RSCS-Q Booklet 1

Symbolic Metrics for Cognitive Systems

Foundational Measures of Entropic Dynamics and Coherence

Entropica Research Collective

research@entropica.org

Version 2.1 — November 2025

Keywords: Symbolic Metrics, Entropic Dynamics, Coherence Index, Integrated Information, Drift Detection, Cognitive Systems, Autonomous Agents, State Observation, EDR, SOC, VSI, Φ

Supplementary Materials: https://github.com/entropica/rscsq

Draft for circulation and publication review

v2.1: Added testing protocols, cross-booklet integration, symbolic index

Abstract

This paper introduces the foundational **symbolic metrics** for the RSCS-Q (Reflex-Symbolic Cognitive System - Quantified) framework. We define four core metrics that characterize cognitive system state: **Entropic Drift Rate (EDR)** measuring state space exploration velocity, **Symbolic Observation Coherence (SOC)** quantifying pattern consistency across observations, **Variance Stability Index (VSI)** tracking distributional stability, and **Integrated Information (Φ)** capturing system-level integration following IIT principles.

These metrics form the observational substrate for higher-level governance mechanisms (Booklets 2–5). We establish mathematical foundations, define measurement protocols, prove key properties including EDR boundedness and SOC transitivity, and provide simulation protocols for validation. The metrics enable real-time monitoring of cognitive system health and provide early warning signals for drift and decoherence.

Keywords: Symbolic Metrics, Entropic Dynamics, Coherence, Integrated Information, Drift Detection

Contents

1 Introduction

Autonomous cognitive systems require quantitative measures of their internal state to enable governance and self-correction. This paper introduces four foundational metrics:

1. **Entropic Drift Rate (EDR)**: Velocity of state space exploration
2. **Symbolic Observation Coherence (SOC)**: Pattern consistency
3. **Variance Stability Index (VSI)**: Distributional stability
4. **Integrated Information (Φ)**: System integration

1.1 Design Principles

Axiom 1.1 (Observability). *All metrics must be computable from observable system outputs without requiring access to internal model weights or hidden states.*

Axiom 1.2 (Boundedness). *All metrics must have bounded ranges to enable threshold-based governance.*

Axiom 1.3 (Composability). *Metrics must compose across subsystems and time windows.*

1.2 Document Organization

Sections 2–5 define each metric formally. Section 6 establishes inter-metric relationships. Section 7 specifies measurement protocols. Section 8 maps metrics to downstream booklets. Section 9 provides simulation and validation procedures.

2 Entropic Drift Rate (EDR)

2.1 Metric Summary

EDR at a Glance

Purpose: Measure velocity of state space exploration
Range: $(-\log_2 n, +\log_2 n)$ bits/step
Interpretation: Positive = exploring, Negative = converging, Zero = stable
Downstream: RCI computation (B2), DriftL2 guard (B3), Fork entropy (B4)

2.2 Definition

Definition 2.1 (State Entropy). *For a discrete state distribution $P = \{p_1, \ldots, p_n\}$ over n states:*

$$H(P) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{1}$$

with $0 \log 0 = 0$ by convention. Units: bits.

Definition 2.2 (Entropic Drift Rate). *The rate of entropy change over a time window $[t, t + \Delta t]$:*

$$EDR(t) = \frac{H(P_{t+\Delta t}) - H(P_t)}{\Delta t} \tag{2}$$

Units: bits per time step.

2.3 Interpretation

- EDR > 0: System exploring (entropy increasing)
- EDR $= 0$: System stable (entropy constant)
- EDR < 0: System converging (entropy decreasing)

Real-World Analog: In a learning system, positive EDR indicates exploration of new hypotheses (akin to high-temperature search), while negative EDR indicates convergence toward a solution (exploitation). Sustained high positive EDR may signal overfitting avoidance or instability; sustained negative EDR may indicate mode collapse.

Proposition 2.3 (EDR Bounds). *For a system with n states:*

$$-\log_2 n \leq EDR \cdot \Delta t \leq \log_2 n \tag{3}$$

Proof. Entropy ranges from 0 (single state with $p = 1$) to $\log_2 n$ (uniform distribution). Maximum change per step is the full range. □ □

2.4 Smoothed EDR

For stability in noisy environments, use exponentially weighted moving average:

$$\overline{\text{EDR}}_t = \alpha \cdot \text{EDR}_t + (1 - \alpha) \cdot \overline{\text{EDR}}_{t-1} \tag{4}$$

with smoothing factor $\alpha \in (0, 1]$, typically $\alpha = 0.1$.

3 Symbolic Observation Coherence (SOC)

3.1 Metric Summary

SOC at a Glance

Purpose: Quantify pattern consistency across observations
Range: $[-1, 1]$ (cosine similarity)
Interpretation: High = consistent patterns, Low = decoherence
Downstream: RCI coherence component (B2), HashAgree (B3), κ_t (B4)

3.2 Definition

Definition 3.1 (Observation Similarity). *For observations o_i, o_j with feature vectors $\mathbf{f}_i, \mathbf{f}_j$:*

$$sim(o_i, o_j) = \frac{\mathbf{f}_i \cdot \mathbf{f}_j}{\|\mathbf{f}_i\| \cdot \|\mathbf{f}_j\|} \tag{5}$$

(Cosine similarity, range $[-1, 1]$)

Definition 3.2 (Symbolic Observation Coherence). *For a window of k observations $\{o_1, \ldots, o_k\}$:*

$$SOC = \frac{2}{k(k-1)} \sum_{i<j} sim(o_i, o_j) \tag{6}$$

(Mean pairwise similarity, range $[-1, 1]$)

3.3 Interpretation

Real-World Analog: In a perception system, high SOC indicates that sequential observations are internally consistent (e.g., tracking a single object). Low SOC indicates perceptual drift or classification ambiguity (e.g., oscillating between incompatible interpretations). Negative SOC indicates anti-correlated observations.

Table 1: SOC Interpretation Guide

Range	Interpretation	Action
$[0.8, 1.0]$	High coherence	Normal operation
$[0.5, 0.8)$	Moderate coherence	Monitor
$[0.2, 0.5)$	Low coherence	Alert
$[-1.0, 0.2)$	Decoherence	Intervene

3.4 Properties

Proposition 3.3 (SOC Transitivity). *If $SOC_{AB} \geq \theta$ and $SOC_{BC} \geq \theta$ for threshold $\theta > 0.5$, then:*

$$SOC_{AC} \geq 2\theta - 1 \tag{7}$$

Proof. By triangle inequality on angular distance. For high similarity ($\theta > 0.5$), the angle between A and C is bounded by the sum of angles A-to-B and B-to-C. See Appendix A for full derivation. □ □

4 Variance Stability Index (VSI)

4.1 Metric Summary

VSI at a Glance

Purpose: Track distributional stability over time
Range: $[0, 1]$ (normalized)
Interpretation: High = stable variance, Low = unstable
Downstream: RCI stability weight (B2), MTTR bound (B3), RA stability (B4)

4.2 Definition

Definition 4.1 (Variance Stability Index). *For a metric M with variance σ_t^2 at time t:*

$$VSI(t) = 1 - \frac{|\sigma_t^2 - \sigma_{t-1}^2|}{\max(\sigma_t^2, \sigma_{t-1}^2, \epsilon)} \tag{8}$$

where $\epsilon > 0$ prevents division by zero (default $\epsilon = 10^{-8}$). Range: $(-\infty, 1]$ with 1 indicating perfect stability.

4.3 Normalized VSI

For bounded range $[0, 1]$:

$$\text{VSI}_{\text{norm}} = \max(0, \text{VSI}) \tag{9}$$

4.4 Multi-Metric VSI

For a vector of metrics $\mathbf{M} = (M_1, \ldots, M_m)$:

$$\text{VSI}_{\text{agg}} = \frac{1}{m} \sum_{i=1}^{m} w_i \cdot \text{VSI}(M_i) \tag{10}$$

with weights $\sum_i w_i = 1$.

Real-World Analog: In control systems, VSI measures whether the system's operating variance remains predictable. A control loop with stable VSI produces consistent outputs; unstable VSI indicates oscillation or chaotic behavior requiring damping.

5 Integrated Information (Φ)

5.1 Metric Summary

Φ at a Glance

Purpose: Capture system-level integration and holistic processing
Range: $[0, \infty)$ bits, normalized to $[0, 1]$
Interpretation: High = integrated system, Low = modular/decomposable
Downstream: RCI integration score (B2), Swarm Φ (B4)

5.2 Background

Integrated Information Theory (IIT) proposes that consciousness corresponds to integrated information, denoted Φ. We adapt this concept for cognitive system coherence measurement.

5.3 Simplified Φ Computation

Definition 5.1 (Partition Information Loss). *For system S partitioned into subsystems (A, B):*

$$PIL(A, B) = I(S) - I(A) - I(B) - I(A; B) \tag{11}$$

where $I(\cdot)$ denotes mutual information.

Definition 5.2 (Integrated Information).

$$\Phi = \min_{partitions} PIL(partition) \tag{12}$$

The minimum information lost across all bipartitions.

5.4 Practical Approximation

Full Φ computation is NP-hard. We use a linear approximation:

$$\Phi_{\text{approx}} = \frac{1}{n} \sum_{i=1}^{n} I(S_i; S \setminus S_i) \tag{13}$$

where S_i is subsystem i and $S \setminus S_i$ is the rest of the system.

5.5 Interpretation

- $\Phi \approx 0$: System is modular/decomposable (independent subsystems)
- $\Phi > 0$: System has integrated structure (subsystems interdependent)
- High Φ: Strong interdependence (holistic processing required)

Real-World Analog: A multi-agent system with high Φ behaves as a unified entity where agent decisions depend on global state. A system with low Φ can be safely decomposed into independent modules for parallel processing or fault isolation.

6 Metric Relationships

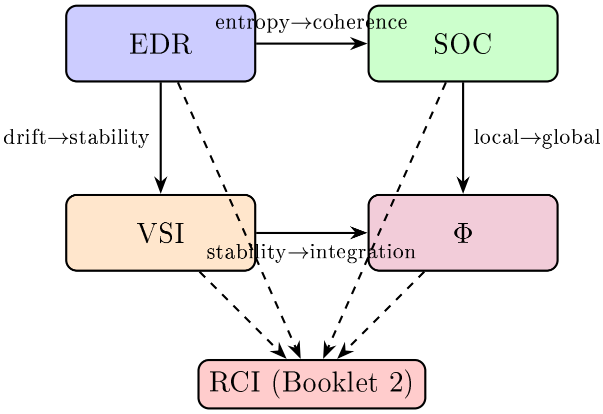

Figure 1: Metric Relationships and Flow to RCI

Proposition 6.1 (Coherence-Entropy Trade-off). *Under typical dynamics:*

$$EDR \cdot SOC \leq C \tag{14}$$

for some constant C. High exploration (positive EDR) typically reduces coherence (SOC).

7 Measurement Protocol

7.1 Sampling Requirements

Table 2: Sampling Parameters

Metric	Window	Frequency	Latency
EDR	10–100 steps	Every step	<10ms
SOC	5–20 observations	Every observation	<50ms
VSI	10–50 steps	Every step	<10ms
Φ	Full system	Every 100 steps	<1s

7.2 Threshold Configuration

Listing 1: Default Metric Thresholds

```
# Drift detection
EDR_THRESHOLD_HIGH = 0.5     # bits/step - excessive exploration
EDR_THRESHOLD_LOW = -0.5     # bits/step - excessive convergence

```

```
# Coherence thresholds
SOC_THRESHOLD_WARN = 0.5     # moderate decoherence
SOC_THRESHOLD_CRIT = 0.2     # critical decoherence

# Stability thresholds
VSI_THRESHOLD = 0.7          # minimum acceptable stability

# Integration threshold
PHI_THRESHOLD = 0.3          # minimum integration
PHI_MAX = 1.0                # normalization ceiling
```

8 Use in RSCS-Q Stack

Cross-Booklet Reference

Each metric feeds into higher-level mechanisms across the RSCS-Q stack. Table 3 summarizes these dependencies.

Table 3: Metric Usage Across Booklets

Metric	B2 (Capsule)	B3 (RSG)	B4 (Swarm)	B5 (ADM)
EDR	Drift input to RCI	DriftL2 guard	Fork entropy	Drift gauge
SOC	Coherence component	HashAgree check	κ_t computation	Coherence chart
VSI	Stability weight	MTTR bound	RA stability	Stability indicator
Φ	Integration score	(indirect)	Swarm Φ	Integration panel

8.1 Detailed Cross-Booklet Integration

Table 4: Metric Integration Details

Booklet	Integration Mechanism
B2: Capsule	$\mathrm{RCI} = w_1 f(\mathrm{EDR}) + w_2\mathrm{SOC} + w_3\mathrm{VSI} + w_4\Phi_{\mathrm{norm}}$ with default weights $(0.25, 0.35, 0.25, 0.15)$
B3: RSG	DriftL2 guard: RCI < 0.55 for $2W$ steps triggers D1 state. VSI used for MTTR_{95} bounding.
B4: Swarm	Fork entropy $S_{\mathrm{fork}} = -\sum p_i \log_2 p_i$ uses EDR formulation. SOC feeds κ_t coherence factor.
B5: ADM	Dashboard displays all four metrics in real-time panels with configurable thresholds.
Capstone	Autonomy Yield (AY) aggregates all metrics: $\mathrm{AY} = 0.25(\mathrm{Recovery} + \mathrm{Latency} + \mathrm{Coherence} + \mathrm{Fidelity})$

9 Simulation and Testing Protocol

This section provides validation procedures for metric implementations.

9.1 Synthetic Test Traces

Listing 2: Test Trace Generation

```
import numpy as np
from typing import List, Tuple

def generate_stable_trace(n_steps: int = 100) -> List[np.ndarray]:
    """Generate trace with stable entropy (EDR ~ 0)."""
    base_dist = np.array([0.25, 0.25, 0.25, 0.25])
    return [base_dist + np.random.normal(0, 0.01, 4)
            for _ in range(n_steps)]

def generate_exploring_trace(n_steps: int = 100) -> List[np.ndarray]:
    """Generate trace with increasing entropy (EDR > 0)."""
    traces = []
    for i in range(n_steps):
        # Gradually move toward uniform distribution
        alpha = i / n_steps
        dist = (1 - alpha) * np.array([0.7, 0.1, 0.1, 0.1]) + \
               alpha * np.array([0.25, 0.25, 0.25, 0.25])
        traces.append(dist)
    return traces

def generate_converging_trace(n_steps: int = 100) -> List[np.ndarray
    ]:
    """Generate trace with decreasing entropy (EDR < 0)."""
    traces = []
    for i in range(n_steps):
        # Gradually move toward concentrated distribution
        alpha = i / n_steps
        dist = (1 - alpha) * np.array([0.25, 0.25, 0.25, 0.25]) + \
               alpha * np.array([0.9, 0.03, 0.03, 0.04])
        traces.append(dist)
    return traces
```

9.2 Expected Values

Table 5: Expected Test Results

Trace Type	EDR	SOC	VSI
Stable	$\approx 0 \pm 0.05$	> 0.9	> 0.95
Exploring	> 0.3	0.5–0.8	0.7–0.9
Converging	< -0.3	0.6–0.9	0.7–0.9
Chaotic	$\lvert\text{EDR}\rvert > 0.5$	< 0.3	< 0.5

9.3 Threshold Calibration

Listing 3: Anomaly Detection Calibration

```
def calibrate_edr_threshold(traces: List, percentile: float = 95):
    """
    Calibrate EDR anomaly threshold from baseline traces.

    Args:
        traces: List of normal operation traces
        percentile: Percentile for threshold (default 95th)

```

```
    Returns:
        (low_threshold, high_threshold) tuple
    """
    edr_values = []
    for trace in traces:
        for i in range(1, len(trace)):
            h_prev = entropy(trace[i-1])
            h_curr = entropy(trace[i])
            edr_values.append(h_curr - h_prev)

    low = np.percentile(edr_values, 100 - percentile)
    high = np.percentile(edr_values, percentile)
    return (low, high)
```

9.4 Integration Test

Listing 4: Full Metric Pipeline Test

```
def test_metric_pipeline():
    """Integration test for all B1 metrics."""
    # Generate test trace
    trace = generate_stable_trace(100)

    # Compute metrics
    edr = compute_edr(trace, window=10)
    soc = compute_soc(trace, window=5)
    vsi = compute_vsi(trace, window=10)
    phi = compute_phi_approx(trace)

    # Assertions
    assert -0.1 < edr < 0.1, f"EDR out of range: {edr}"
    assert soc > 0.8, f"SOC too low: {soc}"
    assert vsi > 0.9, f"VSI too low: {vsi}"
    assert phi >= 0, f"Phi negative: {phi}"

    print("All B1 metric tests passed.")
    return {'edr': edr, 'soc': soc, 'vsi': vsi, 'phi': phi}
```

9.5 Reference to Harness Code

Cross-Booklet Reference

For production testing, see Booklet 2 harness implementations:

- `compute_slos.py`: SLO computation with metric inputs
- `dsl_harness.py`: DSL evaluation using B1 metrics
- `test_booklet1.py`: Unit tests for all metric functions

10 Related Work

Entropy-based metrics draw on information theory Shannon (1948). Integrated Information Theory provides the foundation for Φ Tononi (2004). Coherence measures relate to phase synchronization in neural systems Varela et al. (2001). The VSI concept relates to stationarity testing in time series analysis Cover & Thomas (2006).

11 Conclusion

This paper established four foundational metrics for RSCS-Q:

1. **EDR**: Entropic drift rate for exploration/convergence detection
2. **SOC**: Symbolic observation coherence for pattern consistency
3. **VSI**: Variance stability index for distributional stability
4. Φ: Integrated information for system-level coherence

These metrics satisfy the design axioms (Observability, Boundedness, Composability) and feed into the Reflex Coherence Index (RCI) defined in Booklet 2, enabling threshold-based governance in Booklets 3–5.

Acknowledgements

We thank the IIT research community for foundational work on integrated information, and colleagues for feedback on threshold calibration methodology.

References

Shannon, C.E. A Mathematical Theory of Communication. Bell System Technical Journal, 27(3):379–423, 1948.

Tononi, G. An Information Integration Theory of Consciousness. BMC Neuroscience, 5(42), 2004.

Varela, F., Lachaux, J.P., Rodriguez, E., Martinerie, J. The Brainweb: Phase Synchronization and Large-Scale Integration. Nature Reviews Neuroscience, 2(4):229–239, 2001.

Cover, T.M., Thomas, J.A. Elements of Information Theory. Wiley-Interscience, 2nd edition, 2006.

Oizumi, M., Albantakis, L., Tononi, G. From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0. PLOS Computational Biology, 10(5):e1003588, 2014.

A Mathematical Derivations

A.1 EDR Bound Derivation

From Definition 2.2:

$$\begin{aligned} \mathrm{EDR} \cdot \Delta t &= H(P_{t+\Delta t}) - H(P_t) && (15) \\ &\leq \max H - \min H && (16) \\ &= \log_2 n - 0 = \log_2 n && (17) \end{aligned}$$

Similarly for the lower bound. $\square$

A.2 SOC Transitivity Derivation

Using angular representation where $\text{sim}(a, b) = \cos(\theta_{ab})$:

If $\cos(\theta_{AB}) \geq \theta$ and $\cos(\theta_{BC}) \geq \theta$, then $\theta_{AB} \leq \arccos(\theta)$ and $\theta_{BC} \leq \arccos(\theta)$.

By triangle inequality: $\theta_{AC} \leq \theta_{AB} + \theta_{BC} \leq 2\arccos(\theta)$.

Therefore: $\cos(\theta_{AC}) \geq \cos(2\arccos(\theta)) = 2\theta^2 - 1 \geq 2\theta - 1$ for $\theta > 0.5$. □

B Glossary

EDR (Entropic Drift Rate)
Velocity of state space exploration measured in bits/step. Positive values indicate system is exploring (entropy increasing); negative values indicate convergence. Key input to DriftL2 guard (B3) and fork detection (B4).

SOC (Symbolic Observation Coherence)
Mean pairwise similarity across observations in a window, range $[-1, 1]$. Measures pattern consistency; low SOC indicates decoherence requiring intervention. Feeds RCI coherence component (B2) and κ_t (B4).

VSI (Variance Stability Index)
Distributional stability measure in range $[0, 1]$. VSI=1 indicates perfect variance stability; low VSI indicates oscillation or chaotic behavior. Used for MTTR bounding (B3) and RA stability (B4).

Φ (Integrated Information)
System-level integration metric derived from IIT. High Φ indicates holistic processing where subsystems are interdependent; low Φ indicates modular/decomposable system. Normalized to $[0, 1]$ for RCI computation.

Entropy
Shannon entropy $H(P) = -\sum p_i \log_2 p_i$ measured in bits. Maximum entropy (uniform distribution) indicates maximum uncertainty; zero entropy indicates deterministic state.

Coherence
Consistency of patterns across observations. High coherence enables reliable prediction; low coherence indicates system instability or phase transition.

Drift
Change in system state distribution over time. Detected via EDR thresholds; sustained drift triggers governance responses (D1 state in B3).

C Symbolic Index

Table 6: Symbol Reference

Symbol	Definition	Reference
$H(P)$	Shannon entropy	Eq. 1
EDR	Entropic Drift Rate	Def. 2.2
$\overline{\text{EDR}}$	Smoothed EDR (EWMA)	Eq. 4
SOC	Symbolic Observation Coherence	Def. 3.2
$\text{sim}(o_i, o_j)$	Cosine similarity	Eq. 5
VSI	Variance Stability Index	Def. 4.1
Φ	Integrated Information	Def. 5.2
Φ_{approx}	Linear approximation of Φ	Eq. 13
PIL	Partition Information Loss	Def. 5.1
α	EWMA smoothing factor	Eq. 4
ϵ	Division-by-zero guard	Def. 4.1
W	Window size parameter	Table 2
n	Number of states	Prop. 2.3
k	Observation window size	Def. 3.2
θ	Similarity threshold	Prop. 3.3

Table 7: Term Cross-Reference

Term	Sections
Coherence	3, 6, B
Convergence	2, 9
Decoherence	3, Table 1
Drift	2, 8, B
Entropy	2, 5, A
Exploration	2, 9
Governance	1, 8
Integration	5, 6
Stability	4, 7
Threshold	7, 9