

Article

Not peer-reviewed version

To Lay a Stone with Six Birds: Finite-State Semantics for Packaging, Directionality, and Coarse-Graining

[Ioannis Tsiokos](#) *

Posted Date: 27 February 2026

doi: 10.20944/preprints202602.1699.v1

Keywords: emergence calculus; minimal substrate; coarse-graining; stochastic dynamics; irreversibility



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

To Lay a Stone with Six Birds: Finite-State Semantics for Packaging, Directionality, and Coarse-Graining

Ioannis Tsiokos 

Automorph Inc., Wilmington, DE 19806, USA; ioannis@automorph.io

Abstract

Work on multi-scale organization and audit-based directionality lacks a minimal-substrate theorem: key primitives are often realized only by smuggling in external schedules, ad hoc audits, or declared quotients, making results hard to verify and compare across substrates. We introduce a no-smuggling minimal-substrate semantics for SPT, operationalize it via matched-control audit tests on finite stochastic machines, and demonstrate separations between protocol artifacts, intrinsic drive, and route-dependent packaging defects. Operationally, we define a canonical machine class S_{\min} of autonomous finite stochastic dynamics with phase-in-state, deterministic lenses, induced packaging endomaps $E_{\tau,f}$, and intrinsic audits (path-reversal KL asymmetry Σ_T and ACC affinities), and we implement exact computations together with a reproducible witness suite and a Lean-checked idempotent/closure backbone for packaging equivalence and saturation (de Moura and Ullrich 2021; The mathlib Community 2020). Under calibrated controls, stroboscopic protocols can exhibit $\Sigma_T > 0$ while their autonomous lifts yield $\Sigma_T \approx 0$ (protocol trap), whereas turning on internal phase drive produces a clean null-vs-driven separation with $\Sigma_T(Z) > 0$ and a detectable projected macro arrow; moreover deterministic coarse-graining never increases Σ_T and can erase it entirely, and packaging along refinement routes exhibits a prototype-driven holonomy defect that is large at moderate timescales while a uniform baseline is near zero and satisfies a gain-plus-mismatch bound. These results supply a machine semantics and separation witnesses for a minimal substrate theorem, reducing false positives and turning SPT from a language into a compilation target up to packaging equivalence. We are explicit that we do not establish a universal physical substrate or continuum limit: our conclusions are semantic and rest on exact finite-state audits and controlled constructions rather than empirical natural-system data.

Keywords: emergence calculus; minimal substrate; coarse-graining; stochastic dynamics; irreversibility

1. Introduction

SPT was introduced in Ioannis (2026) as an emergence calculus organized around six primitives (P1–P6). The Six Birds thesis is that these primitives are not optional modeling choices: they are forced by composability, limited access, and refinement. This paper takes the next step and asks a core semantic question:

What is the weakest mathematical substrate on which all six primitives can be realized without smuggling them in?

The phrase *without smuggling* is the key constraint. A general emergence calculus should not rely on external schedules, externally declared quotients, or ad hoc audit functionals that are not induced by the substrate itself. If SPT is to be more than a descriptive language, it needs a minimal *machine semantics*: a canonical representation class and a sense in which any admissible realization can be compiled into it.

1.1. From an Emergence Language to a Substrate Machine

The Six Birds framework is intentionally substrate-agnostic: it identifies primitives that must appear whenever agents interact with dynamics through limited interfaces. That substrate-agnostic stance is a strength, but it creates a natural ambiguity: if the primitives are “forced,” then *forced on what?* In other words, what is the minimal structure that supports all of the following *internally*: (i) staged access via lenses, (ii) packaging as an endomap with stable fixed points, (iii) intrinsic audits of directionality and drive, and (iv) route-dependent composition defects (holonomy) when packaging is performed along different refinement routes?

This paper answers by proposing a canonical minimal machine model: an autonomous finite stochastic dynamics on an explicit extended state, equipped with deterministic coarse-grainings (lenses), internal packaging endomorphisms, and intrinsic audits. The goal is not to privilege Markov chains as physics; rather, finite autonomous stochastic dynamics provide a compact normal form in which the SPT primitives can be made explicit and testable. Our representation goal is therefore semantic: *give a minimal substrate class that can host P1–P6, and show which features are necessary.*

1.2. No-Smuggling Semantics

We make the “no smuggling” constraint operational by treating the primitives as *realizability requirements*. A primitive is *realized* only if it arises from internal structure on the substrate: endomaps, constraints, lenses, and induced audits. Conversely, we call a construction *smuggled* if it requires external oracular input not represented in state or in the admitted structure. Three recurring smuggling modes motivate our minimality claims.

External protocols masquerading as autonomy.

If a system is driven by an externally staged schedule (“apply K_0 , then K_1 , then K_0, \dots ”), stroboscopic composition can generate apparent arrows of time that disappear once the schedule is encoded into state and the dynamics is made autonomous. This is the *protocol trap* from [Ioannis \(2026\)](#). In our witness suite, a stroboscopic Markov closure on a microstate space X exhibits $\Sigma_T > 0$ (e.g. $\Sigma_T = 0.173783$ under the experiment settings), yet the autonomous lifted system on $Z = X \times \Phi$ yields $\Sigma_T = 0$, and the exact macro audit induced by the lifted chain yields $\Sigma_T = 6.93335 \times 10^{-33}$. The lesson is semantic: an admissible substrate must represent protocol state internally (phase-in-state) if it is to claim autonomy.

External directionality audits.

Directionality is not a free label; it must be audited in a way that respects limited access. In particular, if $f : Z \rightarrow X$ is a deterministic coarse-graining, an admissible audit should not report *more* arrow of time after information is discarded. This monotonicity is a no-false-positives constraint. We adopt the path-space KL asymmetry Σ_T as a canonical audit ([Cover and Thomas 2006](#); [Kullback and Leibler 1951](#)) and empirically verify data processing: $\Sigma_T^{\text{macro}} \leq \Sigma_T^{\text{micro}}$ across randomized test cases. The witness suite also shows that genuine micro irreversibility can be invisible at the macro level: for a biased 4-cycle we observe $\Sigma_T^{\text{micro}} = 10.5467$ while a parity lens yields $\Sigma_T^{\text{macro}} = 0$.

Externally declared quotients and objects.

Packaging is the mechanism by which stable objects emerge from dynamics and observation. In Six Birds, packaging appears as a concrete operator $E_{\tau,f}$ with defect measures; abstractly, packaging behaves like an idempotent endomap whose fixed points are the packaged objects. Declaring macro-objects by fiat is smuggling; admissible substrates must induce packaging equivalence internally (“two states are equivalent if they package to the same representative”) and support composition and refinement in that internal language. A small Lean development in this project mechanizes the idempotent and closure algebra (fixed points, packaging equivalence, and lens saturation as a closure operator) ([de Moura and Ullrich 2021](#); [The mathlib Community 2020](#)).

1.3. Drive Restores Genuine Arrows

The protocol trap makes a negative point: some apparent arrows are artifacts of external scheduling. The complementary positive point is equally important for minimality: to obtain a genuine arrow in an autonomous substrate, one needs internal *drive* ($P6_{\text{drive}}$ in the Six Birds taxonomy), detectable for example by a nonzero ACC affinity class (Schnakenberg 1976; Seifert 2012). Our driven-protocol witness makes this explicit. With a driven phase register Φ (nonreversible internal phase dynamics), the autonomous lifted system yields a strong arrow $\Sigma_T = 2.10934$, and the projected macro process becomes directionally biased as well ($\Sigma_T = 0.00638045$), whereas a reversible (null) phase control yields $\Sigma_T = 0$ and $\Sigma_T = 1.00148 \times 10^{-32}$. This sharp separation supports a minimality claim: autonomy plus protocols do not imply arrows; arrows require internal drive.

1.4. What This Paper Adds and How to Read It

Relative to Ioannis (2026), this paper contributes a machine semantics and a minimal-substrate perspective.

- We define a no-smuggling realizability criterion and propose a canonical minimal machine substrate in which P1–P6 are implemented as internal operators and constraints.
- We state a representation/compilation perspective: admissible substrates simulate each other up to packaging equivalence by passage through the canonical machine model.
- We state minimality results showing that removing specific structural features forces at least one primitive to fail or to re-enter as an external oracle.
- We provide a reproducible witness suite (protocol trap, driven protocols, hidden irreversibility, refinement non-monotonicity, and route mismatch/holonomy).

The rest of the paper is organized as follows. Section 2 summarizes the Six Birds foundation. Section 3 introduces the no-smuggling realizability vocabulary (lenses, packaging equivalence, and audits). Section 4 presents the canonical minimal machine substrate and the internal realizations of P1–P6. Section 5 states the main representation and minimality theorems. Section 6 describes the witness suite, including route mismatch at moderate time scales (e.g. $\tau = [6, 3, 3]$ under prototype scheme `f0_rep03_vs_rep25_f1_uniform`) with mismatch norm 2 versus a uniform baseline of 5.02202×10^{-16} , and an empirically verified gain-plus-mismatch bound with slack 2. Section 7 distills these results into a user-facing audit workflow and a failure-mode checklist. Appendices collect reproducibility notes (Appendix A) and the Lean algebraic backbone (Appendix B).

2. From Six Birds to a Minimal Substrate Question

This paper is a direct extension of the Six Birds foundation (Ioannis 2026). Six Birds introduced SPT as an emergence calculus organized around six primitives (P1–P6) and argued that these primitives are structurally forced by limited access and refinement. Here we summarize only the components we will reuse as modules: (i) the “theory package” interface (lens, completion, audit), (ii) canonical packaging operators and their defects, (iii) intrinsic audits of directionality with monotonicity under coarse-graining, and (iv) the autonomy discipline behind the protocol trap. We then state precisely what the present paper adds.

2.1. Six Primitives and the Separation of Certificates

A central design choice in Ioannis (2026) is to keep three certificate families logically independent: *stability* (existence/robustness of packaged fixed points), *novelty* (strict theory extension via non-definable predicates), and *directionality* (auditable arrows of time or drive). This separation prevents category mistakes such as treating route dependence (holonomy) as a directionality certificate, or treating stability as evidence of novelty. The present paper keeps the same separation, but asks a new question: what minimal substrate must exist so that each certificate family can be realized internally and cannot be faked by external oracular additions?

2.2. Theory Packages: Lens, Completion, and Audit

Six Birds formalizes a “theory at a scale” as a package consisting of (i) a carrier of microdescriptions, (ii) a deterministic coarse-graining (lens) that defines what is expressible, (iii) a completion/packaging rule whose fixed points are the packaged objects at that scale, and (iv) an audit functional that is monotone under further coarse-graining. The lens is a deterministic map $f : Z \rightarrow X$ (finite sets in the base development), and it induces a notion of definability: a predicate $h : Z \rightarrow \{0, 1\}$ is definable from f iff it is constant on fibers of f . Novelty is then modeled as strict theory extension by adjoining a non-definable predicate, and a finite forcing lemma shows that non-definability is generic in the finite setting (Ioannis 2026).

Completion is intentionally treated abstractly as an endomap E that is (exactly or approximately) idempotent, so that $\text{Fix}(E)$ behaves like a set of packaged objects. This view does not require an ambient order. It is also compatible with order-theoretic closures when an order is present. In the present paper, this abstraction becomes one of the key semantic constraints: packaged objects must arise as fixed points of an internally constructed endomap, not from externally declared quotients.

2.3. Canonical Empirical Packaging via Dynamics and a Lens

Six Birds gives a concrete, reusable packaging construction from autonomous finite dynamics. Let P be a time-homogeneous Markov kernel on a finite microstate set Z , let $f : Z \rightarrow X$ be a deterministic lens, and fix a timescale $\tau \geq 1$. Define the pushforward $Q_f : \Delta(Z) \rightarrow \Delta(X)$ by coarse-graining a distribution and a lift $U_f : \Delta(X) \rightarrow \Delta(Z)$ by selecting prototypes u_x supported on each fiber $f^{-1}(x)$. The empirical packaging endomap is then

$$E_{\tau, f}(\mu) := U_f(Q_f(\mu P^\tau)). \quad (1)$$

This endomap “evolve–forget–reinststantiate” construction produces packaged representatives at scale (f, τ) . Six Birds develops quantitative defect measures (e.g. total-variation idempotence defect and prototype stability scores) and emphasizes a key semantic warning: saturation diagnostics (small idempotence defect) do not by themselves certify multiple objects (a constant map is perfectly idempotent). The present paper reuses the construction (1) as the canonical way to realize P5 internally in a minimal machine substrate.

2.4. Audits of Directionality: KL Arrow and ACC Drive

Six Birds supplies two complementary intrinsic audits of directionality.

Path-reversal asymmetry and data processing.

For a finite Markov chain with initial distribution ρ and horizon T , Six Birds defines the path-space reversal asymmetry

$$\Sigma_T(\rho) := D_{\text{KL}}(\mathbb{P}_{\rho, T} \parallel \mathcal{R}_* \mathbb{P}_{\rho, T}),$$

where D_{KL} denotes the Kullback–Leibler divergence (Kullback and Leibler 1951), and proves a data-processing inequality under deterministic coarse-graining: if $f : Z \rightarrow X$ is a lens, then the observed macro path law cannot have larger reversal asymmetry than the micro path law (Cover and Thomas 2006). This monotonicity is a “no false positives” constraint: discarding information cannot create an arrow of time. Our present paper treats such monotonicity as part of the no-smuggling semantics for audits (P6): admissible directionality certificates must be induced by the substrate and must not be amplifiable by forgetting. This viewpoint is compatible with stochastic-thermodynamic interpretations of irreversibility (Lebowitz and Spohn 1999; Seifert 2012).

ACC affinity classes ($P_{6_{\text{drive}}}$).

Under bidirected support (microreversibility on support), Six Birds defines an antisymmetric edge 1-form

$$a_{ij} := \log \frac{P_{ij}}{P_{ji}},$$

and the cycle integral (affinity) on an oriented cycle as the sum of a_{ij} along the cycle. Nonzero cycle integrals are the obstruction to exactness and witness internal drive (Kelly 1979; Schnakenberg 1976). This is the $P_{6_{\text{drive}}}$ specialization: a basis-independent nonzero affinity class indicates a sustained nonequilibrium drive that cannot be removed by reparameterization. The present paper uses this audit both as a semantic ingredient (a substrate feature that can realize genuine arrows) and as a minimality witness (what must be present to avoid protocol artifacts).

2.5. The Protocol Trap and Autonomy Discipline

A crucial point in Ioannis (2026) is that protocols must be represented autonomously if the substrate claims to be autonomous. A stroboscopic description on X (“apply K_0 then K_1 then ...”) is non-autonomous unless the protocol phase is included in state. Six Birds constructs an autonomous lifted system on $Z = X \times \Phi$ with time-homogeneous dynamics in which the phase Φ evolves internally. In the null (reversible) regime, apparent stroboscopic arrows on X vanish when the dynamics is made autonomous on Z . This is the protocol trap: directionality inferred from externally scheduled composition is not an intrinsic substrate property. Our present paper elevates this to a no-smuggling axiom: if a substrate uses protocol staging, then the phase must live in state, and directionality must be audited on the autonomous system.

2.6. What This Paper Adds

Six Birds showed that the primitives are structurally forced by limited access and refinement and supplied canonical constructions for packaging and audits. The present paper asks and answers a semantic minimality question that Six Birds did not attempt:

- We define a no-smuggling realizability criterion and propose a canonical minimal machine substrate (autonomous finite stochastic dynamics on an explicit extended state) in which P1–P6 are realized internally.
- We state a representation/compilation viewpoint: admissible substrates simulate each other up to packaging equivalence by passage through the canonical machine model.
- We state minimality results showing that removing specific substrate features forces at least one primitive to fail or to re-enter as an external oracle (smuggling).
- We provide a reproducible witness suite that sharpens the protocol trap into a minimality test and shows that internal drive ($P_{6_{\text{drive}}}$) restores genuine arrows in autonomous protocols.

The next section (Section 3) upgrades these ideas into a no-smuggling vocabulary: substrates, realizations, lenses and refinement, packaging equivalence, and admissible audits.

3. Definitions: Substrates, Realizations, and No-Smuggling Semantics

This section fixes notation and states the semantic discipline that will underlie our minimal substrate theorems. The guiding idea is realizability: a primitive is realized only if it arises from *internal* structure on the substrate, not from external schedules or externally declared oracles.

3.1. Substrates and Realizations

A *substrate* is a mathematical object intended to host realizations of the SPT primitives. In this paper we keep the definition modular and intentionally weak.

Definition 1 (Substrate (informal)). A substrate \mathcal{S} consists of: (i) a state carrier Z (finite in our canonical machine model), (ii) a composable dynamics primitive (autonomous in the canonical model), and (iii) an admissible family of deterministic observational maps (lenses) from Z to coarser description spaces. Additional internal structure may be present (e.g. registers, ledgers, prototype selectors), but must be represented as part of state or as derived operators.

We will often speak of a *realization* of primitives P1–P6 on a substrate.

Definition 2 (Realization (informal)). A realization of a primitive on \mathcal{S} is a specification of internal operators and constraints on \mathcal{S} that instantiate the primitive without introducing external oracles. Concretely, realizations may be given by endomaps on distributions or predicates, deterministically induced coarse-grainings, internal audit functionals, and internal composition rules.

The point of these informal definitions is not to fix a single foundational framework, but to isolate the no-smuggling criterion below.

3.2. No-Smuggling Semantics

We distinguish between internal structure (admissible) and external oracular input (smuggling).

Definition 3 (No smuggling). A construction smuggles a primitive if it requires information or structure not representable within the substrate state and its admitted operators. Equivalently, a primitive is realized without smuggling if it is definable using only: (i) the substrate state carrier, (ii) the admitted compositional dynamics, (iii) deterministic lenses and their induced constructions, and (iv) internal endomaps/audits constructed from the above. External schedules, externally declared equivalence relations, and externally imposed audits that are not induced by the substrate are disallowed.

Remark 1 (Operational reading). A convenient way to read Definition 3 is syntactic: treat a substrate as presenting a signature of admissible constructors (state, dynamics, lenses, induced linear maps, endomaps, and audits), and treat “without smuggling” as definability in the term algebra generated by that signature. We do not develop a full syntactic metatheory here; the witness suite and the theorems in Section 5 use only this disciplined notion of “no external oracles.”

We will use this notion operationally: a minimality theorem will exhibit a substrate feature whose removal forces a primitive to fail *unless* it is smuggled back in.

3.3. Lenses, Refinement, and Definability

A lens is a deterministic coarse-graining.

Definition 4 (Lens). A lens is a deterministic map $f : Z \rightarrow X$. We interpret f as the observational interface: two microstates $z, z' \in Z$ are observationally indistinguishable at scale f if $f(z) = f(z')$.

The lens induces a natural notion of definability of predicates (“questions”) on Z .

Definition 5 (Definability from a lens). A predicate $h : Z \rightarrow \{0, 1\}$ is definable from a lens $f : Z \rightarrow X$ if h is constant on fibers of f , i.e. if $f(z) = f(z')$ implies $h(z) = h(z')$. Equivalently, there exists $\tilde{h} : X \rightarrow \{0, 1\}$ such that $h = \tilde{h} \circ f$.

Refinement is defined by factorization.

Definition 6 (Refinement). A lens $f_2 : Z \rightarrow X_2$ refines $f_1 : Z \rightarrow X_1$ if there exists a deterministic map $g : X_2 \rightarrow X_1$ such that $f_1 = g \circ f_2$. We write $f_2 \succeq f_1$.

Refinement is the semantic notion of “more expressive observation”. In Six Birds, strict refinement supports novelty by adjoining a predicate non-definable at the coarser scale (Ioannis 2026).

3.4. Packaging Endomaps and Packaging Equivalence

Packaging is modeled abstractly as an endomap whose fixed points represent packaged objects.

Definition 7 (Packaging endomap). *A packaging endomap is a function $e : \alpha \rightarrow \alpha$ (for an appropriate carrier α) that is exactly or approximately idempotent. Exact idempotence is the equation $e \circ e = e$. When α is a set of descriptions, $\text{Fix}(e) = \{x \in \alpha : e(x) = x\}$ are the packaged objects.*

Packaging induces an internal equivalence relation, which we call *packaging equivalence*.

Definition 8 (Packaging equivalence). *Given $e : \alpha \rightarrow \alpha$, define a relation \sim_e on α by*

$$x \sim_e y \iff e(x) = e(y).$$

We call \sim_e the packaging equivalence induced by e .

The Lean development in this project proves that \sim_e is an equivalence relation and that $x \sim_e e(x)$, capturing the idea that every state is equivalent to its package.

3.5. Packaging-Respecting Maps and “Up to Packaging” Comparison

To make statements like “simulate up to packaging equivalence” precise without committing to a full categorical development, we use a minimal notion of structure preservation.

Definition 9 (Packaging presentation and packaging-respecting map). *A packaging presentation is a pair (α, e) consisting of a carrier α and an endomap $e : \alpha \rightarrow \alpha$. Given two presentations (α, e) and (β, e') , a map $h : \alpha \rightarrow \beta$ is packaging-respecting if it commutes with packaging:*

$$h \circ e = e' \circ h.$$

Lemma 1 (Packaging-respecting maps preserve packaging equivalence). *If h is packaging-respecting, then $x \sim_e y$ implies $h(x) \sim_{e'} h(y)$.*

Proof. If $e(x) = e(y)$, then applying h and using $h \circ e = e' \circ h$ gives $e'(h(x)) = e'(h(y))$. \square

Remark 2 (What is and is not claimed). *A packaging-respecting map is the minimum structure needed for “up to packaging equivalence” reasoning. We do not claim that packaging equivalence classes are invariant under arbitrary changes of packaging structure (e.g. changing prototypes can change the endomap and therefore its equivalence relation); such dependence is part of what route mismatch diagnostics quantify.*

3.6. Audits and Monotonicity Under Coarse-Graining

Audits are functionals intended to certify directionality or drive. The no-smuggling discipline requires audits to be induced by the substrate and to respect limited access.

Definition 10 (Audit monotonicity). *Let A be an audit functional that assigns a nonnegative number to a process observed at a given scale. We say A is monotone under deterministic coarse-graining if for any lens $f : Z \rightarrow X$, the audited value at the macro scale is no larger than at the micro scale:*

$$A(\text{process on } X) \leq A(\text{process on } Z).$$

A canonical instance is the path-reversal KL asymmetry Σ_T from Ioannis (2026), which satisfies a data-processing inequality (Cover and Thomas 2006) under deterministic coarse-graining. In our

minimal substrate program, such monotonicity is not merely a fact about KL; it is a semantic guardrail against false positives: forgetting information should not create directionality certificates.

3.7. Where These Definitions Are Used

Section 4 introduces a canonical minimal machine substrate in which: lenses are explicit deterministic maps, packaging is realized by an internal operator $E_{\tau,f}$, and audits include Σ_T and ACC affinity classes. Section 5 then states representation and minimality theorems in this vocabulary.

4. The Canonical Minimal Machine Substrate

This section defines a concrete canonical substrate class that will serve as a minimal machine semantics for SPT. The purpose is semantic rather than ontological: finite autonomous stochastic dynamics provide a compact normal form in which lenses, internal packaging endomaps, and intrinsic audits can all be made explicit and checked. The construction is a direct operationalization of the Six Birds program (Ioannis 2026) under the no-smuggling discipline of Section 3.

4.1. Carrier and Dynamics

Fix a finite state carrier Z . A canonical machine substrate is equipped with an autonomous time-homogeneous dynamics on Z , represented as a Markov kernel (Norris 1997)

$$P : Z \times Z \rightarrow [0, 1], \quad \sum_{z' \in Z} P(z, z') = 1 \text{ for all } z \in Z.$$

We use the convention that distributions are row vectors $\mu \in \Delta(Z)$ and evolve by right-multiplication: $\mu \mapsto \mu P$ and $\mu \mapsto \mu P^\tau$. Autonomy means that P does not depend on an external time index. Any externally scheduled protocol must therefore be represented internally, typically by enlarging state (phase-in-state).

Extended state (phase/ledger).

In many uses (including our witness suite), it is natural to factor the carrier as an explicit product

$$Z \cong X \times \Phi \times L,$$

where X is a “microstate” register, Φ is a protocol/phase register, and L is an optional ledger/register used to make internal bookkeeping explicit. This factorization is not an additional assumption—any finite Z can encode such registers—but it makes the no-smuggling intent concrete: if a protocol stage matters, it must live in Φ (or in an equivalent internal component of Z).

4.2. Lenses and Refinement as Deterministic Maps

A *lens* is a deterministic coarse-graining $f : Z \rightarrow X$ in the sense of Definition 4. A canonical machine substrate comes with an admissible family of such lenses, often organized into refinement chains

$$f_2 \succeq f_1 \succeq f_0 \text{ meaning } f_0 = g_{01} \circ f_1, \quad f_1 = g_{12} \circ f_2$$

for some deterministic maps g_{01}, g_{12} (Definition 6). These chains represent staged access: at each stage one may observe only through the currently admitted lens.

4.3. Internal Packaging as an Endomap

Packaging is realized as an internal endomap on distributions induced by (P, f, τ) and a choice of prototypes. Fix a lens $f : Z \rightarrow X$ and a timescale $\tau \geq 1$.

Pushforward (coarse-grain).

The lens induces a pushforward $Q_f : \Delta(Z) \rightarrow \Delta(X)$ by

$$(Q_f(\mu))(x) = \sum_{z: f(z)=x} \mu(z).$$

Lift (prototype reinjection).

To map macrodescriptions back to microdescriptions, we choose for each $x \in X$ a prototype distribution $u_x \in \Delta(Z)$ supported on the fiber $f^{-1}(x)$. This defines a linear lift $U_f : \Delta(X) \rightarrow \Delta(Z)$ by

$$(U_f(v))(\cdot) = \sum_{x \in X} v(x) u_x(\cdot).$$

A canonical default is the uniform distribution on each fiber, but other prototype choices encode different “representatives” of the same observational class.

Definition 11 (Empirical packaging endomap). *Given (P, f, τ) and prototypes U_f , define the empirical packaging endomap $E_{\tau, f} : \Delta(Z) \rightarrow \Delta(Z)$ by*

$$E_{\tau, f}(\mu) := U_f(Q_f(\mu P^\tau)). \quad (2)$$

Equivalently, in matrix form (with Q the $Z \times X$ lens matrix and U the $X \times Z$ prototype matrix),

$$E_{\tau, f} = P^\tau Q U, \quad \mu \mapsto \mu E_{\tau, f}.$$

In the ideal regime, $E_{\tau, f}$ is idempotent and its fixed points are the packaged objects at scale (f, τ) . In practice, idempotence may hold approximately, and Six Birds develops defect diagnostics (e.g. total-variation idempotence defect and prototype stability scores) to measure packaging quality (Ioannis 2026). Our codebase implements these diagnostics as part of the canonical machine semantics.

Packaging equivalence.

Any endomap e induces packaging equivalence $x \sim_e y \iff e(x) = e(y)$ (Definition 8). In the canonical machine model, packaging equivalence is an internal relation on the chosen carrier (e.g. on distributions, or on microdescriptions) and is used as the notion of “simulation up to packaging” in our representation claims.

Route mismatch (holonomy defect).

When lenses form a chain, packaging may be performed along different routes. For example, with $f_2 \succeq f_1 \succeq f_0$, one can package directly at f_0 or package via f_1 and then f_0 . The canonical machine substrate supports an internal mismatch diagnostic using any induced operator norm:

$$\text{RM} := \|\text{Pack}_{0 \leftarrow 2} - \text{Pack}_{1 \leftarrow 2} \circ \text{Pack}_{0 \leftarrow 1}\|.$$

Our witness suite includes prototype-driven cases where route mismatch is large at moderate timescales, while the uniform-prototype baseline has near-zero mismatch. This separation is important semantically: holonomy defects need not be confused with directionality.

4.4. Intrinsic Audits: KL Arrow and ACC Drive

A canonical machine substrate provides intrinsic audits of directionality and drive that are induced by (ρ, P) and are monotone under deterministic coarse-graining (Definition 10).

Path-reversal asymmetry.

Given an initial distribution $\rho \in \Delta(Z)$ and horizon T , the path-space KL asymmetry

$$\Sigma_T(\rho) := D_{\text{KL}}(\mathbb{P}_{\rho,T} \parallel \mathcal{R}_* \mathbb{P}_{\rho,T})$$

is the canonical “arrow” audit from [Ioannis \(2026\)](#); see also ([Cover and Thomas 2006](#); [Kullback and Leibler 1951](#); [Seifert 2012](#)).

Proposition 1 (Markov-chain expansion and finiteness of Σ_T). *Let P be a time-homogeneous Markov kernel on a finite set Z , let $\rho \in \Delta(Z)$, and let $(Z_0, \dots, Z_T) \sim \mathbb{P}_{\rho,T}$. If $P_{Z_{t+1}Z_t} > 0$ whenever (Z_t, Z_{t+1}) occurs with positive probability (in particular, if P has bidirected support), then*

$$\Sigma_T(\rho) = \mathbb{E} \left[\log \frac{\rho(Z_0)}{\rho(Z_T)} \right] + \sum_{t=0}^{T-1} \mathbb{E} \left[\log \frac{P_{Z_t Z_{t+1}}}{P_{Z_{t+1} Z_t}} \right]. \quad (3)$$

If the event $\{P_{Z_{t+1}Z_t} = 0\}$ occurs with positive probability for some t , then $\Sigma_T(\rho) = +\infty$.

Remark 3 (Reversibility and interpretation guardrail). *If $\rho = \pi$ is stationary and P satisfies detailed balance with respect to π , then $\Sigma_T(\pi) = 0$ for all T . In nonequilibrium steady state, $\Sigma_T(\pi)$ typically grows linearly in T and is closely related (up to conventions) to trajectory-level entropy production in stochastic thermodynamics ([Lebowitz and Spohn 1999](#); [Seifert 2012](#)). In this paper, however, Σ_T is used only as an information-theoretic directionality audit; we do not assume local detailed balance or a physical reservoir interpretation.*

Exact macro path law for a deterministic lens.

Let $f : Z \rightarrow X$ be a deterministic lens. Write E_x for the diagonal selector matrix with $(E_x)_{zz} = \mathbf{1}_{\{f(z)=x\}}$ and let $\mathbf{1}$ be the all-ones column vector. For any macro sequence $x_{0:T} = (x_0, \dots, x_T) \in X^{T+1}$, the induced macro path probability is

$$\mathbb{P}_{\rho,T}^f(x_{0:T}) = \rho E_{x_0} P E_{x_1} \cdots P E_{x_T} \mathbf{1}. \quad (4)$$

Equation (4) is the standard forward recursion for a hidden Markov model with deterministic emissions; we use it to compute Σ_T^{macro} exactly in the finite witness suite ([Rabiner 1989](#)). Replacing the induced macro path law by a one-step Markov closure (e.g. compressing a multi-phase protocol into a single kernel on X) can introduce spurious arrows; this is precisely the protocol trap. Given a lens $f : Z \rightarrow X$, we can also evaluate the macro-level asymmetry Σ_T^{macro} on the pushed-forward path law; data processing implies $\Sigma_T^{\text{macro}} \leq \Sigma_T^{\text{micro}}$. Our implementation computes both quantities exactly for finite chains, and uses this monotonicity as a semantic guardrail against false positives.

ACC affinities (drive).

Under bidirected support (i.e. $P_{ij} > 0$ and $P_{ji} > 0$ on each support edge), define the antisymmetric 1-form $a_{ij} = \log(P_{ij}/P_{ji})$ and its cycle integrals (affinities) on a fundamental cycle basis ([Kolmogoroff 1936](#)). Nonzero cycle integrals witness a non-exact affinity class and therefore an internal nonequilibrium drive ($P6_{\text{drive}}$) ([Schnakenberg 1976](#)). This audit distinguishes genuine driven arrows from protocol artifacts: autonomy plus staging does not imply arrows; arrows require internal drive ([Kelly 1979](#)).

4.5. Where P1–P6 Live Inside the Canonical Machine

The canonical machine substrate is designed so that the Six Birds primitives are realized as internal structure rather than external oracles. At a high level:

- **P1 (dynamics/composition):** realized by autonomous state evolution on Z via the time-homogeneous kernel P and composition by multiplication (P^T , concatenation of stages by kernel products).

- **P2 (limited access / interface):** realized by deterministic lenses $f : Z \rightarrow X$ and the induced definability constraints (predicates constant on fibers).
- **P3 (protocols / staged interaction):** realized by explicit internal protocol state (phase-in-state) as a component of Z (e.g. Φ) so that staged protocols become autonomous dynamics on Z .
- **P4 (staging/timescale):** realized by the explicit timescale parameter τ and refinement chains of lenses that control which descriptions are admitted at which stage.
- **P5 (packaging/objects):** realized by the internal packaging endomap $E_{\tau,f}$ and its fixed-point/defect calculus (idempotence defect, retention error, prototype stability).
- **P6 (audits of directionality/drive):** realized by intrinsic audits induced by the autonomous dynamics (path-reversal KL asymmetry Σ_T and ACC affinity classes), with monotonicity under coarse-graining as a semantic constraint.

This mapping is intentionally minimal: we do not assume an ambient geometry, an order structure, or a continuum limit. Instead, the machine semantics makes explicit the minimal operators that will appear in the main theorems (Section 5) and that are exercised by the witness suite (Section 6).

4.6. A Schematic View

Figure 1 summarizes the components and data flow defined in this section.

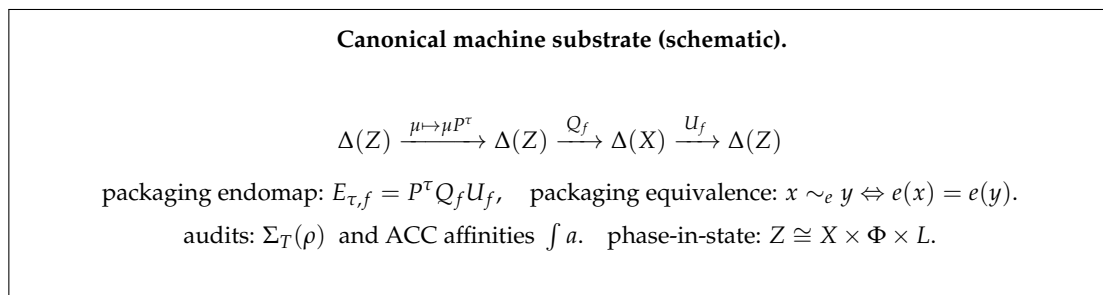


Figure 1. Schematic of the canonical minimal machine substrate. The diagram is structural: all operators are internal and induced by (Z, P) plus deterministic lenses and prototype choices.

5. Main Theorems: The Minimal Substrate Package

We now state the main claims of the paper as a “minimal substrate theorem package.” The theorems are stated at a semantic level: they identify a canonical class of machine substrates that realize P1–P6 internally, give a normal-form/compilation viewpoint, and show that specific structural features are necessary to avoid smuggling. Proofs are given as constructive sketches, with separation witnesses instantiated by the reproducible experiment suite (Section 6) and algebraic lemmas mechanized in Lean (Appendix B).

5.1. The Canonical Minimal Substrate Class

Definition 12 (Canonical minimal machine class \mathcal{S}_{\min}). *A substrate belongs to \mathcal{S}_{\min} if it consists of: (i) a finite carrier Z , (ii) an autonomous time-homogeneous Markov kernel P on Z , (iii) an admissible family of deterministic lenses $f : Z \rightarrow X$ closed under composition (refinement chains), (iv) a family of internal packaging endomaps $E_{\tau,f}$ induced by (P, f, τ) and prototype choices (Definition 11), and (v) intrinsic audits of directionality/drive induced by (ρ, P) (e.g. Σ_T and ACC affinities) that are monotone under deterministic coarse-graining (Definition 10).*

The class \mathcal{S}_{\min} is intentionally small: it assumes neither geometry nor an ambient order, and it requires that protocols be represented autonomously (phase-in-state) whenever staged interaction is used (Section 4).

Remark 4 (Minimality-by-separation). The term “minimal” in \mathcal{S}_{\min} is used in a feature-minimal sense: it is a small list of structural ingredients sufficient to realize P1–P6 under the no-smuggling discipline. We do not claim that \mathcal{S}_{\min} is an initial object or a unique smallest class under inclusion; our lower bounds are separation statements (Theorem 3) rather than lattice-theoretic minimality.

5.2. Realizability on the Canonical Machine

Theorem 1 (Realizability of P1–P6 on \mathcal{S}_{\min}). Every substrate in \mathcal{S}_{\min} admits realizations of primitives P1–P6 without smuggling. More concretely, for any (Z, P) with lenses and induced packaging operators as in Definition 12, there exist internal operators/constraints that instantiate: dynamics/composition (P1), limited access via lenses (P2), autonomous protocols via phase-in-state (P3), staging via timescales and refinement chains (P4), packaging via $E_{\tau, f}$ and its fixed-point/defect calculus (P5), and intrinsic audits of arrow/drive via Σ_T and ACC affinities with monotonicity under coarse-graining (P6).

Sketch. P1 is realized by autonomous evolution $\mu \mapsto \mu^{P^\tau}$ and composition by kernel multiplication. P2 is realized by deterministic lenses and the induced definability constraint (predicates constant on fibers). P3 is realized by enforcing autonomy: any staged protocol is encoded by enlarging state (introducing a phase register Φ so that the dynamics becomes time-homogeneous on $Z \cong X \times \Phi \times L$). P4 is realized by the explicit timescale parameter τ and lens refinement chains. P5 is realized by the internal endomap $E_{\tau, f}$ (Definition 11), with packaged objects represented as (approximate) fixed points and quality quantified by defect measures. P6 is realized by intrinsic audits induced by (ρ, P) : Σ_T is computed on path space and respects data processing under deterministic coarse-graining, while ACC affinities witness internal drive via nonzero cycle integrals. All of these constructions are internal to the substrate data, hence no smuggling is required. \square

5.3. Normal Form and Compilation Up to Packaging

Realizability alone does not turn a language into a machine: we also want a notion of normal form and mutual simulation. The appropriate equivalence notion is *packaging equivalence* induced by idempotent(-like) endomaps (Definition 8). The Lean appendix mechanizes the corresponding algebra: idempotent splitting (fixed points, inclusion/retraction), packaging equivalence as an equivalence relation, and lens saturation as a closure operator (Appendix B).

Theorem 2 (Compilation to \mathcal{S}_{\min} up to packaging equivalence). Let \mathcal{S} be a substrate that realizes P1–P6 without smuggling in the sense of Definition 3. Assume further that the realized operators are finitary (state and registers can be encoded in a finite carrier) and autonomous once all protocol state is represented. Then there exists a substrate $\mathcal{M} \in \mathcal{S}_{\min}$ and an encoding of \mathcal{S} into \mathcal{M} such that:

1. the lens/refinement structure of \mathcal{S} is simulated by deterministic lenses on \mathcal{M} ,
2. packaging in \mathcal{S} is simulated by internal packaging endomaps on \mathcal{M} up to packaging equivalence, and
3. admissible audits in \mathcal{S} are represented by intrinsic audits on \mathcal{M} (in particular, directionality audits respect monotonicity under coarse-graining).

Moreover, if two compilations choose the same canonical prototype-selection scheme for each lens (e.g. uniform-on-fiber) and the same auditing conventions, then they are mutually simulable up to packaging equivalence via packaging-respecting maps in the sense of Definition 9.

Sketch. The compilation is constructive: represent all externally referenced “context” as explicit internal registers. Protocols become autonomous by adding a phase register (phase-in-state). Any deterministic observation interface becomes a lens $f : Z \rightarrow X$. Packaging is represented as an internal endomap on an appropriate carrier (e.g. distributions or descriptions); packaging equivalence then provides the quotient notion used for comparison. Finally, audits are required by no-smuggling semantics to be induced by the substrate and monotone under coarse-graining; in the canonical machine, Σ_T and ACC affinities provide canonical such induced audits. Mutual simulation up to

packaging equivalence is mediated by packaging-respecting maps (Definition 9), which make precise what it means to compare two presentations “up to packaging.” We treat prototype choice as part of the packaging structure rather than as a gauge that leaves equivalence classes invariant; indeed the witness suite (Section 6) shows prototype-driven route mismatch effects. \square

5.4. Minimality: No-Smuggling Lower Bounds

We now state minimality claims as *separations*: if certain structural features are removed, then at least one primitive cannot be realized without reintroducing the missing feature as an external oracle. Each item is witnessed by a small finite construction in the reproducible suite (Section 6).

Theorem 3 (No-smuggling separations). *The following substrate features are necessary (up to equivalent internal encodings) to realize the corresponding primitives without smuggling:*

1. **Autonomy requires phase-in-state (P3).** *If a substrate permits staged protocols but does not represent protocol state internally, then stroboscopic composition can produce apparent arrows of time that vanish once the dynamics is made autonomous by lifting state. Therefore, any autonomous semantics for protocols must encode phase-in-state (or an equivalent internal mechanism).*
2. **Genuine arrows require internal drive (P6_{drive}).** *Autonomy plus staging does not imply directionality. In an autonomous lifted model with reversible internal phase dynamics, protocol-induced arrows can disappear ($\Sigma_T \approx 0$). To obtain a genuine arrow in an autonomous substrate, one needs internal drive detectable by a nonzero affinity class (ACC cycle integrals).*
3. **Audits must be monotone under coarse-graining (P6).** *There exist processes with real micro irreversibility but macro observations whose path law is reversal-invariant ($\Sigma_T^{\text{macro}} = 0$). Any audit that reports increased directionality after coarse-graining violates limited-access semantics and is therefore smuggled.*
4. **Packaging must be an internal (approximate) idempotent endomap (P5).** *Without an internal endomap e (or closure operator) that induces packaging equivalence $x \sim_e y \iff e(x) = e(y)$, there is no internal notion of “same packaged object” and no internal fixed-point criterion for stability. Declaring macrostates by fiat is smuggling.*
5. **Route mismatch is distinct from directionality (P3 vs. P6).** *Packaging along different refinement routes can yield a measurable holonomy defect (route mismatch), even when directionality audits obey data processing. Therefore, route mismatch must be treated as its own certificate family and cannot be conflated with an arrow audit.*

Sketch. Each item is witnessed by a finite explicit construction. (1) is the protocol trap: a stroboscopic kernel on X yields $\Sigma_T > 0$ while an autonomous lift on $Z = X \times \Phi$ yields $\Sigma_T \approx 0$; the discrepancy disappears once phase is represented in state. (2) is shown by driven protocols: adding a nonreversible internal phase dynamics (nonzero ACC affinity) restores $\Sigma_T > 0$ in the autonomous lifted system and can make the arrow visible under projection, while a reversible (null) phase yields $\Sigma_T \approx 0$. (3) is shown by hidden irreversibility: a biased cycle has $\Sigma_T^{\text{micro}} > 0$ while a coarse lens yields $\Sigma_T^{\text{macro}} = 0$, hence any “macro-only” arrow certificate must be smuggled. (4) follows from the algebra of idempotents and closure operators: packaging equivalence is induced by an endomap, and the existence of packaged objects is expressed as fixed points; this algebra is mechanized in Lean (Appendix B). (5) is shown by prototype-driven route mismatch at moderate timescales together with an empirically verified gain-plus-mismatch bound; large mismatch does not imply directionality and must be accounted for separately. \square

Remark 5 (Status and scope). *The theorems above isolate the semantic content needed to turn SPT into a machine semantics. They are stated at a level that cleanly separates: (i) algebraic infrastructure (idempotents/closure), (ii) audit infrastructure (monotone directionality/drive certificates), and (iii) autonomy discipline*

(phase-in-state for protocols). The witness suite provides concrete finite constructions and quantitative diagnostics that support the separation claims, while the Lean appendix checks the reusable algebraic lemmas.

6. Experimental Witness Suite (Separation Exhibits)

This section collects a small witness suite of finite constructions that support the separation claims in Theorem 3. These are not “fit-to-data” experiments. They are executable counterexample-style exhibits: each construction isolates a semantic feature (autonomy discipline, audit monotonicity, internal drive, refinement/stability tradeoffs, and route mismatch) and demonstrates a sharp effect.

All reported numbers are frozen into the build via auto-generated macros (see Appendix A for reproducibility notes). Unless stated otherwise, Σ_T is evaluated from a matched stationary initial distribution for the audited kernel (so nonzero values reflect steady-state directional asymmetry rather than transient boundary terms). Figures under `paper/figures/` are committed artifacts produced by deterministic scripts.

6.1. Protocol Trap: Stroboscopic Arrows Vanish Under Autonomy

The protocol trap (from (Ioannis 2026)) illustrates a no-smuggling constraint on autonomy. A stroboscopic protocol on a microstate space X can exhibit an apparent arrow of time when it is compressed into a one-step Markov closure (a single composed kernel) that forgets the protocol phase. However, if the protocol stage is not represented in state, the construction is non-autonomous; enforcing autonomy by lifting to $Z = X \times \Phi$ can remove the apparent arrow.

In our witness, the one-step Markov closure on X yields a positive path-reversal asymmetry $\Sigma_T = 0.173783$, while the correctly autonomous lifted kernel on $Z = X \times \Phi$ yields $\Sigma_T = 0$; moreover the exact macro audit induced by the lifted chain yields $\Sigma_T = 6.93335 \times 10^{-33}$, consistent with audit monotonicity on the induced path law. The semantic implication is direct: an “autonomous” substrate that relies on external scheduling is smuggling protocol state.

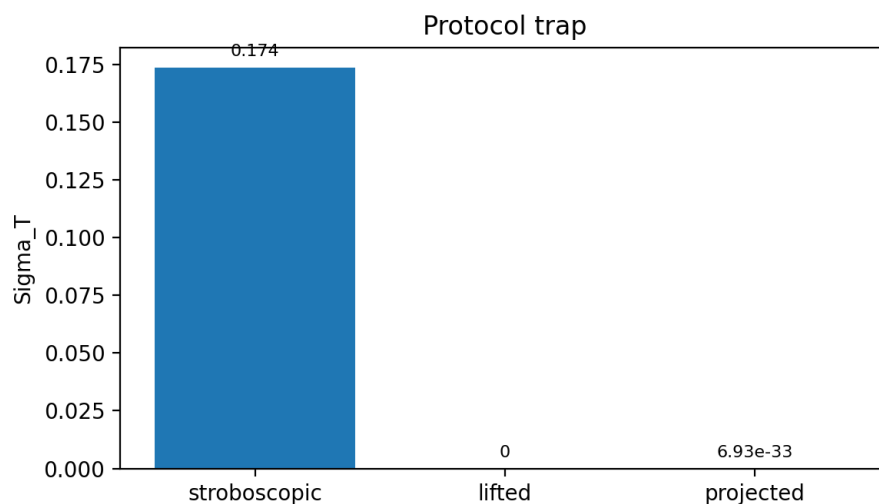


Figure 2. Protocol trap witness (Markov closure vs. autonomous audit): a stroboscopic description on X exhibits $\Sigma_T = 0.173783$, while the autonomous lifted dynamics on $Z = X \times \Phi$ yields $\Sigma_T = 0$ and its projection yields $\Sigma_T = 6.93335 \times 10^{-33}$.

6.2. Driven Protocols: Internal Drive Restores Genuine Arrows

The protocol trap makes a negative point (some arrows are artifacts of external schedules). A minimal substrate story also needs the complementary positive point: in an autonomous substrate, genuine arrows require internal drive ($P_{6_{\text{drive}}}$), detectable for example by nonzero ACC affinity.

Our driven-protocol witness uses an explicit phase register Φ with nonreversible internal dynamics. For the chosen parameters ($T = 4$ and mixing weight $\alpha = 0.3$ in the experiment), the autonomous lifted system exhibits a strong arrow $\Sigma_T = 2.10934$, and the projection to X becomes directionally

biased as well ($\Sigma_T = 0.00638045$). By contrast, a reversible (null) phase control yields $\Sigma_T = 0$ on the lift and $\Sigma_T = 1.00148 \times 10^{-32}$ under projection. The internal drive is certified by the phase-only audit values, including an ACC affinity magnitude 6.59167.

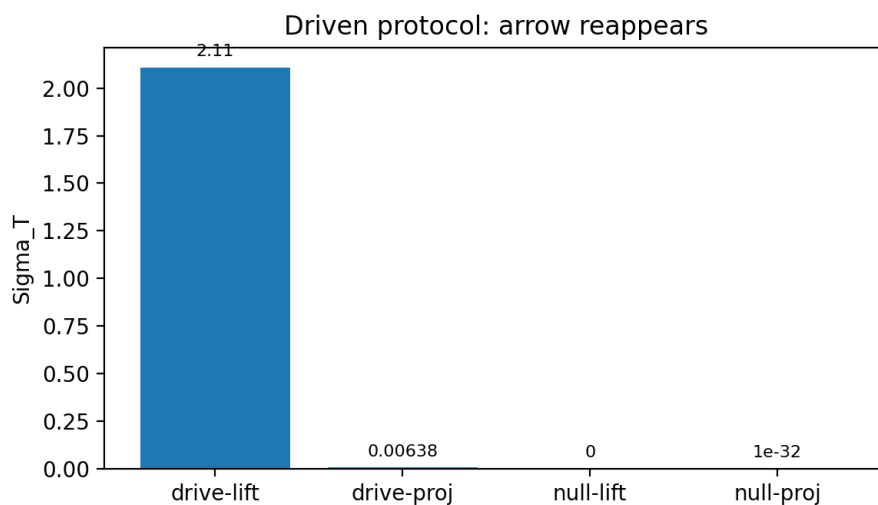


Figure 3. Driven protocol witness: with internal phase drive (nonreversible Φ dynamics), the autonomous lift exhibits $\Sigma_T = 2.10934$ and the projection exhibits $\Sigma_T = 0.00638045$. Under a reversible (null) phase, the corresponding values are $\Sigma_T = 0$ and $\Sigma_T = 1.00148 \times 10^{-32}$.

6.3. Hidden Irreversibility Under Coarse-Graining (DPI Witness)

Audit monotonicity under deterministic coarse-graining (Definition 10) is a no-false-positives guardrail. There exist processes with real micro irreversibility whose macro observation has a reversal-invariant path law. Such cases witness that “macro-only” directionality certificates cannot be required to be complete, only sound.

In our hidden-irreversibility witness, the micro chain has a large reversal asymmetry $\Sigma_T^{\text{micro}} = 10.5467$ while a parity lens yields $\Sigma_T^{\text{macro}} = 0$. This is consistent with data processing: forgetting information cannot increase Σ_T , and it may erase it entirely. Related stochastic-thermodynamic discussions of coarse-graining and hidden contributions to dissipation appear in (Esposito 2012; Puglisi et al. 2010).

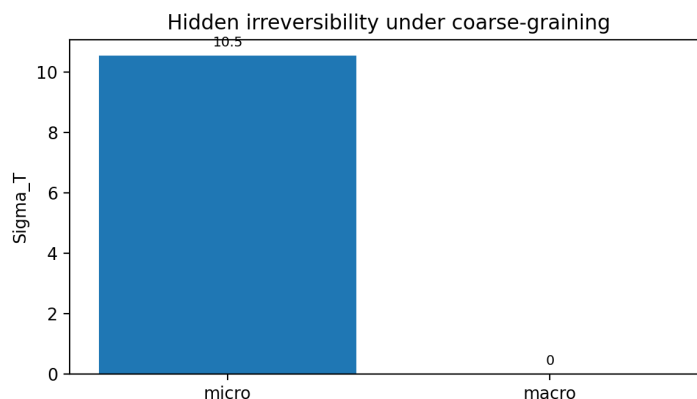


Figure 4. Hidden irreversibility witness: a micro process has $\Sigma_T^{\text{micro}} = 10.5467$ but a coarse observation yields $\Sigma_T^{\text{macro}} = 0$. This exemplifies audit monotonicity and the possibility of undetectable micro irreversibility at the macro scale.

6.4. Refinement Can Help or Hurt: Packaging Stability vs. Timescale

Six Birds emphasizes that packaging depends on both the observational lens and the timescale τ (Ioannis 2026). Our refinement witness makes the point operational: refining a lens can *increase* or *decrease* the number of stable packaged objects depending on τ .

In the witness construction, we compare a coarse lens (two basins) to a refined lens (splitting one basin) across $\tau \in \{1, \dots, 20\}$ with stability threshold $\varepsilon = 0.1$. We observe both directions: refinement yields more stable prototypes for some τ (e.g. the first such τ is 1, with 5 total instances), and fewer stable prototypes for other τ (first at 6, with 15 total instances). At the end of the sweep, the stable counts are coarse 2 versus refined 1 under this threshold. This non-monotonicity is exactly why staging (P4) cannot be collapsed into “more refinement is always better.”

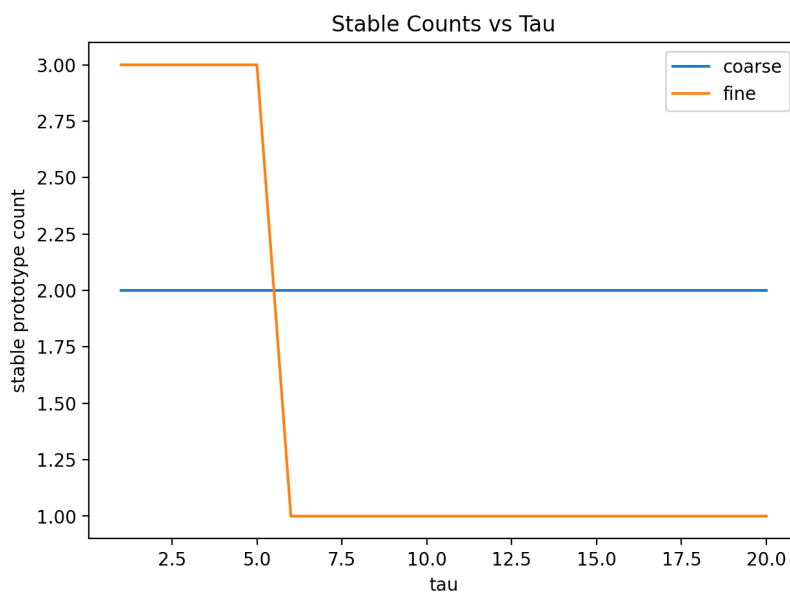


Figure 5. Refinement witness: stable prototype counts as a function of timescale τ for a coarse lens versus a refined lens (same underlying dynamics). Refinement can help or hurt depending on τ .

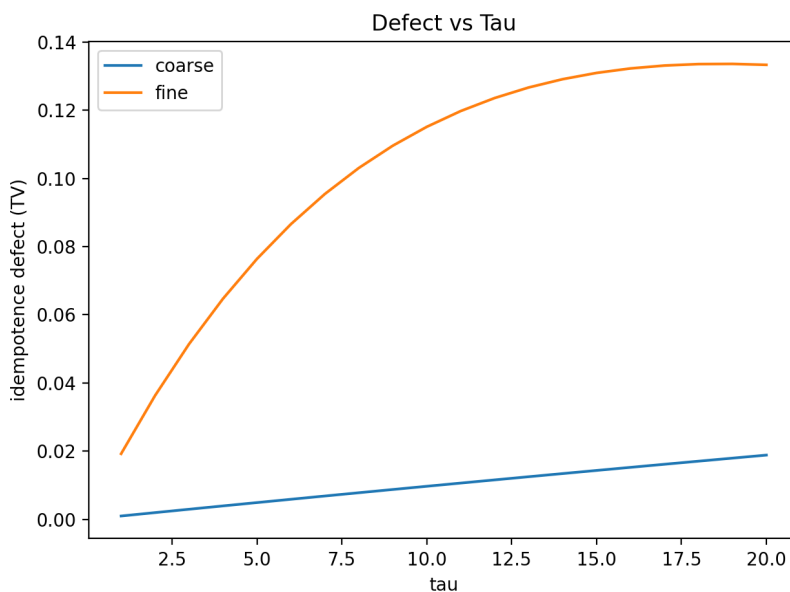


Figure 6. Refinement witness: idempotence defect diagnostic versus timescale τ for the same two lenses as Figure 5.

6.5. Route Mismatch (Holonomy Defect) and the Gain-Plus-Mismatch Bound

When lenses form a refinement chain, packaging can be performed along different routes. A canonical route mismatch (holonomy defect) compares direct packaging to packaging via an intermediate scale, using an induced operator norm (here, max row- ℓ_1). Importantly, route mismatch is a *separate certificate family* from directionality: it can be large even when directionality audits obey data processing.

Our prototype-driven route mismatch witness uses a moderate timescale tuple $\tau = [6, 3, 3]$ and a prototype scheme `f0_rep03_vs_rep25_f1_uniform`. The mismatch norm is 2, while a uniform-prototype baseline at the same τ has near-zero mismatch 5.02202×10^{-16} . Thus the holonomy defect is not merely a long-timescale artifact; it can arise from representative choice (prototype selection) even at moderate staging.

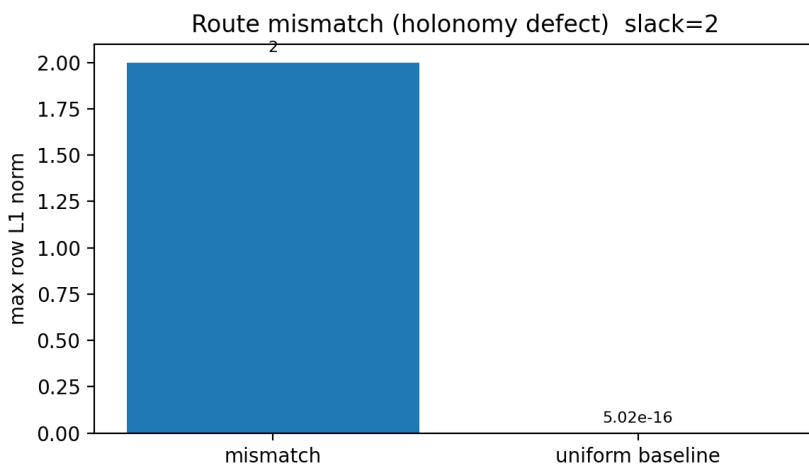


Figure 7. Route mismatch witness: prototype-driven holonomy defect at $\tau = [6, 3, 3]$ under scheme `f0_rep03_vs_rep25_f1_uniform`. The mismatch is 2 versus a uniform baseline 5.02202×10^{-16} .

Finally, we empirically verify an Appendix-E style inequality relating gains, mismatch, and direct packaging. Let $\|\cdot\|$ denote the induced max row- ℓ_1 norm. Define gains $\text{Gain}_{j \leftarrow k} := \|\text{Pack}_{j \leftarrow k}\|$ and route mismatch $\text{RM} := \|\text{Pack}_{0 \leftarrow 2} - \text{Pack}_{1 \leftarrow 2} \circ \text{Pack}_{0 \leftarrow 1}\|$. Our run satisfies

$$\text{Gain}_{0 \leftarrow 2} \leq \text{Gain}_{0 \leftarrow 1} \text{Gain}_{1 \leftarrow 2} + \text{RM},$$

with $\text{Gain}_{0 \leftarrow 2} = 1$, $\text{Gain}_{0 \leftarrow 1} = 1$, $\text{Gain}_{1 \leftarrow 2} = 1$, $\text{RM} = 2$, and slack 2 (i.e. the right-hand side minus the left-hand side). This bound makes the route mismatch diagnostic quantitative and compositional.

7. Discussion: Practical Workflow and Failure-Mode Safeguards

The main technical content of this paper is semantic: a canonical minimal machine substrate \mathcal{S}_{\min} and separation-style minimality claims (Theorem 3). For readers coming from complex systems and representation learning, the practical value is a disciplined workflow for avoiding false positives when diagnosing multi-scale organization and directionality.

7.1. A Practical Audit Workflow (User-Facing)

Given a candidate substrate/model (simulator, learned dynamics, or a hand-built mechanism), the following workflow implements the paper’s “no smuggling” discipline:

1. **Make autonomy explicit.** If the model uses staged protocols or time-varying rules, encode protocol state internally (phase-in-state) so that the resulting dynamics is autonomous on an extended carrier Z (Section 4).

2. **Choose lenses that represent actual access.** Specify deterministic coarse-grainings $f : Z \rightarrow X$ that match what an observer/agent can actually read out; avoid conflating “what the simulator knows” with what is observable (Definition 4).
3. **Package at explicit timescales.** For each (f, τ) , construct packaging endomaps $E_{\tau, f}$ and compute defect/stability diagnostics; compare coarse vs refined lenses across τ rather than assuming refinement is monotone beneficial (Definition 11, Figure 5).
4. **Audit directionality with controls.** Evaluate Σ_T (and, when appropriate, ACC affinities) in matched controls: (i) reversible/null controls to calibrate “artifact” levels, (ii) driven regimes to test for genuine internal drive, and (iii) micro vs macro audits to respect coarse-graining monotonicity (Figure 2, Figure 3, Figure 4).
5. **Treat route mismatch separately.** If packaging is performed along refinement routes, quantify route mismatch as its own defect certificate; do not treat it as an arrow-of-time audit (Figure 7).

7.2. Threat Model: Common Failure Modes and What the Exhibits Test

The witness suite is best read as a “failure mode catalogue”. The point is not that these are rare; it is that these are easy traps in multi-scale analysis if one does not enforce the semantic discipline.

Table 1. Failure-mode checklist: what each certificate is meant to detect, and what can go wrong without matched controls or semantic discipline.

Certificate	Diagnostic / operator	Failure mode + witness
Directionality / arrow (P6)	Σ_T on induced path laws (micro and macro)	Markov closure / external scheduling can create spurious arrows; enforcing autonomy (phase-in-state) removes them (protocol trap; Figure 2).
Drive (P6 _{drive})	ACC affinity (cycle integrals / exactness obstruction)	Autonomy plus protocols does not imply arrows; internal nonreversible drive cleanly separates from null controls (driven protocol; Figure 3).
Limited-access soundness (P2/P6)	Audit monotonicity under lenses	Coarse-graining can hide real micro irreversibility; macro audits can be zero even when micro is positive (hidden irreversibility; Figure 4).
Packaging / objects (P5)	$E_{\tau, f}$; idempotence defect and stability scores	Refinement can help or hurt depending on timescale; stability is not monotone in expressivity (refinement witness; Figure 5).
Route mismatch (holonomy)	$\ \text{Pack}_{0 \leftarrow 2} - \text{Pack}_{1 \leftarrow 2} \circ \text{Pack}_{0 \leftarrow 1} \ $	Representative/prototype choices can induce large route mismatch at moderate timescales even when uniform baselines are near zero (route mismatch; Figure 7).

7.3. Scope and Limits

We emphasize two scope points. First, the “minimal substrate” claims are semantic and separation-based: they identify a small set of internal operators sufficient to realize P1–P6 without smuggling, and show that removing specific features forces primitive failure or external oracles (Theorem 3). Second, the witness suite is intentionally finite and controlled; it is not intended as evidence that any particular natural system implements \mathcal{S}_{\min} , nor as a claim that Markovian models are fundamental. Rather, finite autonomous stochastic machines serve as a compact normal form in which the primitives and their failure modes can be made explicit, computed exactly, and compared reproducibly.

On novelty (P1/P2/P4).

Six Birds develops novelty as strict theory extension via non-definable predicates (Ioannis 2026). This paper does not add a new novelty exhibit; instead it focuses on the semantic substrate conditions that allow novelty, packaging, and directionality to coexist without smuggling. Adding novelty-focused witness suites is a natural next step once the minimal machine semantics is fixed.

Author Contributions: I.T. conceived the study, implemented experiments and formal verification, and wrote the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data, code, and reproducibility artifacts are available in the companion repository.

Acknowledgments: The author acknowledges the use of Lean 4/mathlib and standard Python scientific libraries.

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

Appendix A Implementation and Reproducibility Notes

This appendix summarizes how the witness suite is produced and how the paper freezes numbers and figures for reproducible builds. The goal is not to prescribe a single environment, but to make it clear what to run and where the relevant artifacts live in the repository.

Appendix A.1 Repository Layout (Relevant Paths)

The main implementation lives in the Python package under `spt/`:

- `spt/core/markov.py`: Markov utilities and exact computation of path-reversal asymmetry Σ_T on micro path space.
- `spt/core/lens.py`: deterministic lenses, exact macro path probabilities, and exact macro Σ_T ; includes DPI checks in tests.
- `spt/core/packaging.py`: empirical packaging operators $E_{\tau,f}$, idempotence defect, retention error, and prototype stability scores.
- `spt/core/acc.py`: ACC-style antisymmetric 1-form, cycle basis integrals (affinities), and exactness residual diagnostics.

Reproducible experiments live under `experiments/` and write per-run artifacts:

- `experiments/run_all.py`: shared experiment runner.
- `spt/utils/io.py`: run directory creation and JSON artifact writing: `config.json`, `metrics.json`, and `README.md`.
- `experiments/summarize_results.py`: produces `docs/experiments/index.md` and `docs/experiments/metrics_summary.json` by scanning `results/`.

The paper sources live under `paper/` and include two important tooling scripts:

- `paper/tools/extract_metrics.py`: reads `docs/experiments/metrics_summary.json` and writes `paper/generated/metrics.tex` (LaTeX macros for frozen numbers).
- `paper/tools/make_figures.py`: reads `docs/experiments/metrics_summary.json` and writes committed figure PNGs under `paper/figures/`.

Appendix A.2 Running the Witness Suite

All experiments write to `results/` by default, using timestamped run directories:

```
results/<experiment>/<tag>/<timestamp>/
```

Each run directory contains: `config.json` (inputs), `metrics.json` (outputs), and a short `README.md` note.

A typical end-to-end run (from the repository root) is:

```
python experiments/run_all.py --tag smoke --outdir results
python experiments/summarize_results.py --results_dir results \
  --index_path docs/experiments/index.md --tag smoke
```

The summary step produces two committed artifacts under `docs/experiments/`: (i) a human-readable index (`index.md`) and (ii) a machine-readable latest-run snapshot (`metrics_summary.json`). The paper build does not depend on `results/` being committed; it depends only on these frozen summaries and committed figure files.

Appendix A.3 Freezing Numbers into LaTeX Macros

To keep the PDF stable and self-contained, the paper freezes numerical values into LaTeX macros. The workflow is:

```
python paper/tools/extract_metrics.py \
  --input docs/experiments/metrics_summary.json \
  --output paper/generated/metrics.tex
```

The generated file `paper/generated/metrics.tex` defines macros such as `\PTSigmaStrob`, `\PDSigmaLiftDrive`, `\HISigmaMicro`, and `route-mismatch` terms used in Section 6. The macros are treated as part of the paper source; thus a snapshot of the repository contains all numbers needed to build the PDF.

Appendix A.4 Freezing Figures

Figures used in the paper are committed under `paper/figures/`. They can be regenerated deterministically (from `docs/experiments/metrics_summary.json`) using:

```
python paper/tools/make_figures.py
```

The refinement plots are copied from the most recent available run directory if present; otherwise the script can run the refinement experiment into a temporary `paper/_runs/` directory (which is gitignored) and then copy the resulting PNGs into `paper/figures/`.

Appendix A.5 Building the PDF

The paper is built from `paper/` using `latexmk`:

```
cd paper
latexmk -pdf -interaction=nonstopmode -halt-on-error main.tex
```

The build inputs are therefore: LaTeX sources, `paper/generated/metrics.tex`, and the committed PNG figures.

Appendix B Lean Formalization: A Minimal Algebraic Backbone

This project includes a small Lean4 development that mechanizes the algebraic infrastructure used repeatedly in the no-smuggling semantics (de Moura and Ullrich 2021; The mathlib Community 2020). The Lean code does *not* attempt to formalize stochastic processes, KL divergence, or the probabilistic directionality audits. Instead, it targets the reusable core algebra behind packaging and lens-induced saturation: idempotent endomaps, induced equivalence relations, and closure operators.

The Lean project lives under `lean/` (mathlib-based) and builds with:

```
cd lean && lake build
```

Appendix B.1 Idempotent Splitting (Packaging as an Idempotent Endomap)

Packaging in the paper is modeled abstractly by an (exact or approximate) idempotent endomap e (Definition 7). In Lean, we formalize the exact algebra that motivates packaging equivalence and fixed-point objects.

File: `lean/SPT/Idempotent.lean`.

Key constructions:

- `Idempotent` ($e : \alpha \rightarrow \alpha$): the predicate $e \circ e = e$.
- `Fix e`: the subtype of fixed points $\{x \mid e(x) = x\}$.
- inclusion $i : \text{Fix } e \rightarrow \alpha$ and retraction $r : \alpha \rightarrow \text{Fix } e$ (given idempotence).

Key lemmas (identifiers in Lean):

- `SPT.r_comp_i`: $r \circ i = \text{id}$ on fixed points.
- `SPT.i_comp_r`: $i \circ r = e$ on the carrier.

These lemmas support the paper’s repeated use of “objects as fixed points” and “comparison up to packaging”: if packaging is an internal idempotent(-like) map, then fixed points are canonical representatives and every state retracts to its packaged form.

Appendix B.2 Packaging Equivalence

The paper uses packaging equivalence \sim_e induced by an endomap e (Definition 8) as the right notion of “simulation up to packaging.” In Lean, we define the relation and prove it is an equivalence relation.

File: `lean/SPT/PackagingEquiv.lean`.

Definition:

$$\text{PackEq } e \ x \ y \ :\iff e(x) = e(y).$$

Key lemmas (identifiers in Lean):

- `SPT.packEq_refl`, `SPT.packEq_symm`, `SPT.packEq_trans`.
- `SPT.packEq_to_package`: given idempotence, x is equivalent to its packaged form $e(x)$.
- `SPT.packEq_iff_packEq_apply`: stability of packaging equivalence under applying e (useful for normal-form reasoning).

These lemmas provide a minimal formal target for the “up to packaging equivalence” language used in Theorem 2.

Appendix B.3 Lens Chains via Equivalence Relations and Saturation as Closure

In the paper, a deterministic lens $f : Z \rightarrow X$ induces an observational equivalence relation on Z (same fiber) and therefore a saturation operation on predicates/sets. In Lean, we model a lens abstractly as a `Setoid` α and define saturation as an existential closure along the relation.

File: `lean/SPT/LensChain.lean`.

Definition (Lean):

$$\text{sat } r \ S \ (x) \ :\iff \exists y, S(y) \wedge r(x, y),$$

where $r : \text{Setoid } \alpha$ and $S : \text{Set } \alpha$.

Key lemmas (identifiers in Lean):

- `SPT.sat_mono`: monotonicity of saturation.
- `SPT.sat_idempotent` and `SPT.sat_idempotent_fn`: idempotence (saturation is a closure operator component).
- `SPT.sat_extensive`: extensiveness $S \subseteq \text{sat}_r(S)$.
- `SPT.Refines` and `SPT.sat_refines`: refinement implies saturation inclusion (“finer relation saturates less”).

Together, `sat_mono`, `sat_idempotent`, and `sat_extensive` show that saturation is a closure operator on predicates. This is the abstract version of “quotient saturation packaging” used in the meta-primitive story: a lens induces a closure/saturation on what can be expressed and stabilized at that scale.

Appendix B.4 Scope

This Lean development is intentionally minimal. It is not a mechanization of the probabilistic audits or the Markov-chain calculations. Its role is to provide a stable, machine-checked algebraic substrate for: (i) packaging as an idempotent(-like) endomap with fixed-point objects, (ii) packaging equivalence as the right comparison notion, and (iii) lens-induced saturation as a closure operator compatible with refinement chains.

References

- Cover, Thomas M. and Joy A. Thomas. 2006. *Elements of Information Theory* (2 ed.). Wiley-Interscience.
- de Moura, Leonardo and Sebastian Ullrich. 2021. The lean 4 theorem prover and programming language (system description). In *Automated Deduction – CADE 28*, Volume 12699 of *Lecture Notes in Computer Science*, pp. 625–635. Springer. https://doi.org/10.1007/978-3-030-79876-5_37.
- Esposito, Massimiliano. 2012. Stochastic thermodynamics under coarse graining. *Physical Review E* 85(4), 041125. <https://doi.org/10.1103/PhysRevE.85.041125>.
- Ioannis. 2026. Six birds: Foundations of emergence calculus. arXiv:2602.00134 [cs.LO], version 1, <https://doi.org/10.48550/arXiv.2602.00134>.
- Kelly, Frank P. 1979. *Reversibility and Stochastic Networks*. Wiley.
- Kolmogoroff, A. 1936. Zur theorie der markoffschen ketten. *Mathematische Annalen* 112, 155–160.
- Kullback, Solomon and Richard A. Leibler. 1951. On information and sufficiency. *The Annals of Mathematical Statistics* 22(1), 79–86. <https://doi.org/10.1214/aoms/1177729694>.
- Lebowitz, Joel L. and Herbert Spohn. 1999. A gallavotti–cohen-type symmetry in the large deviation functional for stochastic dynamics. *Journal of Statistical Physics* 95(1–2), 333–365. <https://doi.org/10.1023/A:1004589714161>.
- Norris, J. R. 1997. *Markov Chains*. Cambridge University Press.
- Puglisi, Andrea, Simone Pigolotti, Lamberto Rondoni, and Angelo Vulpiani. 2010. Entropy production and coarse graining in markov processes. *Journal of Statistical Mechanics: Theory and Experiment* 2010(05), P05015. <https://doi.org/10.1088/1742-5468/2010/05/P05015>.
- Rabiner, Lawrence R. 1989. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77(2), 257–286. <https://doi.org/10.1109/5.18626>.
- Schnakenberg, J. 1976. Network theory of microscopic and macroscopic behavior of master equation systems. *Reviews of Modern Physics* 48(4), 571–585. <https://doi.org/10.1103/RevModPhys.48.571>.
- Seifert, Udo. 2012. Stochastic thermodynamics, fluctuation theorems and molecular machines. *Reports on Progress in Physics* 75(12), 126001. <https://doi.org/10.1088/0034-4885/75/12/126001>.
- The mathlib Community. 2020. The lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP '20)*, New York, NY, USA, pp. 367–381. Association for Computing Machinery. <https://doi.org/10.1145/3372885.3373824>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.