions, then:

```
S(n, H) = (b(n), b(n+1), ..., b(n+H-1))
```

The behavioral signature is therefore:

```
behavior(n) = S(n, H)
```

This mapping is:

• Deterministic
• Horizon-bounded
• Finite-dimensional
• Replay-verifiable

SBM operates entirely through this **signature extraction discipline**.

2.3 Behavioral Alphabet

Define the **behavioral alphabet** up to scale `N`:

```
A(N, H) = { S(n, H) | 0 <= n < N }
```

Define the **alphabet growth function**:

```
alpha(N, H) = |A(N, H)|
```

If there exists a finite bound such that:

`alpha(N, H)` stabilizes or grows sub-maximally as `N` increases,

then the system exhibits **behavioral compression**.

Since:

```
alpha(N, H) <= 2^H
```

the signature space is finite by construction.

The empirical question becomes:

How quickly and in what topology does the operator fill the finite signature space?

This is the **core measurable quantity** in SBM.

---

3. Core Empirical Observation

Across arithmetic, feedback, cryptographic, and AI transition operators, a recurring structural pattern is observed:

1. Rapid novelty growth at small `N`
2. Deceleration of novelty emergence at moderate `N`
3. Occasional deep-scale novelty events
4. Long stabilization intervals
5. Bounded alphabet growth under fixed `H`

More precisely:

• `alpha(N, H)` increases quickly at early scales
• Growth slows as signature reuse increases
• Novelty gaps widen
• Long stable runs appear
• The signature space remains finite

This pattern defines:

Behavioral Compression Under Deterministic Evolution

SBM does not assume this behavior.
It measures it.

This empirical regularity holds across:

• Simple arithmetic maps
• Parity feedback systems
• Cryptographic hash feedback
• Pseudorandom generators
• Artificial transition streams

This universality suggests that deterministic operators possess **structural fingerprints** independent of magnitude or probabilistic interpretation.

SBM formalizes this fingerprint.

4. Foundational Axioms of SBM

SBM is governed by structural axioms that define the scope and admissibility of all claims.
These axioms are not philosophical statements.
They are operational constraints.

Axiom 1 — Deterministic Observability

All behavioral claims in SBM must arise from **deterministic operators**.

There shall be:

- No randomness
- No probabilistic inference
- No statistical smoothing
- No entropy estimation
- No Monte Carlo methods

Every result must be:

- Exactly reproducible
- Parameter-declared
- Replay-verifiable

If an operator is not deterministic, it is outside SBM scope.

Axiom 2 — Finite Behavioral Compression

Under structurally defined operators and fixed horizon `H`, the behavioral alphabet may exhibit bounded growth.

Formally:

There exists `N0` such that for all `N > N0`:

`|A(N + delta, H)| - |A(N, H)|` is small and rare.

Equivalently, the novelty rate decreases as `N` increases.

This does not imply strict convergence.
It implies compression under deterministic evolution.

The upper bound is structurally defined:

```
alpha(N, H) <= 2^H
```

Axiom 3 — Rare Deep Emergence

Late-scale novelty events may occur.

However, such emergence must satisfy:

• Sparse occurrence
• Deterministic origin
• Finite amplitude
• No explosive superlinear alphabet growth

Formally:

Deep emergence may increase `alpha(N, H)`, but shall not violate:

```
alpha(N, H) <= 2^H
```

This preserves the finite structural regime.

Axiom 4 — Structural Recurrence

Behavioral signatures may recur across the domain.

Let `x_i` be indices where a given signature occurs.
Define the gap sequence:

```
gap_i = x_(i+1) - x_i
```

The gap sequence is:

• Deterministic
• Measurable
• Reproducible
• Structurally classifiable

Gap statistics form part of the operator fingerprint vector.

Recurrence is not random repetition.
It is structured reuse within a constrained alphabet.

---

5. Behavioral Arithmetic

Classical arithmetic evaluates magnitude:

```
a + b = c
```

SBM extends arithmetic into the behavioral domain.

Let:

`S(n, H)` = behavioral signature of `n` under fixed horizon `H`.

Then arithmetic evolution induces signature evolution:

```
S(n + 1, H) = F(S(n, H))
```

Where `F` is a deterministic structural transition rule induced by the operator.

Thus arithmetic becomes:

• Signature propagation
• Structural state transition
• Alphabet evolution

SBM does not alter numeric magnitude.
It overlays a structural layer while preserving magnitude exactly.

Rather than treating signatures as algebraic objects, SBM interprets magnitude propagation as an induced structural walk:

Arithmetic is interpreted as a deterministic walk through a constrained behavioral alphabet under fixed horizon extraction.

The expression:

```
behavior(a) + behavior(b) -> behavior(c)
```

does not represent algebraic addition of signatures.

It represents deterministic structural propagation under the declared operator.

This perspective enables:

• Structural compression analysis
• Emergence topology classification
• Recurrence measurement
• Closure-front detection
• Fracture regime identification

SBM reframes arithmetic as finite structural dynamics under disciplined horizon extraction.

---

6. Behavioral Geography

SBM introduces the concept of **structural geography** in arithmetic and transition domains.

Structural geography refers to deterministic clustering patterns in signature space across the index domain.

These structures include:

• **Belts** — contiguous regions where signature classes dominate
• **Corridors** — transition pathways between structural regimes
• **Epochs** — extended intervals of signature stability
• **Fracture clusters** — localized high-density novelty regions
• **Oscillatory symmetry zones** — regions exhibiting structured alternation

These are not metaphors.

They are **deterministic classifications derived from measurable behavioral clustering** under fixed horizon `H`.

Structural geography is inferred from:

- Alphabet growth patterns
- Recurrence structure
- Gap distributions
- Fracture density
- Stabilization intervals

Geography is therefore a topological mapping of signature reuse across index space.

---

7. Stabilization Phenomenon

Under fixed horizon `H`, many deterministic operators exhibit extended stabilization behavior.

Stabilization refers to prolonged intervals where:

`alpha(N, H)` remains constant.

Define **stabilization depth**:

```
D = N_end - N_last_growth
```

Where:

- `N_last_growth` = index of most recent novelty event

- N_{end} = final observed index

Large D indicates deep structural closure relative to the observed domain.

Stabilization does not imply global convergence.

It indicates local exhaustion of novelty within the current scale.

Stabilization depth is a core structural indicator in SBM classification.

---

8. Behavioral Metrics

SBM defines a minimal set of deterministic, measurable quantities.
All metrics are derived directly from signature extraction and contain **no probabilistic components**.

8.1 Alphabet Size

```
alpha(N, H) = |A(N, H)|
```

Where:

```
A(N, H) = { S(n, H) | 0 <= n < N }
```

This measures the number of distinct signatures observed up to index N.

Alphabet size is the primary structural observable.

8.2 Emergence Index

Let n_k denote indices where new signatures first appear.
Each such n_k is an emergence index.

Formally:

If $S(n_k, H)$ has not appeared for any $m < n_k$, then n_k is a novelty event.

Emergence indices define the growth trajectory of $\alpha(N, H)$.

8.3 Recurrence Gaps

For a fixed signature occurring at indices `x_i`, define the recurrence gap sequence:

```
G_i = x_(i+1) - x_i
```

Gap sequences characterize:

• Recurrence density
• Structural periodicity
• Local compression strength

Gap statistics form part of the deterministic operator fingerprint.

8.4 Plateau Length

A plateau is an interval where no new signatures appear.

If `N_stable_start` and `N_stable_end` define such an interval, then:

```
P = N_stable_end - N_stable_start
```

Plateau length quantifies local stabilization strength.

8.5 Oscillatory Symmetry Check

Certain operators exhibit structural symmetry.

If a transition operator `T` satisfies:

`T(x + d)` mirrors `T(x - d)` under declared deterministic extraction rules,

then the signature sequence may exhibit oscillatory symmetry.

Symmetry detection is determined by deterministic comparison of extracted signatures under structured offsets.

No probabilistic symmetry tests are used.

Core SBM Structural Invariants

(all deterministic and replay-verifiable)

`alpha(N, H)`
Number of distinct signatures observed up to scale `N` under horizon `H`.

`gamma(H, N) = log2(alpha(N, H)) / H`
Structural Growth Exponent (horizon-normalized structural expansion rate).

`R_alpha(H, N) = alpha(N, H) / (2^H)`
Normalized Occupancy (fraction of signature capacity filled).

`last_emergence_n`
Closure-front position (index of final novelty event within tested domain).

`R_cf(H, N) = last_emergence_n / (2^H)`
Closure-Front Alignment Ratio (closure-front location relative to theoretical boundary).

`E_N = emergence_count / N`
Emergence Density (novelty rate within tested scale).

These invariants replace legacy compression indices and are used consistently throughout the SBM taxonomy, reporting standard, and cross-domain classification framework.

---

9. Behavioral Closure Principle

SBM defines **behavioral closure** within a tested domain when the following conditions hold under fixed horizon `H`:

- `alpha(N, H)` stabilizes or grows sub-maximally
- Emergence events become sparse
- No explosive growth of `alpha(N, H)` occurs
- Structural classification remains robust across increasing `N`

Formally, closure is observed if:

For sufficiently large `N` within the tested domain,

`alpha(N + delta, H) - alpha(N, H)` is small and infrequent.

Behavioral closure is an **empirical structural condition**, not a universal claim.

SBM does not assert global convergence.
SBM does not extrapolate beyond tested scale.

SBM claims only:

Deterministic stabilization within verified ranges.

Closure is therefore domain-bounded and replay-verifiable.

10. Extension Domains

SBM is domain-agnostic with respect to deterministic transition systems.

It may be applied to:

• Integer transition systems
• Arithmetic propagation chains
• Recurrence relations
• Deterministic dynamical systems
• Feedback generators
• Cryptographic transforms (structural analysis only)
• AI transition streams (deterministic extraction only)
• Structural invariant overlays

SBM requires only:

1. A deterministic transition rule
2. A defined signature extraction function
3. A fixed horizon `H`

No domain-specific probabilistic model is required.

---

11. Relationship to Classical Mathematics

SBM is a **conservative overlay framework**.

It does not modify:

• Arithmetic results
• Algebraic identities
• Numerical magnitude
• Classical theorems
• Analytical conclusions

SBM operates orthogonally to magnitude.

If magnitude preservation is expressed as:

```
phi((m, a, s)) = m
```

then:

- `m` represents classical magnitude
- `(a, s)` represent structural state components

SBM studies the structural components `(a, s)`
while preserving `m` exactly.

No algebraic statement is altered.
No theorem is redefined.

SBM adds a deterministic structural classification layer
without interfering with classical truth.

---

12. Distinction from Existing Behavioral Mathematics

The term “behavioral mathematics” is used in other domains to describe:

• Modeling of human decision processes
• Agent-based systems
• Probabilistic reasoning
• Uncertainty frameworks
• Game-theoretic dynamics

SBM is fundamentally different.

SBM:

• Models deterministic structural behavior of mathematical objects
• Requires no probability
• Requires no agents
• Requires no stochastic modeling
• Operates entirely on reproducible transition systems

In SBM, “behavior” means:

Structured transition patterns under deterministic evolution.

No psychological, economic, or cognitive interpretation is implied.

---

13. Research Discipline

SBM enforces strict structural research discipline.

All implementations must satisfy:

• Deterministic execution pipelines
• Fixed operator definitions
• No parameter mutation mid-run
• Explicit horizon declaration `H`

• Explicit domain limit `N`
• Full reproducibility
• SHA-256 verifiable output artifacts
• Explicit reproduction commands

No extrapolation beyond validated ranges is permitted.

All claims must be:

• Domain-bounded
• Replay-verifiable
• Parameter-declared

This discipline is non-negotiable.

---

14. Open Research Questions

SBM raises several structural questions.

These remain open and formally unresolved:

1. Does every deterministic operator admit a finite behavioral alphabet under fixed `H`?
2. What operator classes yield near-maximal or explosive signature growth?
3. Can behavioral alphabets predict structural stability properties?
4. Do universal bounds exist for specific operator families?
5. Can behavioral compression be formally proven beyond empirical measurement?

These questions define the forward research frontier of SBM.

SBM does not claim answers beyond demonstrated domains.

---

15. Terminology Clarification

SBM uses the term “behavior” in a strictly mathematical sense.

Behavior means:

Deterministic transition pattern under a fixed signature extraction rule.

SBM does not:

• Anthropomorphize numbers
• Attribute intent to operators
• Assign symbolic meaning
• Invoke metaphysical interpretations

All terminology is operational and measurable.

SBM studies:

• Signature topology
• Emergence structure
• Recurrence density
• Stabilization regimes

Nothing more.

---

16. Structural Expansion Architecture

SBM is architected as a layered structural framework.

The full standard is organized into:

Layer 1 — Foundational Formalism

- Signature extraction
- Alphabet growth
- Emergence indices
- Stabilization depth

Layer 2 — Operator Taxonomy

- Finite Attractor (FA)
- Structured Compression (SC)
- Closure-Front Aligned (CFA)
- Delayed Closure-Front (DCF)
- Additional subclasses (reserved)

Layer 3 — Cross-Domain Operator Analysis

- Arithmetic systems
- Feedback generators
- Cryptographic transforms
- Deterministic AI transition streams

Layer 4 — Structural Fracture Detection (Phase C)

- Stability plateau construction
- Rule-shift injection
- Fracture gate formalization
- Replay verification

Layer 5 — Conformance & Replay Standard

- Required artifact structure
- Profile vector specification
- Deterministic reproduction protocol

This architecture allows SBM to scale into a complete structural classification standard without altering its foundational axioms.

---

17. Empirical Behavioral Stabilization Example

Collatz Parity Operator (H = 16)

This section records a deterministic large-scale structural analysis of the Collatz parity operator under fixed horizon extraction.

All values are directly derived from replay-verifiable execution artifacts.

17.1 Operator Definition

Define the deterministic transition operator:

```
T(x) = x / 2        if x is even
T(x) = 3x + 1       if x is odd
```

Define the behavioral signature under fixed horizon `H = 16`:

```
B(x) = (parity(T^0(x)), parity(T^1(x)), ..., parity(T^15(x)))
```

Where:

```
parity(y) ∈ {0,1}
```

The maximum signature space size is:

```
2^16 = 65536
```

This is the absolute structural capacity under horizon `H = 16`.

17.2 Experimental Setup

Three deterministic runs were performed:

- Run A: `N = 20000`
- Run B: `N = 50000`
- Run C: `N = 100000`
- Horizon: `H = 16`

Execution characteristics:

- Deterministic engine
- Fixed operator definition
- Fixed horizon
- No parameter mutation
- Replay-verifiable artifact bundle

Independent executions produced byte-identical outputs.

Replay invariant:

```
B_A = B_B
```

17.3 Observed Results

Run A — N = 20000

- `alpha(20000) = 2568`
- `last_emergence = 16384`
- `E(20000) = 0.1284`
- `C(20000) = 0.7928`

Run B — N = 50000

- `alpha(50000) = 2583`
- `last_emergence = 32768`
- `E(50000) = 0.05166`
- `C(50000) = 0.72618`

Increase in distinct signatures:

```
2583 - 2568 = 15
```

Observed capacity usage:

```
2583 / 65536 ≈ 3.9%
```

Run C — N = 100000

- `alpha(100000) = 2584`
- `last_emergence = 65536`
- `E(100000) = 0.02584`
- `C(100000) = 0.68249`

Incremental growth:

```
From 50000 to 100000: 2584 - 2583 = 1
From 20000 to 100000: 2584 - 2568 = 16
```

17.4 Structural Observations

1. Strong Behavioral Compression

Alphabet size grows extremely slowly relative to theoretical capacity.

Observed usage at `N = 100000`:

```
2584 / 65536 ≈ 3.94%
```

Less than 4% of signature space is occupied.

---

2. Binary Scaling Alignment

New emergence boundaries occurred at:

```
16384 = 2^14
32768 = 2^15
65536 = 2^16
```

Emergence aligns exactly with binary power thresholds.

This alignment is deterministic and reproducible.

---

3. Declining Emergence Density

Emergence density decreases monotonically:

```
E(20000)  = 0.1284
E(50000)  = 0.05166
E(100000) = 0.02584
```

This is consistent with:

```
lim_{N→∞} E(N) = 0
```

within tested domain.

---

4. Sublinear Compression Trend

Compression metric declines with scale:

```
C(20000)  = 0.7928
C(50000)  = 0.72618
C(100000) = 0.68249
```

This indicates sublinear alphabet growth relative to domain expansion.

---

5. Sparse Deep Emergence

From `N = 50000` to `N = 100000`, only one new signature emerged.

Late-scale novelty is:

• Rare
• Deterministic
• Scale-aligned

No explosive behavior was observed.

17.5 Structural Interpretation

Under fixed horizon `H = 16`, the Collatz parity operator exhibits:

• Highly constrained signature growth
• Sublinear alphabet expansion
• Binary-aligned late emergence
• Increasing stabilization depth

Within the tested domain, the operator demonstrates **behavioral compression** consistent with SBM Axiom 2.

No claim is made beyond tested range.

17.6 Behavioral Closure Conjecture (Domain-Bounded)

Within the tested domain:

The Collatz parity operator demonstrates:

• Sublinear alphabet expansion
• Monotonic decline in emergence density
• Late emergence aligned to `2^H` boundary

Under horizon `H = 16`, the operator exhibits near-closure behavior at:

```
last_emergence = 65536 = 2^H
```

This alignment suggests structural exhaustion of novelty at the horizon boundary within tested scale.

This constitutes empirical support for a **domain-bounded Behavioral Closure Conjecture** under parity extraction.

No universality is claimed.

---

18. Cross-Operator Validation Example

18.1 SSNT Closure Signature (H = 10) at N = 100000

This section demonstrates that SBM behavioral compression is not limited to Collatz-style feedback dynamics.

A structurally distinct operator family — integer closure structure — also compresses deterministically into a finite alphabet under disciplined signature extraction.

18.2 Operator and Signature Definition

For each integer `n`, define:

Minimum structural closure divisor

```
d_min(n) = min{ d >= 2 | d divides n }, if such d exists with d <=
floor(sqrt(n))
```

If no such divisor exists, define:

```
d_min(n) = 0
```

(Prime-like under SSNT horizon definition.)

Band Classification

Define structural band:

- `band = P` if `d_min(n) = 0`
- otherwise `band ∈ {A, B, C, D, E}` according to fixed threshold cutoffs

Total band categories: 6.

Hardness Ratio

For composite integers:

```
h(n) = d_min(n) / sqrt(n)
```

Bucket hardness into `H = 10` bins:

```
bucket(h(n)) ∈ {0..9}
```

Behavioral Signature

```
B(n) = (band(n), bucket(h(n)))
```

Theoretical maximum alphabet size:

```
|Sigma| <= 6 * 10 = 60
```

This is a strictly bounded finite signature space.

18.3 Experimental Setup

Deterministic run:

- `N = 100000`
- `H = 10` buckets
- Fixed thresholds `(t1, t2, t3, t4)`

• Deterministic extraction engine
• Replay-verifiable manifest

No parameter mutation mid-run.
Independent executions are byte-identical.

18.4 Observed Results

- `alpha(100000) = 47`
- `E(100000) = 0.00047`
- `C(100000) = 0.33625`
- `last_emergence = 66641`

Theoretical maximum alphabet:

```
60
```

Observed occupancy:

```
47 / 60 ≈ 78.3%
```

The alphabet is near-complete under the defined signature.

Emergence events are:

• Extremely sparse
• Deterministic
• Widely separated

No explosive regime was observed.

18.5 Structural Interpretation

Two structurally different operator families now demonstrate compression:

1. **Collatz parity**
   - Feedback-style dynamic operator
   - Large theoretical space (`2^H`)
   - Extremely low occupancy
2. **SSNT closure operator**
   - Factor-closure structural operator
   - Strictly finite theoretical space (<= 60)
   - Near-complete finite stabilization

This cross-family result supports a domain-bounded structural principle:

Under disciplined deterministic signature definitions, many operators compress into finite or sublinearly expanding behavioral alphabets within tested scale.

No universality beyond tested domains is claimed.

18.6 Addendum — Horizon Stress Test (Collatz Parity, H = 20)

A horizon stress test was performed to evaluate how alphabet size scales with increased signature depth.

Run

- `N = 200000`
- `H = 20`

Observed

- `alpha(200000, 20) = 17619`
- `last_emergence = 196608`
- `E = 0.088095`

Theoretical capacity:

```
2^20 = 1048576
```

Normalized occupancy:

```
O = alpha / 2^H
O = 17619 / 1048576 ≈ 0.0168
```

Approximately 1.7% of possible signature space is occupied.

Binary Block Alignment

```
last_emergence = 196608 = 3 * 65536 = 3 * 2^16
```

Emergence aligns with structured binary block boundaries.

Novelty does not expand uniformly across full capacity.

18.7 Horizon-Aware Normalized Occupancy

Define:

```
O(N, H) = alpha(N, H) / 2^H
```

This provides:

- Cross-horizon comparability
- Resolution-independent compression measure
- Operator-agnostic structural density metric

`O(N, H)` becomes a key comparative invariant across deterministic systems.

---

19. Controlled Horizon vs Scale Analysis — Collatz Parity

Two controlled experiments were performed at fixed scale:

```
N = 200000
```

Case A — H = 16

- `alpha(200000,16) = 2584`
- `last_emergence = 65536 = 2^16`

Case B — H = 20

- `alpha(200000,20) = 17619`
- `last_emergence = 196608`

19.1 Normalized Occupancy

Define horizon-normalized occupancy:

```
O(N,H) = alpha(N,H) / 2^H
```

Compute:

- `O(200000,16) ≈ 0.0395`
- `O(200000,20) ≈ 0.0168`

19.2 Structural Interpretation

Increasing behavioral horizon:

• Increases absolute alphabet size `alpha`
• Decreases fractional occupancy `O(N,H)`

Thus, higher-resolution behavioral extraction reveals more structure,
but occupies a smaller fraction of the theoretical signature space.

Additionally, at fixed `H = 16`:

Alphabet size remained constant between:

```
N = 100000
N = 200000
```

This indicates complete stabilization once:

```
N >= 2^H
```

19.3 Empirical Properties Observed

From controlled scale–resolution separation:

1. **Scale Closure at Fixed Horizon**
   For fixed `H`, stabilization occurs once `N` exceeds a threshold near `2^H`.
2. **Horizon Compression Across Resolution**
   As `H` increases, normalized occupancy `O(N,H)` decreases.

These are empirical observations within tested domain.

No universal claim is made.

---

20. Empirical Horizon Scaling Law — Collatz Parity

Controlled experiments were performed at fixed:

```
N = 200000
```

for varying horizon `H`.

20.1 Observed Values

H	alpha(N,H)	last_emergence
12	377	4096 = 2^12
14	987	16384 = 2^14
16	2584	65536 = 2^16
18	6764	131072
20	17619	196608

For horizons satisfying:

```
2^H <= N
```

full stabilization occurred at:

```
last_emergence = 2^H
```

within tested scale.

20.2 Occupancy Behavior

Using:

```
O(N,H) = alpha(N,H) / 2^H
```

Observed occupancy:

- `H = 12` → ~0.092
- `H = 14` → ~0.060
- `H = 16` → ~0.039
- `H = 18` → ~0.0258
- `H = 20` → ~0.0168

Occupancy decreases monotonically as `H` increases.

20.3 Empirical Growth Pattern

Observed growth within the tested post-closure regime is consistent with a stable structural growth exponent.

Using the structural growth definition:

```
gamma(H) = log2(alpha(H)) / H
```

empirical measurements for Collatz parity yield:

```
gamma(H) ≈ 0.705
```

Equivalently, horizon scaling may be expressed as:

```
alpha(H) ≈ 2^(gamma(H) * H)
```

Within the tested range, this becomes:

```
alpha(H) ≈ 2^(0.705 H)
```

The full combinatorial signature space grows as:

```
2^H
```

Since:

```
2^(0.705 H) << 2^H
```

the Collatz parity alphabet grows sub-exponentially relative to total signature capacity.

This statement is domain-bounded to the tested horizon range and does not assert asymptotic universality.

20.4 Empirical Scaling Laws (Domain-Bounded)

Two empirical regularities are observed within tested range:

1. Closure Threshold Law

```
N_closure(H) ≈ 2^H
```

Stabilization occurs once scale exceeds signature capacity boundary.

---

2. Horizon Compression Law

```
alpha(N,H) / 2^H
```

decreases as `H` increases.

This indicates resolution-dependent compression.

20.5 Structural Interpretation

Across both dimensions:

• Scale (`N`)
• Resolution (`H`)

behavioral compression persists.

The system exhibits:

• Scale stabilization at fixed horizon
• Sub-capacity expansion across horizon
• Monotonic occupancy decay

These findings support deterministic behavioral compression across both scale and resolution dimensions within the tested domain.

No asymptotic claim is made beyond measured ranges.

---

21. Operator Class Contrast — Finite Attractor Example (Digitsum)

The `digitsum_mod9` operator was evaluated at:

```
N = 200000
H = 12, 14, 16
```

21.1 Observed

• `alpha = 9`
• `last_emergence = 10`
• Metrics identical across all tested horizons

No horizon sensitivity was observed.

Alphabet size remained constant under increasing `H`.

21.2 Structural Interpretation

This behavior defines a **Finite Attractor Operator**.

Definition — Finite Attractor Operator

An operator for which:

```
alpha(N,H) = constant
```

for sufficiently small `N`, and remains invariant under further increases in `H`.

Behavior collapses into a small fixed alphabet independent of resolution.

21.3 Contrast with Other Operators

Operator Type	Observed Behavior
Digitsum_mod9	Immediate finite attractor (alpha = 9)
SSNT Closure	Finite structured closure (near-complete bounded alphabet)
Collatz Parity	Expanding feedback with horizon compression

Digitsum demonstrates strict attractor behavior.

Collatz demonstrates horizon-sensitive expansion with compression.

SSNT demonstrates bounded structured completion.

21.4 Structural Operator Taxonomy

Empirical evidence within tested domains supports the following operator classes:

1. Finite Attractor

• Small fixed alphabet
• Independent of horizon `H`
• Immediate stabilization

Example: `digitsum_mod9`

---

2. Finite Structured Closure

• Strictly bounded theoretical alphabet
• Near-complete structural occupation
• Sparse late emergence

Example: `SSNT closure signature`

---

3. Expanding Feedback with Horizon Compression

• Alphabet grows with increasing `H`
• Growth sub-exponential relative to `2^H`
• Normalized occupancy `alpha(N,H) / 2^H` decreases with `H`

Example: `Collatz parity`

This taxonomy is:

• Empirically derived
• Deterministically reproducible
• Domain-bounded

No claim is made that these classes are exhaustive or universal.

---

22. Pre-Closure vs Post-Closure Regimes (Horizon Law Refinement)

Horizon scaling must be interpreted relative to scale–resolution boundary.

Define empirical closure threshold:

```
N_closure(H) ≈ 2^H
```

22.1 Post-Closure Regime

Condition:

```
N >= 2^H
```

In this regime:

• Behavioral emergence frontier can complete
• `last_emergence` may align with `2^H`

Empirical example (Collatz parity, H = 18):

Extending scale from:

```
N = 200000
to
N = 400000
```

Produced:

- `last_emergence = 262144 = 2^18`
- `alpha(400000,18) = 6765`

Closure completed once scale exceeded the horizon boundary.

22.2 Pre-Closure Regime

Condition:

```
N < 2^H
```

In this regime:

- Run terminates before potential closure boundary
- Novelty may still be occurring at end of tested range
- `last_emergence` may equal `N`

Empirical example at fixed `N = 200000`:

- `H = 22` → `last_emergence = 200000`
- `H = 24` → `last_emergence = 200000`

These runs are resolution-pre-closure.

22.3 Reporting Requirement

Horizon scaling curves must be interpreted with regime awareness.

Any civilization-grade SBM reporting must explicitly state:

- `N`
- `H`
- Whether run is pre-closure or post-closure
- Whether `2^H <= N` holds

Without regime classification, horizon scaling conclusions are incomplete.

22.4 Structural Significance

The regime distinction explains:

• Why occupancy decreases with increasing `H`
• Why stabilization appears at power-of-two boundaries
• Why some runs show apparent ongoing novelty

Scale and resolution must always be jointly interpreted.

---

23. Empirical Closure-Front Law — Collatz Parity

A decisive empirical pattern was observed for the Collatz parity signature of length `H`.

Define:

```
T(x) = x/2 if even, else 3x + 1
```

Signature:

```
B(x) = (parity(T^0(x)), ..., parity(T^(H-1)(x)))
```

Let:

```
last_emergence(H,N)
```

be the largest `n <= N` for which a new signature first appears.

23.1 Post-Closure Results (N >= 2^H)

Observed:

• `H = 18, N = 400000`
→ `last_emergence = 262144 = 2^18`

• `H = 19, N = 600000`
→ `last_emergence = 524288 = 2^19`

• `H = 20, N = 1200000`
→ `last_emergence = 1048576 = 2^20`

23.2 Empirical Closure-Front Law

Within tested regime:

```
If N >= 2^H
then last_emergence(H,N) = 2^H
```

This is the **empirical closure-front law** for Collatz parity under SBM.

No universality beyond tested range is claimed.

23.3 Post-Closure Horizon Scaling

Observed alphabets:

- `alpha(18) = 6765`
- `alpha(19) = 10946`
- `alpha(20) = 17711`

Define:

```
gamma_bits(H) = log2(alpha(H)) / H
```

Observed:

```
gamma_bits ≈ 0.705
```

Thus empirically:

```
alpha(H) ≈ 2^(0.705 H)
```

Since theoretical capacity is:

```
2^H
```

Normalized occupancy:

```
alpha(H) / 2^H ≈ 2^(-0.295 H)
```

This quantifies **horizon compression** under SBM for Collatz parity.

---

24. Operator Class Distinction via Closure-Front Alignment

Define:

```
F(H) = 2^H
```

Define Closure-Front Alignment Ratio:

```
R = alpha(H) / F(H)
```

Classify:

• If `R ≈ 1` and `last_emergence ≈ F(H)`
→ **Closure-Front Aligned (CFA)**

• If `R ≈ 1` but `last_emergence >> F(H)`
→ **Delayed Closure-Front (DCF)**

• If `R << 1` and `last_emergence = N`
→ **Dense Expansion / Pre-Closure**

This becomes a structural SBM operator taxonomy dimension.

---

25. PRNG Structural Subspace Detection — Xorshift Parity

A deterministic xorshift32 operator was evaluated under SBM parity-signature observation.

Update rule:

```
x = x ^ (x << 13)
x = x ^ (x >> 17)
x = x ^ (x << 5)
```

with 32-bit masking.

Signature:

```
B(n) = (parity(x0), ..., parity(x(H-1)))
```

Theoretical capacity:

```
|S(H)| = 2^H
```

25.1 Full-Space Saturation with Delayed Closure-Front

Empirical results:

H = 18, N = 1000000

- `alpha = 262144 = 2^18`
- `last_emergence = 458751`

H = 19, N = 3000000

- `alpha = 524288 = 2^19`
- `last_emergence = 1507327`

H = 20, N = 7000000

- `alpha = 1048576 = 2^20`
- `last_emergence = 3604479`

25.2 Structural Interpretation

1. Xorshift parity fills full theoretical signature space:

```
alpha(H) = 2^H
```

(empirically confirmed for H = 18, 19, 20)

2. Closure-front does **not** align with `2^H`.

Example for `H = 20`:

```
2^20 = 1048576
last_emergence = 3604479
```

3. Classification:

Full-Space Saturation with Delayed Closure-Front

25.3 Structural Contrast

Operator	alpha(H) behavior	closure-front
Collatz parity	Sub-exponential vs 2^H	aligned at 2^H
Xorshift parity	Full 2^H	delayed far beyond 2^H

SBM therefore distinguishes:

• **Closure-Front Aligned (CFA)** operators
• **Delayed Closure-Front (DCF)** operators

This distinction is:

• Deterministic
• Signature-based
• Independent of statistical randomness tests

It is purely structural.

---

26. Structural Growth Exponent and Closure-Front Formalization

26.1 Preliminaries

Let:

• `T : X -> X` be a deterministic operator
• `H >= 1` be a fixed behavioral horizon
• `S(x, H)` be a deterministic signature extraction function
• `B(x) = S(x, H)`

Define behavioral alphabet:

```
A(N, H) = { B(x) | x <= N }
```

Define alphabet size:

```
alpha(N, H) = |A(N, H)|
```

Define theoretical signature capacity:

```
F(H) = 2^H
```

All definitions are deterministic and replay-verifiable.

26.2 Structural Growth Exponent

Define the **Structural Growth Exponent**:

```
gamma(H) = log2(alpha_post(H)) / H
```

Where:

- `alpha_post(H)` is alphabet size in post-closure regime
- Post-closure means `N >= F(H)`

Interpretation

- `gamma(H) = 1` → full-space saturation
- `0 < gamma(H) < 1` → structured subspace compression
- `gamma(H) = 0` → finite attractor

`gamma` becomes a primary SBM structural invariant for post-closure behavior.

26.3 Closure-Front Alignment Ratio

Define:

```
R_cf(H, N) = last_emergence(H, N) / F(H)
```

Classification

If `N >= F(H)`:

- If `last_emergence = F(H)`
→ **Closure-Front Aligned (CFA)**

- If `alpha(H) = F(H)` but `last_emergence >> F(H)`
→ **Delayed Closure-Front (DCF)**

If `last_emergence = N`
→ **Pre-Closure Regime**

This formalizes the empirical closure law observed in Collatz parity.

26.4 Occupancy Ratio

Define structural occupancy:

```
R_alpha(H, N) = alpha(N, H) / F(H)
```

This measures subspace occupancy within signature capacity.

Observed Examples

Collatz parity

```
alpha(H) ≈ 2^(0.705 H)
gamma ≈ 0.705
R_alpha(H) ≈ 2^(-0.295 H)
```

Monotonic exponential decay in occupancy.

Xorshift parity (H = 18, 19, 20)

```
gamma ≈ 1
```

Full-space saturation.

Digitsum operator

```
gamma = 0
```

Finite attractor.

`gamma` and `R_alpha` together separate operator classes.

26.5 SBM Operator Fingerprint Vector

Define deterministic operator fingerprint:

```
Index(T) = (gamma, R_cf, R_alpha, stabilization_depth, mean_gap, var_gap)
```

Where:

- `gamma` — structural growth exponent
- `R_cf` — closure-front alignment ratio
- `R_alpha` — occupancy ratio
- `stabilization_depth` — scale closure measure
- `mean_gap`, `var_gap` — recurrence statistics

This fingerprint deterministically characterizes structural growth behavior under SBM extraction.

It is a formally defined structural descriptor.

27. Formal SBM Operator Taxonomy (Upgraded)

27.1 Class F — Finite Attractor

Definition

```
exists M such that alpha(N, H) <= M for all H, N
```

Properties

- `gamma = 0`
- Immediate stabilization
- Horizon invariant

Example: `digitsum_mod9`

27.2 Class SC — Structured Compression

Definition

```
alpha(H) ≈ 2^(kH), where 0 < k < 1
```

Properties

- Sub-exponential occupancy relative to `2^H`
- Post-closure alignment
- `gamma = k`

Example: Collatz parity (`gamma ≈ 0.705`)

27.3 Class CFA — Full-Space Closure-Front Aligned

Definition

```
alpha(H) = F(H)
last_emergence = F(H)
```

Novelty completes exactly at theoretical boundary.

27.4 Class DCF — Full-Space Delayed Closure-Front

Definition

```
alpha(H) = F(H)
last_emergence >> F(H)
```

Full-space saturation occurs, but closure-front is delayed deep into domain.

Example: Xorshift parity.

27.5 Class FI — Fracture-Inducing (Unverified)

Definition

```
alpha(N) grows explosively with N
No stabilization plateau observed
```

Not yet observed among tested deterministic operators.

This class remains hypothetical pending empirical confirmation.

---

28. Post-Closure Reporting Standard

Every civilization-grade SBM report must include:

1. Operator definition
2. Horizon `H`
3. Domain scale `N`
4. `alpha(N,H)`
5. `gamma(H)`
6. `R_alpha(H,N)`
7. `R_cf(H,N)`
8. Closure regime (pre or post)
9. Deterministic manifest hash

All structural claims must be accompanied by invariant values.

No narrative without invariants.

This defines:

SBM Reporting Standard 1.0

28.1 Phase A Status

The following structural components are now formally defined and integrated into the SBM standard:

- Structural Growth Exponent `gamma(H)`
- Closure-Front Alignment Ratio `R_cf(H, N)`
- Occupancy Ratio `R_alpha(H, N)`
- Emergence Density `E_N`
- Deterministic Operator Fingerprint `Index(T)`
- Formal SBM Operator Taxonomy Framework

All invariants are:

- Deterministic
- Parameter-declared
- Replay-verifiable
- Manifest-bound

Phase A (**Arithmetic Domain Structural Formalization**) is complete.

This phase establishes the core structural observability layer over deterministic arithmetic operators without altering classical magnitude.

No extrapolation beyond tested domains is claimed.

28.2 Phase B Objective

Cross-Domain Structural Universality Testing

Objective:

Introduce a non-arithmetic deterministic feedback operator into `sbm_test.py`.

Candidate:

SHA1 Prefix Feedback Iteration

Define:

```
x0 = n
x_{k+1} = SHA1(x_k) mod 2^32
```

Signature:

```
B(n,H) = (parity(x0), ..., parity(x(H-1)))
```

Test whether SBM classifies SHA1 feedback distinctly from arithmetic and PRNG operators.

29. Cryptographic Feedback (SHA1) Structural Classification

29.1 Test Declaration

Operator: `sha1_parity`

Definition:

```
x0 = n mod 2^32
For k = 0..H-1:
    b_k = parity(x_k)
    x_{k+1} = int32(SHA1(bytes4_be(x_k))[0:4])
```

Signature:

```
B(n,H) = (b_0, b_1, ..., b_{H-1})
```

Fixed horizon:

```
H = 18
```

Theoretical capacity:

```
F(H) = 2^H = 262144
```

Runs Executed

Run A: `N = 400000`
Run B: `N = 1200000`
Run C: `N = 3000000`

29.2 Observed Profile Metrics

Run A — N = 400000

- `alpha_N = 205043`
- `last_emergence_n = 399999`

Run B — N = 1200000

- `alpha_N = 259449`
- `last_emergence_n = 1199980`

Run C — N = 3000000

- `alpha_N = 262141`
- `last_emergence_n = 2981531`

29.3 Derived Structural Invariants

For each run:

```
R_alpha(H,N) = alpha_N / 2^H
R_cf(H,N) = last_emergence_n / 2^H
gamma(H,N) = log2(alpha_N) / H
```

Run A (N = 400000)

```
R_alpha = 205043 / 262144 ≈ 0.78218
R_cf = 399999 / 262144 ≈ 1.52588
gamma ≈ 0.98031
```

Run B (N = 1200000)

```
R_alpha ≈ 0.98972
R_cf ≈ 4.57756
gamma ≈ 0.99897
```

Run C (N = 3000000)

```
R_alpha ≈ 0.9999886
R_cf ≈ 11.37497
gamma ≈ 0.9999991
```

29.4 Structural Interpretation

Observed pattern:

1. As `N` increases:

```
alpha_N -> 2^H
gamma(H) -> 1
```

2. Closure-front remains significantly beyond theoretical boundary:

```
last_emergence_n >> 2^H
R_cf >> 1
```

3. Novelty continues deep into domain even after near-saturation.

This indicates:

• Full-space saturation
• Delayed closure-front

29.5 SBM Class Assignment

Under SBM taxonomy:

This operator satisfies:

```
alpha(H) ≈ F(H)
last_emergence >> F(H)
```

Classification:

Delayed Closure-Front (DCF)

SHA1 parity feedback is structurally aligned with DCF class, similar to xorshift parity.

29.6 Structural Significance

SBM has classified a cryptographic feedback operator using only:

• Deterministic signature extraction
• Alphabet growth `alpha(N,H)`
• Closure-front `last_emergence`
• Scale–resolution invariants

No statistical randomness tests were used.

This establishes a deterministic structural axis for:

• PRNG evaluation
• Cryptographic feedback analysis
• Feedback-system classification

29.7 Phase B Status

Phase B objectives achieved:

- Cryptographic operator integrated
- Cross-scale deterministic evaluation performed
- Full-space saturation confirmed (`gamma -> 1`)
- Delayed closure-front behavior confirmed (`R_cf >> 1`)
- Deterministic class assignment: **DCF**

This extends SBM classification beyond arithmetic systems into cryptographic feedback systems within tested domain.

No claim is made beyond tested ranges.

30. SBM Operator Taxonomy Table

30.1 Operator Classes (Empirically Verified)

Finite Attractor (FA)

Definition

There exists `M` such that:

```
alpha(N, H) <= M
```

for all tested `N` and `H`.

Structural Invariants

- `gamma(H) = 0`
- `R_alpha(H, N)` rapidly stabilizes at a small constant
- Horizon invariant behavior
- Immediate or near-immediate stabilization

Example

```
digitsum_mod9
```

Structured Compression (SC)

Definition

Alphabet growth follows:

```
alpha(H) ≈ 2^(kH)
```

for some constant `0 < k < 1` within the tested post-closure regime.

Structural Invariants

- `0 < gamma(H) < 1`
- Sub-exponential occupancy relative to `2^H`
- `R_alpha(H, N)` decreases with increasing `H`
- Post-closure alignment typically observed

Example

`collatz_parity` (`gamma ≈ 0.705`)

Full-Space Closure-Front Aligned (CFA)

Definition

```
alpha(H) = 2^H
last_emergence_n = 2^H
```

Novelty completes exactly at the theoretical boundary `F(H) = 2^H`.

Structural Invariants

- `gamma(H) = 1`
- `R_alpha(H, N) ≈ 1`
- `R_cf(H, N) ≈ 1` in post-closure regime

Example

None observed in tested operator set so far.

Full-Space Delayed Closure-Front (DCF)

Definition

```
alpha(H) = 2^H
last_emergence_n >> 2^H
```

Full-space saturation occurs, but closure-front is delayed deep into the tested domain.

Structural Invariants

- `gamma(H) = 1`
- `R_alpha(H, N) ≈ 1`
- `R_cf(H, N) >> 1`

Examples

```
xorshift_parity
sha1_parity
```

30.2 Operator Fingerprint Vector (Standardized)

Define deterministic operator fingerprint:

```
Index(T) = (gamma, R_alpha, R_cf, emergence_count, mean_gap, var_gap)
```

This vector is a required artifact for civilization-grade SBM reporting.

All values must be derived deterministically and accompanied by manifest hash.

---

31. AI Transition Sequence Monitoring

31.1 Core Idea

SBM need not be restricted to arithmetic operators.

Instead, it can process deterministic transition sequences extracted from AI systems.

Examples of extractable sequences:

- Token ID transitions
- Attention-head activation patterns
- Gradient sign transitions
- Weight update direction sequences
- Layer output threshold crossings

SBM does not evaluate accuracy or performance.

SBM evaluates **structural behavioral evolution** of transition sequences.

31.2 Deterministic Transition Signature Definition

Let a system produce a sequence:

```
X = (x_0, x_1, x_2, ..., x_T)
```

Define deterministic transition:

```
delta_t = f(x_t, x_{t+1})
```

Define binary structural signal:

```
b_t = parity(delta_t)
```

Define sliding signature:

```
B(t,H) = (b_t, b_{t+1}, ..., b_{t+H-1})
```

SBM then measures:

```
alpha(N,H)
```

over sliding window transitions.

This converts any deterministic transition stream into an SBM behavioral operator.

31.3 Structural Collapse Detection Law

Define novelty increment:

```
Delta_alpha(N) = alpha(N) - alpha(N-1)
```

Define stabilization depth:

```
D = N - last_emergence_n
```

A structural fracture event is flagged when:

1. `D` is large (stable plateau),
2. `gamma(H)` is stable across scale, and
3. A sudden spike occurs in `Delta_alpha(N)`.

This defines deterministic structural anomaly detection.

No probability.
No confidence intervals.
No statistical modeling.

Pure novelty emergence spike under fixed extraction.

31.4 Prototype Implementation Path

Full transformer introspection is not required initially.

Step 1 — Deterministic Stream Simulation

Define generator:

```
x_t = (a * t + b) mod M
```

Extract:

```
delta_t = x_{t+1} - x_t
```

Apply parity signature extraction.

Feed into SBM pipeline.

Step 2 — Controlled Structural Shift

At time `T_shift`, change rule:

```
x_t = (c * t + d) mod M
```

Observe whether SBM detects:

- New alphabet growth
- Closure reset
- Novelty spike
- Change in `gamma`

If these are detected deterministically, structural fracture detection is validated.

---

32. AI Structural Monitoring Formalization

32.1 AI Behavioral Growth Exponent

For a deterministic transition stream extracted from an AI system, define:

```
gamma_AI(H) = log2(alpha_AI(H)) / H
```

Where:

- `alpha_AI(H)` is post-closure alphabet size under AI transition extraction
- `H` is fixed signature horizon

Interpretation

• Increase in `gamma_AI` → structural expansion of behavioral diversity
• Sharp decrease in `gamma_AI` → structural compression or collapse
• Stable `gamma_AI` → steady behavioral regime

This is a structural drift indicator independent of performance metrics.

32.2 Drift Regime Classification

Under deterministic monitoring, transition streams can be classified into regimes.

Stable Regime

• `last_emergence_n < N`
• Small `Delta_alpha(N)`
• Increasing stabilization depth

Exploration Regime

• Frequent novelty
• Large `E(N)`
• Sustained alphabet growth

Fracture Regime

• Long stabilization plateau
• Sudden novelty burst
• Large `Delta_alpha` spike

These regimes are defined purely through deterministic signature evolution.

32.3 Structural Monitoring Principle

Modern AI monitoring commonly evaluates:

- Loss curves
- Validation accuracy
- Statistical anomaly detection
- Gradient norms

SBM evaluates:

- Behavioral emergence density
- Alphabet growth topology
- Closure-front movement
- Structural novelty spikes

SBM monitoring is orthogonal to performance evaluation.

It measures structural transition geometry.

No probabilistic modeling is required.

---

33. Phase C — Executable Proof (SBM AI Monitor)

33.1 Deterministic Evidence Standard

All Phase C results are valid only if independent replays produce byte-identical artifacts:

```
B_A = B_B
```

If replay differs, structural claim is invalid.

33.2 Transition Stream Construction (Deterministic)

Base transition rule:

```
x_{t+1} = (a * x_t + c) mod M
```

Inject deterministic rule-shift at index `shift_n`:

```
For t < shift_n:  use (a1, c1)
For t >= shift_n: use (a2, c2)
```

This creates a controlled structural regime transition.

33.3 Signature Extraction

Define transition difference:

```
d_t = (x_{t+1} - x_t) mod M
```

Define structural bit:

```
b_t = parity(d_t)
```

Define sliding window signature:

```
B(t,H) = (b_t, b_{t+1}, ..., b_{t+H-1})
```

Define alphabet:

```
A(N,H) = { B(t,H) | 0 <= t < N }
alpha(N,H) = |A(N,H)|
```

All computations are deterministic.

33.4 Output Bundle Specification

Each Phase C run produces:

- `sbm_ai_results.csv`
- `sbm_ai_alphabet.csv`
- `sbm_ai_metrics.csv`
- `sbm_ai_profile.json`
- `sbm_ai_manifest.sha256`

These files constitute the deterministic evidence artifact.

33.5 Fracture Candidate Indicators (Deterministic Artifacts)

The following indicators are recorded:

- `alpha_before_shift`
- `alpha_at_shift`
- `alpha_after`
- `max_stable_run`
- `max_spike`
- `spike_at_n`

- `fracture_candidate_count`
- `fracture_first_at_n`

Fracture detection is based solely on these invariants.

No statistical thresholds are used.

---

34. Phase C Test 2 — Plateau to Burst Fracture Results

34.1 Run Declaration

This section records Phase C Test 2 outputs and deterministic classification.

All values are inserted directly from `sbm_ai_profile.json`.

Operator: Deterministic transition stream with rule-shift injection
Pre-shift mode: plateau
Post-shift mode: lcg
Observation mapping: `obs = popcnt_parity`

Declared parameters:

```
N = 400000
H = 18
M = 2^32
shift_n = 200000
L = long_stable_L = 2000
```

34.2 Reported Deterministic Fields

```
alpha_N = 140035
emergence_count = 140035
last_emergence_n = 399998

alpha_before_shift = 18
alpha_at_shift = 19
alpha_after = 140035

max_stable_run = 199982
fracture_candidate_count = 1
fracture_first_at_n = 199983

mean_gap = 2.856434865818
var_gap = 285592.2526798245
```

All values are replay-verifiable artifacts.

34.3 Fracture Classification Gate

A structural fracture event is detected if:

1. `max_stable_run >= L`
2. `fracture_candidate_count >= 1`
3. `fracture_first_at_n` occurs at or structurally adjacent to `shift_n` under horizon-window measurement.

All three conditions must hold deterministically.

34.4 Outcome — Plateau to Burst Transition

Observed:

```
max_stable_run = 199982
L = 2000
fracture_candidate_count = 1
fracture_first_at_n = 199983
```

Since:

```
199982 >> 2000
```

a long stabilization plateau is confirmed prior to rule shift.

Immediately following the plateau, a novelty burst occurs.

Horizon-Window Alignment Note

With horizon:

```
H = 18
```

each signature window spans 18 consecutive transitions.

Because the generator shifts at:

```
shift_n = 200000
```

the earliest signature window containing post-shift transitions may begin up to `H-1` positions before `shift_n`.

Thus:

```
fracture_first_at_n = 199983
```

is structurally adjacent to `shift_n = 200000` under windowed measurement.

This alignment is consistent with deterministic sliding-window extraction.

34.5 Alphabet Expansion Due to Rule Shift

```
alpha_after - alpha_before_shift
= 140035 - 18
= 140017
```

This confirms large-scale structural novelty following rule injection.

34.6 Deterministic Classification

This run satisfies the Phase C fracture gate.

Classification:

Deterministic Fracture Regime

34.7 Replay Verification

Independent replay under identical parameters produced an identical profile vector:
```
Index_AI = (alpha_N, last_emergence_n, alpha_before_shift, alpha_after,
max_stable_run, fracture_candidate_count, fracture_first_at_n, mean_gap,
var_gap)
```

Replay verification performed at manifest level:
`sbm_ai_manifest.sha256` is identical between primary and replay runs.

Replay invariant satisfied:
`B_A = B_B`

Phase C evidence is replay-stable and deterministic.

34.8 Operator Identity Record (OIR) and Registry Discipline

To elevate Phase C from executable proof to civilization-grade review standard, each fracture run must produce an **Operator Identity Record (OIR)**.

This formalizes the packaging layer required under **SBM Reporting Standard 1.0**.

34.8.1 Operator Identity Record (OIR)

The OIR is a human-readable identity declaration that binds:

- Operator name
- Deterministic parameters
- Horizon `H`
- Domain `N`
- `shift_n`
- Pre-mode and post-mode
- Observation mapping
- Required artifact bundle
- Determinism claim
- Verification method
- Profile summary invariants

The OIR does **not** replace artifact files.
It is a structural identity anchor for external review.

An OIR must:

1. Match the exact parameters used in `RUN.txt`.
2. Declare required artifacts explicitly.
3. Declare replay determinism requirement.
4. Include profile summary values copied directly from `sbm_ai_profile.json`.
5. Include replay equivalence statement for replay runs.

OIR files must be stored alongside artifact bundles.

34.8.2 Replay OIR Requirement

For any Phase C run claiming deterministic fracture classification:

A replay execution must be performed.

The replay folder must contain:

- Full artifact bundle
- Replay OIR
- `RUN.txt`
- Identical `sbm_ai_manifest.sha256`

Replay determinism is satisfied if:

```
manifest_primary == manifest_replay
```

(byte-identical comparison)

Failure of manifest equality invalidates structural claim.

34.8.3 Operator Registry (Canonical Evidence Set)

All validated operators shall be recorded in:

```
outputs\OPERATOR_REGISTRY.txt
```

The registry must declare:

- Operator name
- Primary artifact folder
- Replay folder (if applicable)
- Structural claim type

The registry is not narrative.

It is an evidence index.

34.8.4 SBM Certificate (Reviewer Snapshot)

Each validated operator must include a one-page **SBM Certificate** containing:

- Operator declaration
- Parameter set
- Artifact list
- Determinism statement
- Structural invariants
- Classification

The certificate is the reviewer's entry point.

It does not replace raw artifacts.

34.8.5 RUN.txt Reproduction Anchor

Each artifact folder must contain a `RUN.txt` file containing the exact execution command used to generate the bundle.

This command must reproduce the artifact set deterministically when executed in the declared environment.

This is the authoritative reproduction anchor.

No reproduction command — no conformance.

34.9 Conformance Verification Ritual (Verifier Discipline)

To enforce structural integrity across operator bundles, SBM defines a deterministic conformance verification ritual.

This ritual ensures:

- Artifact completeness
- Manifest presence
- Replay equivalence
- Structural reproducibility

34.9.1 Required Artifact Set (Phase C)

Each Phase C operator folder must contain:

- `sbm_ai_results.csv`
- `sbm_ai_alphabet.csv`
- `sbm_ai_metrics.csv`
- `sbm_ai_profile.json`
- `sbm_ai_manifest.sha256`
- `RUN.txt`
- OIR file
- SBM Certificate

Missing artifacts invalidate conformance.

34.9.2 Manifest Integrity Standard

The file `sbm_ai_manifest.sha256` must:

- Contain SHA-256 hashes of all artifact files.
- Be generated deterministically.
- Match exactly across replay runs.

Replay verification condition:

```
hash_primary == hash_replay
```

Byte-level equality required.

No tolerance.

34.9.2A Manifest Self-Hash Rule

The file `sbm_ai_manifest.sha256` is required and must be byte-identical across replay runs.

The manifest must list hashes for all Phase C artifacts:

```
sbm_ai_results.csv
sbm_ai_alphabet.csv
sbm_ai_metrics.csv
sbm_ai_profile.json
```

The manifest may optionally include a hash entry for itself (`sbm_ai_manifest.sha256`). Self-hash is not required.

Replay determinism is evaluated using byte-identical manifest comparison and artifact hash validation.

34.9.3 Deterministic Replay Gate

A Phase C structural fracture classification is valid **only if**:

1. Fracture gate conditions hold.
2. Profile vector values are identical across replay.
3. Manifest files are byte-identical.
4. Artifact sets are complete.
5. `RUN.txt` reproduces the bundle deterministically.

If any condition fails:

Structural claim is invalid.

34.9.4 Evidence Hierarchy

SBM evidence hierarchy is:

Level 1 — Raw Artifact Files
Level 2 — Manifest Hash
Level 3 — Profile Vector Invariants
Level 4 — OIR Identity Declaration
Level 5 — Certificate Summary
Level 6 — Registry Entry

All six levels must exist for civilization-grade conformance.

34.9.5 Structural Integrity Principle

SBM does not rely on narrative validation.

SBM relies on:

- Deterministic artifacts
- Hash equality
- Parameter declaration
- Exact reproduction

All structural claims must be reducible to:

```
replay(primary_run) == replay(secondary_run)
```

No statistical confidence intervals.
No probabilistic tolerance.
No approximation.

Deterministic identity is the conformance standard.

34.9.6 Registry Segmentation Rule

Operator registries may be segmented by phase to prevent schema collisions between bundle formats.

If a phase-specific registry exists, it must be preferred over the global registry.

Example precedence:

`outputs\OPERATOR_REGISTRY_PHASEC.txt`
then fallback to
`outputs\OPERATOR_REGISTRY.txt`

Phase C conformance verification must use the Phase C registry when present.

This prevents non-Phase C evidence folders (Phase A/B operators) from being evaluated under Phase C artifact requirements.

34.9.7 Tamper Detection Ritual (Integrity Demonstration)

SBM conformance verification must detect any unauthorized artifact modification.

Tamper ritual:

1. Backup a Phase C artifact file.
2. Modify exactly one digit (one-byte change) in a Phase C artifact file.
3. Run verifier and confirm FAIL due to SHA-256 mismatch between the manifest hash and recomputed hash.
4. Restore the original artifact.
5. Run verifier and confirm PASS.

This demonstrates that SBM evidence integrity is enforced by deterministic hash verification.

34.9.8 Single-Command Conformance Invocation

SBM provides a single-command conformance invocation to support reviewer execution.

Canonical command file:

```
verify\RUN_ALL_VERIFY.cmd
```

This command:

1. Executes the Phase C verifier using the preferred registry.
2. Writes the latest report to `verify\last_verify_report.txt`.
3. Prints `OVERALL_STATUS: PASS` or `OVERALL_STATUS: FAIL`.
4. Prints `EXIT_CODE: 0` on PASS and `EXIT_CODE: 1` on FAIL.
5. Returns exit code `0` on PASS and `1` on FAIL.

This enables deterministic conformance validation with one command.

34.9.9 Phase C Release Capsule (Reviewer Package)

SBM Phase C evidence may be distributed as a minimal release capsule to enable independent verification.

Canonical release folder:

```
release_phasec\
```

Required contents:

```
release_phasec\outputs\ai_fracture\
release_phasec\outputs\ai_fracture_replay\
release_phasec\outputs\OPERATOR_REGISTRY_PHASEC.txt
release_phasec\verify\verify_sbm.py
release_phasec\verify\RUN_ALL_VERIFY.cmd
```

Verification procedure:

From inside `release_phasec`, run:

```
verify\RUN_ALL_VERIFY.cmd
```

Expected output:

```
OVERALL_STATUS: PASS
EXIT_CODE: 0
```

This capsule provides deterministic reproducibility, manifest integrity validation, replay equivalence, and audit-grade proof packaging.

---

35. Multi-Phase Deterministic Replay Demonstrations

Phase C establishes executable fracture proof.

To elevate SBM from phase-specific validation to a civilization-grade multi-phase deterministic standard, replay sealing must be demonstrated for additional operator classes.

Replay sealing enforces:

```
B_A = B_B
```

Byte-identical artifact bundles across independent executions.

35.1 Phase A Replay Proof — digitsum_mod9

Under parameters:

```
H = 12
N = 20000
```

Two independent executions were performed.

Primary bundle:

OUT_SBM_DIGITS_A_PRIMARY

Replay bundle:

OUT_SBM_DIGITS_A_REPLAY

Replay invariant enforced:

```
B_A = B_B
```

The file `sbm_manifest.sha256` is byte-identical across both runs (verified via `fc`).

Conclusion:

Phase A deterministic replay identity is proven.

This confirms that Phase A operators satisfy the same replay discipline as Phase C fracture operators.

35.2 Phase B Replay Proof — xorshift_parity

Under parameters:

```
H = 19
N = 3000000
```

Two independent executions were performed.

Primary bundle:
`OUT_SBM_XORSHIFT_B_PRIMARY`

Replay bundle:
`OUT_SBM_XORSHIFT_B_REPLAY`

Replay invariant enforced:

```
B_A = B_B
```

The file `sbm_manifest.sha256` is byte-identical across both runs (verified via `fc`).

Profile Summary (from `sbm_profile.json`)

```
alpha_N = 524288
emergence_count = 524288
last_emergence_n = 1507327
```

Derived Structural Invariants:

```
gamma(H, N) = log2(alpha_N) / H ≈ 1.000

R_alpha(H, N) = alpha_N / (2^H) ≈ 1.000

R_cf(H, N) = last_emergence_n / (2^H) ≈ 2.88

E_N = emergence_count / N ≈ 0.174762666667
```

Structural Interpretation

• Full-space saturation observed: `alpha_N = 2^19`
• Structural growth exponent `gamma ≈ 1` confirms complete signature capacity occupation
• Closure-front significantly exceeds theoretical boundary (`R_cf >> 1`)
• Classification: **Full-Space Delayed Closure-Front (DCF)**

Conclusion

Phase B deterministic replay identity is proven.

Structural invariants are identical across independent executions.

Deterministic conformance condition satisfied:

```
B_A = B_B
```

---

36. Unified Multi-Phase Verification Orchestrator

To support civilization-grade independent review, SBM provides a single-command multi-phase verification orchestrator that validates Phase A, Phase B, and Phase C evidence under strict deterministic conformance discipline.

Canonical command file:

```
verify\RUN_VERIFY_ALL_PHASES.cmd
```

Purpose:

The orchestrator enforces deterministic replay validation across all validated operator classes.

Execution Model:

The orchestrator performs the following checks in sequence:

1. Phase A — Replay identity verification
   Manifest equality check between primary and replay bundles.
2. Phase B — Replay identity verification
   Manifest equality check between primary and replay bundles.
3. Phase C — Capsule conformance verification
   Invocation of the Phase C verifier enforcing:
   - Artifact completeness
   - SHA-256 integrity validation
   - Replay manifest byte identity
   - Deterministic fracture classification gate

Success Criteria:

The following output must be produced:

PHASE_A_STATUS: PASS
PHASE_B_STATUS: PASS
PHASE_C_STATUS: PASS
OVERALL_STATUS: PASS
EXIT_CODE: 0

Failure Criteria:

If any phase returns FAIL, the orchestrator must return:

OVERALL_STATUS: FAIL
EXIT_CODE: 1

No tolerance or probabilistic interpretation is permitted.

Deterministic Conformance Standard:

All structural claims are valid only if independent replay executions satisfy:

```
B_A = B_B
```

where replay artifacts are byte-identical at the manifest level.

Result:

SBM operates as a one-command, multi-phase, deterministic conformance system suitable for audit-grade independent verification.

---

37. Core Structural Freeze Certification

SBM Deterministic Core Freeze Snapshot

The deterministic core of SBM has been frozen under structural hash discipline.

The freeze boundary includes:

- **scripts**
- **verify**
- **reference_outputs**
- **release_phasec**

Top-level `outputs/` and the presentation layer (docs, README, FAQ, diagrams, explanatory materials) are intentionally excluded from the structural freeze boundary and may evolve independently without affecting deterministic conformance.

Core Structural Hash

STRUCTURAL_FILE_LIST.txt SHA256

```
0fb12f1921a78fe9c33d5d178c0f44548c7f892f462c8267c44f3eaaf19bc8f6
```

This hash defines the **certified deterministic boundary** of the SBM core.

Any modification inside the frozen boundary changes this hash and therefore invalidates the freeze certification.

Multi-Phase Deterministic Verification Result

```
PHASE_A_STATUS: PASS
PHASE_B_STATUS: PASS
PHASE_C_STATUS: PASS
OVERALL_STATUS: PASS
EXIT_CODE: 0
```

Structural Integrity Conditions Confirmed

• **Replay invariant satisfied:** `B_A = B_B`
• **Manifest equality enforced across replay bundles**
• **Artifact completeness enforced**
• **Capsule reproducibility verified**
• **Cross-phase orchestrator validation enforced**

Deterministic Conformance Standard

All structural claims contained within this release are valid only under the condition of deterministic replay identity.

Independent executions under identical parameters must reproduce **byte-identical artifacts** within the frozen core boundary.

The replay identity condition is formally expressed as:

```
B_A = B_B
```

Conclusion

SBM Phase A, Phase B, and Phase C deterministic validation complete.
Deterministic core frozen under structural hash certification.
Presentation layer decoupled from structural verification boundary.

SBM is certified under deterministic conformance discipline.

Appendix A — Formal Structural Propositions and Research Conjectures

A.1 Behavioral Stabilization

Given:

```
alpha(N,H)
```

Stabilization over `[N1, N2]` is defined as:

```
alpha(N2,H) - alpha(N1,H) = 0
```

This indicates closure within tested domain.

A.2 Rare Emergence Event

A rare emergence occurs at `k` if:

```
alpha(k,H) - alpha(k-1,H) = 1
```

and the next emergence gap:

```
k_next - k
```

is large relative to local scale.

A.3 Structural Recurrence Definition

For signature `b` occurring at:

```
x1, x2, ..., xn
```

Define recurrence gaps:

```
G_i = x(i+1) - x(i)
```

If `G_i` exhibits deterministic structure rather than explosive divergence, recurrence is structural.

A.4 Empirical Behavioral Closure Conjecture

For certain deterministic operators with fixed horizon `H`:

```
exists M such that alpha(N,H) <= M
```

within sufficiently large tested domain.

No universality claim is made.

A.5 Delayed Emergence Constraint

Deep-scale emergence:

- Sparse
- Deterministic
- Does not induce super-exponential alphabet growth

A.6 Behavioral Envelope Principle

There may exist a function:

```
E(N)
such that:
alpha(N,H) <= E(N)
```

where `E(N)` grows sublinearly or stabilizes within tested range.

This remains an open research direction.

OMP