Technical Note

A Reproducible Kernel-Damping Evidence Protocol for SORT-AI Core-3 Structural Coupling Regimes

Gregor Wegener *

Posted Date: 28 May 2026

doi: 10.20944/preprints202605.1992.v1

Keywords: SORT-AI; kernel damping; structural diagnostics; evidence protocol; reproducibility; runtime control coherence; interconnect stability; agentic system stability; MOCK v4; structural assessment; Core-3; risk-transition analysis

 doi:10.20944/preprints202605.1992.v1

Technical Note

A Reproducible Kernel-Damping Evidence Protocol for SORT-AI Core-3 Structural Coupling Regimes

Gregor Herbert Wegener

Independent Research & Systems Modeling, Friedrichstrasse 4, 10969 Berlin, Germany;
gregor.wegener@independent-research-systems-modeling.com

Abstract

This technical note introduces a reproducible kernel-damping evidence protocol for the SORT-AI Core-3 applications AI.01 (Interconnect Stability Control), AI.04 (Runtime Control Coherence), and AI.13 (Agentic System Stability). These applications span complementary structural coupling regimes in advanced AI systems: physical/interconnect coupling, logical/runtime-control coupling, and semantic/agentic coupling. The protocol evaluates whether declared structural risk-transition scenarios admit a Gaussian kernel-damping reconstruction under the declared canonical SORT scale parameter $\sigma_0 = 0.00190643$. The analysis is restricted to the structural analysis layer and does not claim production deployment, vendor-specific measurement, empirical benchmarking, runtime optimization, or execution by MOCK v4. MOCK v4 is treated as the frozen structural reference architecture, not as a runtime engine. The accompanying archived evidence release contains machine-readable scenario inputs, declared risk-transformation rules, executable scripts, expected outputs, generated outputs, and a reproduction manifest sufficient to reproduce all reported κ, ξ, scenario-level means, sample dispersions, and coefficients of variation. The contribution is methodological: the note formalizes a reproducibility protocol through which SORT-AI Core-3 applications can be tested as structurally defined damping regimes without converting MOCK v4 into an execution environment or introducing a new MOCK version.

Keywords: SORT-AI; kernel damping; structural diagnostics; evidence protocol; reproducibility; runtime control coherence; interconnect stability; agentic system stability; MOCK v4; structural assessment; Core-3; risk-transition analysis

1. Introduction

Advanced AI systems are increasingly realized as coupled execution structures rather than isolated model instances. Their observable behaviour arises across interconnect fabrics, runtime-control layers, schedulers, orchestration surfaces, memory paths, policy boundaries, and agentic execution chains. In such systems, local component correctness does not by itself guarantee global stability, reproducibility, or cost coherence. Established analyses of large-scale datacenter systems, tail behaviour, foundation-model deployment, and AI evaluation limits show that component-level metrics remain necessary but are often insufficient for explaining system-level behaviour under scale [6,7,21–24,29,30,33].

This creates a methodological requirement that is distinct from production validation. A structural diagnostic framework can define problem classes and analytical vocabulary, but such a framework requires a reproducible evidence layer if its internal calculations are to be inspected, repeated, and bounded. Computational reproducibility has become a central requirement for scientific artefacts, especially when claims depend on declared data transformations, executable scripts, software environments, and archived outputs [35,36,38–40]. This requirement is also relevant for machine-learning and AI-system research, where evaluation gaps, experimental instability, and hidden technical debt can obscure the relation between observed metrics and system-level interpretation [20,32,43].

 doi:10.20944/preprints202605.1992.v1

The present technical note documents a reproducible kernel-damping evidence protocol for the SORT-AI Core-3 applications AI.01, AI.04, and AI.13. These applications represent physical/interconnect coupling, logical/runtime-control coupling, and semantic/agentic coupling, respectively. The associated archived evidence release provides the machine-readable inputs, declared risk transformations, scripts, expected outputs, generated outputs, and reproduction manifest used in the protocol [5].

The contribution of this note is methodological. It does not validate SORT-AI as a whole, does not report vendor-specific telemetry, does not benchmark production infrastructure, and does not claim runtime optimization. It documents how declared structural risk-transition scenarios can be transformed into risk variables, reconstructed through a Gaussian kernel-damping protocol, and reproduced as analysis-layer artefacts under the declared canonical scale parameter $\sigma_0 = 0.00190643$.

1.1. Why an Evidence Protocol Is Required

The SORT-AI domain architecture defines the structural vocabulary through which advanced AI systems are read as coupled systems rather than isolated components [1]. Within that architecture, applications are recurrent structural problem forms. The Core-3 application papers define three such forms: interconnect-induced instability for AI.01, runtime-control incoherence for AI.04, and agentic system instability for AI.13 [2–4].

What remains necessary is an archive-grade evidence package that connects those declared structural problem forms to a deterministic reconstruction protocol. The evidence protocol serves this role. It specifies how scenario metrics are converted into risk variables, how risk transitions are mapped to kernel quotients, how implied structure-mode magnitudes are computed, and how scenario-level dispersion and classification are reproduced. The protocol is therefore not an additional application paper and not a production test. It is the reproducibility layer that makes the Core-3 kernel-damping calculation inspectable and repeatable.

The archived evidence release is the citable artefact for this layer [5]. The GitHub development surface is used as a data source for the manuscript, while the archived release defines the versioned reference state used for citation, reproduction, and comparison.

1.2. Contribution and Structure

This note makes four methodological contributions. First, it formalizes the reproducibility protocol for Core-3 kernel-damping reconstruction. Second, it documents a scenario-class architecture that distinguishes core, boundary, and overlap regimes within each application. Third, it links the reported calculations to an archived evidence release containing declared inputs, executable artefacts, expected outputs, generated outputs, and a reproduction manifest. Fourth, it fixes the boundary between MOCK v4 as frozen structural reference architecture and the evidence release as an analysis-layer reproducibility artefact.

The remainder of the note is organized as follows. Section 2 positions the protocol relative to SORT-AI and MOCK v4. Section 3 explains the Core-3 application selection. Section 4 defines the scenario-class architecture. Sections 5 and 6 introduce the kernel-damping protocol and the risk-transformation rules. Section 7 defines the scenario statistics and classification bands. Section 8 describes the archived reproducibility package. Section 9 reports the Core-3 results. Sections 10, 11, and 12 discuss the interpretation, limitations, and conclusion. Appendices A–E provide the scenario tables, transformation table, reproduction manifest, archive information, and source-material boundary.

The contribution of this note is the conversion of the Core-3 calculation into a versioned, archived, manifest-bound reproducibility artefact. The systems and AI literature cited above motivates the relevant coupling regimes; the reproducibility protocol itself is documented by the archived release. Accordingly, the manuscript should be read as a reproducibility-method Technical Note for a declared scenario-fixture archive. It is not a benchmark paper, a systems experiment, a workload-replay or tracing study, a model-serving benchmark, an AI safety evaluation, or a runtime optimization study.

 doi:10.20944/preprints202605.1992.v1

2. Framework Positioning

2.1. SORT as a Level-0 Structural Assessment Framework

SORT is used in this note as a Level-0 structural assessment framework. In this role, it provides a structural description layer for consistency, projection, coupling, and boundary analysis prior to any Level-1 dynamical model, field equation, or implementation-specific system model. The protocol developed below does not introduce new physical laws, empirical degrees of freedom, or additional fitted parameters. Its object is the reproducible reconstruction of declared structural risk transitions under a fixed kernel-damping rule.

This positioning is important for the interpretation of the evidence release. The reported calculations do not evaluate whether a physical, operational, or vendor-specific system follows a SORT dynamic. They evaluate whether declared scenario-level risk vectors can be transformed, reconstructed, and classified within a common structural protocol. The relevant criteria are therefore structural and reproducibility-oriented: coherence of the declared transformations, consistency of the kernel reconstruction, stability of the implied structure-mode distribution, and traceability of the input–output relation.

The mathematical language used in the protocol is compatible with standard operator-theoretic and matrix-analytic settings, but the present note does not attempt to derive or extend those foundations. The domain-level SORT-AI architecture supplies the specific application hierarchy used here [1], while standard references on functional analysis, perturbation theory, Hilbert-space methods, and matrix analysis provide the broader mathematical background for operator and projection language [51–54].

2.2. SORT-AI Domain Architecture

SORT-AI organizes advanced AI systems through a domain hierarchy in which each level has a distinct methodological role:

$$\text{Domain} \rightarrow \text{Cluster} \rightarrow \text{Application} \rightarrow \text{Scenario Class} \rightarrow \text{Metric Set}.$$

The domain specifies the system class under analysis. The cluster identifies the structural problem class. The application denotes a recurrent structural problem form within that cluster. The scenario class defines a controlled structural slice of the application, and the metric set provides the declared quantities used for risk-transition reconstruction.

This distinction prevents two common category errors. An application is not a use case: it denotes a recurrent structural form rather than a concrete deployment instance. A scenario is not an application: it is a typed manifestation within an application and does not create a new application object. The Core-3 group follows this hierarchy. AI.01 belongs to Cluster A, Coupling, and represents physical/interconnect coupling. AI.04 belongs to Cluster C, Control, and represents logical/runtime-control coupling. AI.13 belongs to Cluster D, Emergence, and represents semantic/agentic coupling. This mapping follows the canonical SORT-AI domain architecture [1].

Domain modules such as SORT-AI provide domain-specific interpretation while consuming the shared SORT public core. Here, the SORT public core denotes the framework-level set of shared formal objects and interface conventions, including operator definitions, projection contracts, kernel conventions, and the canonical scale parameter. Domain modules do not modify these shared objects. The kernel form and the canonical scale parameter used in the present protocol therefore belong to the shared SORT public core and are not domain-internal to SORT-AI.

2.3. MOCK v4 as Frozen Structural Reference Architecture

MOCK, the Model–Operator Consistency Kernel, denotes the frozen structural and contractual reference architecture of SORT. It defines operator topology, projection contracts, domain-isolation boundaries, and the public structural consistency interfaces through which domain modules such as SORT-AI consume the shared SORT public core. MOCK v4 is the version of this reference architecture

 doi:10.20944/preprints202605.1992.v1

used by the present evidence layer. It is not an execution engine, runtime, benchmark harness, numerical solver, or production system. The Core-3 evidence release is a separate analysis-layer artefact that operates on top of MOCK v4, without modifying MOCK v4 and without defining a new MOCK version [5]. The intended boundary relation is therefore:

$$\text{MOCK v4 structural architecture} \rightarrow \text{Core-3 kernel-damping evidence} \rightarrow \text{archived evidence release.}$$

This boundary is maintained throughout the manuscript: the protocol documents reproducible structural calculations, not execution by MOCK v4.

2.4. Evidence Status and Non-Claims

The evidence status of this note is deliberately narrow. The protocol supports deterministic reproducibility of declared scenario calculations and evaluates whether declared risk-transition vectors are compatible with a Gaussian kernel-damping reconstruction under the canonical scale parameter $\sigma_0 = 0.00190643$. It also reports the resulting scenario-level coherence classifications within the tolerances and classification bands defined by the archived evidence release [5].

The protocol does not claim production validation of SORT or SORT-AI, benchmark superiority, vendor-specific measurement, telemetry-derived inference, runtime optimization, or empirical re-estimation of σ_0 from AI production data. It also does not claim coverage of all SORT-AI applications, automatic transfer to other SORT domains, or uniqueness of the Gaussian kernel form. Table 1 summarizes the corresponding claim boundary.

Table 1. Evidence status and claim boundary of the Core-3 kernel-damping evidence protocol.

Supported by this note	**Not claimed by this note**
Deterministic reproducibility of declared Core-3 scenario calculations.	Production validation of SORT, SORT-AI, or any deployed AI system.
Compatibility of declared risk-transition vectors with a Gaussian kernel-damping reconstruction under $\sigma_0 = 0.00190643$.	Benchmark or performance comparison against any vendor system.
Scenario-level structural coherence within the tolerances and bands of the archived evidence release.	Vendor-specific or telemetry-derived measurements.
Analysis-layer documentation of core, boundary, and overlap regimes for AI.01, AI.04, and AI.13.	Runtime optimization, scheduler improvement, or operational intervention in any deployed system.
Use of MOCK v4 as frozen structural reference architecture.	Execution by MOCK v4 or introduction of a new MOCK version.

The same boundary can be expressed as a validity hierarchy, in which each level is restricted to what the protocol actually documents (Table 2). This hierarchy is descriptive of the present release; it does not extend or weaken the claim boundary defined in Table 1.

Table 2. Validity hierarchy of the Core-3 kernel-damping evidence protocol. Each level is restricted to the artefact actually documented in the archived release.

Validity level	**Claimed here?**	**Meaning**
Arithmetic reproducibility	yes	Scripts reproduce κ, ξ, means, dispersions, and CV within manifest tolerances.
Scenario-fixture consistency	yes	Declared scenarios remain internally consistent under the fixed protocol.
Empirical system validity	no	No claim that production systems follow the same distribution or the same σ_0.

 doi:10.20944/preprints202605.1992.v1

3. Core-3 Application Selection

The Core-3 group is selected as a structurally diverse starting set for the evidence protocol. The selection is not intended to cover all SORT-AI applications. It is designed to test whether the same kernel-damping reconstruction can be applied across three qualitatively different coupling regimes: physical/interconnect coupling, logical/runtime-control coupling, and semantic/agentic coupling. This provides a compact but non-trivial application set for evaluating reproducibility across distinct structural axes.

3.1. AI.01 — Physical / Interconnect Coupling

AI.01 belongs to Cluster A, Coupling, in the SORT-AI domain architecture [1]. Within the Core-3 set, it represents the physical and topological coupling axis. Its structural focus is interconnect-induced instability in large-scale AI and HPC systems, including synchronization drift, straggler cascades, topology-induced capacity loss, and cost-per-performance degradation under fabric saturation [2]. These phenomena are consistent with broader observations in large-scale distributed systems, where tail latency, straggler effects, multi-tenant GPU workloads, accelerator hardware integration, distributed-systems observability, and warehouse-scale infrastructure constraints can dominate system-level behaviour even when local components operate correctly [6–9,16–18]. In the present protocol, AI.01 provides the physical/interconnect reference axis for testing whether declared risk transitions admit coherent kernel-damping reconstruction.

3.2. AI.04 — Logical / Runtime-Control Coupling

AI.04 belongs to Cluster C, Control, in the SORT-AI domain architecture [1]. Within the Core-3 set, it represents the logical and runtime-control coupling axis. Its structural focus is the loss of coherence between schedulers, orchestrators, runtime engines, retry mechanisms, admission or routing layers, and policy surfaces [3]. Large-scale serving and orchestration systems already show that control decisions across batching, scheduling, placement, memory management, and phase separation can materially affect effective throughput and latency behaviour, and broader system-design principles on end-to-end functional placement remain relevant to the layering of these control surfaces [10–15,19]. In the present protocol, AI.04 provides the logical/control reference axis for scenario classes involving cross-layer control conflict, retry amplification, control oscillation, SLA-boundary occupation, and overlaps with infrastructure or agentic execution regimes.

3.3. AI.13 — Semantic / Agentic Coupling

AI.13 belongs to Cluster D, Emergence, in the SORT-AI domain architecture [1]. Within the Core-3 set, it represents the semantic and agentic coupling axis. Its structural focus is instability in multi-agent, tool-using, and autonomous planning workflows, where system behaviour emerges through planning, tool selection, execution, verification, and context propagation [4]. Recent work on reasoning-and-acting methods, tool-using language models, open-ended agentic systems, and generative-agent simulations illustrates how agentic workflows extend beyond isolated model calls into temporally coupled execution structures [25–28]. In the present protocol, AI.13 provides the semantic/agentic reference axis for scenario classes involving intent divergence, tool-use amplification, recursive planning drift, context saturation, and verification/execution boundaries. In this note, AI.13 is treated as a structural scenario family for semantic/agentic coupling, not as a safety certification, threat model, dangerous-capability evaluation, or governance-compliance method.

3.4. Rationale for Core-3 Selection

The Core-3 group is selected because it spans physical, logical, and semantic coupling regimes within advanced AI systems. This makes the set suitable for an initial reproducibility protocol: the same mathematical reconstruction is applied to scenario classes that differ in structural interpretation, metric families, and coupling surface. The purpose is not to infer complete SORT-AI coverage, but to test

 doi:10.20944/preprints202605.1992.v1

whether a common declared kernel-damping protocol remains reproducible across three representative coupling axes. Table 3 summarizes the application-to-axis mapping.

Table 3. Core-3 coupling-axis mapping. Cluster identifiers follow the SORT-AI domain architecture [1].

Application	Cluster	Coupling Axis	Canonical Reference
AI.01	A (Coupling)	physical / interconnect	[2]
AI.04	C (Control)	logical / runtime-control	[3]
AI.13	D (Emergence)	semantic / agentic	[4]

4. Scenario-Class Architecture

4.1. Application-to-Scenario Hierarchy

The evidence protocol uses a nested scenario architecture. An application denotes a recurrent structural problem form within the SORT-AI domain, while a scenario denotes a controlled structural slice within that application. The scenario is therefore not a new application and not a deployment use case. It is the unit at which declared metric sets are transformed into risk variables and evaluated through the kernel-damping protocol. The hierarchy used in the archived release is:

$$\text{Domain} \rightarrow \text{Cluster} \rightarrow \text{Application} \rightarrow \text{Scenario Type} \rightarrow \text{Metric Set} \rightarrow \text{Kernel-Damping Test}.$$

Three scenario types are used: core, boundary, and overlap. Core scenarios represent the central structural regime of an application. Boundary scenarios represent operating-margin or transition-zone regimes. Overlap scenarios represent cross-application coupling regimes in which two structural axes interact within a single scenario class.

4.2. Core Scenarios

Core scenarios isolate the dominant structural mode of the corresponding application. They are designed to express the axis-defining regime with minimal cross-application mixing. In the Core-3 release, each application contains three core scenarios, yielding nine core scenarios in total. Because core scenarios are intended to represent the most internally homogeneous structural slice of each application, they are expected to produce low coefficients of variation in the implied structure-mode values.

4.3. Boundary Scenarios

Boundary scenarios describe regimes in which the application remains within its own structural axis but operates near a margin, capacity limit, composition boundary, or service-level transition. These scenarios may exhibit larger implied structure-mode magnitudes than core scenarios because they represent stronger or more constrained risk transitions. Their role in the protocol is not to introduce separate applications, but to test whether structurally marginal cases remain internally coherent under the same kernel-damping reconstruction. In the Core-3 release, AI.01 contains two boundary scenarios, AI.04 contains one, and AI.13 contains two.

4.4. Overlap Scenarios

Overlap scenarios test structured coupling between two Core-3 applications. They are included because advanced AI systems often exhibit mixed regimes in which physical, logical, and semantic coupling surfaces interact. The release defines six overlap scenarios: AI.01.O1 for interconnect–runtime-control coupling, AI.01.O2 for interconnect–agentic orchestration coupling, AI.04.O1 for control–infrastructure coupling, AI.04.O2 for control–agentic execution coupling, AI.13.O1 for agentic–runtime-control coupling, and AI.13.O2 for agentic–infrastructure coupling. These scenarios are expected to produce more mixed coherence than core scenarios, typically reflected in moderate coefficients of variation within the acceptable mixed/overlap band defined by the archived evidence release.

 doi:10.20944/preprints202605.1992.v1

4.5. Scenario Inventory

Table 4 lists the twenty scenario classes included in the Core-3 evidence release. The table provides the scenario identifier, structural class, scenario type, and intended structural slice for each application. The inventory is descriptive: it defines the declared scenario taxonomy used by the evidence protocol and does not introduce additional metrics beyond those included in the archived release.

Table 4. Scenario inventory for the Core-3 kernel-damping evidence release. Source: archived evidence release [5].

ID	Scenario Class	Type	Purpose (structural slice)
AI.01.C1	Synchronization-Latency Drift	core	Synchronization / latency drift
AI.01.C2	Straggler Cascade Propagation	core	Straggler amplification
AI.01.C3	Topology-Induced Capacity Loss	core	Topology-induced capacity loss
AI.01.B1	Interconnect Saturation Boundary	boundary	Fabric-utilization boundary
AI.01.B2	Heterogeneous Fabric Boundary	boundary	Heterogeneous fabric composition boundary
AI.01.O1	Interconnect ∩ Runtime Control	overlap	Interconnect–control mixed regime
AI.01.O2	Interconnect ∩ Agentic Orchestration	overlap	Interconnect–agentic mixed regime
AI.04.C1	Cross-Layer Control Conflict	core	Scheduler/orchestrator/runtime/policy conflict
AI.04.C2	Retry Amplification	core	Local retry logic produces global cost / attempt amplification
AI.04.C3	Control Oscillation	core	Control loops mutually amplify in time
AI.04.B1	SLA Boundary Occupation	boundary	System near SLA / capacity / margin boundary
AI.04.O1	Control ∩ Infrastructure	overlap	AI.04 ∩ AI.01
AI.04.O2	Control ∩ Agentic Execution	overlap	AI.04 ∩ AI.13
AI.13.C1	Multi-Agent Intent Divergence	core	Intent / goal divergence between agents
AI.13.C2	Tool-Use Amplification	core	Tool-call / cost / execution amplification
AI.13.C3	Recursive Planning Drift	core	Recursive planning instability
AI.13.B1	Context Saturation Boundary	boundary	Context / memory / state-carryover boundary
AI.13.B2	Verification / Execution Boundary	boundary	Boundary between verification, release, and execution
AI.13.O1	Agentic ∩ Runtime Control	overlap	Agentic execution coupled with runtime-control
AI.13.O2	Agentic ∩ Infrastructure	overlap	Agentic orchestration coupled with infrastructure

Inventory summary. Three applications, twenty scenarios, 104 declared metrics, decomposed by scenario type as: 9 core, 5 boundary, 6 overlap.

5. Kernel-Damping Protocol

The evidence release uses a kernel-damping protocol rather than a new dynamical model. Its purpose is to reconstruct declared risk transitions through a fixed Gaussian damping form and to report the implied structure-mode values produced by that reconstruction. The protocol is applied

 doi:10.20944/preprints202605.1992.v1

uniformly across all Core-3 applications and scenario classes. No parameter is fitted from AI telemetry in this procedure.

5.1. Risk Transition Form

Each metric in the evidence release is represented as a pair of declared risk values, $r_i^{(0)}$ and $r_i^{(1)}$, where $r_i^{(0)}$ denotes the baseline risk state and $r_i^{(1)}$ denotes the comparison risk state after the declared transition. All metrics are first expressed in risk form, so that lower values correspond to lower structural risk. The per-metric damping quotient is then defined as

$$\kappa_i = \frac{r_i^{(1)}}{r_i^{(0)}}. \tag{1}$$

For risk reductions, $0 < \kappa_i \leq 1$. Values $\kappa_i > 1$ would indicate risk increase rather than damping. The archived evidence release restricts the reported reconstruction inputs to $0 < \kappa_i < 1$, so that the implied structure mode ξ_i in Equation 3 is real-valued and strictly positive. The boundary case $\kappa_i = 1$ would yield $\xi_i = 0$; it is not included in the v0.1.0 reconstruction inputs because the release documents strictly damped risk transitions.

Admissibility conditions. The reconstruction inputs are admissible only if $r_i^{(0)} > 0$, $r_i^{(1)} > 0$, and $0 < \kappa_i < 1$. Zero-risk cases after normalization are excluded from the v0.1.0 reconstruction inputs.

5.2. Gaussian Kernel Form

The damping quotient is interpreted through a fixed Gaussian kernel form,

$$\kappa_{\sigma_0}(\xi_i) = \exp\left[-\frac{(\sigma_0 \xi_i)^2}{2}\right]. \tag{2}$$

The Gaussian form is not specific to the SORT-AI domain. It belongs to the shared SORT public core used across domain modules, while SORT-AI provides the application-specific interpretation. The Gaussian form is used because the archived protocol treats damping as a symmetric scale-suppression rule in the shared SORT projection notation; alternative kernel families are not evaluated in this release. In this note, the Gaussian kernel functions as a reproducibility rule for the archived scenario calculations, not as a claim of uniqueness or as a fitted physical law.

5.3. Implied Structure Mode

Solving Equation 2 for the mode variable gives the implied structure-mode magnitude for each metric:

$$\xi_i = \frac{\sqrt{-2\ln(\kappa_i)}}{\sigma_0}. \tag{3}$$

In this protocol, ξ_i is a protocol-internal, dimensionless scalar index of declared risk-transition magnitude after mapping κ_i onto the fixed Gaussian family. It is not interpreted as a measured physical frequency, a hardware-specific signal, or a production telemetry variable. Scenario-level statistics are computed from the collection of ξ_i-values associated with each scenario.

5.4. Canonical Scale Parameter

The scale parameter used throughout the protocol is fixed as

$$\sigma_0 = 0.00190643. \tag{4}$$

In the present evidence release, σ_0 is a declared canonical scale parameter inherited from the SORT-AI reference framing [1]. It is not re-fitted from AI production telemetry, benchmark traces, or vendor measurements. Any future attempt to estimate, vary, or re-calibrate σ_0 from domain-specific data would constitute a separate evidence release and is outside the scope of this note.

 doi:10.20944/preprints202605.1992.v1

Scale-parameter role. Because $\xi_i = \sqrt{-2\ln(\kappa_i)}/\sigma_0$, the canonical scale parameter fixes the absolute scale of reported ξ_i-values but does not enter the scenario-level coefficient of variation. As a result, σ_0 sets the numerical scale of $\bar{\xi}_S$ and $s_{\xi,S}$ but is not a classification determinant. The corresponding invariance of CV_S under uniform rescaling by σ_0 is recorded in Section 7.3.

6. Risk Transformation and Metric Normalization

The kernel-damping protocol requires each declared metric to be expressed as a risk variable r before κ_i and ξ_i are evaluated. The normalization convention is fixed throughout the evidence release: lower values of r correspond to lower structural risk. This convention permits heterogeneous metric families to be compared through a common risk-transition form without treating the original metrics as directly commensurable.

6.1. Direct Risk Metrics

Direct risk metrics already have the required orientation. For these quantities, larger observed values correspond to larger structural risk, and the transformation is the identity map $r = x$. This family includes metrics such as drift indicators, tail-latency measures, attempt counts, retry rates, and failure rates. In the archived evidence release, such metrics enter the kernel-damping calculation without sign inversion or offset correction.

6.2. Health / Coherence Metrics

Health or coherence metrics have the opposite orientation: larger values indicate better structural condition. For normalized scores $x \in (0, 1]$, where $x = 1$ denotes the best attainable state within the declared metric convention, the corresponding risk variable is $r = 1 - x$. This transformation is used for quantities such as coherence indicators, success rates, agreement scores, and normalized goodput measures. It converts improvement in a health score into reduction of the associated risk variable.

6.3. Overhead / Multiplier Metrics

Overhead or multiplier metrics are expressed relative to a neutral baseline $x = 1$, where 1 denotes no additional overhead. For these quantities, the risk variable is defined as $r = x - 1$. This family includes cost multipliers, attempt multipliers, latency multipliers, and related amplification factors. The transformation isolates the excess component above the neutral baseline and makes it compatible with the risk-reduction quotient in Equation 1.

Table 5. Risk-transformation rules used in the evidence release. Each metric is pre-declared as belonging to exactly one transformation family. Source: archived evidence release [5].

Family	Rule	Typical metric types
identity	$r = x$	direct risk indicators (drift, tail latency)
risk	$r = x$	failure rates, attempt counts
health	$r = 1 - x$	coherence scores, success rates, normalised goodput
multiplier	$r = x - 1$	cost multipliers, attempt multipliers, overhead factors

The *identity* and *risk* families share the same algebraic rule $r = x$ but are distinct registry labels in the archived metric taxonomy. The labels record the declared semantic role of each metric in the registry and are preserved separately so that the transformation provenance remains traceable, even though the numerical mapping is identical.

6.4. Pre-Declaration Rule

Pre-declaration rule. Risk transformations are declared before kernel evaluation and are not adjusted after observing the resulting structure-mode distribution.

This rule is essential to the reproducibility character of the protocol. It prevents post-hoc selection of transformations that would improve the apparent coherence of the ξ_i-distribution after the fact. In

 doi:10.20944/preprints202605.1992.v1

the archived release, the transformation family of each metric is fixed in the metric registry before κ_i, ξ_i, scenario means, sample dispersions, or coefficients of variation are computed [5]. The reported results therefore reflect the declared metric normalization scheme rather than an adaptive fitting procedure.

7. Scenario Statistics and Regime Classification

Scenario-level classification is based on the distribution of metric-level structure-mode values within each declared scenario. The protocol first computes ξ_i for every metric using Equation 3. It then aggregates those values at the scenario level through the arithmetic mean, sample dispersion, and coefficient of variation. These quantities provide the basis for the three-band regime classification used in the archived evidence release.

7.1. Mean Structure Mode

For each scenario S, the mean structure mode is defined as the arithmetic mean of the metric-level ξ_i-values assigned to that scenario:

$$\bar{\xi}_S = \frac{1}{N_S} \sum_{i \in S} \xi_i, \tag{5}$$

where S indexes the scenario and N_S is its metric count. The value $\bar{\xi}_S$ summarizes the central damping-mode magnitude implied by the declared risk transitions in that scenario.

7.2. Sample Dispersion

Scenario-level dispersion is computed using the sample standard deviation of the metric-level structure modes:

$$s_{\xi,S} = \sqrt{\frac{1}{N_S - 1} \sum_{i \in S} (\xi_i - \bar{\xi}_S)^2}, \tag{6}$$

using the denominator $N_S - 1$. The sample dispersion measures how tightly the metric-level damping reconstruction clusters around the scenario mean. Lower dispersion indicates that the declared metrics imply a more internally coherent structure-mode regime.

7.3. Coefficient of Variation

The coefficient of variation normalizes the sample dispersion by the scenario mean:

$$CV_S = \frac{s_{\xi,S}}{\bar{\xi}_S}. \tag{7}$$

The coefficient of variation provides a dimensionless dispersion measure suitable for cross-scenario and cross-application comparison [55]. In the present protocol, CV_S is used as the classification statistic because it compares relative, rather than absolute, dispersion of the implied structure-mode distribution. Because $\xi_i = \sqrt{-2\ln(\kappa_i)}/\sigma_0$, the scenario-level coefficient of variation $CV_S = s_{\xi,S}/\bar{\xi}_S$ is invariant under uniform rescaling by σ_0; σ_0 fixes the absolute scale of reported ξ_i-values but does not enter the band classification. The reported CV precision follows the archived output precision and the manifest rounding tolerance; it is not used here for broader statistical inference.

7.4. Classification Bands

The archived evidence release uses the three-band classification scheme shown in Table 6. A scenario is classified as coherent within the release-local CV classification scheme when the implied structure-mode distribution has low relative dispersion. It is classified as acceptable mixed / overlap when the distribution remains structured but shows the increased dispersion expected in cross-application or mixed-regime cases. Values above the upper threshold are treated as unstable or outlier-dominated within the declared release protocol.

 doi:10.20944/preprints202605.1992.v1

Table 6. Classification bands used in the archived evidence release. Lower CV corresponds to greater internal coherence of the implied structure mode within the scenario.

CV range	Classification
$CV \leq 0.15$	coherent
$0.15 < CV \leq 0.25$	acceptable mixed / overlap
$CV > 0.25$	unstable / outlier-dominated

Only this three-band scheme is used in the present note. No additional classification bands are introduced here, because the purpose of the manuscript is to document the archived Evidence Release v0.1.0 rather than to extend its classification protocol. The bands are release-local classification bands of Evidence Release v0.1.0 and are not proposed as general statistical thresholds.

8. Reproducibility Package and Archive Structure

The evidence protocol is accompanied by an archived reproducibility package. The package is designed to make the declared Core-3 calculations inspectable at the level of inputs, transformation rules, executable scripts, expected outputs, generated outputs, and manifest tolerances. This structure follows established reproducibility and software-citation principles for computational research artefacts [34–37,41,42].

8.1. Repository Path

The reproducibility artefact is archived as a software release on Zenodo [5] (DOI 10.5281/zenodo.20400331). The evidence package is contained within the archived release at the path:

`evidence_releases/sort_ai_core3_kernel_damping_v1/`

The GitHub repository functions as the development surface and data source for the present manuscript, while the Zenodo archive is the citable reference state. The manuscript therefore refers to the archived software release when citing the evidence artefact, rather than citing individual files or subdirectories in the development repository. This distinction preserves a stable reference target for reproduction and avoids ambiguity between the evolving repository state and the versioned evidence release.

8.2. Input Artefacts

The input artefacts define the declared scenario and metric space used by the protocol. They include the consolidated metric table, the scenario registry, application-specific scenario-metric files, and methodological documentation. The central input files are:

- `data/core3_metrics.csv`, containing the declared metric rows;
- `data/scenarios.json`, containing the scenario registry;
- `data/ai01/scenario_metrics.json`, containing the AI.01 scenario metrics;
- `data/ai04/scenario_metrics.json`, containing the AI.04 scenario metrics;
- `data/ai13/scenario_metrics.json`, containing the AI.13 scenario metrics;
- `docs/methodology.md`, documenting the protocol method;
- `docs/mock_v4_reference_boundary.md`, documenting the MOCK v4 boundary;
- `docs/non_claims.md`, documenting the claim limitations;
- `docs/source_notes.md`, documenting the source-material boundary.

These artefacts define the declared analysis-layer inputs. They are not production telemetry, benchmark measurements, or vendor-specific operational traces.

 doi:10.20944/preprints202605.1992.v1

8.3. Executable Artefacts

The executable artefacts implement the deterministic reconstruction workflow. They compute κ_i, ξ_i, scenario-level means, sample dispersions, coefficients of variation, and manifest checks from the declared inputs. The principal scripts are:

- `scripts/kernel_damping.py`, which implements the κ_i- and ξ_i-calculation;
- `scripts/run_all.py`, which serves as the end-to-end reproduction driver;
- `scripts/validate_manifest.py`, which performs manifest-integrity checks;
- `scripts/build_core3_evidence.py`, which supports package construction.

The scripts reproduce the archived analysis-layer calculations. They do not execute MOCK v4 as a runtime engine and do not operate on external production systems.

8.4. Expected Outputs

The directory `outputs_expected/` contains the committed reference state of the release. These files define the expected outputs against which a reproduction run can be compared:

- `outputs_expected/core3_summary.json`;
- `outputs_expected/ai01_results.csv`;
- `outputs_expected/ai04_results.csv`;
- `outputs_expected/ai13_results.csv`;
- `outputs_expected/reproducibility_report.md`.

The expected-output directory should not be overwritten during a local reproduction run. It functions as the versioned reference state of the archived evidence release.

8.5. Generated Outputs

The directory `outputs_generated/` is reserved for outputs produced by a local reproduction run. In version control it is kept empty except for a placeholder file. Running the reproduction workflow writes generated artefacts to this directory without modifying the committed reference outputs in `outputs_expected/`. The intended validation procedure is to compare the generated outputs against the expected outputs within the tolerances declared in the manifest.

The release declares three relevant tolerances: κ absolute tolerance $= 0.001$, ξ absolute tolerance $= 1.0$, and CV rounding tolerance $= 0.001$. These tolerances define the numerical comparison boundary for the reproduction procedure. Agreement within these tolerances supports deterministic reproduction of the archived calculations; it does not constitute production validation or benchmark evidence.

8.6. GitHub Release and Zenodo DOI

The development surface for the evidence package is the SORT repository, release tag `v0.1.0-core3-evidence`. The citable artefact is the Zenodo software release with DOI 10.5281/zenodo.20400331 [5]. This manuscript cites the archived Zenodo release as the stable reference object. The GitHub subdirectory is used only as the development and data surface associated with that archive. Licensing and repository-specific notices are governed by the archived release metadata and the included repository files.

This distinction follows software-citation practice: a manuscript should cite a persistent archived artefact rather than an evolving repository path when the purpose is scientific reproduction [37,42]. The archived release therefore fixes the version boundary for the calculations reported in this note.

9. Results

All numerical values in this section are taken from the archived evidence release [5], specifically from `outputs_expected/core3_summary.json` and the corresponding application-level result files. They are reported as data of the archived evidence release. They are not independent empirical measurements, vendor telemetry, benchmark results, or production-system observations. The manuscript

 doi:10.20944/preprints202605.1992.v1

reports the protocol definitions, scenario-level summaries, and aggregate results; full metric-level reproduction requires the archived release.

9.1. Core-3 Aggregate

Table 7 summarizes the aggregate quantities of the Core-3 evidence release. The release contains three applications, twenty scenarios, and 104 declared metrics evaluated under the fixed scale parameter $\sigma_0 = 0.00190643$. The overall coefficient of variation is $CV = 0.141$, placing the aggregate release within the coherent band of the archived classification scheme.

Table 7. Core-3 aggregate quantities. Source: archived evidence release [5], `outputs_expected/core3_summary.json`.

Quantity	Value
Applications	3
Scenarios	20
Declared metrics	104
Canonical σ_0	0.00190643
Overall $\bar{\xi}$	969.35
Overall s_ξ (sample)	137.0
Overall CV	0.141
Overall classification	coherent

9.2. AI.01 Results

Table 8 reports the scenario-level results for AI.01. The three core scenarios are all classified as coherent, with CV-values between 0.035 and 0.044. The two boundary scenarios remain coherent while exhibiting higher mean structure-mode values. The overlap scenario AI.01.O1, which represents interconnect–runtime-control coupling, enters the acceptable mixed/overlap band with $CV = 0.205$. This behaviour is consistent with its declared role as a cross-axis coupling scenario in the archived release.

Table 8. Scenario-level results for AI.01 (Cluster A, physical/interconnect coupling). $\bar{\xi}$, s_ξ, and CV are computed from per-metric ξ values within each scenario. Source: archived evidence release [5].

Scenario	Type	$\bar{\xi}$	s_ξ	CV	Classification
AI.01.C1	core	782.00	34.39	0.044	coherent
AI.01.C2	core	920.00	33.35	0.036	coherent
AI.01.C3	core	854.00	30.08	0.035	coherent
AI.01.B1	boundary	1087.00	96.80	0.089	coherent
AI.01.B2	boundary	1064.00	66.09	0.062	coherent
AI.01.O1	overlap	958.33	196.82	0.205	acceptable mixed / overlap
AI.01.O2	overlap	1023.00	75.55	0.074	coherent
All metrics	—	955.56	137.40	0.144	coherent

9.3. AI.04 Results

Table 9 reports the scenario-level results for AI.04. The three core runtime-control scenarios are highly coherent, with CV-values between 0.030 and 0.038. The boundary scenario AI.04.B1 remains coherent at an elevated mean structure-mode value. Both overlap scenarios, AI.04.O1 and AI.04.O2, are classified as acceptable mixed/overlap. This is consistent with the role of AI.04 as the logical control-axis application that connects structurally to both physical/infrastructure coupling and semantic/agentic execution coupling.

 doi:10.20944/preprints202605.1992.v1

Table 9. Scenario-level results for AI.04 (Cluster C, logical/runtime-control coupling). Source: archived evidence release [5].

Scenario	Type	$\bar{\xi}$	s_{ξ}	CV	Classification
AI.04.C1	core	768.40	28.64	0.037	coherent
AI.04.C2	core	952.00	36.50	0.038	coherent
AI.04.C3	core	823.00	24.90	0.030	coherent
AI.04.B1	boundary	1082.00	92.64	0.086	coherent
AI.04.O1	overlap	973.33	158.32	0.163	acceptable mixed / overlap
AI.04.O2	overlap	1024.17	195.05	0.190	acceptable mixed / overlap
All metrics	—	941.00	153.17	0.163	acceptable mixed / overlap

9.4. AI.13 Results

Table 10 reports the scenario-level results for AI.13. All core and boundary scenarios are classified as coherent. The overlap scenario AI.13.O1, which represents agentic–runtime-control coupling, falls into the acceptable mixed/overlap band with $CV = 0.173$. By contrast, AI.13.O2, representing agentic–infrastructure coupling, remains coherent. This asymmetry is reported as part of the archived scenario structure and is not interpreted here as evidence of production behaviour or domain-wide generality.

Table 10. Scenario-level results for AI.13 (Cluster D, semantic/agentic coupling). Source: archived evidence release [5].

Scenario	Type	$\bar{\xi}$	s_{ξ}	CV	Classification
AI.13.C1	core	878.00	33.28	0.038	coherent
AI.13.C2	core	1038.00	45.08	0.043	coherent
AI.13.C3	core	924.00	31.10	0.034	coherent
AI.13.B1	boundary	1150.00	49.12	0.043	coherent
AI.13.B2	boundary	1079.00	46.82	0.043	coherent
AI.13.O1	overlap	956.67	165.73	0.173	acceptable mixed / overlap
AI.13.O2	overlap	1043.00	57.84	0.055	coherent
All metrics	—	1008.33	114.37	0.113	coherent

9.5. Core / Boundary / Overlap Matrix

Table 11 summarizes the structural regime architecture across the Core-3 applications. Core scenarios cluster tightly in the coherent band across all three applications, with $CV < 0.05$. Boundary scenarios remain coherent while generally appearing at elevated mean structure-mode values in the application-level result tables. Overlap scenarios are more dispersed: some remain coherent, while others fall into the acceptable mixed/overlap band. No scenario in the archived release falls into the unstable/outlier-dominated band. This pattern supports the internal regime distinction used by the evidence protocol without extending the claim beyond the declared scenario set.

Table 11. Core / Boundary / Overlap classification matrix across the three Core-3 applications. Counts in each cell are the number of scenarios of that type in each application; the parenthesised range gives the observed CV range. Source: archived evidence release [5].

Application	Core	Boundary	Overlap
AI.01	3 ($CV \in [0.035, 0.044]$)	2 ($CV \in [0.062, 0.089]$)	2 ($CV \in [0.074, 0.205]$)
AI.04	3 ($CV \in [0.030, 0.038]$)	1 ($CV = 0.086$)	2 ($CV \in [0.163, 0.190]$)
AI.13	3 ($CV \in [0.034, 0.043]$)	2 ($CV \in [0.043, 0.043]$)	2 ($CV \in [0.055, 0.173]$)
Total	9	5	6

 doi:10.20944/preprints202605.1992.v1

9.6. Cross-Application Interpretation

Across the archived Core-3 release, all three applications admit kernel-damping reconstruction under the declared scale parameter $\sigma_0 = 0.00190643$. The resulting ξ-distributions are application-specific, but they are structurally compatible under the same protocol. The aggregate Core-3 coefficient of variation is $CV = 0.141$, which lies within the coherent band of the release classification scheme.

This result should be interpreted narrowly. It does not establish that σ_0 is universal across AI systems, does not prove uniqueness of the Gaussian kernel form, and does not provide empirical evidence about production infrastructure. It shows that the archived Core-3 scenario values reproduce compatibly under the declared protocol and that the physical, logical, and semantic coupling axes can be evaluated within a common analysis-layer kernel-damping framework.

10. Discussion

10.1. What the Evidence Supports

The evidence release supports a bounded reproducibility claim. The declared Core-3 scenario values are reproducibly kernel-testable under the protocol defined in Sections 5–7, and the reported results can be regenerated from the archived inputs, scripts, and manifest tolerances [35,36]. Within this scope, the Core-3 applications can be read as structural damping regimes: AI.01 for physical/interconnect coupling, AI.04 for logical/runtime-control coupling, and AI.13 for semantic/agentic coupling. The results also show that the core, boundary, and overlap distinction is operationalisable as an internal scenario architecture. The protocol is deterministic within the declared tolerances, and the evidence layer remains separated from MOCK v4 as frozen structural reference architecture [1].

Failure conditions. The reproduction check fails if generated outputs exceed manifest tolerances, if declared transformation families are altered after evaluation, if admissibility constraints are violated, or if scenario files are inconsistent with the manifest. These failure conditions define the operational meaning of "reproducible" for the present release. The release provides a citable, versioned, manifest-bound reproducibility surface for future SORT-AI work.

10.2. What the Evidence Does Not Support

The evidence does not support production deployment claims, vendor comparisons, live telemetry validation, runtime optimisation claims, or benchmark conclusions. It also does not establish structural necessity for SORT, does not independently re-derive σ_0 from AI telemetry, and does not claim that the Core-3 group covers all SORT-AI applications. These limitations are part of the methodological boundary of the protocol rather than an after-the-fact qualification. They preserve the status of the release as a reproducibility artefact and prevent the reported scenario calculations from being read as empirical performance evidence, a distinction that is particularly important in light of known benchmark, evaluation, and reproducibility limitations in AI-system research [29,31,32,43,50].

10.3. Why the Protocol Matters

The protocol matters because it provides an archivable evidence standard for SORT-AI without expanding the claim beyond analysis-layer reproducibility. It connects the Core-3 applications to a versioned evidence release, a fixed reproduction manifest, and a persistent software archive. This makes the calculation inspectable and citable while preserving the separation between structural-axiomatic development and computational evidence. By fixing inputs, transformations, scripts, expected outputs, generated-output conventions, and manifest tolerances, the release converts the Core-3 calculation into a citable, versioned, manifest-bound reproducibility artefact. In this sense, numerical reproducibility is supportive: it documents that the declared risk-transition calculations can be reproduced under the protocol, but it is not equivalent to structural necessity. The package also aligns SORT-AI evidence practice with broader principles for FAIR research artefacts, software citation, computational reproducibility, and artefact review [34,37,40–42].

 doi:10.20944/preprints202605.1992.v1

11. Limitations

The evidence release uses synthetic but structurally grounded scenario inputs. These inputs are declared analysis-layer values and are not production telemetry, benchmark measurements, vendor-specific traces, or live operational data. *Structurally grounded* here means not empirical sampling, but declared alignment with the SORT-AI scenario architecture: scenario-class alignment, pre-declared metric families, fixed orientation rules, declared admissibility constraints, and archive-level reproducibility. It does not mean telemetry-derived. The scope is also limited to the Core-3 set of AI.01, AI.04, and AI.13; it does not cover all 52 SORT-AI applications and does not imply automatic transfer to SORT-CX, SORT-QS, SORT-COSMO, or other domains. The protocol also does not operate as an AI risk-management process in the sense of operational risk-management frameworks [44]; it remains restricted to analysis-layer reproducibility of declared scenario calculations.

The Gaussian kernel is the declared protocol kernel for this release and is not presented as a proof of uniqueness. Likewise, $\sigma_0 = 0.00190643$ is used as the canonical declared scale parameter of the protocol and is not re-fitted from AI telemetry. The classification bands are intentionally coarse and reproduce the three-band scheme of the archived release. Future refinements of the classification scheme would require a separate release version. The v0.1.0 release does not contain a scenario classified as unstable/outlier-dominated; exercising that band would require a separate release rather than reinterpretation of the present archive. Finally, the number of scenarios per application is small, with six or seven scenarios per Core-3 application. For this reason, the note reports reproducible scenario statistics and coefficients of variation, but does not attempt broader statistical inference beyond the declared evidence protocol. Future releases may map scenario fixtures to telemetry analogues, but no such mapping is claimed here.

12. Conclusions

This technical note formalizes a reproducible kernel-damping evidence protocol for the SORT-AI Core-3 applications AI.01, AI.04, and AI.13. The protocol establishes deterministic reproducibility for declared risk-transition scenarios under the canonical scale parameter $\sigma_0 = 0.00190643$, using the archived evidence release as the citable reference artefact. The reported results show that the declared Core-3 scenarios admit reproducible kernel-damping reconstruction within the protocol boundaries, including core, boundary, and overlap regimes.

The note does not introduce a new MOCK version, does not claim production performance, and does not treat MOCK v4 as an execution engine. Its contribution is instead to provide a citable, versioned, manifest-bound reproducibility surface for future SORT-AI work, while preserving the boundary between MOCK v4 as frozen structural reference architecture and the Core-3 evidence release as an analysis-layer reproducibility artefact.

Author Contributions: The author is the sole contributor to this manuscript and was responsible for conceptualization, methodology, formal analysis, manuscript drafting, and final editing.

Funding: This research received no external funding.

Data Availability Statement: The reproducibility artefact for this Technical Note is archived as a Zenodo software release under DOI `10.5281/zenodo.20400331`. The archive contains the Core-3 kernel-damping evidence package, including declared scenario inputs, risk-transformation rules, executable scripts, expected outputs, generated-output conventions, and the reproduction manifest. The associated GitHub repository functions as the development surface and data source, while the Zenodo archive is the citable versioned reference state for the calculations reported in this manuscript. The data are synthetic but structurally grounded analysis-layer inputs and are not production telemetry, benchmark measurements, or vendor-specific operational traces.

Acknowledgments: The author acknowledges the use of standard scientific software for document preparation and consistency checking.

Conflicts of Interest: The author declares no conflict of interest.

 doi:10.20944/preprints202605.1992.v1

Use of Artificial Intelligence: Large language models were used as editorial aids for language refinement, structural editing, and LaTeX formatting. All scientific content, including the conceptual structure, mathematical definitions, diagnostic formulations, evidence-boundary statements, and interpretation of the protocol, was produced and verified by the author, who takes full responsibility for the content of this manuscript.

Appendix A. Full Scenario Tables

This appendix summarizes the full scenario inventory used in the archived Core-3 evidence release. The table reports the scenario identifier, SORT-AI cluster, scenario type, metric count, mean implied structure mode, coefficient of variation, and classification. These values are reproduced from the archived evidence release and correspond to the scenario-level data reported in Section 9.

Table A1. Scenario inventory with metric counts. Source: archived evidence release [5].

Scenario	Cluster	Type	Metrics	$\bar{\xi}$	CV	Classification
AI.01.C1	A	core	5	782.00	0.044	coherent
AI.01.C2	A	core	5	920.00	0.036	coherent
AI.01.C3	A	core	5	854.00	0.035	coherent
AI.01.B1	A	boundary	5	1087.00	0.089	coherent
AI.01.B2	A	boundary	5	1064.00	0.062	coherent
AI.01.O1	A	overlap	6	958.33	0.205	acceptable mixed / overlap
AI.01.O2	A	overlap	5	1023.00	0.074	coherent
AI.04.C1	C	core	5	768.40	0.037	coherent
AI.04.C2	C	core	5	952.00	0.038	coherent
AI.04.C3	C	core	5	823.00	0.030	coherent
AI.04.B1	C	boundary	5	1082.00	0.086	coherent
AI.04.O1	C	overlap	6	973.33	0.163	acceptable mixed / overlap
AI.04.O2	C	overlap	6	1024.17	0.190	acceptable mixed / overlap
AI.13.C1	D	core	5	878.00	0.038	coherent
AI.13.C2	D	core	5	1038.00	0.043	coherent
AI.13.C3	D	core	5	924.00	0.034	coherent
AI.13.B1	D	boundary	5	1150.00	0.043	coherent
AI.13.B2	D	boundary	5	1079.00	0.043	coherent
AI.13.O1	D	overlap	6	956.67	0.173	acceptable mixed / overlap
AI.13.O2	D	overlap	5	1043.00	0.055	coherent

Appendix B. Risk Transformation Table

The four transformation families used in the evidence release are listed in Table 5 of the main text. Each metric in `data/core3_metrics.csv` of the archived release [5] is pre-declared as belonging to exactly one family. The transformation is fixed at metric-registration time and is not adjusted on the basis of the resulting κ, ξ, or CV values. This appendix therefore does not introduce an additional transformation scheme; it records the rule that the main-text transformation table is the operative normalization standard for the release.

Appendix C. Reproduction Manifest

The archived release [5] declares the validation tolerances listed in Table A2. These tolerances define the comparison boundary between committed expected outputs and locally generated reproduction outputs.

 doi:10.20944/preprints202605.1992.v1

Table A2. Validation tolerances declared in the reproduction manifest.

Quantity	Tolerance
κ absolute tolerance	0.001
ξ absolute tolerance	1.0
CV rounding tolerance	0.001

The manifest further declares the canonical scale parameter $\sigma_0 = 0.00190643$, the operator count $n = 22$, and the MOCK reference as "MOCK v4 frozen structural reference architecture; not execution engine". The expected reproduction report records maximum absolute deviations of 0.000189 for κ and 0.427778 for ξ, both within the declared manifest tolerances. These deviations reflect numerical rounding in the reproduced artefacts and do not alter the scenario-level classifications reported in Section 9.

Appendix D. Repository and Zenodo Archive

The reproducibility artefact is archived as a software release on Zenodo under DOI 10.5281/zenodo.20400331 [5]. The development surface is the GitHub repository `gregorwegener/SORT`, release tag `v0.1.0-core3-evidence`. The evidence package is contained at:

`evidence_releases/sort_ai_core3_kernel_damping_v1/`

The repository subdirectory is used as the data source for the present note but is not cited directly. The citable artefact for the evidence release is the Zenodo archive. Licensing and repository-specific notices are governed by the archived release metadata and the included repository files.

Appendix E. Source Material and Version Boundary

The archived evidence release was assembled from three analysis-layer source documents:

- `AI.01 Kernel-Damping Evidence Set v1.md` — source for AI.01;
- `AI.04 Kernel-Damping Evidence Set v3.md` — source for AI.04;
- `AI.13 Kernel-Damping Evidence Set v1.md` — source for AI.13.

Source values are synthetic but structurally grounded scenario inputs prepared for kernel-damping consistency testing. They are not demo replays and not production measurements [5].

Version boundary. The present Technical Note refers to the `v0.1.0-core3-evidence` release tag of the Core-3 kernel-damping evidence package. Subsequent versions of the release will be linked through the Zenodo concept record. The protocol described here is the `v0.1.0` protocol; any future modification of the kernel form, the classification bands, or the risk-transformation families constitutes a separate release.

References

1. Wegener, G. H. (2026). SORT-AI: Domain Architecture and Structural Diagnostics for Advanced AI Systems. *MDPI Preprints*. doi:10.20944/preprints202512.1334.v3
2. Wegener, G. H. (2026). SORT-AI: Interconnect Stability and Cost per Performance in Large-Scale AI Infrastructure. *MDPI Preprints*. doi:10.20944/preprints202601.0161.v1
3. Wegener, G. H. (2026). SORT-AI: Runtime Control Coherence in Large-Scale AI Systems — Structural Causes of Cost, Instability, and Non-Determinism Beyond Interconnect Failures. *MDPI Preprints*. doi:10.20944/preprints202601.0298.v1
4. Wegener, G. H. (2026). SORT-AI: Agentic System Stability in Large-Scale AI Systems — Structural Causes of Cost, Instability, and Non-Determinism in Multi-Agent and Tool-Using Workflows. *MDPI Preprints*. doi:10.20944/preprints202601.1741.v1
5. Wegener, G. H. (2026). *SORT-AI Core-3 Kernel-Damping Evidence Release*. Zenodo Software Release v0.1.0-core3-evidence. doi:10.5281/zenodo.20400331
6. Barroso, L. A.; Hölzle, U.; Ranganathan, P. (2018). *The Datacenter as a Computer: Designing Warehouse-Scale Machines*, 3rd ed. Morgan & Claypool. doi:10.2200/S00874ED3V01Y201809CAC046

 doi:10.20944/preprints202605.1992.v1

7. Dean, J.; Barroso, L. A. (2013). The Tail at Scale. *Communications of the ACM*, **56**(2), 74–80. doi:10.1145/2408776.2408794
8. Ananthanarayanan, G.; Ghodsi, A.; Shenker, S.; Stoica, I. (2013). Effective Straggler Mitigation: Attack of the Clones. *Proceedings of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 185–198. usenix.org/conference/nsdi13
9. Jouppi, N. P.; Young, C.; Patil, N.; Patterson, D.; et al. (2017). In-Datacenter Performance Analysis of a Tensor Processing Unit. *Proceedings of the 44th Annual International Symposium on Computer Architecture (ISCA)*, 1–12. doi:10.1145/3079856.3080246
10. Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; et al. (2023). Efficient Memory Management for Large Language Model Serving with PagedAttention. *Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP)*. doi:10.1145/3600006.3613165
11. Patel, P.; Choukse, E.; Zhang, C.; Goiri, Í.; et al. (2024). Splitwise: Efficient Generative LLM Inference Using Phase Splitting. *Proceedings of the 51st International Symposium on Computer Architecture (ISCA)*. doi:10.1109/ISCA59077.2024.00019
12. Zhong, Y.; Liu, S.; Chen, J.; Hu, J.; et al. (2024). DistServe: Disaggregating Prefill and Decoding for Goodput-Optimized Large Language Model Serving. *Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. usenix.org/conference/osdi24
13. Yu, G.-I.; Jeong, J. S.; Kim, G.-W.; Kim, S.; Chun, B.-G. (2022). Orca: A Distributed Serving System for Transformer-Based Generative Models. *Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 521–538. usenix.org/conference/osdi22
14. Verma, A.; Pedrosa, L.; Korupolu, M.; Oppenheimer, D.; Tune, E.; Wilkes, J. (2015). Large-Scale Cluster Management at Google with Borg. *Proceedings of the 10th European Conference on Computer Systems (EuroSys)*. doi:10.1145/2741948.2741964
15. Saltzer, J. H.; Reed, D. P.; Clark, D. D. (1984). End-to-End Arguments in System Design. *ACM Transactions on Computer Systems*, **2**(4), 277–288. doi:10.1145/357401.357402
16. Sigelman, B. H.; Barroso, L. A.; Burrows, M.; Stephenson, P.; et al. (2010). Dapper, a Large-Scale Distributed Systems Tracing Infrastructure. *Google Technical Report*. research.google/pub36356
17. Jeon, M.; Venkataraman, S.; Phanishayee, A.; Qian, J.; Xiao, W.; Yang, F. (2019). Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads. *Proceedings of the 2019 USENIX Annual Technical Conference*, 947–960. usenix.org/conference/atc19
18. Meta Engineering (2024). Taming Tail Utilization of Ads Inference at Meta Scale. *Meta Engineering Blog*. engineering.fb.com/2024/10/30
19. Databricks Engineering (2024). LLM Inference Performance Engineering: Best Practices. *Databricks Engineering Blog*. databricks.com/blog/llm-inference-performance-engineering-best-practices
20. Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; et al. (2015). Hidden Technical Debt in Machine Learning Systems. *Advances in Neural Information Processing Systems 28 (NeurIPS)*.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30 (NeurIPS)*.
22. Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; et al. (2020). Language Models Are Few-Shot Learners. *Advances in Neural Information Processing Systems 33 (NeurIPS)*.
23. Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. (2024). The Llama 3 Herd of Models. *arXiv preprint*. arXiv:2407.21783
24. Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; et al. (2021). On the Opportunities and Risks of Foundation Models. *arXiv preprint*. arXiv:2108.07258
25. Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *International Conference on Learning Representations (ICLR)*. openreview.net/forum?id=WE_vluYUL-X
26. Schick, T.; Dwivedi-Yu, J.; Dessì, R.; Raileanu, R.; et al. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. *Advances in Neural Information Processing Systems 36 (NeurIPS)*. arXiv:2302.04761
27. Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. (2023). Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint*. arXiv:2305.16291
28. Park, J. S.; O'Brien, J. C.; Cai, C. J.; Morris, M. R.; Liang, P.; Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST)*. doi:10.1145/3586183.3606763

 doi:10.20944/preprints202605.1992.v1

29. Kapoor, S.; Widder, D. G.; Ensmenger, N.; Narayanan, A. (2025). Can We Trust AI Benchmarks? An Interdisciplinary Review of Current Issues in AI Evaluation. *arXiv preprint*. arXiv:2502.06559
30. Wei, J.; Tay, Y.; Bommasani, R.; Raffel, C.; et al. (2022). Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research (TMLR)*.
31. Schaeffer, R.; Miranda, B.; Koyejo, S. (2023). Are Emergent Abilities of Large Language Models a Mirage? *Advances in Neural Information Processing Systems 36 (NeurIPS)*. arXiv:2304.15004
32. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. (2018). Deep Reinforcement Learning That Matters. *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*. doi:10.1609/aaai.v32i1.11694
33. Maslej, N.; Fattorini, L.; Perrault, R.; et al. (2025). *The AI Index 2025 Annual Report*. Stanford Institute for Human-Centered Artificial Intelligence.
34. Wilkinson, M. D.; Dumontier, M.; Aalbersberg, I. J.; Appleton, G.; et al. (2016). The FAIR Guiding Principles for Scientific Data Management and Stewardship. *Scientific Data*, **3**(1), 160018. doi:10.1038/sdata.2016.18
35. Stodden, V.; McNutt, M.; Bailey, D. H.; Deelman, E.; Gil, Y.; Hanson, B.; Heroux, M. A.; Ioannidis, J. P. A.; Taufer, M. (2016). Enhancing Reproducibility for Computational Methods. *Science*, **354**(6317), 1240–1241. doi:10.1126/science.aah6168
36. Sandve, G. K.; Nekrutenko, A.; Taylor, J.; Hovig, E. (2013). Ten Simple Rules for Reproducible Computational Research. *PLoS Computational Biology*, **9**(10), e1003285. doi:10.1371/journal.pcbi.1003285
37. Smith, A. M.; Katz, D. S.; Niemeyer, K. E.; FORCE11 Software Citation Working Group (2016). Software Citation Principles. *PeerJ Computer Science*, **2**, e86. doi:10.7717/peerj-cs.86
38. Peng, R. D. (2011). Reproducible Research in Computational Science. *Science*, **334**(6060), 1226–1227. doi:10.1126/science.1213847
39. Donoho, D. L. (2010). An Invitation to Reproducible Computational Research. *Biostatistics*, **11**(3), 385–388. doi:10.1093/biostatistics/kxq028
40. National Academies of Sciences, Engineering, and Medicine (2019). *Reproducibility and Replicability in Science*. The National Academies Press, Washington, DC. doi:10.17226/25303
41. Association for Computing Machinery (2020). *Artifact Review and Badging (Version 1.1)*. ACM Policy Document. acm.org/publications/policies/artifact-review-and-badging-current
42. Katz, D. S.; Gruenpeter, M.; Honeyman, T. (2021). Taking a Fresh Look at FAIR for Research Software. *Patterns*, **2**(3), 100222. doi:10.1016/j.patter.2021.100222
43. Hutchinson, B.; Rostamzadeh, N.; Greer, C.; Heller, K.; Prabhakaran, V. (2022). Evaluation Gaps in Machine Learning Practice. *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. doi:10.1145/3531146.3533233
44. National Institute of Standards and Technology (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1. doi:10.6028/NIST.AI.100-1
45. National Institute of Standards and Technology (2024). *Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile*. NIST AI 600-1. doi:10.6028/NIST.AI.600-1
46. European Parliament and Council of the European Union (2024). Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, L 2024/1689. eur-lex.europa.eu/eli/reg/2024/1689
47. Bengio, Y.; Clare, S.; Prunkl, C.; Murray, M.; et al. (2026). *International AI Safety Report 2026*. Department for Science, Innovation and Technology (DSIT).
48. Anderljung, M.; Barnhart, J.; Korinek, A.; Leung, J.; O'Keefe, C.; Whittlestone, J.; et al. (2023). Frontier AI Regulation: Managing Emerging Risks to Public Safety. *arXiv preprint*. arXiv:2307.03718
49. Shevlane, T.; Farquhar, S.; Garfinkel, B.; Phuong, M.; et al. (2023). Model Evaluation for Extreme Risks. *arXiv preprint*. arXiv:2305.15324
50. Phuong, M.; Aitchison, M.; Catt, E.; Cogan, S.; et al. (2024). Evaluating Frontier Models for Dangerous Capabilities. *arXiv preprint*. arXiv:2403.13793
51. Reed, M.; Simon, B. (1980). *Methods of Modern Mathematical Physics I: Functional Analysis*, revised and enlarged ed. Academic Press. ISBN 978-0-12-585050-6.
52. Kato, T. (1995). *Perturbation Theory for Linear Operators* (Reprint of 1980 ed.). Springer-Verlag. ISBN 978-3-540-58661-6.
53. Halmos, P. R. (1982). *A Hilbert Space Problem Book*, 2nd ed. Springer-Verlag. ISBN 978-0-387-90685-0.
54. Bhatia, R. (1997). *Matrix Analysis*. Springer. ISBN 978-0-387-94846-1.

 doi:10.20944/preprints202605.1992.v1

55. Reed, J. F.; Lynn, F.; Meade, B. D. (2002). Use of Coefficient of Variation in Assessing Variability of Quantitative Assays. *Clinical and Diagnostic Laboratory Immunology*, **9**(6), 1235–1239. doi:10.1128/CDLI.9.6.1235-1239.2002