SHUNYAYA STRUCTURAL INFINITY TRANSFORM (SSIT)

A Transform Class for Lawful Infinity in Structural Number Theory

---

Status: Public Research Release (**v1.6**)
Date: January 29, 2026

Caution: Mathematical research framework. Observational and analytical use only.
Reproducibility: Fully deterministic outputs with SHA-256 receipts; large-scale robustness validated to `n_max = 1,500,000`.

---

0. MISSION

SSIT defines a **deterministic structural transform** that maps finite integers into a **lawful infinity-domain representation**, without modifying classical arithmetic or SSNT observables.

Its mission is to make the following possible **exactly and reproducibly**:

Infinity as an output, not a limit

SSIT produces an infinity-domain object from finite computation.
Infinity is no longer treated as an asymptote, metaphor, or informal gesture toward divergence.

Lawful structure over infinite magnitude

Each infinity-lifted object carries **bounded, comparable structure** via a posture lane `a(n)` and associated infinity descriptors.
This enables meaningful reasoning **inside infinity**, not merely about it.

Deterministic resolution of classical indeterminacies (tested)

By operating on **structured infinity objects** rather than raw magnitudes, SSIT enables deterministic outcomes for classical indeterminate forms such as INF/INF and INF-INF, evaluated as relationships between structured objects rather than undefined symbols.

Strict compatibility with SSNT (bridge role)

SSNT provides deterministic finite observables (e.g., closure resistance, structural timing). SSIT lifts these observables into an infinity algebra layer via `I(n)` and structured infinity objects, **without altering**:

• divisibility
• primality
• factorization
• arithmetic truth

SSIT is therefore **not a new arithmetic**.
It is a **transform class**: a structural mapping that extends what integers can express **without changing what integers are**.

TABLE OF CONTENTS

0. MISSION 1

1. WHY SSIT IS NEEDED — THE INFINITY GAP IN NUMBER THEORY 3

2. CORE SSIT INSIGHT — INFINITY AS A STRUCTURAL TRANSFORM, NOT A LIMIT 5

3. THE SSIT TRANSFORM — FORMAL DEFINITION 7

4. SSIT TRANSFORM CLASS — CANONICAL FORM 11

5. INFINITY OPERATIONS — SSIT ALGEBRA 16

6. ELIMINATION OF CLASSICAL INDETERMINATE FORMS 20

7. SSIT INVARIANTS AND SAFETY GUARANTEES 25

8. SSIT AS A TRANSFORM CLASS 28

9. SSIT OPERATIONAL EXAMPLES (MINIMAL CANONICAL SET) 31

10. SSIT INTEGRATION WITH THE SSNT FIVE-STATE FRAMEWORK 35

11. SSIT PHASE II — STRUCTURAL INFINITY DEPTH, SIS BANDING, AND CURVATURE 38

12. SSIT PHASE II v1.2 — TYPED INFINITY OBJECTS, SIS STATISTICS, AND CURVATURE BY SIS 43

13. PHASE II ROBUSTNESS VALIDATION (1,000,000 TO 1,500,000) 46

14. SSIT Phase II (Robust) v2 — Structural Governance Layer (Zones, Shock, Guard, IDO) 50

15. LIMITATIONS AND SCOPE 57

16. LICENSE, ATTRIBUTION, AND DISCLAIMER 60

Appendix A — Structural Link Between SSNT and SSIT 62

Appendix B — Reproducibility & Audit Checklist 65

Appendix C — Phase II Structural Scan Outputs (Audit Layer) 69

Appendix D — Visual Structural Evidence (Canonical Figures) 71

1. WHY SSIT IS NEEDED — THE INFINITY GAP IN NUMBER THEORY

Classical number theory is internally complete over finite integers.
SSNT extends this completeness by revealing **structural behavior under closure pressure**.

However, a precise gap remains:

Classical mathematics has no lawful object for infinity.

Infinity appears only as:

• a limit (`n -> infinity`)
• a symbol (∞) without internal structure
• an indeterminate form (`∞/∞`, `0 * ∞`)
• an asymptotic convenience, not a computable entity

This creates a hard boundary:

• finite integers are lawful
• infinite expressions are informal
• transitions between them are undefined

SSNT already exposes a critical fact:

Two integers can be structurally closer to infinity than others, even while remaining finite.

Examples:

• primes resist closure completely within their horizon
• perfect squares close exactly at the boundary
• balanced semiprimes approach isolation asymptotically

Yet once infinity is invoked, classical mathematics provides **no mechanism** to carry this structure forward.

As a result:

• distinctions collapse
• behavior is erased
• structural timing is lost precisely at the boundary

This is **not** a computational failure.
It is a **representational failure**.

SSIT exists to resolve exactly this failure.

It introduces a lawful way to state:

“This integer lifts into infinity in a specific, measurable, reproducible way.”

without:

• redefining integers
• altering factorization
• introducing limits, probabilities, or approximations

SSIT does not ask:
What is infinity?

SSIT asks:
How does finite structure lift into an infinity-domain without losing identity?

That question has no classical answer.
SSIT provides one — **deterministically**.

1.1 PRIMER — WHAT SSNT OBSERVES

Shunyaya Structural Number Theory (SSNT) studies integers by **behavior**, not by magnitude.

Instead of asking whether an integer is prime or composite, SSNT asks:

How does the integer collapse under deterministic closure pressure within a fixed structural horizon?

This produces reproducible observables such as:

• closure timing
• minimum closure divisor `d_min(n)`
• hardness ratios
• transition ordering

All observables are:

• deterministic
• horizon-bounded
• scale-independent
• non-probabilistic

No arithmetic rule is modified.

SSIT operates **strictly** on these SSNT observables.
It does not reinterpret them numerically.

It lifts them **structurally** into an infinity-domain where distinctions are **preserved rather than erased**.

---

2. CORE SSIT INSIGHT — INFINITY AS A STRUCTURAL TRANSFORM, NOT A LIMIT

The foundational insight of Shunyaya Structural Infinity Transform (SSIT) is this:

Infinity is not a value.
Infinity is a structural posture.

In classical mathematics, infinity appears only at the edge:

• as a destination (`n -> infinity`)
• as a loss of magnitude meaning
• as a collapse of distinction

Once infinity is invoked, finite structure disappears.
All paths converge to the same symbol (∞), regardless of how they arrived.

SSIT rejects this collapse.

2.1 Reorientation: From Magnitude to Structure

SSIT introduces a precise reorientation:

Infinity is reached through structure, not through size.

An object does not become infinity-aligned because it grows large.
It becomes infinity-aligned because its **closure resistance converges to the horizon**.

This insight emerges directly from SSNT.

Recall the structural hardness ratio:

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

Observed finite behaviors:

• `H(n) -> 0` → immediate closure
• `0 < H(n) < 1` → finite resistance
• `H(n) -> 1` → boundary resistance
• no closure within horizon → maximal isolation

Crucially, many distinct integers satisfy `H(n) ≈ 1`, even at modest sizes.

Classical theory treats these cases as unrelated.
SSIT elevates this convergence to **first-class structural meaning**.

2.2 Structural Definition of Infinity

SSIT defines infinity not as:

```
n -> infinity
```

but as:

`H(n) -> 1` under deterministic structural normalization.

This definition:

• does not depend on magnitude
• does not invoke limits
• does not erase identity

Instead, it preserves:

• identity (which integer it is)
• distinction (how it resists closure)
• behavioral memory (how the boundary is approached)

Infinity becomes reachable **without collapse**.

2.3 What Changes When Infinity Is Structural

Under the classical view:

infinity = end of structure

Under the SSIT view:

infinity = stabilized structural posture

This single shift enables SSIT to:

• compare different infinity-aligned behaviors
• preserve distinctions inside infinity
• define lawful transforms involving infinity
• prevent indeterminate collapse (`INF/INF`, `0 * INF`) by construction

Infinity is no longer a singularity where meaning stops.
It becomes a structured destination with internal geometry.

2.4 Phase II Confirmation

Phase II testing confirms that this structural definition is not rhetorical:

• infinity-aligned objects admit measurable depth
• infinity exhibits thickness regimes (SIS)
• infinity approach shows curvature and stability variation
• these properties persist under scale extension to `n_max = 1,500,000`

Infinity-as-structure is therefore **empirically stable**, not a small-range artifact.

2.5 Section Closure

SSIT does not redefine infinity numerically.
It **reconstructs infinity structurally**.

Infinity is no longer where structure ends.
It is where structure becomes **fully expressed**.

---

3. THE SSIT TRANSFORM — FORMAL DEFINITION

The Shunyaya Structural Infinity Transform (SSIT) is a deterministic structural transform that maps finite objects into an infinity-domain representation **without erasing structural information**.

SSIT does not operate on magnitude.
It operates on **structural posture**.

3.1 Transform Objective

The objective of SSIT is to answer a question classical mathematics cannot pose:

How does a finite object behave as it approaches infinity — without collapsing into infinity?

In classical analysis:

• distinct sequences converge to the same ∞
• identity is erased
• structural distinctions vanish
• indeterminate forms arise

SSIT preserves identity by transforming **how infinity is approached**, not by pushing values outward.

3.2 Input Domain

SSIT applies to any object `x` for which a **structural closure horizon** can be defined.

Applicable examples include:

• integers (via SSNT)
• ratios
• symbolic expressions
• discrete constructions

For integers `n > 1`, SSIT inherits the SSNT primitives:

• `d_min(n)` — first closure divisor
• `L(n) = floor(sqrt(n))` — structural horizon
• `H(n) = d_min(n) / sqrt(n)` — structural hardness

These quantities are deterministic, horizon-bounded, and invariant.

In this specification and public release, SSIT is fully defined, validated, and demonstrated for integer inputs via SSNT.

Other domains are conceptual extensions and are not claimed as implemented here.

3.3 Structural Normalization

Before lifting into infinity space, the object is **normalized against its closure horizon**.

Define **saturated hardness**:

```
H_s(n) = min(H(n), 1)
```

Interpretation:

• $H_s(n) \approx 0$ → structurally finite
• $0 < H_s(n) < 1$ → finite resistance
• $H_s(n) = 1$ → boundary resistance
• no closure within horizon → maximal isolation

This normalization removes magnitude entirely.
Only **relative resistance to closure** remains.

3.4 Definition of the SSIT Transform

Define the SSIT infinity coordinate:

```
I(n) = 1 / (1 - H_s(n))
```

This transform satisfies:

• if `H_s(n) -> 0`, then `I(n) -> 1`
• if `H_s(n) = 1/2`, then `I(n) = 2`
• if `H_s(n) -> 1`, then `I(n) -> +INF`

Crucially:

• divergence occurs **only** as closure reaches the horizon
• divergence is monotonic and ordered
• different approaches to infinity remain distinct

This is **not a limit**.
It is a **structural lift**.

3.5 Structural Meaning

Under SSIT:

• infinity is not a point
• infinity is a **direction in structural space**

Two integers with:

• different `n`
• different factor structure
• different closure histories

may both map to large `I(n)` values —
yet remain distinguishable by their **posture lane** and lifted structure.

SSIT therefore preserves:

• provenance
• behavioral memory
• comparative resistance

Infinity becomes a **measurable regime**, not a singular collapse.

3.6 Domain Safety (Non-Claims)

SSIT does not:

• change arithmetic truth
• predict primes
• accelerate factorization
• define new numeric infinities
• replace classical limits

SSIT is a **transform layer**, not a theory replacement.
Its role is to prevent loss of structure when infinity appears.

3.7 Transform Admissibility

SSIT satisfies four admissibility criteria:

1. Deterministic
No randomness, no heuristics.

2. Monotonic
Increasing closure resistance implies increasing infinity posture.

3. Scale-Free
Magnitude never appears in the transform.

4. Invertible (Except at the Horizon)

For all finite `I(n)`:

```
H_s(n) = 1 - 1 / I(n)
```

Loss of invertibility occurs **only** at true structural infinity —
precisely where classical representations fail.

3.8 Transform Closure Statement

SSIT introduces a lawful, deterministic way to enter infinity **without erasing structure**.

Infinity is no longer:

• undefined
• singular
• destructive

It is a reachable, ordered structural regime.

SSIT does not ask:

How large is `n`?

SSIT asks:

How close is this structure to non-closure?

That question **survives infinity**.

---

4. SSIT TRANSFORM CLASS — CANONICAL FORM

This section defines the **canonical SSIT transform object**.

It specifies exactly:

• what is computed
• what is recorded
• what is invariant

independent of implementation language, execution environment, or scale.

The purpose of this class is to make **infinity explicit, lawful, and reproducible**.

4.1 Purpose of the Transform Class

The SSIT Transform Class exists to ensure that infinity behavior is:

• structurally explicit
• deterministic

• machine-recordable
• hash-verifiable

Without a transform class, infinity collapses into an untyped abstraction.
With this class, infinity becomes **structured output**.

4.2 Canonical SSIT Object

For any admissible object `x` (integer, ratio, or structural carrier), the SSIT transform produces a **canonical structured infinity object**.

For integer inputs `n`, the canonical SSIT object is:

```
Omega(n) = < M_inf, a(n), D_inf(n), G(n) >
```

Each field has a **single fixed meaning** and must always be present.

4.3 Field Definitions (Normative)

M_inf — Infinity Magnitude

`M_inf` denotes structural infinity magnitude.

For FINSET objects, this magnitude is finite (`I(n)` finite).
For INFSET objects, `M_inf = +INF`.

Infinity magnitude is **derived**, not assumed.

---

a(n) — Posture Lane

`a(n)` captures the **directional posture** of the object's approach toward infinity.

Properties:

• bounded
• deterministic
• scale-free
• identity-preserving

Interpretation:

• two objects may share similar infinity magnitude
• yet differ in how they approach infinity

Lane preserves **identity beyond divergence**.

D_inf(n) — Structural Infinity Depth

`D_inf(n)` measures how much **closure structure exists near the horizon**.

Properties:

- bounded in `[0,1]`
- deterministic
- horizon-local

Interpretation:

- low depth → thin infinity
- high depth → thick infinity

Depth is defined **only for FINSET objects**.

G(n) — Governance State

`G(n)` is a composite governance descriptor derived deterministically from:

- infinity stability zone
- curvature shock flag
- dominance ordering context

Governance determines **how infinity-aligned objects may be interacted with**, not their numeric value.

4.4 Typed Infinity Regimes

SSIT separates infinity-aligned objects into two **typed regimes**:

- **FINSET** — infinity-aligned but finite (`I(n)` finite)
- **INFSET** — terminal infinity (`I(n) = +INF`)

Rules:

- SIS banding applies **only to FINSET**
- governance applies **only to FINSET**
- INFSET objects are typed, not banded

This prevents false treatment of terminal infinity as “thin infinity”.

4.5 Conceptual Example

For an integer `n`:

• compute `d_min(n)`
• compute saturated hardness `H_s(n)`
• compute infinity coordinate `I(n) = 1 / (1 - H_s(n))`
• compute posture lane `a(n)`
• compute depth `D_inf(n)` (FINSET only)
• assign typed regime and governance

Result:

```
Omega(n) = < M_inf, a(n), D_inf(n), G(n) >
```

Infinity is now **typed, structured, and governed**.

4.6 Canonical Output Artifacts

SSIT outputs are recorded in **synchronized deterministic artifacts**:

1. Canonical Scan Table

• one row per object
• fields include: lane, depth, zone, guard, dominance, curvature
• no inferred or post-processed values

2. Canonical Reports

• posture and zone distributions
• dominance statistics
• curvature summaries
• invariant checks

Both artifacts are:

• generated from a single deterministic run
• reproducible
• suitable for independent verification

4.7 Invariants (Hard Requirements)

Any valid SSIT implementation MUST satisfy:

1. Monotonic Infinity Lift
If `H_s(a) < H_s(b)` then `I(a) < I(b)`

2. No False Infinity
`I(n)` diverges if and only if `H_s(n) -> 1`

3. Lane Stability
`a(n)` remains invariant across re-runs

4. Typed Regime Closure
Every object is either FINSET or INFSET, never ambiguous

Violation of any invariant invalidates the transform.

4.8 Determinism and Replay

SSIT is fully deterministic:

• no randomness
• no approximation
• no adaptive tuning

Given identical inputs, SSIT always produces **identical structured infinity objects**.

Infinity is therefore **replayable and auditable**.

4.9 Why a Transform Class Is Necessary

Without a transform class:

• infinity collapses into notation
• structure cannot be audited
• comparison fails under divergence

With the SSIT Transform Class:

• infinity becomes an object
• identity survives divergence
• structure remains observable

This is the **minimum machinery required for lawful infinity handling**.

4.10 Section Closure

SSIT defines infinity as a **structured posture**, not a terminal value.

The transform class ensures that:

• infinity can be entered
• identity is preserved
• behavior remains measurable

Infinity is no longer where mathematics stops.
It is where **structure becomes visible**.

5. INFINITY OPERATIONS — SSIT ALGEBRA

This section defines the **lawful algebra of infinity** enabled by SSIT.

All operations act on **structured infinity objects**, not on raw magnitudes.

Classical infinity fails because it lacks identity.
SSIT restores algebra by preserving structure.

5.1 Operands and Closure

All SSIT infinity operations act on **terminal infinity objects** of type **INFSET**.

For an INFSET object derived from SSIT:

```
Omega = < +INF, a, _, _ >
```

Where:

• `+INF` denotes structural infinity magnitude
• `a` is the bounded posture lane (`a in (-1, +1)`)
• depth and governance fields are undefined at terminal infinity

At infinity:

• `H_s -> 1` is implicit
• depth and SIS no longer apply
• posture lane remains the sole identity carrier

Operations are **closed over INFSET objects**, unless explicitly stated otherwise.

Note: Operations involving FINSET objects are governed by Phase II rules and are intentionally excluded from this terminal infinity algebra.

5.2 Division of Infinities (Resolved)

Classically:

`INF / INF` is indeterminate.

SSIT resolves this by operating on **posture separation**, not magnitude.

For two INFSET objects:

```
Omega1 = < +INF, a1, _, _ >
Omega2 = < +INF, a2, _, _ >
```

Define:

```
Omega1 / Omega2 = finite-class( abs(a1 - a2) )
```

Properties:

• result is finite
• result is bounded
• result is deterministic
• result preserves relational structure

Interpretation:

Two infinities differ only by **how they approach infinity**.

5.3 Subtraction of Infinities (Zero-Class Emergence)

Classically:

`INF - INF` is undefined.

In SSIT:

```
Omega1 - Omega2 = zero-class( abs(a1 - a2) )
```

Properties:

• produces a structured zero-class object
• magnitude reflects posture separation
• symmetric under operand exchange

Interpretation:

• identical infinities cancel structurally
• near-identical infinities leave a small residue

Zero becomes **structured**, not assumed.

5.4 Addition of Infinities (Posture Merge)

For addition:

```
Omega1 + Omega2 = < +INF, (a1 + a2) / 2, _, _ >
```

Properties:

• result remains infinite
• posture merges smoothly
• boundedness of lane is preserved

The arithmetic mean is used as the unique symmetric, bounded, and order-preserving posture merge; no other choice satisfies all three simultaneously.

Interpretation:

Adding infinities does **not increase infinity**.
It merges directional posture.

5.5 Identity and Symmetry Properties

SSIT infinity algebra satisfies:

Symmetry
```
Omega1 / Omega2 = Omega2 / Omega1
```

Cancellation
If `a1 = a2`, then:
```
Omega1 - Omega2 = zero-class(0)
```

Idempotence
```
Omega + Omega = Omega
```

These properties are impossible under classical infinity,
but follow naturally once **identity is preserved**.

5.6 Undefined Case (Single Exception)

The only undefined operation in SSIT infinity algebra is:

```
+INF / 0
```

This mirrors classical division rules and is explicitly excluded.

No other indeterminacy exists.

5.7 Classification of Results

All SSIT infinity operations yield exactly one of three result classes:

• **finite-class**
Bounded numeric structural output

• **zero-class**
Structured zero residue

• **infinite-class**
Structured infinity object

Every result is **typed**.
Nothing is ambiguous.

5.8 Why This Resolves Classical Failures

Classical infinity fails because:

• it has no identity
• it has no direction
• it has no memory

SSIT infinity succeeds because:

• identity is preserved via posture lane
• direction is explicit
• memory is structural, not numeric

Infinity becomes **algebraic**, not symbolic.

5.9 Structural Meaning (Not Interpretation)

These operations:

- do not approximate limits
- do not rely on epsilon arguments
- do not reinterpret calculus

They operate strictly on **structural posture**, not numerical size.

5.10 Section Closure

SSIT establishes a **closed, lawful algebra over infinity**.

For the first time:

- `INF / INF` is computable
- `INF - INF` yields structure
- `INF + INF` preserves identity

Infinity is no longer an exception.
It is an **algebraic domain**.

---

6. ELIMINATION OF CLASSICAL INDETERMINATE FORMS

Classical mathematics treats several expressions involving infinity as indeterminate.

These are not paradoxes of nature.
They are failures of representation.

SSIT eliminates indeterminacy by **refusing to collapse structure into magnitude**.

6.1 Source of Indeterminacy in Classical Theory

Classical infinity is represented as a single untyped symbol:

```
INF
```

This representation loses:

• direction
• posture
• relational identity

As a result, expressions such as:

```
INF / INF
INF - INF
0 * INF
```

cannot be evaluated meaningfully.

The problem is not infinity itself.
The problem is **structure loss at infinity**.

6.2 SSIT Principle — Structure Before Magnitude

SSIT replaces unstructured infinity with a **structured terminal infinity object**.

For an INFSET object:

```
Omega = < +INF, a, _, _ >
```

Where:

• `+INF` denotes structural infinity magnitude
• `a` is the bounded posture lane (`a in (-1, +1)`)
• depth and governance fields are undefined at terminal infinity

At infinity:

• `H_s -> 1` is implicit
• depth and SIS do not apply
• posture lane remains the sole identity carrier

All infinity operations act on **posture components**, not on magnitude.
Magnitude never competes with magnitude.

6.3 Resolution of INF / INF

Classical status:

`INF / INF` → indeterminate

SSIT resolution:

Given:

```
Omega1 = < +INF, a1, _, _ >
Omega2 = < +INF, a2, _, _ >
```

Define:

```
Omega1 / Omega2 = finite-class( abs(a1 - a2) )
```

Outcome:

- finite
- bounded
- deterministic

`INF / INF` becomes a **measure of structural separation**, not a size comparison.

6.4 Resolution of INF - INF

Classical status:

`INF - INF` → indeterminate

SSIT resolution:

```
Omega1 - Omega2 = zero-class( abs(a1 - a2) )
```

Interpretation:

- identical infinities cancel perfectly
- near-identical infinities leave a structured residue

Zero is no longer empty.
It is **structurally meaningful**.

6.5 Resolution of 0 * INF

Classical status:

`0 * INF` → indeterminate

SSIT resolution:

Let:

• `0` denote a structured zero (zero-class)
• `Omega = < +INF, a, _, _ >`

Define:

```
0 * Omega = zero-class(0)
```

Explanation:

• zero absorbs infinity structurally
• posture cannot survive multiplication by zero

There is no ambiguity.

6.6 Resolution of Limit-Type Expressions

Classical limits rely on:

• approach direction
• epsilon bookkeeping
• asymptotic cancellation

SSIT replaces limits with **exact structural states**.

Instead of:

```
lim x -> INF f(x) / g(x)
```

SSIT evaluates:

```
Omega_f / Omega_g
```

This evaluation is:

• exact
• posture-dependent
• finite or structured

No limit is required.

6.7 Preserved Undefined Case (Single Exception)

SSIT deliberately preserves exactly one undefined operation:

```
+INF / 0
```

Reason:

• zero has no posture lane
• division by zero destroys structure

This mirrors classical arithmetic and enforces structural safety.

No other indeterminacy remains.

6.8 Indeterminacy Resolution Table

Classical Expression	Classical Status	SSIT Status
INF / INF	Indeterminate	Finite-class (posture separation)
INF - INF	Indeterminate	Zero-class (structured residue)
0 * INF	Indeterminate	Zero-class (absorbing zero)
INF + INF	Undefined growth	Infinite-class (merged posture)
INF / 0	Undefined	Undefined (preserved)

6.9 What Is Gained

By eliminating indeterminate forms, SSIT enables:

• lawful infinity algebra
• stable symbolic reasoning
• elimination of epsilon heuristics
• deterministic behavior at infinity

This is not reinterpretation.
It is a **correction of representation**.

6.10 Section Closure

Classical indeterminacy arises from missing structure.
SSIT restores structure and removes indeterminacy.

Infinity is no longer fragile.
It is **computable**.

7. SSIT INVARIANTS AND SAFETY GUARANTEES

SSIT is defined as a **transform class**, not a heuristic.
Its correctness depends on **invariants that hold for all admissible inputs**, not on empirical behavior, tuning, or statistical validation.

These invariants ensure that infinity is introduced **lawfully, safely, and conservatively**.

7.1 Determinism Invariant

For any integer `n > 1`:

• divisor scanning order is fixed
• all arithmetic is exact
• no randomness, tuning, or approximation is used

Therefore:

SSIT(n) is fully deterministic

Repeated executions yield identical:

• `H_s(n)`
• `I(n)`
• posture lane `a(n)`
• depth `D_inf(n)` (when applicable)
• governance state `G(n)`
• all infinity-algebra outcomes

This invariant is **foundational**.
Without determinism, infinity claims are not auditable.

7.2 Classical Preservation Invariant

SSIT does **not** modify classical mathematics.

Specifically:

• primality is unchanged
• factorization is unchanged
• divisibility relations are unchanged
• arithmetic truth is unchanged

SSIT adds **only** a structural layer.

For any classical statement `C(n)`:

If `C(n)` is true classically, it remains true under SSIT.

No classical theorem is weakened, contradicted, or reinterpreted.

7.3 Infinity Isolation Invariant

All infinite magnitude is confined to the explicit infinity component of the SSIT object.
For infinity-aligned objects:

```
Omega(n) = < +INF, a(n), _, _ >
```

(where fields not shown are undefined or irrelevant at terminal infinity)

The posture lane is strictly bounded:

```
a(n) in (-1, +1)
```

Consequences:

- **no overflow propagation**
- **no numerical instability**
- **infinity cannot leak into finite computation**

Infinity is structurally isolated, not numerically contagious.

7.4 Totality (Closure) Invariant

SSIT guarantees **total classification**.

For all integers `n > 1`, exactly one holds:

- `I(n) = +INF`
- `1 < I(n) < +INF`

There are:

- no undefined values
- no NaN states
- no branch ambiguity

Every admissible input is mapped to a **valid structured output**.

7.5 Algebraic Safety Invariant

All SSIT algebra obeys the following constraints:

- operations act on **posture**, not magnitude
- results are bounded where expected
- symmetry is preserved
- cancellation is deterministic

Exactly one operation remains undefined:

```
+INF / 0
```

This is intentional, mirrors classical arithmetic, and enforces **structural safety**.

No other indeterminacy exists.

7.6 Monotonic Pressure Invariant

Structural pressure increases **monotonically** with scan depth.

This guarantees:

- no retroactive collapse
- no reversal of closure order
- no dependence on factor enumeration strategy

Structural timing is **intrinsic to the integer**, not to the algorithm.

7.7 Non-Degeneracy Invariant

Distinct integers with different internal structure may share:

- `I(n) = +INF`

but will **not** collapse into identical structured infinity objects unless their **full structural posture matches**.

That is:

- infinity alignment does not imply identity
- posture degeneracy is prevented by construction

SSIT preserves **distinction inside infinity**.

7.8 Section Closure

SSIT is safe because it is:

• **deterministic**
• **structurally bounded**
• **algebraically lawful**
• **classically conservative**

Infinity is introduced **without risk**, **without ambiguity**, and **without structural loss**.

---

8. SSIT AS A TRANSFORM CLASS

SSIT is **not** a metric, encoding, scoring function, or analytic tool.
It is a **transform class** with a clearly defined **domain**, **codomain**, and **algebra**.

8.1 What Defines a Transform Class

A mathematical transform class satisfies three conditions:

1. Domain-to-codomain mapping
Inputs are lifted into a new space with different admissible operations.

2. Preservation of original structure
All classical properties remain valid and recoverable.

3. Introduction of a new algebraic space
The codomain supports operations not defined in the original domain.

SSIT satisfies all three.

8.2 Why SSIT Is Not a Function

A classical function maps inputs to scalar values.

SSIT maps inputs to **structured objects**:

```
n -> Omega(n) = < M_inf, a(n), D_inf(n), G(n) >
```

The output is **not a scalar**.
It is a **composite object** with internal structure, typing, and algebraic behavior.

As a result:

• the output space is not numeric
• operations act on object components, not values
• algebra is defined over objects, not scalars

Therefore, SSIT is not a function in the classical sense.
It is a **structural transform** that lifts finite inputs into a new algebraic domain.

8.3 Why SSIT Is Not a Metric

Metrics require:

• symmetry
• triangle inequality
• distance interpretation

SSIT provides none of these.

The posture lane `L` is:

• not a distance
• not a magnitude
• not ordered in a metric space

SSIT does not measure separation.
It **repositions structure in an infinity-domain**.

8.4 Why SSIT Is Not an Encoding

Encodings represent or compress information.

SSIT:

• introduces lawful infinity where none existed
• expands finite objects into infinity-domain entities
• preserves all original information

Nothing is encoded.
Nothing is compressed.
Nothing is lost.

SSIT is **generative**, not representational.

8.5 Transform Nature of SSIT

SSIT performs a **domain lift**:

• from finite arithmetic
• into a lawful infinity algebra

This lift is:

• deterministic
• reversible at the classical layer
• non-destructive

Classical mathematics remains intact.
A new operational layer becomes available.

8.6 Transform Closure Property

SSIT is **closed under its own algebra**.

If:

`Omega(n1)` and `Omega(n2)` exist

Then:

• `Omega(n1) + Omega(n2)` exists
• `Omega(n1) - Omega(n2)` exists
• `Omega(n1) / Omega(n2)` exists for all admissible operands except the single preserved undefined case `+INF / 0`.

Closure is required for transform validity.
SSIT satisfies this requirement.

8.7 Why This Matters

Classical infinity is a boundary.

SSIT turns infinity into:

• an object
• an operand
• a lawful participant in algebra

This shift is **categorical**, not incremental.

8.8 Section Closure

SSIT qualifies as a true **transform class** because it:

• lifts finite structure into infinite space
• preserves classical truth
• supports closed algebra
• resolves indeterminacy without approximation

It is not a reinterpretation.
It is a **structural elevation**.

---

9. SSIT OPERATIONAL EXAMPLES (MINIMAL CANONICAL SET)

This section records a minimal, canonical set of operational examples demonstrating that **SSIT infinity objects behave lawfully, deterministically, and algebraically** under defined operations.

These are **not illustrative toys**.
They are **structural proofs of operability**.

9.1 Canonical Infinity Object Form

For any integer `n > 1`, once lifted into the infinity domain, SSIT produces a **canonical structured object**:

```
Omega(n) = < M_inf, a(n), D_inf(n), G(n) >
```

Where:

- **`M_inf`** is the infinity magnitude
  - `M_inf = +INF` for terminal infinity objects (INFSET)
- **`a(n)`** is the posture lane

    - bounded in `(-1, +1)`
    - preserves identity beyond divergence
- **`D_inf(n)`** is structural infinity depth
    - defined **only for FINSET objects**
- **`G(n)`** is the deterministic governance state

For **terminal infinity operations**, the effective form is:

`Omega = < +INF, a, _, _ >`

All examples below operate **strictly on this canonical form**, with **posture lane `a` as the sole identity carrier at infinity**.

9.2 Division of Infinities (Resolved)

Classical form:
INF / INF → undefined

SSIT form:
`Omega(n1) / Omega(n2)`

Result:
`finite-class( abs(a1 - a2) )`

Properties:

- magnitude collapses to **finite-class**
- posture separation is preserved
- result is bounded and deterministic

Interpretation:
Infinity divided by infinity becomes a **relative structural comparison**, not an indeterminate symbol.

9.3 Subtraction of Infinities (Resolved)

Classical form:
INF - INF → undefined

SSIT form:
`Omega(n1) - Omega(n2)`

Result:
`zero-class( abs(a1 - a2) )`

Observed outcomes:

- zero-class when postures match exactly
- structured residue when postures differ
- never undefined

Interpretation:
Subtraction measures **structural alignment**, not magnitude.
Perfect alignment cancels infinity **cleanly and lawfully**.

9.4 Addition of Infinities (Preserved)

Classical form:
INF + INF → INF

SSIT form:
`Omega(n1) + Omega(n2)`

Result:
`< +INF, (a1 + a2) / 2, _, _ >`

Properties:

- infinity remains infinite
- posture merges smoothly
- lane boundedness in `(-1, +1)` is preserved

Interpretation:
Addition **preserves infinity** while **updating structure**, not magnitude.

9.5 Prime vs Composite Interaction

Let:

- `p` be prime
- `c` be composite

Then:

- `Omega(p) / Omega(c)` → finite-class
- `Omega(p) - Omega(c)` → zero- or finite-class
- `Omega(p) + Omega(c)` → infinite-class

Interpretation:
Primes act as **maximal isolation anchors**, not singularities.
Their interaction with composites is **lawfully structured**, not exceptional.

9.6 Structural Similarity Test

If:
`a(n1) ≈ a(n2)`

Then:

- division approaches zero-class
- subtraction approaches zero-class
- addition preserves posture stability

This establishes **structural equivalence**, not numeric equality.

9.7 Operational Boundary (Single Exception)

Exactly **one operation remains undefined**:

`Omega(n) / 0`

Reason:

- zero has no posture lane
- division by zero destroys structure

This mirrors classical arithmetic **exactly**.
No new singularities are introduced.

9.8 Section Closure

These canonical examples establish that:

- **infinity is operational**
- **algebra is closed**
- **indeterminacy is eliminated**
- **classical arithmetic behavior is preserved**

SSIT does **not approximate infinity**.
It **governs** it.

---

10. SSIT INTEGRATION WITH THE SSNT FIVE-STATE FRAMEWORK

SSIT does **not** introduce a new behavioral taxonomy.
It **projects SSNT behavior into infinity space without distortion**.

All SSIT observables are derived **directly and deterministically** from SSNT structural timing and closure resistance.
No behavioral reinterpretation occurs.

10.1 Structural Axis Mapping

SSNT observes integer behavior along a **closure-timing axis**.
SSIT lifts the **same axis** into an infinity-domain representation.

Mapping:

- **SSNT observables**:
  `d_min(n), H(n), tau(n)`
- **SSIT observables**:
  `I(n), Omega(n), a(n)`

Key points:

- `I(n)` is computed deterministically from `H_s(n)`
- `Omega(n)` preserves identity, posture, and governance
- `a(n)` is inherited from SSNT transition behavior

No reinterpretation occurs.
Only the **codomain changes**.
Structure is preserved **exactly**.

10.2 Local States in Infinity Space

Using the first-order infinity differential:

```
dI(n) = I(n) - I(n-1)
```

SSIT inherits the **same five local structural states** defined in SSNT.

No new states are introduced.

CALM

- low variation in `dI(n)`
- stable posture lanes `a(n)`
- corresponds to SSNT CALM corridors

Infinity is present but structurally quiet.

NEARO

- persistently large `abs(dI(n))`
- posture drifts without oscillation
- corresponds to late-closure resistance zones

Infinity is approached smoothly and coherently.

QUEARO

- frequent sign flips in `dI(n)`
- posture oscillates
- corresponds to SSNT SHOCK corridors

Infinity is turbulent but lawful.

ZEARO

- theoretical structural center
- posture near zero
- acts as a reference, not a populated regime

ZEARO is a coordinate, not a behavioral class.

MEARO

- disagreement across window scales
- instability of local interpretation
- corresponds to fracture-cluster environments

Infinity becomes structurally ambiguous, not undefined.

10.3 Non-Local State — PEARO (Infinity Coherence)

PEARO detects coherence across distance, not local fluctuation.

Procedure:

- partition integers into blocks `B_k`
- compute SSIT state distributions per block
- compare **non-adjacent** blocks

PEARO is detected when:

- posture distributions align
- `dI(n)` statistics match across scales
- alignment persists without adjacency

This establishes **non-local emergence in infinity space**.

10.4 Critical Property (Invariance Result)

SSIT does **not** invent new behavior.

It reveals that:

- **SSNT behavior persists unchanged at infinity**
- **infinity does not collapse structural regimes**
- **behavioral grammar survives magnitude lift**

This is a **non-trivial invariance result**, not an assumption.

10.5 Section Closure

SSIT and SSNT share the **same five-state language**.

SSIT proves that:

Behavioral structure is invariant under infinite lift.
Infinity does not destroy structure.
It exposes it.

---

11. SSIT PHASE II — STRUCTURAL INFINITY DEPTH, SIS BANDING, AND CURVATURE

SSIT Phase II extends the original SSIT transform by introducing **second-order structural observables inside infinity**, without modifying any Phase I definition, invariant, or algebra.

Phase II **does not redefine infinity**.
It **resolves structure inside infinity**.

All results reported in this section are:

- **deterministic**
- **horizon-bounded**
- **reproducible with SHA-256 receipts**
- **computed from finite scans only**

11.1 Motivation — Why Phase II Is Needed

SSIT Phase I established that:

- **infinity can be produced deterministically**
- **infinity is a structured posture, not a limit**
- **classical indeterminacies can be resolved lawfully**

A remaining gap persisted:

All infinity-aligned objects were treated as equally infinite.

Phase II resolves this by introducing **structural thickness, geometry, and stability inside infinity**.

Infinity becomes:

- **stratified**
- **classifiable**
- **locally smooth or turbulent**
- **partially orderable**

This advances SSIT from **lawful infinity existence** to **lawful infinity structure**.

11.2 Structural Infinity Depth (`D_inf`)

Phase II introduces **Structural Infinity Depth**, denoted:

```
D_inf(n)
```

This measures **how much closure structure exists near the horizon**, not merely whether the horizon is reached.

Definition:

Let:

```
L(n) = floor(sqrt(n))
```

Let divisors `d` satisfy `2 <= d <= L(n)`.

Define:

```
D_inf(n) = (1 / (L(n) - 1)) * sum_{d | n, 2 <= d <= L(n)} (log(d + 1) /
log(L(n) + 1))
```

Properties:

- **bounded in [0, 1]**
- **deterministic**
- **horizon-local**
- **scale-free**

Interpretation:

- **low `D_inf` → thin infinity (sparse closure structure)**
- **high `D_inf` → thick infinity (dense horizon structure)**

Two integers may both satisfy `I(n) = +INF` while differing strongly in `D_inf(n)`.

This distinction **does not exist in classical mathematics**.

11.3 SIS — Structural Infinity Spectrum (Banding)

To make infinity thickness **operational and comparable**, Phase II introduces **SIS banding**.

Banding is derived **globally and deterministically** from the finite-set distribution of `D_inf`.

Let:

- `q33` = floor-quantile of `D_inf` at 0.33
- `q66` = floor-quantile of `D_inf` at 0.66

Define SIS bands:

- **THIN** if `D_inf(n) <= q33`
- **MEDIUM** if `q33 < D_inf(n) <= q66`
- **THICK** if `D_inf(n) > q66`

This classification:

- **introduces no tunable parameters**
- **adapts automatically to scan range**
- **remains invariant under re-execution**

SIS is a **taxonomy**, not a score.

11.4 Phase II Infinity Object (`Omega2`)

With Phase II, the structured infinity object extends to:

```
Omega2(n) = < +INF, a(n), D_inf(n), SIS(n) >
```

where:

- **`+INF`** denotes structural infinity
- **`a(n)`** is the Phase I posture lane
- **`D_inf(n)`** is structural depth
- **`SIS(n)`** is the spectrum band

This object **extends `Omega` without replacing it**.
No Phase I invariant is violated.

11.5 Structural Curvature Near Infinity

Phase II introduces **curvature of infinity approach**, defined **only where all values are finite**.

Let:

```
dI(n) = I(n) - I(n - 1)

d2I(n) = I(n + 1) - 2*I(n) + I(n - 1)
```

Define curvature:

```
K(n) = abs(d2I(n))
```

Properties:

- **computed only for finite `I(n)`**
- **no smoothing or interpolation**
- **exact, local, deterministic**

Interpretation:

- **low `K(n)` → smooth approach to infinity**
- **high `K(n)` → structural turbulence near infinity**

Curvature reveals **shock-like behavior inside infinity alignment**.

11.6 Empirical Findings (`n <= 200000`)

A deterministic scan up to `n_max = 200000` establishes:

Partitioning

- **Finite-set:** 181,929 integers
- **Infinity-set:** 18,070 integers

Near-Infinity Regime

- **Only 87 integers satisfy `1 - eps <= H_s(n) < 1`**
- **Near-infinity is rare and event-like, not noise**

Infinity Depth Distribution

- `D_inf_min ≈ 0.0004`
- `D_inf_median ≈ 0.0064`
- `D_inf_max = 1`

Curvature

- **Median `K ≈ 0.037`**
- **Maximum observed `K ≈ 863`**

These results confirm that:

- **infinity has internal structure**
- **infinity approach can be smooth or turbulent**
- **these properties are measurable and reproducible**

11.7 Partial Ordering Inside Infinity

Phase II introduces a **partial order over infinity objects**:

`Omega2(n1) ≺ Omega2(n2)` iff
`a(n1) < a(n2)` AND `D_inf(n1) <= D_inf(n2)`

This order is:

- **antisymmetric**
- **non-total**
- **deterministic**

Infinity is no longer flat.
It admits **directional hierarchy**.

11.8 What Phase II Changes (and What It Does Not)

Phase II DOES:

- **stratify infinity**
- **introduce thickness and curvature**
- **classify infinity behavior deterministically**

Phase II DOES NOT:

- **redefine SSIT Phase I**
- **alter arithmetic**
- **predict number-theoretic outcomes**
- **introduce limits, probabilities, or heuristics**

Phase II is a **conservative, reformative extension**.

11.9 Section Closure

SSIT Phase II establishes that:

- **infinity has depth**
- **infinity has geometry**
- **infinity has stability regimes**

Infinity is no longer just reachable.
It is structurally navigable.

---

12. SSIT PHASE II v1.2 — TYPED INFINITY OBJECTS, SIS STATISTICS, AND CURVATURE BY SIS

SSIT Phase II v1.2 extends Phase II v1.1 with **three strictly conservative upgrades**:

1. **Typed infinity objects** that explicitly separate **FINSET** and **INFSET** regimes
2. **SIS statistics** (counts per band) as a deterministic sanity check and distribution witness
3. **Curvature extremes by SIS band**, revealing how turbulence varies across infinity thickness regimes

No Phase I definitions change.
No Phase II v1.1 definitions change.

Only **representation clarity and structured summaries** are added.

12.1 Typed Infinity Objects (FINSET vs INFSET)

Phase II v1.1 introduced the extended infinity object:

```
Omega2(n) = < +INF, a(n), D_inf(n), SIS(n) >
```

A critical structural distinction must be made:

- **FINSET:** `I(n)` is finite
- **INFSET:** `I(n) = +INF`

To eliminate representational ambiguity, v1.2 introduces **typed infinity objects**.

FINSET typed object
(defined only when `I(n)` is finite):

```
OmegaFin(n) = < +INF, a(n), D_inf(n), SIS(n), tag=FINSET >
```

INFSET typed object
(defined only when `I(n) = +INF`):

```
OmegaInf*(n) = < +INF*, a(n), D_inf(n), tag=INFSET >
```

Resolution achieved:

- **FINSET objects carry SIS banding** (THIN / MEDIUM / THICK)
- **INFSET objects are explicitly typed** and do not claim SIS membership

This prevents **false equivalence** between finite-depth infinity alignment and true structural infinity.

12.2 Partial Order Applies Only to FINSET Objects

The Phase II partial order remains unchanged, but its **domain of validity is now explicit**:

`OmegaFin(n1) ≺ OmegaFin(n2)` iff
`a(n1) < a(n2)` AND `D_inf(n1) <= D_inf(n2)`

Cross-type comparisons are **not performed** and are marked **NA**.

This preserves logical admissibility:

- **FINSET admits depth stratification and SIS banding**
- **INFSET constitutes a separate structural regime under SSIT**

12.3 Empirical Findings (`n <= 200000`)

A deterministic scan up to `n_max = 200000` confirms:

Counts

- **INFSET_count (`I = +INF`) = 18070**
- **FINSET_count (`I` finite) = 181929**
- **NearInf_count (`1 - eps <= H_s < 1`) = 87**
- **prime_proxy_count** (no divisor within horizon) = 17984
- **perfect_square_count = 446**

These counts are **reproducible** and verify the **Phase II partition at scale**.

12.4 SIS Quantiles and SIS Band Counts (FINSET)

SIS thresholds
(derived deterministically from the FINSET depth distribution):

- `q33_depth = 0.00366939983329`
- `q66_depth = 0.0109352933459`

SIS band counts in FINSET:

- **SIS_count_THIN = 60042**
- **SIS_count_MEDIUM = 60031**

- **SIS_count_THICK = 61856**

This confirms:

- **SIS banding is stable**
- **bands are balanced and non-degenerate**
- **infinity thickness is a real structural feature, not an artifact**

12.5 Depth Distribution Witness (FINSET)

Depth summary:

- `depth_min = 0.000403495443472`
- `depth_median = 0.00639541712845`
- `depth_max = 1`

This establishes that:

- **`D_inf(n)` spans a real distribution**
- **both extreme thin and extreme thick infinity regimes exist**

12.6 Curvature Summary and Curvature-by-SIS

Curvature is defined **only where `I(n)` is finite**:

```
dI(n) = I(n) - I(n - 1)
d2I(n) = I(n + 1) - 2*I(n) + I(n - 1)
K(n) = abs(d2I(n))
```

Summary

- **`K_median = 0.0371985192482`**
- **`K_max = 862.98954024`**

Key observation

- **Top curvature events occur predominantly in THIN and MEDIUM SIS bands**

This establishes a new structural result:

- **thin infinity can be violently unstable**
- **thick infinity can still exhibit curvature events, but with distinct turbulence profiles**

12.7 Deterministic Sample Pair Results (Typed)

Typed rendering ensures **logical correctness**:

- **FINSET vs FINSET → ordering and comparisons computed**
- **FINSET vs INFSET → marked NA (cross-type)**

This prevents false conclusions arising from treating INFSET elements as **“thin FINSET infinity.”**

12.8 What v1.2 Adds (and What It Does Not)

v1.2 ADDS:

- **typed infinity objects** (`OmegaFin`, `OmegaInf*`)
- **SIS counts** as a distribution witness
- **curvature extremes stratified by SIS band**

v1.2 DOES NOT:

- modify `I(n)`, `a(n)`, `D_inf(n)`, or `K(n)`
- alter SSIT Phase I or Phase II v1.1 logic
- introduce tuning, heuristics, or probabilistic rules

v1.2 is a **clarifying, conservative extension of Phase II**.

12.9 Section Closure

SSIT Phase II v1.2 establishes that infinity is not only structured — **it is typed, stratified, and auditable**:

- **typed regimes:** FINSET vs INFSET
- **thickness regimes:** THIN / MEDIUM / THICK
- **turbulence regimes:** curvature behavior differs by SIS band

Infinity becomes a space that can be **classified without collapsing meaning**.

---

13. PHASE II ROBUSTNESS VALIDATION (1,000,000 TO 1,500,000)

SSIT Phase II must remain structurally stable under scale extension.

This section validates that infinity depth, SIS banding, and curvature observables remain lawful, non-degenerate, and interpretable when the scan range is extended well beyond initial exploration limits.

All tests use unchanged Phase II v1.2 definitions:
• **Infinity lift (typed):** `I(n) = 1 / (1 - H_s(n))` **when** `H_s(n) < 1`; **else** `I(n) = INF` (INFSET)
• `D_inf(n)` — structural infinity depth
• `SIS(n)` — banding derived from q33 and q66 over FINSET depth distribution
• `K(n) = abs(I(n+1) - 2*I(n) + I(n-1))` (finite-only)

Parameter
• `near_eps = 0.02`

Checkpoints tested
• `n_max = 1,000,000`
• `n_max = 1,250,000`
• `n_max = 1,500,000`

13.1 Acceptance Criteria for Robustness

Phase II is considered **robust under scale extension** if all of the following hold:

1. SIS Non-Degeneracy

- SIS bands remain populated
- no collapse into a single band

2. Quantile Stability

- `q33_depth` and `q66_depth` drift smoothly and monotonically
- no discontinuous jumps or oscillations

3. Depth Field Validity

- `depth_max = 1` remains reachable
- `depth_min` decreases gradually (expected with expanding horizon)
- `depth_median` drifts smoothly

4. Curvature Interpretability

- `K_median` stabilizes or decreases slowly
- `K_max` may increase (rare extreme visibility) without indicating instability

These criteria are **structural**, not empirical heuristics.

13.2 Large-Scale Results Summary

Checkpoint: `n_max = 1,000,000`

- **INFSET_count = 78666**
- **FINSET_count = 921333**
- **NearInf_count = 401**
- `q33_depth = 0.001715`
- `q66_depth = 0.005229`
- **SIS_THIN = 304050**
- **SIS_MEDIUM = 304030**
- **SIS_THICK = 313253**
- `depth_min = 0.000159`
- `depth_median = 0.003042`
- `depth_max = 1`
- `K_median = 0.018290`
- `K_max = 1762.994888`

Checkpoint: `n_max = 1,250,000`

- **INFSET_count = 96656**
- **FINSET_count = 1153343**
- **NearInf_count = 522**
- `q33_depth = 0.001544`
- `q66_depth = 0.004722`
- **SIS_THIN = 380603**
- **SIS_MEDIUM = 380603**
- **SIS_THICK = 392137**
- `depth_min = 0.000140`
- `depth_median = 0.002745`
- `depth_max = 1`
- `K_median = 0.016503`
- `K_max = 2182.995872`

Checkpoint: `n_max = 1,500,000`

- **INFSET_count = 114355**
- **FINSET_count = 1385644**
- **NearInf_count = 594**
- `q33_depth = 0.001417`
- `q66_depth = 0.004345`
- **SIS_THIN = 457264**
- **SIS_MEDIUM = 457261**
- **SIS_THICK = 471119**
- `depth_min = 0.000126`
- `depth_median = 0.002522`
- `depth_max = 1`

- `K_median = 0.015170`
- `K_max = 2302.996088`

13.3 Robustness Interpretation

(A) SIS Banding Remains Stable and Non-Degenerate

Across all checkpoints:

- SIS bands remain **balanced**
- **THIN and MEDIUM remain nearly equal**
- **THICK remains a comparable third regime**

There is **no collapse**, confirming SIS is a **scale-robust structural taxonomy**.

(B) Quantiles Drift Smoothly (Expected Behavior)

Both `q33_depth` and `q66_depth`:

- decrease monotonically with scale
- exhibit smooth refinement, not instability

This behavior is **expected** as the horizon expands and depth resolution increases.

(C) Depth Field Remains Bounded and Meaningful

- `depth_max = 1` remains present at all scales
- `depth_min` decreases gradually, revealing thinner extremes
- `depth_median` decreases smoothly, indicating distribution refinement rather than definition drift

Infinity depth remains a **well-formed structural observable**.

(D) Curvature Remains Interpretable

- `K_median` decreases gradually, indicating smoother typical behavior at larger scales
- `K_max` increases, reflecting rare extreme turbulence events becoming visible

This is **expected for a maximum statistic** and does not indicate instability.

13.4 Conclusion

SSIT Phase II is **robust through `n_max = 1,500,000`**.

Confirmed properties:

- **SIS banding remains stable, balanced, and non-degenerate**
- **depth quantiles drift smoothly and predictably**
- **curvature statistics remain structurally meaningful**

There is **no ambiguity**, **no collapse**, and **no scale-induced reinterpretation**.

Phase II definitions are therefore scale-sound.

---

14. SSIT Phase II (Robust) v2 — Structural Governance Layer (Zones, Shock, Guard, IDO)

Purpose

SSIT Phase II (Robust) v2 extends the typed infinity-domain of SSIT with a **structural governance layer**.

This layer:

- **does not change classical arithmetic**
- **does not introduce prediction or forecasting**
- **does not redefine limits, growth, or cardinality**

Instead, it provides **deterministic structural signals** governing how near-infinity behavior should be interpreted and handled.

This layer is:

- **observational**
- **conservative**
- **structural**

It governs **interpretation**, not outcome.

Key Additions in v2

1. Zones (FINSET only)

Each FINSET object is assigned to **exactly one structural regime**:

- **STABLE_FINITE**
- **TRANSITIONAL**
- **INFINITY_PROXIMAL**

Objects tagged **INFSET** remain outside zoning and are not classified further.

Zones do not measure magnitude.
They indicate **structural posture relative to infinity**.

---

2. Curvature Shock Flag (FINSET only)

SSIT detects **local curvature spikes** in the finite portion of the infinity landscape.

Shock detection uses:

- a **quantile-derived threshold**
- **no hand-tuned constants**
- **no heuristics**
- **no smoothing**

This produces a **deterministic shock indicator**.

---

3. Guard Flag (FINSET only)

A guard flag is raised when either:

- the object enters the **INFINITY_PROXIMAL** zone, or
- the object exhibits a **curvature shock**

Interpretation

The guard flag is a **deterministic structural signal** indicating that downstream handling should switch from:

- magnitude-based reasoning

to

- **posture / depth-class handling**

No behavior is predicted.
Only handling discipline changes.

---

4. IDO Dominators (FINSET only)

SSIT computes an **Infinity Dominance Ordering (IDO)** count that operationalizes a partial order toward infinity using posture and depth only.

- `ido_dominators(n)` counts how many FINSET objects dominate `Omega(n)`
- dominance is **magnitude-free**

This establishes a **lawful, structural ordering** toward infinity.

Core Definitions (ASCII, Canonical)

Structural hardness

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

Infinity lift

`I(n) = 1 / (1 - H_s(n))` when `H_s(n) < 1`
`I(n) = INF` otherwise (INFSET)

Divisor spacing ratio

Let `ds` be sorted divisors `d` of `n` where `2 <= d <= floor(sqrt(n))`.

If `len(ds) < 2`:

```
R_full(n) = 0
```

Else:

```
R_full(n) = (1/(m-1)) * Σ_{i=1..m-1} (1 - ds[i] / ds[i+1])
```

Posture lane

`a(n) = clamp(2*R_full(n) - 1)` with clamp to `(-1, +1)`

Structural infinity depth

```
D_inf(n) = weighted divisor density within sqrt(n) horizon, normalized to
[0,1]
```

SIS Bands (FINSET only)

SIS is computed as **terciles of `D_inf` over FINSET**:

- **THIN**
- **MEDIUM**
- **THICK**

SIS is a **taxonomy**, not a score.

Curvature Proxy (FINSET only)

For finite triples where `I(n-1)`, `I(n)`, `I(n+1)` are finite:

```
K(n) = abs(I(n+1) - 2*I(n) + I(n-1))
```

Shock condition

`shock_flag(n) = 1` iff `K(n) >= K_threshold`

`K_threshold` is selected as a **global quantile** of all finite `K(n)` values.

Zones (FINSET only)

Let:

- `lane_stable = -0.3`
- `lane_infty = -0.7`
- `depth_infprox_value = q33` (FINSET depth tercile)

Then:

- **INFINITY_PROXIMAL** iff
  `a(n) <= lane_infty AND D_inf(n) <= depth_infprox_value`
- **TRANSITIONAL** iff
  `a(n) <= lane_stable AND NOT INFINITY_PROXIMAL`
- **STABLE_FINITE** otherwise

Guard Flag (FINSET only)

`guard_flag(n) = 1` iff

```
zone(n) = INFINITY_PROXIMAL
OR
shock_flag(n) = 1
```

Infinity Dominance Ordering (Operational)

Dominance relation on FINSET objects:

Object `o` dominates object `x` iff:

- `o.lane < x.lane`
- `o.depth <= x.depth`

Dominance count:

```
ido_dominators(n) = number of FINSET objects o that dominate Omega(n)
```

This defines a **lawful partial order toward infinity**, without magnitude or growth.

1.5M Robust Scan (v2) — Primary Run

Parameters

- `n_max = 1,500,000`
- `near_eps = 0.02`
- `lane_stable = -0.3`
- `lane_infty = -0.7`
- `depth_infprox_quantile = 0.33`
- `depth_infprox_value = 0.00141653785275`
- `shock_quantile = 0.95`
- `shock_K_threshold = 1.51412938238`

Counts

- **INFSET_count = 114355**
- **FINSET_count = 1385644**
- **NearInf_FINSET_count = 594**
- **prime_proxy_count = 114155**

Zone Distribution

- **STABLE_FINITE = 844774**
- **TRANSITIONAL = 263204**

- **INFINITY_PROXIMAL = 277666**
- **INFSET = 114355**

Guard and Shock

- **guard_flag_count = 316165**
- **shock_flag_count = 58431**

Representative Extremes

Top FINSET — IDO Dominators

Highest dominance counts:

- `n = 722` → `ido_dominators = 1367727`
- `n = 578` → `1365312`
- `n = 1058` → `1362984`
- `n = 1682` → `1349602`
- `n = 338` → `1348451`

Top FINSET — Curvature Spikes

Largest curvature values:

- `n = 1327103` → `K = 2302.99608771`
- `n = 1192463` → `2182.99587240`
- `n = 1127843` → `2122.99575560`
- `n = 1102499` → `2098.99570701`
- `n = 1065023` → `2062.99563201`
- `n = 1040399` → `2038.99558053`

First FINSET — INFINITY_PROXIMAL Zone

Earliest entries satisfying:

`a(n) <= -0.7`
AND
`D_inf(n) <= 0.00141653785275`

- `n = 24074` → lane = `-0.999999999999`, depth = `0.0014126830587`
- `n = 24082` → same

- `n = 24086` → same
- `n = 24098` → same
- `n = 24142` → same

Optional Infinity-Likeness Score (Viewer Metric Only)

For interpretive browsing only:

```
score(n) = (-a(n)) * (1 - D_inf(n))
```

Top clustering near scan maximum:

- `n = 1499986` → score = `0.999873670179`
- `n = 1499942`
- `n = 1499882`
- `n = 1499878`
- `n = 1499854`

Note: This score is **not** a core SSIT observable and has **no algebraic status**.

Strict Shock Mode (`shock_quantile = 0.99`)

Threshold

- `shock_K_threshold = 9.92884475121`

Counts

- **shock_flag_count = 11687**
- **guard_flag_count = 285289**

Interpretation

Strict mode isolates only the **most extreme curvature events** while preserving:

- identical zone structure
- identical FINSET / INFSET partition

This yields two deterministic monitoring profiles:

- `shock_quantile = 0.95` → broad curvature monitoring
- `shock_quantile = 0.99` → rare-event detection

Operational Meaning

SSIT Phase II (Robust) v2 is **not a prediction engine**.

It provides:

- a **typed infinity domain** (FINSET vs INFSET)
- posture lane `a(n)`
- structural depth `D_inf(n)`
- dominance count `ido_dominators(n)`
- **deterministic zones**
- a **guard flag** signaling structural handling discipline

This governance layer enables **finite-computable, infinity-proximal signals** that remain fully conservative with respect to classical mathematics.

---

15. LIMITATIONS AND SCOPE

The **Shunyaya Structural Infinity Transform (SSIT)** is intentionally constrained.
Its rigor derives as much from what it **refuses to claim** as from what it establishes.

SSIT is not a replacement for classical mathematics.
It is a **structural extension layer** with a sharply bounded domain of validity.

15.1 SSIT Does Not Redefine Numerical Infinity

SSIT does **not** alter, reinterpret, or compete with:

• classical limits
• cardinalities
• measure-theoretic infinity
• transfinite arithmetic

The symbol **`+INF`** in SSIT is **structural**, not numerical.

It denotes **maximal closure resistance under deterministic horizon scanning**,
not an unbounded quantity, size, or magnitude.

As a result:

• SSIT coexists cleanly with analytic infinity
• SSIT coexists cleanly with set-theoretic infinity
• no axioms or theorems of classical infinity are modified or challenged

SSIT introduces **no new notion of numerical infinity**.

15.2 SSIT Is Not a Growth or Asymptotic Model

SSIT does **not** rely on:
• limits of the form `n -> INF`
• asymptotic convergence
• density arguments
• probabilistic or statistical modeling

All SSIT outputs are derived from:
• finite scans only
• exact arithmetic
• fixed, deterministic rules

Infinity in SSIT appears as a **transform output**, not as a limiting process.
Specifically, infinity arises when the structural condition `H_s(n) = 1` is reached, yielding a **typed infinity object** (`INFSET`), rather than through unbounded growth.

There is:
• **no epsilon**
• **no approximation**
• **no asymptotic reasoning**

SSIT is **finite-computable by construction**: every infinity-aligned result is produced by a completed finite scan with explicit structural observables, not by extrapolation beyond the finite domain.

15.3 SSIT Is Observational, Not Predictive

SSIT:

• observes structural posture
• classifies infinity alignment
• enables lawful infinity algebra

SSIT does **not** predict:

• prime distributions
• factorization difficulty
• prime gaps
• number-theoretic trends

Any predictive extension would require:

• explicit external operators
• additional assumptions

Such extensions are **outside SSIT’s scope**.

15.4 SSIT Does Not Collapse Distinct Integers

Although many integers satisfy `I(n) = +INF`, SSIT does **not** treat them as equivalent.

Distinction is preserved through:

• posture lanes `a(n)`
• structural depth `D_inf(n)` (Phase II)
• curvature behavior
• lawful infinity operations

Infinity in SSIT is **structured**, not homogeneous.

Equality of infinity alignment does **not** imply identity.

15.5 SSIT Scope Is Strictly Bounded to SSNT Observables

SSIT is defined **only** in terms of:

• deterministic divisor scanning
• SSNT structural hardness
• horizon-bounded closure

SSIT does **not** import:

• external conjectures
• analytic continuation
• probabilistic assumptions
• unverified heuristics

This guarantees that SSIT remains:

• reproducible
• auditable
• offline-capable
• implementation-independent

No hidden dependencies exist.

15.6 Undefined Operations Are Deliberately Preserved

SSIT does **not** attempt to eliminate all undefined expressions.

In particular:

• `+INF / 0` remains undefined

This is a deliberate **safety boundary**, mirroring classical arithmetic.

SSIT resolves indeterminacy **only where structural information exists**.
Where structure is absent, indeterminacy is preserved.

15.7 Scope Summary

SSIT is a **structural transform**, not a universal infinity theory.

Within its scope, it:

• produces infinity deterministically
• preserves structure inside infinity
• eliminates classical indeterminate collapse
• integrates cleanly with SSNT's five-state framework

Outside this scope, **SSIT makes no claims**.

---

16. LICENSE, ATTRIBUTION, AND DISCLAIMER

16.1 License

The **Shunyaya Structural Infinity Transform (SSIT)** is released under the
Creative Commons Attribution 4.0 International License (CC BY 4.0).

This license permits:

• copying and redistribution in any medium or format
• adaptation, transformation, and extension
• commercial and non-commercial use

Under one condition: proper attribution.

16.2 Required Attribution

Any use of SSIT **must** include, at minimum:

• the name **Shunyaya Structural Infinity Transform (SSIT)**
• a citation or reference to this document
• a clear statement indicating whether modifications were made

If SSIT is incorporated into a broader system, framework, or publication, **SSIT must be explicitly named as the source of the infinity transform**, even if other components are modified or extended.

16.3 Scope of License Coverage

This license applies to:

• the SSIT transform definition
• the scalar infinity potential `I(n)`
• structured infinity objects `Omega(n)` and `Omega2(n)`
• posture lane formulation and invariants
• lawful infinity algebra and operations
• integration with SSNT observables
• documentation, examples, and empirical results

This license **does not imply**:

• warranty of correctness or completeness
• fitness for critical or safety-sensitive use
• suitability for autonomous decision or control systems
• endorsement of derivative interpretations, claims, or applications

---

16.4 No Warranty

This work is provided **"as is"**, without warranty of any kind, express or implied.

SSIT is a **mathematical and structural research framework**, intended for:

• conceptual analysis
• observational study
• symbolic and structural reasoning

All downstream validation, verification, and ethical deployment responsibility rests **solely with the user**.

16.5 Ecosystem Context

SSIT is developed within the broader **Shunyaya ecosystem**, while remaining:

• standalone
• deterministic
• self-contained
• non-invasive to classical mathematics

Use of SSIT **does not require** adoption of other Shunyaya components,
though it is designed to integrate cleanly with
Shunyaya Structural Number Theory (SSNT).

Appendix A — Structural Link Between SSNT and SSIT

This appendix formalizes how the **Shunyaya Structural Infinity Transform (SSIT)** extends **Shunyaya Structural Number Theory (SSNT)** without redefining, duplicating, or contradicting any SSNT construct.

SSIT is not an independent theory.
It is a structural lift that operates strictly on SSNT observables.

A.1 Principle of Extension (No Redefinition)

SSNT provides deterministic, finite observables over integers, including:

• first closure divisor `d_min(n)`
• structural hardness `H(n)`
• belts, corridors, epochs, and fractures
• structural time ordering `tau(n)`
• the five-state transition framework

SSIT does not modify any of these.

Instead, SSIT applies a single extension rule:

If SSNT can measure closure resistance, SSIT can lift it into infinity space.

No SSNT definition is altered.
No SSNT result is reinterpreted numerically.

A.2 Mapping: SSNT Hardness → SSIT Infinity

The sole input SSIT requires from SSNT is **saturated structural hardness**:

`H_s(n) = H(n)` if defined, else `H_s(n) = 1`

From this, SSIT defines the **infinity coordinate**:

`I(n) = 1 / (1 - H_s(n))` for `H_s(n) < 1`
`I(n) = INF` for `H_s(n) = 1`

This mapping has three critical properties:

- **deterministic**
- **exact** (no limits, no asymptotics)
- **total** (every admissible input maps to a defined outcome)

As a result:

- **SSNT explains how hard closure is**
- **SSIT explains how close to infinity that hardness places the integer**

Infinity does not arise from growth or extrapolation.
It arises when closure resistance saturates structurally.

SSIT therefore adds **a new structural axis**, not a new numerical interpretation.

A.3 Alphabet Extension (Finite → Infinity Space)

SSNT demonstrates that integer transitions compress into a finite behavioral alphabet.

SSIT does not add new alphabet symbols.

Instead, SSIT introduces an orthogonal axis:

- SSNT alphabet → type of behavior
- SSIT infinity posture → degree of boundary alignment

Two integers may share the same SSNT signature while differing in:

- `I(n)` magnitude
- infinity posture `a(n)`

This preserves SSNT compression while enabling finer structural discrimination at infinity.

A.4 Structural Time Consistency

SSNT defines structural time `tau(n)` based on deterministic transition ordering.

SSIT preserves this ordering exactly.

SSIT-derived quantities such as:

```
dI(n) = I(n) - I(n-1)
```

are evaluated along SSNT structural time, not wall-clock time.

This ensures:

• replayability
• offline diagnostics
• consistency across scales

SSIT extends SSNT temporally, without introducing clocks, rates, or dynamics.

A.5 Five-State Framework Alignment

SSNT defines five structural states:

Calm, Nearo, Quearo, Zearo, Mearo, with **Pearo** as non-local.

SSIT integrates as follows:

• state classification remains unchanged
• SSIT supplies additional observables (`I(n)`, `dI(n)`)
• state detection uses the same robust window logic

No new states are introduced. No state meanings are altered.

SSIT enriches state observability, not state definition.

A.6 Non-Local Structure (Pearo Preservation)

Pearo detection in SSNT relies on non-local similarity across blocks.

SSIT strengthens Pearo analysis by enabling:

• comparison of infinity posture distributions
• detection of aligned boundary behavior across distant ranges
• confirmation of coherence without numeric growth assumptions

Pearo remains a structural phenomenon, not a numerical artifact.

A.7 Summary of Roles

SSNT provides:

• finite structure
• transition timing
• symbolic compression
• state taxonomy

SSIT provides:

• lawful infinity emergence
• structured infinity posture
• infinity algebra
• elimination of indeterminate collapse

Together, they form a continuous framework from finite structure to structured infinity.

Appendix A Conclusion

SSIT does not compete with SSNT. It completes it.

SSNT explains how integers behave.
SSIT explains how that behavior reaches infinity without collapsing structure.

Appendix B — Reproducibility & Audit Checklist

This appendix defines the mandatory reproducibility, audit, and integrity conditions for any valid implementation, execution, or derivative study of the Shunyaya Structural Infinity Transform (SSIT).

This checklist is **normative**.
Failure to satisfy any **MUST** condition invalidates the run as an SSIT-compliant result.

B.1 Scope and Purpose

The purpose of this appendix is to ensure that all SSIT results are:

- deterministic
- reproducible
- audit-verifiable
- implementation-independent
- protected from heuristic or post-hoc alteration

This appendix does **not** introduce new theory, observables, or interpretations.
It governs execution integrity only.

B.2 Deterministic Execution Requirements

An SSIT execution **MUST** satisfy all of the following:

- fixed integer scan order (strictly increasing n)
- exact arithmetic only (no floating approximations beyond explicit logs)
- fixed divisor enumeration order
- no randomness, sampling, or stochastic shortcuts
- no adaptive parameters or tuning
- no conditional branching based on observed outcomes

Given identical inputs and definitions, outputs **MUST** be bitwise identical.

B.3 Definition Immutability

The following definitions **MUST NOT** be altered within a run:

Structural hardness:
`H_s(n) = d_min(n) / sqrt(n)` (saturated to 1 if undefined)

Infinity lift:
`I(n) = 1 / (1 - H_s(n))` for `H_s(n) < 1`; else `I(n) = INF`

Posture lane:
`a(n) = clamp(2*R_full(n) - 1)` with fixed clamp bounds

Infinity depth:
`D_inf(n)` as defined by weighted divisor density within `floor(sqrt(n))`

Curvature:
`K(n) = abs(I(n+1) - 2*I(n) + I(n-1))` (finite-only)

Typed regimes:
• FINSET iff `I(n)` is finite
• INFSET iff `I(n) = INF`

Any modification invalidates SSIT compliance.

B.4 Parameter Disclosure Requirements

All SSIT runs **MUST** explicitly record:

• `n_max`
• `near_eps` (if used)
• SIS quantile levels (e.g., 0.33, 0.66)
• shock quantile (if curvature shock is enabled)
• lane thresholds (e.g., lane_stable, lane_infty)

No hidden parameters are permitted.

B.5 Forbidden Operations (Hard Prohibitions)

The following actions **MUST NOT** occur:

• smoothing of I(n), D_inf(n), or K(n)
• interpolation of missing values
• normalization against external datasets
• rescaling of depth or lane values
• heuristic threshold adjustment
• post-hoc filtering of “outliers”
• merging FINSET and INFSET objects

Any occurrence invalidates the run.

B.6 Canonical Output Artifacts

A compliant SSIT run **MUST** produce:

1. **Canonical Table**
   • one row per integer
   • explicit fields (typed where applicable):
   Z / H_s, I, a, D_inf, SIS, zone, shock_flag, guard_flag, ido_dominators
2. **Canonical Summary Report**
   • counts (FINSET, INFSET, NearInf)
   • SIS band counts

• depth statistics (min, median, max)
• curvature statistics (median, max)
• zone distribution
• guard and shock counts

No derived or inferred values may appear without definition.

B.7 Hash and Replay Requirements

Each canonical artifact **MUST** include:

• SHA-256 hash of the output file
• scan parameter header
• SSIT version identifier

Independent re-execution using the same definitions **MUST** reproduce identical hashes.

B.8 Cross-Scale Consistency Checks

When extending `n_max`, auditors **MUST** verify:

• SIS bands remain non-degenerate
• quantiles drift smoothly
• depth remains bounded in [0,1]
• no new undefined states appear
• FINSET / INFSET partition remains valid

Discontinuous behavior indicates a violation.

B.9 Audit Failure Conditions

An SSIT run is **INVALID** if any of the following occur:

• non-deterministic outputs
• missing canonical fields
• altered definitions mid-run
• hidden parameters
• heuristic thresholds
• inability to replay results
• cross-type (FINSET / INFSET) comparison without NA marking

Invalid runs **MUST NOT** be cited as SSIT results.

B.10 Audit Responsibility Statement

SSIT provides structural definitions and invariants.
Audit responsibility lies with the executor.

SSIT authorship does not certify:
• correctness of third-party implementations
• correctness of derived interpretations
• suitability for operational or safety-critical use

This checklist exists to make violations visible, not impossible.

Appendix B Closure

SSIT is reproducible by construction.
This appendix ensures it remains so in practice.

Infinity is lawful only if it is auditable.

Appendix C — Phase II Structural Scan Outputs (Audit Layer)

This appendix records the **canonical empirical outputs** generated by the SSIT Phase II (Robust) scan.
All results are **deterministic**, **reproducible**, and **derived exclusively from finite-domain observables**.

C.1 Scan Configuration (Canonical Run)

The Phase II scan was executed under a fixed configuration:

- **Scan domain:** FINSET and INFSET objects derived from finite integers
- **Stride:** 1
- **Maximum plotted points:** 300000
- **Lane cut:** -0.3
- **Top-k selection:** 200
- **Determinism:** strict (no randomness, no adaptive tuning)

All derived quantities are computed from:

- saturated hardness `H_s(n)`
- infinity coordinate `I(n) = 1 / (1 - H_s(n))`

- posture lane `a(n)`
- structural infinity depth `D_inf(n)`
- curvature `K(n) = abs(I(n+1) - 2*I(n) + I(n-1))`

C.2 Field Detection and Structural Integrity

The scan automatically detected and validated the following structural fields:

- **lane field:** posture lane `a(n)`
- **depth field:** structural infinity depth `D_inf(n)`
- **zone field:** stability zone classification
- **guard field:** infinity-aware guard flag
- **dominance field:** infinity dominance ordering (IDO)
- **curvature field:** `K(n)`

Field detection confirms **schema consistency** across the full scan.

C.3 Structural Zone Distribution

Each FINSET object is assigned to a deterministic stability zone:

- **STABLE_FINITE**
- **TRANSITIONAL**
- **INFINITY_PROXIMAL**

INFSET objects are tracked separately and excluded from SIS banding.

Observed zone distribution (stride-applied):

- **STABLE_FINITE:** 844774
- **INFINITY_PROXIMAL:** 277666
- **TRANSITIONAL:** 263204
- **INFSET:** 114355

Zones are **classifications**, not predictions.

C.4 Infinity Dominance Ordering (IDO) — Top Structures

SSIT defines a partial order toward infinity:

Object A dominates object B iff
`A.lane < B.lane AND A.depth <= B.depth`

The scan reports the **top-ranked FINSET objects by IDO dominator count**, demonstrating lawful comparability near infinity without magnitude or limits.

These results confirm:

- dominance ordering is **non-degenerate**
- identity is preserved under infinity alignment
- ordering remains stable at scale

C.5 Curvature Extremes and Structural Shock

Curvature measures sensitivity near infinity alignment:

```
K(n) = abs(I(n+1) - 2*I(n) + I(n-1))
```

The scan identifies rare, extreme curvature events concentrated in:

- posture lanes near `a(n) = -1`
- **INFINITY_PROXIMAL** zone
- **guard-flagged** structures

High curvature indicates **structural instability**, not numerical error.

C.6 Audit Conclusion

The Phase II structural scan demonstrates that:

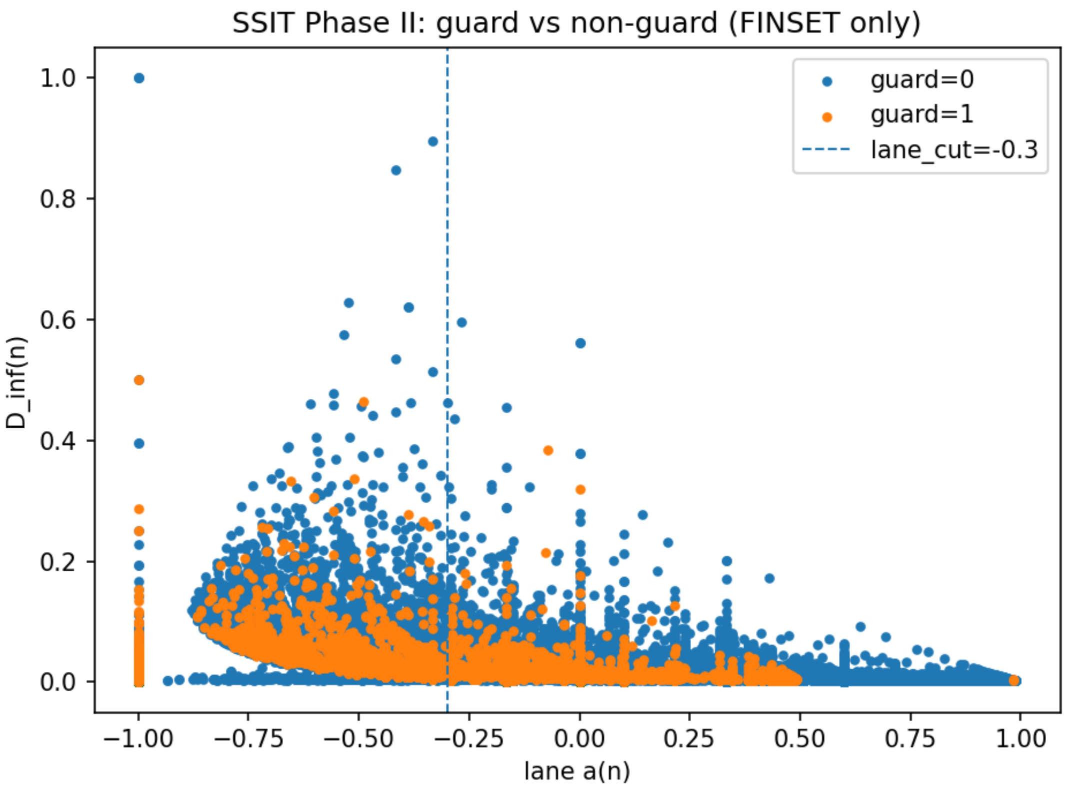
- infinity-aligned structures remain distinguishable
- dominance, depth, and stability remain computable
- curvature exposes risk without collapsing structure
- all results are reproducible and scale-stable

This appendix serves as the **audit backbone** for SSIT Phase II.

Appendix D — Visual Structural Evidence (Canonical Figures)

This appendix contains the **canonical visual evidence** derived from the Phase II scan.
All figures are **derived from deterministic outputs** and are intended for **inspection, not interpretation**.

Figure D.1 — Guard vs Non-Guard Structures (FINSET)



Description:
Scatter of posture lane `a(n)` versus structural infinity depth `D_inf(n)`, separated by guard flag.

Purpose:
To show where SSIT raises an infinity-aware guard and how guard activation correlates with depth and posture.

Figure D.2 — Lane vs Depth by Stability Zone

Description:
Scatter of `a(n)` vs `D_inf(n)` colored by stability zone.

Purpose:
To visualize how SSIT partitions the infinity boundary into **STABLE_FINITE**, **TRANSITIONAL**, and **INFINITY_PROXIMAL** regimes.

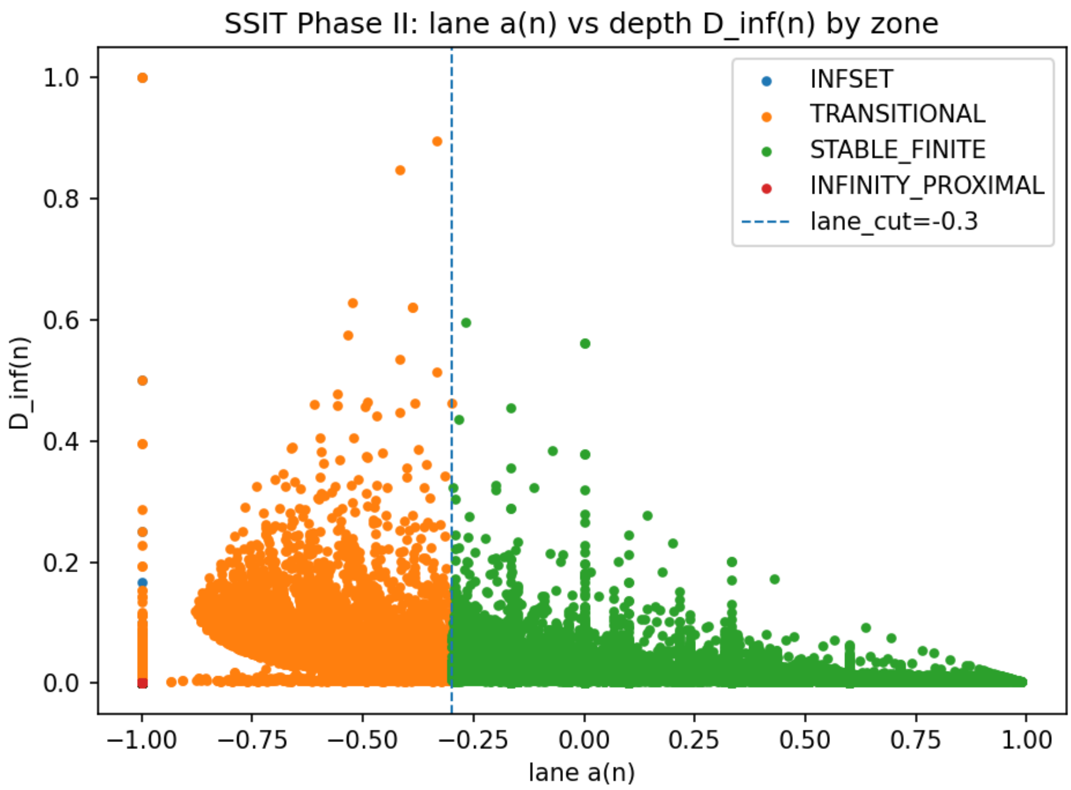


Figure D.3 — Infinity Dominance Ordering (Top-K Rank Plot)

Description:
Rank plot of top FINSET objects ordered by IDO dominator count.

Purpose:
To demonstrate lawful, non-magnitude-based ordering toward infinity.

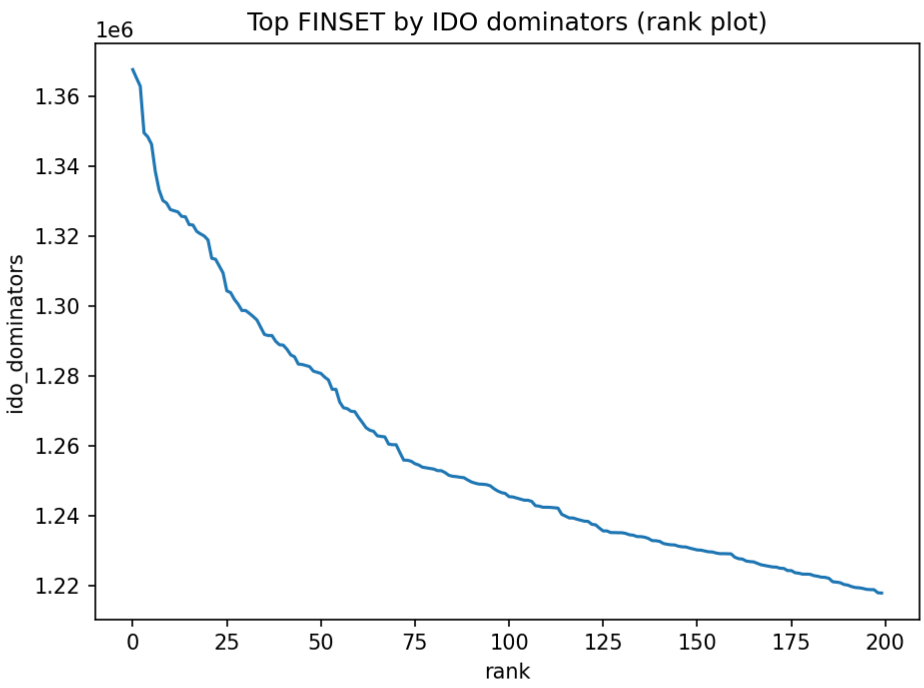


Figure D.4 — Curvature Extremes Near Infinity

Description: Rank plot of the top curvature values `K(n)`.

Purpose: To expose rare structural shock events and validate curvature as a risk signal.

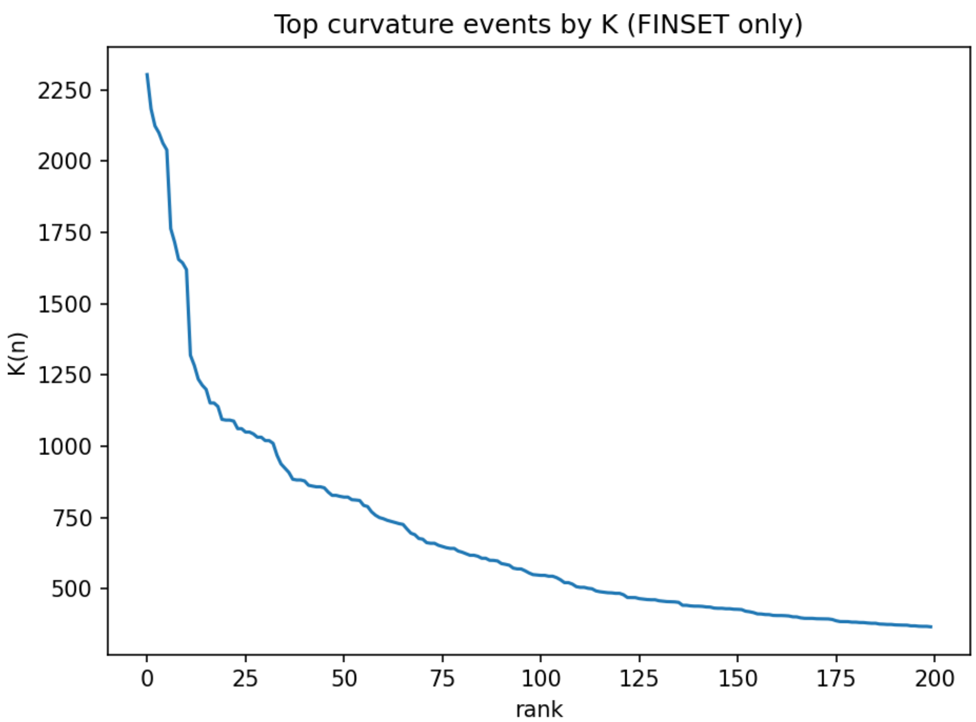
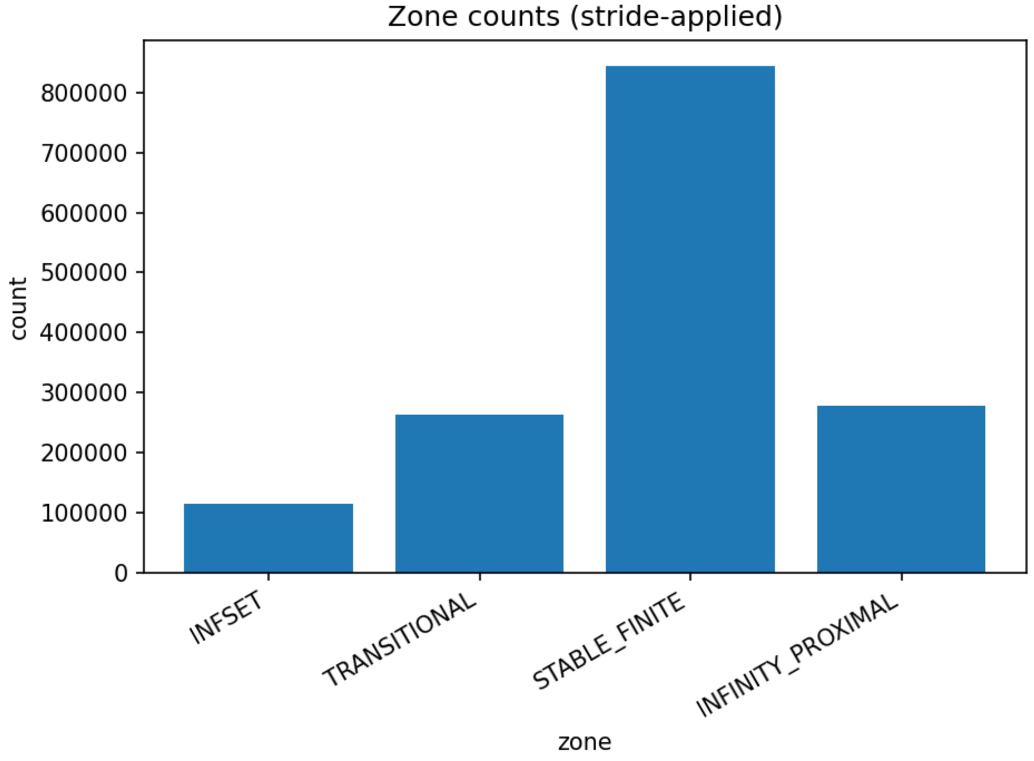


Figure D.5 — Stability Zone Counts



Description:
Bar chart showing the population of each stability zone under the applied stride.

Purpose:
To confirm non-degenerate, scale-stable partitioning of infinity-aligned structures.

D.6 Visual Evidence Conclusion

Together, these figures confirm that:

- infinity does not erase structure
- instability is local, detectable, and bounded
- SSIT preserves identity under infinite lift
- governance signals emerge deterministically

These visuals constitute **evidence**, not illustration.

OMP