The SERD Framework: Formalization, Restricted Kernels, and Frozen-Regime Observer Recovery

Tommy Wood

Unconventional Computing Laboratory, University of the West of England

RESEARCH Gauge Freedom Journal (2026) 1:2

https://doi.org/10.65323/gfj.2026.002

GAUGE FREEDOM JOURNAL

Correspondence: Tommy Wood (Tommy.Wood@uwe.ac.uk)

Received: 10 April 2026 / *Accepted:* 14 May 2026

Published online: 19 May 2026

Abstract

The *Space Element Reduction Duplication* (SERD) framework is a discrete, background-free relational framework in which observer-indexed separation information is reconstructed from local updates and transported records on a typed latent substrate rather than imposed by an external manifold or metric. The parent framework is built from three primitive element types—point particles, space elements, and information gaps—together with a small family of strictly local update operations and a deterministic propagation phase for gap-based records.

This paper develops the formal middle layer of the current SERD programme. Its aim is not to establish complete theorem-level closure of the full SERD rule space, but to define a common parent framework, separate microscopic law from scheduler family, identify restricted kernels derived from that framework, and state the strongest currently available rigorous results within those kernels. The paper studies two implementation-faithful restricted kernels: a split-enabled structural kernel and a constant-point-particle metric kernel.

The strongest results are obtained for the frozen regime of the constant-point-particle kernel. After local updates have been switched off, the remaining receiver-indexed transport dynamics are formalized as a deterministic frozen queue system. In this regime the paper proves deterministic evolution after freeze, one-generation-only rebroadcast, finite-horizon settling of residual transported records, and, for observer-symmetric frozen states, an exact terminal frozen correction theorem expressing the eventual observer-side correction as a finite functional of the full frozen state. On that basis, it gives a conditional characterization of late-time observer recovery and observer agreement in terms of recovery compatibility of the full frozen state. A separate conditional bridge theorem identifies when a structurally quiescent late regime of the split-enabled structural kernel may be explicitly identified with a constant-point-particle frozen-state description, and hence brought under the same frozen-regime theorem package.

The paper does not claim a complete treatment of unrestricted active dynamics, a final continuum identification, or a theorem-level account of every numerical phenomenon discussed in companion computational work. Its contribution is narrower and more precise: it formalizes the parent framework, isolates its current restricted kernels, and proves a bounded frozen-regime theorem package together with a conditional bridge result from the split-enabled structural sector.

Keywords— SERD; relational dynamics; emergent geometry; restricted kernels; frozen transport; observer recovery; AI-assisted research provenance

1 Introduction

A central problem in foundational physics is how distance-like and geometry-like structure might arise from a substrate that is itself discrete and background free. The *Space Element Reduction Duplication* (SERD) framework is one attempt to address that problem. Rather than beginning with a manifold, metric, or field on an external spacetime, SERD begins with a purely relational latent substrate and asks what observer-indexed separation information can be reconstructed from the history of local changes within that substrate.

The SERD framework is built from three primitive element types:

(i) **point particles**, which act as discrete loci and observers in the latent state;

(ii) **space elements**, which form filamentary chains of separation between loci;

(iii) **information gaps**, which are explicit local boundary sites at which records are written, stored, and propagated.

Loci are therefore not embedded in an already existing space. Instead, separation is represented by chains of space elements, while the local history of how those chains change is written at neighbouring information gaps and transported along the filaments at a rate of one space element per discrete time step. Observer-indexed separation summaries are then reconstructed from this internal bookkeeping at point-particle interfaces rather than read off from an external geometry.

Parent framework and restricted kernels. At the parent level, SERD is a disciplined family of local, background-free dynamics on this typed substrate. The broader move vocabulary includes point-particle splits, space-element duplication, space-element reduction, and broader coarsening sectors involving point particle merging- or fusion/fission-type behaviour on compact or fused configurations. The present paper does not attempt a theorem-level treatment of that full move family at once. Instead, it studies two implementation-faithful restricted kernels derived from the same parent architecture.

Kernel A is a split-enabled structural kernel. It contains point-particle splits, space-element duplication and reduction, and deterministic propagation of structural markers; in the computational-paper implementations it may also include an optional localized fission sweep on fused-gap motifs. Kernel B is a constant-point-particle metric kernel. It fixes the point-particle set, evolves pairwise latent filament lengths by local duplication and reduction, allows an optional compact-bond reopening pre-pass in the implementation-faithful update stage, and transports signed metric records deterministically on receiver-indexed faces.

Microscopic law and scheduler family. A central distinction in the paper is the distinction between the *microscopic law* and the *scheduler family*. The microscopic law consists of the typed latent-state class, the admissible move vocabulary, and the deterministic propagation architecture. Scheduler families specify how compatible local updates are selected and executed in explicit trajectories or simulations. This distinction matters because it allows the formal framework to remain microscopically parameter free even when numerical experiments introduce update probabilities, freeze times, or execution choices at the scheduler level.

Main results. The strongest rigorous results in the paper are obtained for the frozen regime of Kernel B. Once the implementation-faithful local updates have been switched off, the remaining receiver-indexed transport dynamics are shown to be deterministic and to settle in finite time under the paper-faithful frozen transport rule extracted from the computational kernel. In that regime, the paper proves one-generation rebroadcast, finite-horizon settling of the residual frozen queue, and, for observer-symmetric frozen states, an exact terminal frozen correction theorem expressing the eventual observer-side correction as a finite functional of the full frozen state.

Kernel A enters the theory more narrowly. The paper does not attempt a full theorem-level treatment of the active split phase. Instead, it proves a conditional bridge result showing when a structurally quiescent late regime of Kernel A may be interpreted in the Kernel B language and hence brought under the same frozen-regime theorem package.

Interpretive scope. The paper does not claim a complete theorem-level treatment of the full parent SERD framework, nor a final continuum identification with gravity, quantum field theory, or particle phenomenology. Its contribution is narrower: it formalizes the parent framework clearly enough that the restricted kernels can be read as subtheories of a common architecture, and it proves a bounded but nontrivial theorem package for the frozen metric regime. Although the exact freeze protocol is introduced as a theorem-level device, it may also be read heuristically as an idealized approximation to a quasi-static or metastable latent sector in which transport settles on a timescale short relative to further structural rewrites. The paper does not prove such an approximation principle, but this interpretation helps explain why frozen-regime results may still be relevant to the broader SERD programme.

Structure of the paper. Section 2 places the SERD framework in the context of relational, discrete-spacetime, and information-based approaches. Section 3 states the paper's formal contract and scope. Section 4 introduces the parent SERD framework. Section 5 defines the restricted kernels. Section 6 separates microscopic law from scheduler family. Section 7 contains the main theorem package. The later sections discuss theorem-target observables, implementation correspondence, limitations, and conclusions.

2 Background and related approaches

The *Space Element Reduction Duplication* (SERD) framework belongs to a broader landscape of attempts to understand spacetime, geometry, and effective dynamics in relational, discrete, and information-bearing terms. The purpose of this section is not to give an exhaustive survey, but to locate the parent SERD framework and the restricted kernels studied here relative to the comparison classes most relevant to the paper's actual results. Those comparison classes are chosen to clarify four aspects of the present model: relations-first ontology, discrete local rewrites, explicit local bookkeeping, and observer-relative reconstruction.

2.1 Relational and structural motivations

A central motivation for the SERD programme is the idea that geometry should be reconstructed from relations internal to the system rather than imposed by an external background. In this respect, SERD belongs to a broad family of relational and structural approaches in which patterns of relation are treated as prior to self-standing objects placed in a fixed geometry. Representative points of contact include process-oriented metaphysical traditions, ontic structural realism, and relational approaches in physics.[1, 2, 3]

What distinguishes SERD within this broader family is that the relational stance is built into a concrete typed latent ontology. The parent framework assumes only three primitive element types—point particles, space elements, and information gaps—and requires that both local dynamics and local memory be realized entirely in terms of those primitives. Background independence is therefore implemented here not only as an interpretive stance, but as a restriction on what the latent substrate is allowed to contain.

This matters for the present paper because its formal results concern not only latent rewrites, but also transported records, frozen transport, and late-time observer recovery. The most relevant comparison class is therefore not relational rhetoric in general, but approaches in which observer-accessible structure is expected to emerge from internal relations rather than from an externally fixed geometry.

2.2 Discrete and combinatorial spacetime programmes

The SERD programme also belongs to the wider landscape of discrete approaches to spacetime, including causal-set theory, causal dynamical triangulations, quantum graphity-style models, spin-network and combinatorial-geometry approaches, and graph-rewriting or hypergraph-based frameworks.[4, 5, 6, 7, 8, 9, 10, 11, 12, 13]

Across these programmes, several themes recur: geometry is treated as emergent rather than fundamental; locality is enforced combinatorially rather than by reference to a prior metric; causal or geometric structure is extracted from patterns of relational update; and continuum behaviour, when it appears at all, must be recovered only after coarse graining, reconstruction, or effective description.

The SERD framework shares these broad aspirations, but differs in several specific ways. First, it uses a typed substrate with an explicit distinction between point particles, space elements, and information gaps. Second, it imposes a two-phase time-step structure separating local update moves from deterministic record propagation. Third, it treats gap-based records as the immediate carriers of observer-relevant geometric information. Fourth, it emphasizes refinement–coarsening balance through the pairing of duplication/reduction and split/broader-coarsening move families in the wider programme.

These distinctions matter here because the theorem-bearing core of the paper is not only about which local rewrites are allowed, but also about what happens to transported records after those rewrites, especially in frozen transport and observer-recovery regimes.

2.3 Information-based and emergent-geometry viewpoints

A second comparison class consists of approaches in which information flow, entropy, or informational structure plays a constitutive role in the emergence of spacetime or effective dynamics.[14, 15, 16, 17, 18, 19, 20, 21]

In SERD, this viewpoint takes a concrete operational form. Local records are stored at information gaps and propagated across the latent substrate according to deterministic rules defined on a typed discrete state. Observer-relative geometry is then reconstructed not from a background metric, but from accumulated local histories encoded in those transported records. Effective distance is therefore not primitive; it is reconstructed from propagated structural or topological information carried internally by the model.

This differs from broader information-first programmes in which entropy, entanglement, or holographic data provide the primary descriptive language. The SERD framework instead begins from a sparse local rewrite system with explicit record carriers and asks what geometry internal observers can recover from those records. In this sense it is closer to an executable information-bearing network dynamics than to an information-theoretic reinterpretation of an already-given continuum structure.

2.4 Biological and morphological analogies

A distinctive feature of the wider SERD programme is its use of filamentary biological transport networks, especially fungal mycelium, as a source of structural analogy and intuition.[22, 23, 24, 25, 26, 27, 28]

The claim is not that spacetime is literally biological, nor that the local rules of SERD are derived from biology. The narrower claim is morphological: branching, reinforcement, pruning, transport, and condensation on filamentary substrates provide a suggestive analogy for how a small family of local refinement–coarsening rules might generate robust large-scale structure together with observer-relevant information flow.

For the present paper this analogy is strictly secondary. It is useful at the level of motivation and interpretation, but it is not part of the theorem-bearing core.

2.5 Prior SERD and SERD-adjacent work

The present paper is part of a wider line of work on the SERD model and related information-propagating network frameworks. Earlier papers and preprints by the present author have reported light-cone-like propagation effects, nonlocal and horizon-like signatures, redshift-scale factor relation correspondence, biologically analogous latent state topology and observed-state mechanics.[29, 30, 31, 26, 32]

The role of the present paper is narrower. It is not intended to replace those conceptual, technical, or numerical studies, but to provide the formal middle layer that turns the wider programme into a more disciplined hierarchy consisting of:

(i) a conceptual and ontological design rationale;

(ii) a parent formal framework together with theorem-bearing restricted kernels;

(iii) executable implementations and numerical tests.

Without such a middle layer, the computational kernels risk appearing ad hoc and the conceptual claims risk remaining underformalized. One of the main contributions of the present paper is therefore to make the parent formalism, the restricted kernels, and the theorem envelope explicit enough that later conceptual and computational claims can be read against a clear formal basis for theorem-bearing claims.

2.6 Scope relative to neighbouring programmes

The SERD framework is closest in spirit to programmes that combine background independence, discrete local rewrites, internal information-bearing transport, and observer-relative reconstruction of effective geometry.

At the same time, SERD is not yet a complete physical theory in the conventional sense. Its analogies to gravity, field theory, effective particle structure, condensation, or large-scale phase behaviour remain partial, regime-dependent, and in many cases still mathematically incomplete.

Accordingly, the contribution of the present paper is not to resolve the larger problem of emergent spacetime in full generality. Its more precise role is to provide a theorem-bearing account of a restricted but nontrivial sector of one such model family, and to clarify how microscopically parameter-free law, scheduler-driven execution, transported records, and observer recovery fit together within a single formal hierarchy.

3 Paper contract and technical scope

This paper has a deliberately restricted role within the current SERD programme. It does not attempt a complete theorem-level treatment of the full parent framework in all of its historically discussed variants. Its purpose is to define a common parent formal setting, isolate the restricted kernels that are currently most amenable to rigorous analysis, and state the strongest currently available results within those kernels. The introduction has already explained the framework at a first-contact level; the present section fixes the interpretive contract that governs everything that follows.

Scope of the paper. The paper is concerned with:

(i) the parent typed latent framework built from point particles, space elements, and information gaps;

(ii) the admissible local move family and deterministic propagation architecture that define the parent microscopic law;

(iii) the separation between that microscopic law and the scheduler families used to generate explicit trajectories;

(iv) two restricted kernels derived from the parent framework;

(v) theorem-level results for those kernels, especially in the frozen transport and observer-recovery regime of Kernel B;

(vi) the formal relation between the parent framework, the restricted kernels, and the executable kernels used in companion computational work.

Parent framework and microscopic law. The wider ontological and design principles of the SERD programme do not single out one uniquely forced microscopic dynamics. Rather, they delimit a family of admissible local, background-free, microscopically parameter-free models built on a common latent ontology and two-phase time-step structure. The parent framework formalized here should therefore be read as a common formal setting from which theorem-level restricted kernels can be selected and compared.

In the sense used throughout this paper, the parent SERD framework is parameter free at the microscopic-law level. Once the latent ontology, admissible local move family, and deterministic propagation rule are fixed, the microscopic law is fixed. Whenever probabilities, time cutoffs, or execution controls appear in explicit implementations, they are to be interpreted as scheduler-level quantities unless explicitly stated otherwise. This does not exclude the emergence of effective large-scale regularities or observer-level parameters in special dynamical regimes; it means only that such quantities are not written into the microscopic law itself as tunable couplings.

Restricted kernels and main formal target. The main formal content of the paper is carried by two restricted kernels derived from the same parent framework:

(i) **Kernel A**, a split-enabled structural kernel;

(ii) **Kernel B**, a constant-point-particle metric kernel.

These kernels are not introduced as independent toy models. They are explicit subtheories of the same parent formalism, selected because they isolate the parts of the dynamics that are currently most amenable to rigorous analysis.

The strongest results proved here are obtained for Kernel B. The central formal target is the exact frozen transport structure that governs the constant-point-particle metric regime once local updates have ceased. On that basis the paper proves deterministic evolution after freeze, one-generation-only rebroadcast, finite-horizon settling of residual transported records, and, for observer-symmetric frozen states, an exact terminal frozen correction theorem expressing the eventual observer-side correction as a finite functional of the full frozen state. It then characterizes late-time observer recovery and observer agreement conditionally, in terms of recovery compatibility of the full frozen state. Kernel A plays a different role: it provides the split-enabled structural regime from which a trajectory may enter a structurally quiescent constant-point-particle regime under stated bridge assumptions. Once that identification is made explicitly, the Kernel B theorem package applies.

What is and is not proved here. Within this contract, the paper proves:

(i) preservation of the relevant state classes for the restricted kernels where explicitly stated;

(ii) shifted-serialisation results showing that simultaneous same-filament SE update schedules are well defined under transported live indices, together with exact commutation for updates supported on distinct filaments;

(iii) deterministic frozen transport, one-generation rebroadcast, and finite settling of residual metric records in Kernel B;

(iv) exact terminal frozen correction results in Kernel B, together with conditional observer-recovery and observer-agreement corollaries formulated in terms of recovery compatibility of the full frozen state;

(v) bridge results identifying when a split-enabled Kernel A trajectory enters a structurally quiescent constant-point-particle regime to which the Kernel B theorem package may be applied.

Only statements explicitly formulated and proved in later sections are to be read as theorem-level claims.

The paper does *not* claim:

(i) a complete theorem-level closure of the full parent SERD move family;

(ii) a full theorem-level treatment of unrestricted split-, merge-, fusion-, or fission-enabled dynamics;

(iii) theorem coverage for every observable discussed in companion computational work;

(iv) a final physical identification of gravity-like, Yukawa-like, condensation, or CPPN-rich behaviour with established continuum theories.

Relation to the conceptual and computational papers. The conceptual paper motivates the relations-first ontology, locality, and ontological minimality. The computational paper studies executable kernels, simulation schedules, and numerical observables across a broader empirical range. The role of the present paper is different: it provides the formal middle layer that makes the parent framework, the restricted kernels, the theorem package, and the implementation correspondence explicit enough that those broader conceptual and computational claims can be read against a clear formal basis for theorem-bearing claims.

Operational reading rule. All later sections are to be read relative to the present contract. The parent framework provides the common formal setting, the restricted kernels fix the theorem-level scope, implementation correspondence remains downstream of the mathematics, and empirical observations are not to be promoted to theorem status without proof.

4 Parent SERD framework

This section defines the parent SERD framework at the level needed for the restricted-kernel formalization developed later in the paper. The aim is not to provide a complete theorem-level treatment of every admissible move sector of the wider programme. Rather, the goal is to state a common typed latent substrate, a common local move vocabulary, and a common propagation architecture from which the restricted kernels of Sec. 5 can be obtained as explicit subtheories.

Why a parent framework is needed. The companion computational paper works with executable restricted kernels, while the wider SERD programme also contains conceptual and ontological arguments about locality, ontological minimality, and refinement–coarsening balance. A parent formal framework is therefore needed to make clear that the restricted kernels are not ad hoc constructions. They are specialized sectors of a common relational architecture with a shared ontology and a shared two-phase time-step structure.

4.1 Typed latent state

Primitive types. The parent SERD framework is built from three primitive element types:

(i) **point particles** (PPs), which serve as distinguished loci of attachment and the sources and sinks of internal observation;

(ii) **space elements** (SEs), which form filamentary adjacency between neighbouring latent sites;

(iii) **information gaps** (IGs), which serve as explicit boundary sites at which local bookkeeping is written, updated, and later propagated.

This typed ontology reflects the programme-level requirement that both local dynamics and local memory be realized entirely in terms of primitive relational elements rather than imposed coordinates or externally supplied metric data.

Latent state as a typed relational object. At each discrete time $t \in \mathbb{N}$, the latent state is a finite typed relational structure

$$G_t = (P_t, S_t, N_t;\, \mathcal{A}_t, \mathcal{T}_t),$$

where:

- P_t is the finite set of point particles;
- S_t is the finite set of space elements;
- N_t is the finite set of information gaps;
- $\mathcal{A}_t$ records the incidence and adjacency relations between these types;
- $\mathcal{T}_t$ records the gap-based transported objects currently stored on the latent state.

Incidence structure. The intended incidence pattern is the following.

(i) A space element joins two neighbouring information gaps and therefore contributes to a filamentary segment.

(ii) A point particle is incident to one or more information gaps (or zero at initialization) and acts as a distinguished endpoint or branching site of the latent structure.

(iii) Information gaps mediate adjacency between point particles and space elements, and they are the sites at which local records are written and from which they subsequently propagate.

The exact combinatorial representation may vary across restricted kernels and implementations, but these type-level roles are fixed throughout the present paper.

Why information gaps are explicit. The explicit inclusion of IGs is a defining feature of the framework. In many discrete network models, local update history is implicit in the graph itself or in external labels. In SERD, by contrast, IGs are the local sites at which refinement–coarsening events leave records. This allows observer-relevant information to be propagated without introducing a background metric or global labels.

4.2 Admissible local move family

At the current stage of the formal programme developed in this paper, the parent SERD framework is built on the typed latent substrate of point particles, space elements, and information gaps, together with a bounded family of strictly local update moves and a deterministic propagation phase. The role of the parent move family is not to assert that one uniquely forced microscopic law follows from the wider conceptual programme. Its role is to define the local move vocabulary from which theorem-bearing restricted kernels can later be selected.

Diagram notation: lowercase p, n, s denote single PP/IG/SE symbols; numeric subscripts (e.g. $n_1, n_2, p_1, p_2, s_a, s_b$) denote concrete single instances created/selected within the local rewrite, while Greek labels (e.g. α, β) denote sets of PPs/IGs/SEs.

Current-stage move types. The broader SERD programme discusses a family of local refinement and coarsening moves acting at both filament and locus level. In the present paper, the theorem-bearing parent vocabulary is organized as follows:

(i) **split:** a local update in which a point particle is replaced by two daughter point particles together with the induced local structural reconfiguration and a newly explicit separating filament;

(ii) **duplication:** a local filament-refinement move in which a space element is replaced by two space elements, with an additional information gap inserted so that the refinement becomes available to later bookkeeping and transport;

(iii) **reduction:** a local filament-coarsening move in which a space element is removed and the neighbouring local boundary structure is consolidated accordingly;

(iv) **broader coarsening sectors:** in the wider SERD programme, complementary coarsening may also be discussed through merge and localized fusion/fission-type channels associated with compact or fused configurations. Fusion-like compact contacts may arise through complete local reduction of separating filament structure, while localized compact-bond reopening plays the corresponding reopening role in the restricted executable kernels considered here.

How this should be read. The theorem-bearing core developed in the present paper is anchored in the split, duplication, reduction, compact-bond reopening, and transport architecture of the restricted kernels. Broader merge, fusion, and fission sectors belong to the larger parent-programme landscape and to executable extensions discussed in companion work, but they are not given a full theorem-level treatment here unless explicitly reintroduced inside a restricted kernel definition. This distinction allows the paper to keep faith with the wider SERD ontology without pretending that

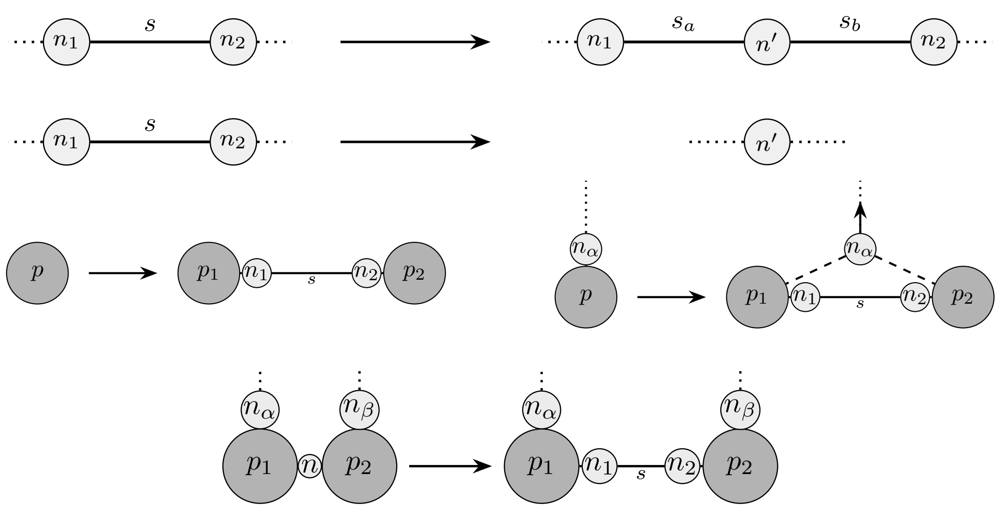

Figure 1: **Core local update motifs of the latent state.** *Top:* filament refinement and coarsening. In the upper row, an SE s between information gaps n_1 and n_2 is duplicated into a longer local filament segment with a newly explicit intermediate gap n'. In the lower row, the same local segment is reduced by removal of s and consolidation of its neighbouring boundary structure into a single gap n'. *Middle:* point-particle split, shown both in the minimal case ($t = 0 \to t = 1$) and in the more general boundary-attached case. A parent point particle p is replaced by daughters p_1 and p_2 separated by a newly explicit local filament. *Bottom:* compact-bond reopening (fission), in which a fused pair regains nonzero separation through insertion of a short filament. These diagrams are schematic locality guides: they indicate type-preserving local reconfiguration on the latent substrate, not metric embedding data.

every historically discussed move sector has already been brought under the same level of formal control.

Why these moves belong together. These moves belong together because they realize the same structural principle at two levels. Duplication and reduction refine and coarsen filaments. Split and broader coarsening sectors refine and coarsen the locus structure. The family is therefore balanced in the precise sense relevant to the wider SERD programme: the law contains both local refinement channels and local coarsening channels, but it does not hard-wire a one-way microscopic drift toward either.

Compact point-particle networks. Some later remarks refer to compact point-particle networks (CPPNs). In the present paper this term is used only in its programme-level sense: a CPPN is a fused or compact point-particle component whose members have no resolved external separation in the relevant observed-state description, although the component may retain internal relational structure. CPPN-rich behaviour, condensation, and effective interaction analogies are not theorem-bearing claims in this paper; they belong to the broader computational and phenomenological programme.

Definition 4.1 (Current-stage parent move family)**.** *Let G_t be a well-typed latent* SERD *state. The current-stage parent move family*

$$\mathcal{M}_{\mathrm{SERD}}$$

is the set of all admissible local moves of theorem-bearing types split, duplication, and reduction,

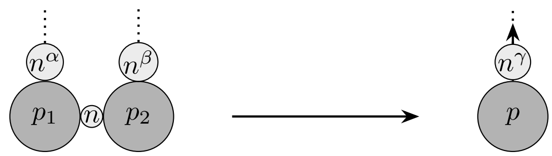

Figure 2: **PP merge.** Two point particles p_1 and p_2, joined by a fused contact across the shared information gap n, merge into a single point particle p. The symbols n^α and n^γ denote the external information-gap neighbourhood sets attached to p_1 and p_2, respectively; after merger these are reattached to the merged PP p as a new local IG neighbourhood structure. This diagram is included to illustrate the broader parent SERD move family; *PP merge is not part of the restricted kernels analyzed in the present paper.*

together with the broader coarsening sectors discussed for the full SERD *programme, where all such moves act on bounded neighbourhoods of G_t and preserve the well-typed latent-state class whenever they are admitted.*

Remark 4.2 (Current theorem envelope)**.** *For the purposes of the present paper, the strongest theorem package is developed for restricted kernels built from split-enabled structural propagation and constant-point-particle metric transport. References to merge, localized fusion, or broader fission sectors should therefore be read as part of the wider* SERD *move landscape unless they are explicitly reintroduced inside a later formal kernel definition.*

Diagrammatic role. The schematic update-rule figures included in this section should therefore be read as locality guides: they display how the main latent motifs refine or coarsen the typed substrate, but they do not by themselves enlarge the theorem envelope beyond the kernel definitions given later in the paper.

4.3 Gap-based transported objects

Why transported objects are needed at all. Once local moves modify the latent substrate, later local dynamics and internal observers must have access to some controlled memory of those modifications. In the SERD framework that memory is not stored in an external ledger. It is written onto the latent structure itself, at information gaps or at receiver-indexed local faces derived from them, and is then transported by deterministic local rules. This is the point at which information gaps become more than passive separators: they become the sites at which local structural and metric histories are made available to the future evolution of the system.

Two transported sectors. At the parent level it is useful to distinguish two broad classes of transported object:

(i) **structural records**, which encode the propagation of split-like branching effects and other local structural reorganizations across the latent substrate;

(ii) **metric records**, which encode the local refinement or coarsening increments associated with duplication and reduction along filaments.

The distinction is schematic at the parent level, but it becomes operationally important in the

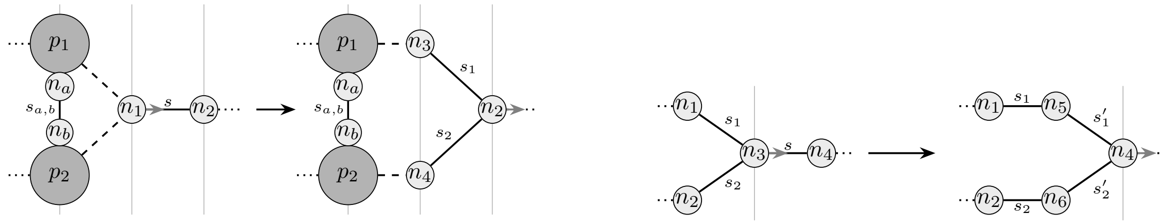

Figure 3: **Split-generated structural propagation motifs.** *Left:* first outward transmission from a split-adjacent boundary region. The local branching created by a split is written at neighbouring gaps and then begins to propagate away from the split site. *Right:* one further transport step along a filament, showing the same structural branching effect advancing by a local rewrite of the neighbouring gap/segment pattern. These panels are included to give intuition for the structural transport sector retained in Kernel A. They should be read as parent-schematic propagation motifs rather than as standalone theorem statements.

restricted kernels. Kernel A retains the split-enabled structural sector, while Kernel B retains the metric sector in a constant-point-particle setting.

Structural records. Structural records are the parent-level carriers of branching history. Their role is to ensure that a local structural change is not forgotten the instant the underlying rewrite is executed. Instead, the effect of that change is written locally and then propagated outward across the latent structure so that later local configurations and internal observers can still register that a branching event has occurred. The diagrams in Figs. 3 and 4 are useful here because they make clear that the structural sector is not a global tag pasted onto the state; it is a local transport process on the same typed substrate.

Metric records. Metric records play a different role. They summarize local refinement and coarsening histories along filaments. When interpreted from the perspective of internal observers, accumulated metric records behave like effective separations or path-length increments. In this sense, metric records provide the minimal parent-level route from local rewrite history to observer-relative distance reconstruction.

What the parent framework commits to, and what it does not. The parent framework commits to the existence of these transported sectors and to their attachment to the latent structure. It does *not* yet commit to one unique low-level representation of every carrier type across all kernels. In some restricted settings, transported records are stored directly on gaps. In others, especially in the constant-point-particle metric specialization, they are organized on receiver-indexed local faces. The invariant content is that they are internal to the latent state and that their subsequent motion is governed by deterministic local transport.

Why the distinction matters for the theorem package. This split between structural and metric transport is one of the reasons the present paper can maintain a bounded theorem envelope. The strongest theorems are obtained in the metric sector after freeze, where deterministic transport and finite settling can be analyzed cleanly. The structural sector remains important, especially for Kernel A and for the bridge into a quiescent constant-point-particle regime, but the present theorem package does not attempt to provide a complete parent-level structural-carrier algebra for every conceivable inverse or coalescing process.

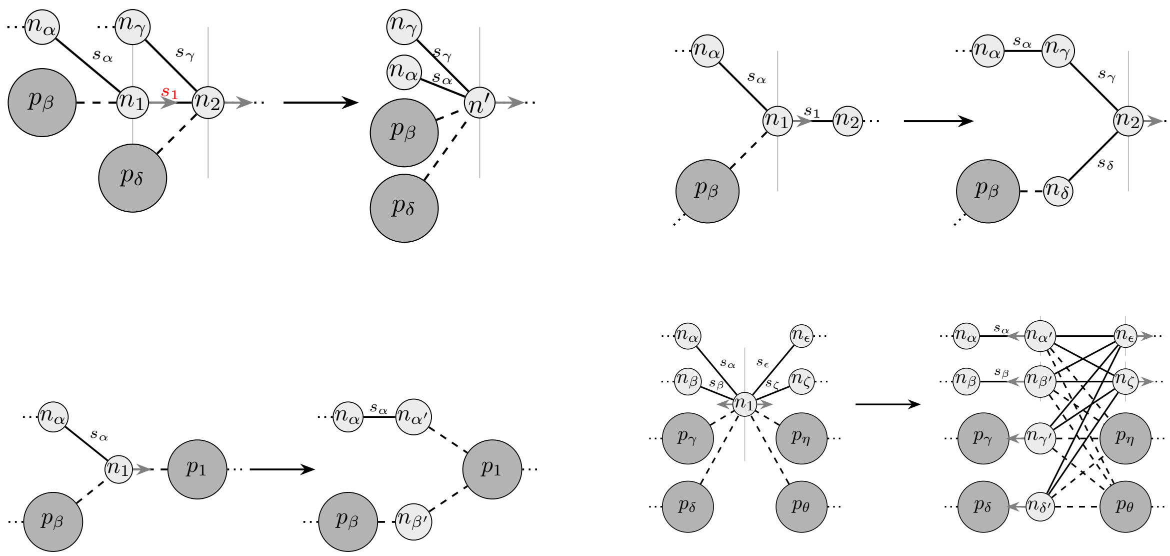

Figure 4: **General structural-carrier motifs on the latent substrate.** *Top left:* local generation of a structural carrier at a branching interface. *Top right:* propagation of that carrier across a filament segment by a further local transport step. *Bottom left:* update on a point-particle boundary, where the incoming structural record is absorbed into new local boundary records. *Bottom right:* collision or resolution of structural carriers at a shared gap, producing a more elaborate local boundary reconfiguration. The role of this figure is explanatory: it gives geometric intuition for the structural transport layer discussed at the parent level and for the split-enabled structural sector of Kernel A. The formal theorem envelope remains the one explicitly stated in the text.

Reader guidance. The best way to read this subsection is therefore as follows. The parent framework needs two kinds of transported memory because local refinement/coarsening and local structural reorganization play different formal roles. The figures are included to give the reader a concrete schematic grip on those two roles before the paper narrows to kernel-specific formalizations. The theorem-bearing claims, however, remain exactly those stated later in the manuscript and should not be read off from the diagrams alone.

4.4 Two-phase time-step structure

Update phase followed by propagation phase. Each discrete time step in the parent SERD framework factorizes into two conceptually distinct phases:

(i) a **local update phase**, in which an admissible family of local rewrite moves is selected and executed on the current latent state;

(ii) a **propagation phase**, in which the records written, modified, or exposed at information gaps during the update phase are advanced deterministically across the updated latent structure.

Why the factorization is essential. This split is not a presentational convenience. It is one of the formal devices that makes the wider architecture intelligible. The update phase is the place where scheduler dependence can enter: compatible local rewrites may be sampled, ordered, or executed according to a chosen scheduler family. The propagation phase is where that dependence stops: once the local updates for the time step have been fixed, the transport of the resulting records is determined by the latent state and the propagation rule itself.

What the figures are meant to show. The schematic transport diagrams included near this subsection should be read precisely in this light. Figures 3 and 4 display local propagation motifs that occur only after the relevant local structural configuration has already been created in the update phase. They are therefore naturally read as illustrations of the second half of the two-phase schedule rather than as standalone rewrite rules.

Parent-level scheduler neutrality. At the parent level, the framework does not prescribe a unique scheduler. What is fixed is the microscopic law: the typed latent ontology, the admissible move vocabulary, and the deterministic propagation architecture. How compatible local moves are selected in finite trajectories is left to scheduler families, which are treated separately in Sec. 6. This is the formal basis for interpreting simulation-level probabilities as execution choices rather than as microscopic couplings.

Consequences for the restricted kernels. The same two-phase architecture is inherited by both restricted kernels, but the active transported sector differs between them. Kernel A couples its split-enabled local update sector to structural propagation. Kernel B couples duplication/reduction updates to deterministic metric transport on a fixed point-particle set. The common two-phase structure is therefore one of the strongest reasons to treat the kernels as realizations of a common parent framework rather than as disconnected constructions.

Conceptual payoff. This factorization also clarifies the observer story. Internal observers never have access to a God's-eye latent state together with all unrealized update possibilities. What they receive are the records that have actually been written locally and then propagated to them through the second phase. In that sense, observer-relative geometry and causal accessibility are downstream of the two-phase architecture itself.

4.5 How the restricted kernels arise

The restricted kernels studied in this paper are obtained by selecting theorem-bearing subfamilies of the current-stage parent move family $\mathcal{M}_{\mathrm{SERD}}$, together with corresponding restrictions on the active propagated record sector and on the admissible scheduler family. In this way, the kernels inherit the parent ontology and two-phase time-step structure while differing in which local moves are retained as part of their theorem-bearing core.

Definition 4.3 (Restricted realization of the current-stage parent framework)**.** *A **restricted realization** of the current-stage parent* SERD *model is specified by:*

(i) a chosen subfamily

$$\mathcal{M}_{\mathrm{res}} \subseteq \mathcal{M}_{\mathrm{SERD}}$$

of admissible local moves;

(ii) a chosen propagated record sector retained during the deterministic propagation phase;

(iii) a scheduler family selecting compatible local update families from $\mathcal{M}_{\mathrm{res}}$.

Current role of Kernels A and B. Kernels A and B are the two restricted realizations studied in this paper. They are not independent models introduced ad hoc, but theorem-bearing realizations of the current-stage parent SERD framework.

Kernel A. Kernel A is the split-enabled structural realization. Its theorem-bearing core retains split, duplication, reduction, and the associated structural propagation sector. In the implemented companion simulations, this realization may also include localized fission activity and the compact-bond states generated by complete filament reduction.

Kernel B. Kernel B is the constant-point-particle metric realization. Its theorem-bearing core retains duplication, reduction, and deterministic metric transport on a fixed point-particle set. In the implemented companion simulations, this realization also admits compact-bond states corresponding to zero-length separation and localized reopening of such states by fission.

Current proof strategy. The present paper does not attempt a uniform theorem-level treatment of the full current-stage parent move family. Instead, it develops the strongest current results for theorem-bearing kernel cores whose dynamics are sufficiently explicit to support rigorous state-class, settling, reconstruction, and bridge results. The broader parent move family remains the formal context in which these kernels are situated.

5 Restricted kernels

The parent SERD framework introduced in Sec. 4 specifies a typed latent-state class, a parent family $\mathcal{M}$ of admissible local moves, and a deterministic propagation architecture for gap-based records. This section extracts from that parent framework the two restricted kernels that carry the theorem package of the paper. These kernels are not introduced as independent models. They are explicit subtheories of the parent framework, chosen because they isolate the parts of the dynamics that are currently most amenable to rigorous analysis and clean correspondence with the companion implementations.

Kernel-selection principle. The restricted kernels are obtained by selecting subfamilies of the parent move family $\mathcal{M}$ and, where necessary, by restricting the transported record sector. In this way, both kernels inherit the parent ontology and the two-phase time-step structure while differing in which local updates and which propagated records are retained.

5.1 Kernel A: split-enabled structural kernel

Informal role. Kernel A is the split-enabled structural kernel used in the split-growth and split-freeze computational experiments. It is the closest implemented kernel to the structural parent SERD picture in which point-particle splits generate propagating structural records on the latent substrate.

Definition 5.1 (Implementation-faithful Kernel A state)**.** *An **implementation-faithful Kernel A state** at time t consists of:*

(i) a finite set P_t of point particles;

(ii) a finite set N_t of information gaps;

(iii) a finite set S_t of space elements, each joining two information gaps;

(iv) a finite set Π_t of propagating structural markers carried at information gaps;

(v) the corresponding incidence and neighborhood data needed to determine the local update rules and the deterministic propagation rule.

Definition 5.2 (Implementation-faithful Kernel A update phase)**.** *A single implementation-faithful Kernel A update phase consists of the following local operations:*

(i) point-particle splits;

(ii) space-element duplications;

(iii) space-element reductions;

(iv) optionally, localized point-particle fission moves on fused-gap motifs when that extension is enabled in the scheduler family.

Definition 5.3 (Implementation-faithful Kernel A propagation phase)**.** *After the update phase, the structural markers in Π_t propagate deterministically by repeated local propagation and local collision rules until no further propagation step remains in the current time step.*

Remark 5.4 (Theorem-bearing emphasis)**.** *The present paper does not attempt a full theorem-level analysis of the active split-enabled regime of Kernel A. The role of Kernel A in the theorem package is narrower: it provides the implemented structural-growth regime from which one may later enter a structurally quiescent constant-point-particle regime that can be identified with a metric Kernel B state.*

Remark 5.5 (Relation to the computational paper)**.** *The computational experiments using Kernel A include an optional localized fission sweep in the update phase. For this reason, the implementation-faithful Kernel A move family is slightly broader than the minimal structural core consisting only of split, duplication, reduction, and structural propagation. The theorem package later in the paper will therefore distinguish the implemented Kernel A regime from the narrower post-quiescence regime to which the Kernel B theory is applied.*

5.2 Kernel B: constant-point-particle metric kernel

Informal role. Kernel B is the constant-point-particle metric kernel used in the frozen-regime observer-recovery experiments of the companion computational work. It fixes the point-particle set, evolves pairwise latent filament lengths by local refinement and coarsening, and transports signed metric records on receiver-indexed faces at information gaps. The strongest theorem-bearing results in the present paper are proved for the frozen transport regime of this kernel.

Definition 5.6 (Implementation-faithful Kernel B state)**.** *An* ***implementation-faithful Kernel B state*** *at time t consists of:*

(i) a fixed finite point-particle label set

$$P = \{1, \ldots, n\};$$

(ii) for each unordered pair $\{p, q\} \subseteq P$ with $p \neq q$, a latent filament length

$$L_{pq}(t) \in \mathbb{N}_0;$$

(iii) for each ordered pair $p \neq q$, a receiver-indexed face array

$$A^{p \leftarrow q}(t) \in \mathbb{Z}^{\{0,\ldots,L_{pq}(t)\}\times P},$$

where row 0 is adjacent to receiver p;

(iv) for each observer $p \in P$, an observer matrix

$$O^p(t) \in \mathbb{Z}^{P\times P}.$$

Definition 5.7 (Compact bond)**.** *For $p \neq q$, the condition*

$$L_{pq}(t) = 0$$

is called a ***compact bond*** *between p and q. It is the theorem-level latent representation of a completely reduced pairwise filament.*

Definition 5.8 (Implementation-faithful Kernel B update phase)**.** *A single implementation-faithful Kernel B update phase consists of:*

(i) an optional compact-bond reopening pre-pass, in which a compact bond $\{p, q\}$ may be reopened by

$$L_{pq} = 0 \longmapsto L_{pq} = 1,$$

with one new row inserted on each receiver-indexed face and boundary deposits

$$A^{p\leftarrow q}_{0,p} \mathrel{+}= 1, \qquad A^{q\leftarrow p}_{0,q} \mathrel{+}= 1.$$

Here row 0 is the receiver-adjacent boundary row, while columns p and q are the receiver-owned radial slots on the corresponding faces;

(ii) a left-to-right edge-wise schedule of space-element duplications and reductions on each positive-length filament, applied at the current live row of the affected space element. Duplications insert a zero face row and deposit $+1$ in the receiver's own column; reductions merge adjacent face rows by full-vector summation and deposit -1 in the receiver's own column.

Definition 5.9 (Implementation-faithful Kernel B propagation phase)**.** *After the update phase, metric records propagate deterministically by receiver-indexed face transport. This transport rule is stated precisely in Sec. 7.5.*

Remark 5.10 (Theorem-bearing focus)**.** *The implemented computational-paper Kernel B includes the optional compact-bond reopening pre-pass described above. The strongest theorems in the present paper, however, concern the frozen regime that begins after all local updates—including compact-bond reopening, duplication, and reduction—have been switched off. For this reason the later theorem package depends only on the frozen state and on the deterministic transport rule, not on whether compact-bond reopening occurred earlier in the trajectory.*

5.3 Relation between the two kernels

Nested viewpoint. Kernel A and Kernel B are not unrelated constructions. Both are derived from the same parent SERD framework, but they isolate different sectors of it. Kernel A retains a structurally active split-enabled regime, whereas Kernel B isolates a constant-point-particle metric regime with no active structural carriers.

Why the distinction matters. The distinction between the two kernels is not merely terminological. It separates:

(i) a structurally active regime in which point-particle branching may still occur and structural carriers may still propagate;

(ii) a constant-point-particle regime in which the theorem-bearing core contains only filamentary duplication/reduction together with metric-record transport.

This separation is the formal basis for the bridge theorem proved later. The theorem does not say that an active Kernel A execution literally becomes a Kernel B execution. Rather, it shows that under finite split stopping and finite structural-carrier lifetime, a Kernel A trajectory may enter a structurally quiescent constant-point-particle latent regime, and that under an explicit identification of that regime with a Kernel B state, the Kernel B theorem package becomes applicable.

Role in the paper. Kernel A therefore plays the role of a split-enabled precursor regime, while Kernel B plays the role of the main theorem-bearing metric regime. The former motivates the bridge theorem; the latter carries the strongest results: deterministic evolution after freeze, one-generation rebroadcast, finite settling, exact terminal frozen correction for observer-symmetric full frozen states, and conditional observer recovery and observer agreement under recovery compatibility of the full frozen state.

Representative visualizations. Before turning to the scheduler layer and theorem package, Figs. 5 and 7 show representative visualization-only trajectories of the two restricted kernels. These figures are included to make the restricted state spaces and their distinct dynamical roles visually legible. They are illustrative only and are not used as premises in the proofs below; all theorem-bearing claims depend solely on the formal state spaces, update rules, and state-transition definitions.

6 Schedulers and implementation architecture

This section distinguishes the microscopic law of the SERD framework from the execution structures used to generate explicit trajectories. In the present paper, the microscopic law consists of the latent-state class, admissible local move family, and deterministic propagation rule of the relevant kernel. By contrast, executable trajectories require additional choices about which compatible local moves are attempted, in what order they are executed, when particular move classes are suppressed, and when a latent state is declared frozen. Those additional choices are treated here as *scheduler families*.

The distinction matters because it preserves the intended parameter-free character of the microscopic law while still allowing explicit Monte Carlo or deterministic execution procedures for restricted kernels. Probabilities, stopping horizons, freeze times, and similar execution controls are therefore interpreted as scheduler-level quantities unless explicitly stated otherwise.

6.1 Microscopic law and scheduler families

Microscopic law. For the purposes of the present paper, the microscopic law consists of:

(i) the typed latent-state class;

(ii) the admissible local move family of the relevant kernel;

(iii) the deterministic propagation rule acting during the second phase of each time step.

Scheduler family. A scheduler family specifies how that microscopic law is sampled or executed in finite trajectories. At this level, a scheduler may determine:

(i) which compatible local moves are proposed at a given time step;

(ii) how overlapping or conflicting proposals are resolved;

(iii) in what order accepted moves are executed;

(iv) whether particular move sectors are suppressed after a finite horizon;

(v) whether a trajectory enters a frozen regime after some finite time.

Interpretive rule. Throughout the paper, if a probability, control threshold, stopping time, or execution-order convention appears in a computational implementation, it is interpreted as part of the scheduler family unless explicitly stated otherwise. In particular, split probabilities, duplication and reduction proposal probabilities, finite split horizons, and freeze times are not treated as microscopic physical parameters of the parent SERD law.

6.2 Two-phase execution structure

Each time step is organized into two phases.

Phase I: local updates. The first phase applies an admissible family of local rewrite moves to the current latent state. The parent framework does not require a unique update scheduler. Different admissible implementations may select and resolve local proposals differently, provided they remain within the move family of the kernel being executed.

Phase II: deterministic propagation. Once the update phase has been completed, the second phase propagates the transported records stored on the latent structure according to the deterministic propagation rule of the kernel. This is the phase in which structural carriers or metric records are advanced across the already updated latent state.

This separation is especially important later in the paper, where the strongest theorem package isolates deterministic post-freeze transport in Kernel B.

6.3 Kernel-specific scheduler families

Kernel A scheduler families. A scheduler family for Kernel A may select among:

(i) point-particle split proposals;

(ii) space-element duplication proposals;

(iii) space-element reduction proposals;

(iv) optional localized boundary-opening proposals in executable variants.

It may also impose a finite split horizon after which no further split moves are attempted. Such a horizon is important for the bridge assumptions, but it is a scheduler choice rather than a change in the meaning of a split.

Kernel B scheduler families. A scheduler family for Kernel B may select among duplication and reduction proposals on a fixed point-particle set. It may also impose a finite freeze time after which no further update moves are attempted. In executable variants, localized boundary-opening moves associated with fused-boundary configurations may also be controlled at the scheduler level, but these remain outside the theorem-bearing core unless explicitly added as an extra hypothesis.

Freeze as a scheduler condition. In the theorem-bearing setting, freeze marks the transition from an update-plus-propagation regime to a propagation-only regime. Once freeze has occurred, no further local update moves are applied and subsequent evolution is governed entirely by the deterministic transport law of the relevant kernel.

6.4 Admissibility and execution order

At the level of the parent framework, an admissible update family is any collection of local proposals that the scheduler is permitted to execute on the current latent state while remaining within the kernel's move family. The theorem package later studies particular restricted classes of such update families because they are mathematically tractable, but the scheduler notion itself is broader.

Two execution viewpoints must be distinguished:

(i) within a single filament, simultaneous SE update schedules are interpreted through transported live indices, so that old SE labels are followed through the insertions and deletions generated earlier in the same tick;

(ii) across distinct filaments, local SE update maps act on disjoint state components and therefore commute exactly.

In the reference implementation this corresponds to sampling start-of-tick SE actions and executing them left-to-right using a cumulative live-index shift. The theorem package formalizes this convention as a shifted-serialisation result rather than as a naive fixed-row-index commutation claim. This prevents implementation-order conventions from being mistaken for unrestricted global order-independence statements.

6.5 Reference implementation architecture

The companion computational work uses explicit reference implementations of the restricted kernels. These implementations instantiate the scheduler family, the latent-state update phase, and the propagation phase using concrete data structures and execution conventions.

Kernel A implementation pattern. In the split-enabled structural sector, the reference implementation maintains explicit latent dictionaries for point particles, space elements, information gaps, and structural carriers. The scheduler samples split, duplication, and reduction proposals, applies them to the latent state, and then propagates structural carriers deterministically. Optional localized boundary-opening moves may also be included in the executable family.

Kernel B implementation pattern. In the constant-point-particle metric sector, the reference implementation maintains a fixed point-particle set, integer filament lengths on every pairwise latent connection, receiver-indexed face arrays carrying transported metric records, and observer-indexed matrices updated during the propagation phase. The update phase acts only on filament refinement and coarsening, while the propagation phase absorbs and rebroadcasts metric records deterministically. This is the implementation-level counterpart of the theorem-bearing Kernel B specialization proved later.

Visualization routines, embedding controls, and output settings belong to the computational presentation layer only and play no role in the theorem package.

6.6 How the scheduler layer interfaces with the theorem package

The theorem package does not require a full classification of all admissible scheduler families. Instead, it uses scheduler assumptions in a limited and explicit way. Typical examples include:

(i) the existence of a finite split horizon in Kernel A;

(ii) the existence of a finite freeze time in Kernel B;

(iii) the use of shifted live-index serialisation for same-filament SE update schedules in Kernel B;

(iv) the restriction to distinct-filament update families when exact cross-filament commutation is invoked.

Many trajectory-level questions remain scheduler-dependent, including active-phase branching histories, the exact timing of quiescence, and the empirical shape of numerical observables in the companion computational paper. The present theorem package therefore does not eliminate scheduler dependence in general; it shows what can be proved once the relevant scheduler conditions have been stated explicitly.

With the parent framework, the restricted kernels, and the scheduler layer now distinguished, the next task is to state the theorem package itself. That package is strongest in the constant-point-particle metric sector, where post-freeze transport becomes deterministic and where finite settling and late-time observer recovery can be formulated precisely. The split-enabled structural kernel enters that theorem package only through a narrower conditional bridge result into a structurally quiescent constant-point-particle latent regime.

7 Main theorem package

This section contains the main theorem-bearing results of the paper. The strongest results are obtained for Kernel B, the constant-point-particle metric kernel, and in particular for its frozen transport regime after local update moves have ceased. In that setting the paper proves deterministic post-freeze evolution, one-generation-only rebroadcast, finite settling of residual metric records, and, for observer-symmetric full frozen states, an exact terminal frozen correction theorem expressing the eventual observer-side correction as a finite functional of the full frozen state. The later recovery and agreement results are then stated conditionally in terms of recovery compatibility of that full frozen state.

Kernel A enters the theorem package in a narrower way. The paper does not attempt a full theorem-level treatment of the active split-enabled phase. Instead, it proves a conditional bridge result showing when a structurally quiescent late regime of Kernel A may be identified with the frozen Kernel B language, so that the frozen-regime Kernel B theorem package applies.

All theorem statements in this section are to be read relative to the contract fixed in Sec. 3. In particular, theorem-level claims are restricted to the explicit kernel scopes stated below; implementation correspondence remains downstream of the mathematics; and empirical observations from companion computational work are not used as substitutes for formal proof.

7.1 Claim-status and proof order

Purpose of this subsection. The present paper combines a parent parameter-free SERD formalism, restricted kernels, code-level implementation correspondences, and a bounded theorem package. To prevent ambiguity, we classify the claims that appear in this section and fix the order in which the results are proved.

Definition 7.1 (Claim-status classes)**.** *Every substantive statement in this section belongs to one of the following classes:*

(i) ***Kernel theorem:*** *a proved statement for a specifically named restricted kernel under explicitly stated assumptions;*

(ii) ***Bridge theorem:*** *a proved conditional statement showing when a trajectory of one restricted kernel enters a regime to which the theorem package of another kernel may be applied under an explicit identification;*

(iii) ***Implementation correspondence statement:*** *a statement identifying how a formal kernel, hypothesis, or transported object is realized in the reference computational code;*

(iv) ***Empirical-only statement:*** *a numerical observation, computational signature, or phenomenological pattern reserved for companion computational work and not claimed here as a theorem.*

Contractual consequence. No empirical-only statement may be used as a premise in a formal proof. Implementation correspondence statements may clarify why a theorem is relevant to a computational experiment, but they do not replace formal definitions, hypotheses, or deductions.

Proof order adopted in this paper. The theorem package is developed in the following order:

Step 1. define the theorem-level specialization of Kernel B;

Step 2. prove state-class preservation under theorem-level Kernel B local-update ticks;

Step 3. prove that shifted serialisation gives a well-defined simultaneous SE update map, and prove exact commutation across distinct filaments;

Step 4. define the frozen Kernel B transport rule and frozen queue space;

Step 5. prove deterministic evolution after freeze, one-generation rebroadcast, and finite settling of residual metric records;

Step 6. define the full frozen state, frozen queue operator, and exact terminal frozen correction;

Step 7. characterize late-time observer recovery and observer agreement in terms of the exact terminal frozen correction and recovery compatibility of the full frozen state;

Step 8. prove the bridge from Kernel A to a structurally quiescent constant-point-particle regime and state the identification under which the Kernel B theorem package applies.

Interpretive note. This proof order reflects the logical dependence of the results. The observer-recovery theorem is not stated as a free-standing claim about arbitrary observer states. It is stated only after the frozen transport rule, one-generation rebroadcast structure, finite-settling result, and exact terminal frozen correction have been defined. Likewise, the bridge result does not say that an active Kernel A execution is already a Kernel B execution. It isolates a later quiescent regime and then states the additional identification needed before the Kernel B results may be invoked.

7.2 Kernel B theorem-level specialization

We now make the theorem-level Kernel B state explicit in the notation used by the theorem package. Throughout this section, P denotes a fixed finite point-particle label set with $|P| = n$. Matrix and vector indices are written using labels in P, rather than by choosing a particular integer encoding of those labels.

Definition 7.2 (Theorem-level Kernel B state)**.** *A theorem-level Kernel B state at time t is a triple*

$$X(t) = \big(L(t), A(t), O(t)\big)$$

consisting of:

(i) a symmetric latent separation matrix

$$L(t) = \big(L_{pq}(t)\big)_{p,q\in P} \in \mathbb{N}_0^{P\times P}, \qquad L_{pp}(t) = 0, \qquad L_{pq}(t) = L_{qp}(t);$$

(ii) a family of receiver-indexed face arrays

$$A(t) = \big\{A^{p\leftarrow q}(t)\big\}_{p\neq q}, \qquad A^{p\leftarrow q}(t) \in \mathbb{Z}^{\{0,\dots,L_{pq}(t)\}\times P};$$

(iii) a family of observer matrices

$$O(t) = \big\{O^{p}(t)\big\}_{p\in P}, \qquad O^{p}(t) \in \mathbb{Z}^{P\times P}.$$

Definition 7.3 (Receiver-indexed face record)**.** *Fix $p \neq q$. The array*

$$A^{p\leftarrow q}(t) \in \mathbb{Z}^{\{0,\dots,L_{pq}(t)\}\times P}$$

is the receiver-indexed face of the filament $\{p, q\}$ whose row coordinate is measured from the receiver p. If

$$A^{p\leftarrow q}_{r,c}(t) \neq 0,$$

then:

(i) $r \in \{0, \dots, L_{pq}(t)\}$ is the face-row position, with $r = 0$ the boundary row adjacent to receiver p;

(ii) $c \in P$ is the carried PP label of the stored metric record;

(iii) the entry is called ***radial at*** p *precisely when* $c = p$.

Thus row 0 *is a boundary-position index, whereas radiality is a column-label condition. The radial slot on the face* $A^{p\leftarrow q}$ *is column* p*, independently of how the label set* P *is encoded in an implementation.*

Definition 7.4 (Theorem-level Kernel B local-update tick)**.** *Fix a theorem-level Kernel B state* $X(t)$*. A* ***theorem-level Kernel B local-update tick*** *consists of the following local update on* $L(t)$ *and* $A(t)$*. Observer matrices are not changed during this local-update tick.*

(i) ***Compact-bond reopening pre-pass.*** *For each unordered pair* $\{p, q\}$ *with* $L_{pq}(t) = 0$*, either do nothing or apply the local reopening move*

$$L_{pq}(t) = 0 \;\longmapsto\; L_{pq}(t)^{+} = 1.$$

On the two receiver-indexed faces, one new row is inserted so that the arrays have shape

$$\mathbb{Z}^{\{0,1\}\times P},$$

and the local metric increment is deposited into the receiver's own column at the receiver-adjacent boundary row:

$$A^{p\leftarrow q}_{0,p} \mathrel{+}= 1, \qquad A^{q\leftarrow p}_{0,q} \mathrel{+}= 1.$$

Here the index 0 *denotes the receiver-adjacent row, while* p *and* q *denote receiver-owned radial columns.*

(ii) ***Filament-wise duplication/reduction schedule.*** *For each unordered pair* $\{p, q\}$ *with current positive length*

$$\ell = L_{pq}(t)^{+},$$

choose an action schedule

$$u_{pq,t} \in \{-1, 0, +1\}^{\{0,\ldots,\ell-1\}}.$$

The value $u_{pq,t}(s) = +1$ *denotes duplication of the old SE at start-of-tick index* s*, the value* $u_{pq,t}(s) = -1$ *denotes reduction of that old SE, and* $u_{pq,t}(s) = 0$ *denotes no local update at that old SE.*

(iii) ***Shifted live-index execution.*** *The schedule* $u_{pq,t}$ *is sampled relative to the start-of-tick SE indexing. It is executed by scanning old SE indices* $s = 0, \ldots, \ell - 1$*. If earlier insertions or deletions on that same filament have shifted the current live location of old index* s *to*

$$s' = s + \sigma_u(s), \qquad \sigma_u(s) := \sum_{h<s} u_{pq,t}(h),$$

then the action $u_{pq,t}(s)$ *is applied at live index* s'*, provided*

$$0 \le s' < L'_{pq},$$

where L'_{pq} *is the current filament length immediately before that action is executed. If this inequality fails, the action is skipped.*

(iv) ***Local duplication bookkeeping.*** *If a duplication is executed at live index s', then one zero row is inserted into each receiver-indexed face at the corresponding local position. On face $A^{p\leftarrow q}$, the metric increment is deposited into the receiver-owned radial column on the row adjacent to the duplicated SE:*

$$A^{p\leftarrow q}_{s',p} \mathrel{+}= 1.$$

On the mirrored face $A^{q\leftarrow p}$, the corresponding mirrored row is used and the deposited column is q. The common latent length $L_{pq} = L_{qp}$ increases by one.

(v) ***Local reduction bookkeeping.*** *If a reduction is executed at live index s', then the two face rows bounding the reduced SE are merged by summing their full row vectors on each directed face. On face $A^{p\leftarrow q}$, the metric decrement is deposited into the receiver-owned radial column on the merged row:*

$$A^{p\leftarrow q}_{s',p} \mathrel{-}= 1.$$

On the mirrored face $A^{q\leftarrow p}$, the corresponding mirrored row is used and the deposited column is q. The removed row is then deleted from each face array, and the common latent length $L_{pq} = L_{qp}$ decreases by one.

Proposition 7.5 (State-class preservation under Kernel B local-update ticks)**.** *Let $X(t)$ be a theorem-level Kernel B state. Then the state obtained after one theorem-level Kernel B local-update tick is again a theorem-level Kernel B state.*

Proof. *The compact-bond reopening pre-pass acts only on unordered pairs $\{p, q\}$ with $L_{pq}(t) = 0$. When it is applied, the updated length is*

$$L_{pq}(t)^+ = L_{qp}(t)^+ = 1,$$

so symmetry of the latent separation matrix is preserved. On the two directed faces $A^{p\leftarrow q}$ and $A^{q\leftarrow p}$, exactly one new row is inserted, so both arrays retain the required receiver-indexed shape. No point particle is created or destroyed, and observer matrices are unchanged during this pre-pass.

Now fix an unordered pair $\{p, q\}$ with current positive length. A duplication executed on that filament increases the common latent length by one and inserts one new row on each directed face. A reduction decreases the common latent length by one and replaces one adjacent face-row pair on each directed face by its merged row, followed by deletion of the removed row. In both cases the resulting arrays have shape

$$\mathbb{Z}^{\{0,\ldots,L'_{pq}\}\times P}$$

for the updated length L'_{pq}, and the latent length remains in $\mathbb{N}_0$.

The shifted live-index rule changes only which current local filament position is updated next. It does not alter the symmetry of the latent matrix, the receiver-indexed array format, the fixed point-particle set, or the dimensions of the observer matrices.

Therefore, after the full local-update tick, the latent separation matrix is still symmetric with zero diagonal, every directed face array still has the required receiver-indexed shape, and the observer family still lies in $\mathbb{Z}^{P\times P}$. The resulting state is again of the form stated in Definition 7.2. □

7.3 Shifted serialisation of simultaneous SE updates

The Kernel B local-update tick treats SE duplications and reductions on a filament as a simultaneous start-of-tick action schedule, but serialises that schedule in a fixed left-to-right order using shifted live indices. This subsection records the mathematical content of that serialisation.

Fix an unordered PP pair $\{p, q\}$, and write its start-of-tick filament length as ℓ. On the directed face with receiver p, the metric-record field is a row-vector sequence

$$a = (a_0, \dots, a_\ell), \qquad a_r \in \mathbb{Z}^P.$$

Let $e_p \in \mathbb{Z}^P$ denote the unit vector in the receiver column p.

For $0 \le r < \ell$, define the facewise duplication and reduction maps

$$D_r^p(a_0, \dots, a_\ell) = (a_0, \dots, a_r + e_p, 0, a_{r+1}, \dots, a_\ell),$$

and

$$R_r^p(a_0, \dots, a_\ell) = (a_0, \dots, a_{r-1}, a_r + a_{r+1} - e_p, a_{r+2}, \dots, a_\ell).$$

Thus D_r^p inserts a new zero IG row after the left boundary of the duplicated SE and deposits a $+1$ receiver-indexed metric contribution on that boundary, while R_r^p merges the two IG rows bounding the reduced SE by vector addition and deposits a -1 receiver-indexed metric contribution on the merged row.

The opposite directed face with receiver q is updated by the same rule in the mirrored coordinate. Thus an action at old SE index r on the (p, q)-oriented filament acts on receiver face p at index r, and on receiver face q at index $\ell - 1 - r$. The two facewise maps together define the full filament update.

Let

$$u = (u_0, \dots, u_{\ell-1}) \in \{-1, 0, +1\}^\ell$$

be a start-of-tick SE action schedule, where $u_r = +1$ denotes duplication, $u_r = -1$ denotes reduction, and $u_r = 0$ denotes no update at old SE index r. Define the live-index shift

$$\sigma_u(r) = \sum_{h<r} u_h.$$

The implementation-faithful serialisation of u is the map obtained by scanning $r = 0, \dots, \ell - 1$ and applying the nontrivial action u_r at the live index

$$r + \sigma_u(r).$$

Equivalently, duplications to the left shift later live indices by $+1$, and reductions to the left shift later live indices by -1.

Lemma 7.6 (Shifted adjacent interchange). *Let $r < s$. Let $X_r \in \{D_r^p, R_r^p\}$ be a duplication or reduction at old SE index r, and let $Y_s \in \{D_s^p, R_s^p\}$ be a duplication or reduction at old SE index s on the same directed face. Let*

$$w(D) = +1, \qquad w(R) = -1.$$

Then, whenever both sides are interpreted with live indices transported by the action already applied, one has

$$Y_{s+w(X)}^p \circ X_r^p = X_r^p \circ Y_s^p.$$

The same identity holds on the mirrored receiver face.

Proof. *It is enough to check the four possible pairs D, D, D, R, R, D, and R, R. All rows outside the local interval involved in the two operations are unchanged.*

For two duplications, inserting a zero row after r and then after the transported position $s+1$ gives the same sequence as inserting first after s and then after r. The deposited contributions are e_p on the same old boundary rows in both orders.

For a duplication followed by a reduction, the first insertion shifts the later reduction index by $+1$. The reduction therefore merges the same two old IG rows that it would have merged had it been performed first. The additional $+e_p$ and $-e_p$ contributions occur on the same resulting rows in both serialisations.

For a reduction followed by a duplication, the first reduction shifts the later duplication index by -1. Again the duplication is applied to the same transported old SE boundary as in the opposite order, and the merged row is the same vector sum.

For two reductions, performing the left reduction first shifts the later reduction index by -1. If the two reductions are adjacent, both orders merge the three involved IG rows into

$$a_r + a_{r+1} + a_{r+2} - 2e_p.$$

If they are non-adjacent, the same argument applies separately on the two disjoint local row blocks. Hence the resulting row sequence is identical in all cases. Since row merging is vector addition in $\mathbb{Z}^P$, no ordering ambiguity is introduced by combining metric records.

The mirrored receiver face is identical after applying the order-reversing coordinate $r \mapsto \ell - 1 - r$. Therefore the full two-face filament update satisfies the same shifted adjacent interchange relation. □

Proposition 7.7 (Implementation-faithful shifted serialisation)**.** *Fix a start-of-tick filament of length ℓ and a simultaneous SE action schedule*

$$u \in \{-1, 0, +1\}^{\ell}.$$

Any serial execution of the nonzero actions in u, provided each action is applied to the live image of its original old SE index under the insertions and deletions already performed, produces the same final pair of directed face arrays as the left-to-right shifted serialisation used in the implementation.

In particular, the Kernel B local-update phase defines a deterministic map from the start-of-tick action schedule u and the start-of-tick face arrays to the post-update face arrays.

Proof. *By Lemma 7.6, any adjacent interchange of two neighbouring actions in a serial execution leaves the final face arrays unchanged, provided the later live index is transported by the earlier insertion or deletion. Since any two serial orderings of the same finite action set are connected by a finite sequence of adjacent interchanges, all transported-index serialisations have the same result. The left-to-right serialisation is therefore a canonical representative of a well-defined simultaneous SE update map.* □

Remark 7.8. *This is not a fixed-row-index commutation statement. If one applies two local moves using their original row indices after an insertion or deletion has already occurred, the second move may act on the wrong live SE. The commutation property is therefore a shifted, implementation-faithful commutation property: old SE labels are preserved conceptually, while their live indices are transported by the shift function σ_u.*

Proposition 7.9 (Exact commutation across distinct filaments)**.** *Let m_1 and m_2 be Kernel B SE update maps supported on two distinct unordered PP pairs,*

$$\{p, q\} \neq \{r, s\}.$$

Then

$$m_2 \circ m_1 = m_1 \circ m_2.$$

Proof. The Kernel B latent state stores one independent pair of directed face arrays for each unordered PP pair. An SE duplication or reduction on the filament $\{p, q\}$ changes only the latent length $L_{pq} = L_{qp}$ and the two directed face arrays attached to that same unordered pair. It does not modify the face arrays or latent length of any other unordered pair, and it does not update any observer matrix during the local-update phase.

Therefore two SE updates supported on distinct unordered PP pairs act on disjoint state components. Their compositions are identical in either order. □

7.4 Dependency map for the frozen Kernel B theorem package

Reader guide. The frozen Kernel B theorem package has a simple logical structure, and it is useful to state that structure explicitly before the formal definitions begin. The results should be read in the following order:

$$\text{full frozen state} \implies \text{frozen queue operator} \implies \text{one-generation rebroadcast}$$

$$\implies \text{finite settling} \implies \text{exact terminal frozen correction}$$

$$\implies \text{conditional observer recovery and agreement.}$$

What each step does. The first step fixes the full frozen state $X(T) = (L(T), A(T), O(T))$ from which the frozen transport regime starts. The second step packages the post-freeze transport dynamics into a deterministic queue evolution. The third step isolates the key structural fact that rebroadcast occurs at most once along any transported metric contribution. The fourth step uses this to show that the residual frozen queue drains in finite time. The fifth step identifies, for observer-symmetric full frozen states, the eventual observer-side correction as a finite functional of $X(T)$. The sixth step then states exactly when that correction yields late-time observer recovery and observer agreement, namely under recovery compatibility of the full frozen state.

Why the dependency map matters. This order is not merely expository. Each later theorem depends on the earlier structure. In particular, the observer-recovery theorem is not an independent convergence statement: it rests on the exact terminal frozen correction theorem for observer-symmetric full frozen states, which itself rests on finite settling, which in turn rests on the one-generation rebroadcast property of the frozen queue dynamics.

Worked example. A finite three-particle worked example tracing this chain is given in Subsection 7.8. It is included to make the abstract queue algebra auditable in a concrete case.

7.5 Frozen Kernel B transport rule

Purpose. The strongest theorem package in Kernel B concerns the regime after all local duplication, reduction, and compact-bond reopening moves have been switched off. We therefore begin by fixing the exact frozen transport data and the paper-faithful frozen transport tick used in the theorem-level constant-point-particle specialization.

Definition 7.10 (Freeze time and full frozen state)**.** *A theorem-level Kernel B trajectory* ***freezes at time*** T *if, for all* $t \geq T$*, no further Kernel B local-update tick is applied and only the frozen transport tick remains active. The resulting state*

$$X(T) = (L(T), A(T), O(T))$$

is called the ***full frozen state****. Its frozen latent separation matrix is denoted*

$$L^* := L(T).$$

Its frozen queue state is the family of receiver-indexed face arrays

$$Q(T) := A(T) = \{A^{p \leftarrow q}(T)\}_{p \neq q}.$$

Definition 7.11 (Frozen queue space)**.** *Fix a frozen latent matrix* L^**. The corresponding* ***frozen queue space*** *is*

$$\mathcal{Q}_{L^*} := \prod_{p \neq q} \mathbb{Z}^{\{0,\ldots,L^*_{pq}\} \times P}.$$

A frozen queue state is an element $Q \in \mathcal{Q}_{L^*}$*. The zero element* $0 \in \mathcal{Q}_{L^*}$ *is the queue state in which every receiver-indexed face array is identically zero.*

Definition 7.12 (Observer-symmetric full frozen state)**.** *A full frozen state* $X(T)$ *is called* ***observer-symmetric*** *if, for every observer* $p \in P$*,*

$$O^p_{q,p}(T) = O^p_{p,q}(T) \qquad \text{for all } q \neq p, \qquad O^p_{p,p}(T) = 0.$$

This is the observer-local radial symmetry condition enforced by the paper-faithful frozen transport rule at each transport tick. No symmetry is imposed here on arbitrary off-diagonal pairs (i, j) *with* $i \neq p \neq j$*.*

Definition 7.13 (Paper-faithful frozen transport tick)**.** *Fix a full frozen state* $X(T)$ *and write* $L^* = L(T)$*. A single* ***paper-faithful frozen transport tick*** *from time* t *to time* $t+1$*, for* $t \geq T$*, is the deterministic update obtained by applying the following operations in order:*

(i) for every ordered pair $p \neq q$*, append one zero staging row at the far end of the array* $A^{p \leftarrow q}(t)$*;*

(ii) for every ordered pair $p \neq q$*, let*

$$a^{p \leftarrow q}(t) := A^{p \leftarrow q}_{0,\bullet}(t) \in \mathbb{Z}^P$$

be the row adjacent to receiver p*, and update the observer matrix by*

$$O^p_{\bullet,q}(t) \longmapsto O^p_{\bullet,q}(t) + a^{p \leftarrow q}(t);$$

(iii) define the absorbed radial scalar by

$$\rho_{p\leftarrow q}(t) := \big(a^{p\leftarrow q}(t)\big)_p;$$

(iv) enforce observer-local radial symmetry by setting

$$O^p_{q,p}(t+1) := O^p_{p,q}(t+1) \qquad \text{for all } q \neq p, \qquad O^p_{p,p}(t+1) := 0;$$

(v) for every triple of distinct point particles q, p, k, rebroadcast the absorbed radial scalar $\rho_{p\leftarrow q}(t)$ by adding it to the staging row at column q of the face array $A^{k\leftarrow p}$. Equivalently, the origin label q is preserved and the immediate back edge to q is excluded;

(vi) delete row 0 from every face array.

Here the row index 0 denotes the receiver-adjacent boundary row of $A^{p\leftarrow q}$; it is not a PP label. The receiver-owned radial component of this absorbed row is $A^{p\leftarrow q}_{0,p}$.

In the remainder of the paper, we may refer to this rule simply as the **frozen transport tick**.

Proposition 7.14 (Frozen state-class preservation)**.** *Let $X(t)$ be a theorem-level Kernel B state with latent matrix L^* fixed. Then one paper-faithful frozen transport tick produces another theorem-level Kernel B state with the same latent matrix L^*.*

Proof. *A frozen transport tick does not change the point-particle set or the latent separation matrix. On each receiver-indexed face array it appends one zero staging row and later deletes the receiver-adjacent row 0. Hence each face array returns to shape*

$$\mathbb{Z}^{\{0,\dots,L^*_{pq}\}\times P}.$$

The observer matrices are updated by integer vector additions and observer-local radial symmetry assignments, so they remain elements of $\mathbb{Z}^{P\times P}$. Therefore the resulting state is again a theorem-level Kernel B state with latent matrix L^*. □

Definition 7.15 (Frozen queue operator)**.** *Fix a freeze time T and latent matrix L^*. The* ***frozen queue operator***

$$F_T : \mathcal{Q}_{L^*} \longrightarrow \mathcal{Q}_{L^*}$$

is the map that sends a frozen queue state $Q(t)$ to the next frozen queue state $Q(t+1)$ by carrying out the queue part of Definition 7.13: append zero staging rows, compute receiver-adjacent arrivals and their radial scalars, stage the corresponding origin-label-preserving rebroadcasts, and delete row 0 from every receiver-indexed face array. The observer matrices are not part of F_T.

Definition 7.16 (Per-tick observer correction map)**.** *Fix a freeze time T, a frozen latent matrix L^*, and an observer $p \in P$. For a frozen queue state $Q \in \mathcal{Q}_{L^*}$, define the* ***queue-determined per-tick observer correction***

$$G^p_T(Q) \in \mathbb{Z}^{P\times P}$$

as the matrix increment produced at observer p by one paper-faithful frozen transport tick when the incoming observer matrix is observer-symmetric. Explicitly, if

$$a^{p\leftarrow q}(Q) := Q^{p\leftarrow q}_{0,\bullet}, \qquad \rho_{p\leftarrow q}(Q) := \big(a^{p\leftarrow q}(Q)\big)_p,$$

then

$$G^p_T(Q) = \sum_{q\neq p}\Big(a^{p\leftarrow q}(Q)\, e_q^\top + \rho_{p\leftarrow q}(Q)\, E_{q,p}\Big),$$

followed by the convention that the diagonal entry (p,p) *is set to* 0*. Here* $e_q \in \mathbb{Z}^P$ *is the* q*-th standard basis vector and* $E_{q,p}$ *is the matrix unit with a single* 1 *in entry* (q,p)*. The second term copies the observer's radial entry* (p,q) *into* (q,p)*; it does not symmetrize arbitrary off-diagonal pairs* (i,j) *with* $i \neq p \neq j$*.*

Remark 7.17 (Code-faithful content)**.** *Definition 7.13 formalizes the transport conventions used in the constant-point-particle kernel for the paper-facing runs: full-vector absorption into observer columns, extraction of the receiver-indexed radial scalar, observer-local radial symmetry only, origin-label-preserving rebroadcast, and exclusion of the immediate back edge.*

7.6 Frozen transport dynamics and finite settling

Definition 7.18 (Residual transport support)**.** *For a frozen theorem-level Kernel B state at time* $t \geq T$*, the* ***residual transport support*** *is*

$$\Sigma(t) := \Big\{(p,q,r,c)\ :\ p\neq q,\ 0\le r\le L^*_{pq},\ c\in P,\ A^{p\leftarrow q}_{r,c}(t)\neq 0\Big\}.$$

Lemma 7.19 (Deterministic evolution after freeze)**.** *Let a theorem-level Kernel B trajectory freeze at time* T*. Then the later state* $X(t+1)$ *is uniquely determined by* $X(t)$ *for every* $t \geq T$*.*

Proof. After time T*, no further Kernel B local-update tick occurs by Definition 7.10. The only remaining dynamics is the paper-faithful frozen transport tick of Definition 7.13, which is deterministic once* L^* *is fixed. Hence* $X(t+1)$ *is uniquely determined by* $X(t)$ *for every* $t \geq T$*.* □

Lemma 7.20 (Linearity of the frozen queue operator)**.** *Fix* T *and* L^**. Then*

$$F_T : \mathcal{Q}_{L^*} \to \mathcal{Q}_{L^*}$$

is linear on the abelian group of frozen queue states.

Proof. Once L^* *is fixed, every receiver-indexed face array has fixed shape. The operator* F_T *is built from appending zero rows, extracting row* 0*, fixed-coordinate additions into staging rows, and deleting row* 0*. Each operation is additive with respect to the face-array entries. Therefore their composition is additive, hence linear, on* $\mathcal{Q}_{L^*}$*.* □

Lemma 7.21 (One-generation rebroadcast)**.** *Let a nonzero frozen queue entry be absorbed at receiver* p *from side* q*. If that absorbed entry contributes to the radial scalar* $\rho_{p\leftarrow q}$*, then every rebroadcast entry generated from it is non-radial at its next receiver and therefore cannot generate any further rebroadcast.*

Proof. Suppose an absorbed entry contributes to $\rho_{p\leftarrow q}$*. By Definition 7.13, rebroadcast from* p *is sent only along faces* $A^{k\leftarrow p}$ *with* $k \notin \{p,q\}$*, and is written into column* q*, not column* k*. Hence the rebroadcast entry is non-radial at its next receiver* k*, because radiality at* k *would require carried label* k*. Therefore when that rebroadcast entry is later absorbed at* k*, it contributes nothing to the radial scalar* $\rho_{k\leftarrow p}$*, so no second rebroadcast generation is produced.* □

Lemma 7.22 (Observer-symmetry preservation after freeze)**.** *If the full frozen state $X(T)$ is observer-symmetric, then every later frozen state $X(t)$, $t \geq T$, is also observer-symmetric.*

Proof. Each paper-faithful frozen transport tick explicitly resets

$$O^p_{q,p}(t+1) := O^p_{p,q}(t+1) \qquad \text{for all } q \neq p, \qquad O^p_{p,p}(t+1) := 0.$$

Hence the observer-symmetry relations are preserved at every frozen tick. □

Lemma 7.23 (Additivity of frozen queue evolution and observer corrections)**.** *Fix a freeze time T and latent matrix L^*. Let $Q_1, Q_2 \in \mathcal{Q}_{L^*}$. Then*

$$F_T(Q_1 + Q_2) = F_T(Q_1) + F_T(Q_2).$$

Moreover, for each observer $p \in P$,

$$G^p_T(Q_1 + Q_2) = G^p_T(Q_1) + G^p_T(Q_2).$$

Hence frozen transport and the associated queue-determined per-tick observer corrections evolve by superposition of individual frozen queue entries.

Proof. The frozen queue operator F_T is linear by Lemma 7.20. The correction map G^p_T is built from row-0 extraction, matrix addition into observer columns, radial-scalar extraction, and observer-local radial copying, each of which is additive in the queue entries under the observer-symmetric convention used in Definition 7.16. Therefore both the queue evolution and the queue-determined observer correction evolve by superposition. □

Proposition 7.24 (Finite-horizon drain of paper-faithful frozen transport)**.** *Let a theorem-level Kernel B trajectory freeze at time T, and define*

$$M := \max_{p \neq q} L^*_{pq}.$$

Then

$$\Sigma(t) = \varnothing \qquad \text{for all } t \geq T + 2(M+1).$$

In particular, the paper-faithful frozen transport settles in finite time.

Proof. By Lemma 7.23, it is sufficient to track the evolution of a single nonzero frozen queue entry and then sum over all such entries.

Consider one nonzero entry of some face array $A^{p\leftarrow q}(T)$. Suppose it initially lies in row r, where

$$0 \leq r \leq L^*_{pq} \leq M.$$

Because row 0 is absorbed and then deleted at each frozen tick, an entry initially in row r is absorbed after at most $r + 1 \leq M + 1$ ticks.

If its carried label is not p, then it contributes nothing to the radial scalar $\rho_{p\leftarrow q}$, so it disappears permanently at that absorption event.

If its carried label is p, then it contributes to $\rho_{p\leftarrow q}$, and hence may generate rebroadcast entries on faces $A^{k\leftarrow p}$ with $k \notin \{p, q\}$. By Lemma 7.21, those rebroadcast entries are non-radial at their next *receivers and therefore cannot generate further rebroadcast. Each such rebroadcast entry is staged at the far end of a face of length at most M, so it is absorbed after at most a further $M + 1$ ticks and then disappears permanently.*

Thus every individual frozen queue entry disappears after at most $2(M + 1)$ ticks. Since the frozen queue state contains only finitely many nonzero entries and the queue evolution is additive, the entire residual transport support vanishes by time $T + 2(M + 1)$. □

Corollary 7.25 (Observer matrices stabilize after settling)**.** *Under the assumptions of Proposition 7.24, there exists a finite time $T_{\mathrm{s}} \le T + 2(M + 1)$ such that*

$$O^p(t + 1) = O^p(t) \qquad \text{for all } p \in P \text{ and all } t \ge T_{\mathrm{s}}.$$

Proof. *Once $\Sigma(t) = \varnothing$, no queue entry remains to be shifted, absorbed, or rebroadcast. Hence no later frozen transport tick can change any observer matrix.* □

7.7 Observer recovery after freeze

Definition 7.26 (Frozen defect)**.** *For each observer $p \in P$, the **frozen defect** at freeze time T is*

$$D^p(T) := L^* - O^p(T).$$

Proposition 7.27 (Exact terminal frozen correction for observer-symmetric frozen states)**.** *Assume the hypotheses of Proposition 7.24, and assume in addition that the full frozen state $X(T)$ is observer-symmetric. Then for each observer $p \in P$ there exists a unique matrix*

$$\Delta^p(X(T)) \in \mathbb{Z}^{P \times P}$$

such that

$$O^p(t) = O^p(T) + \Delta^p(X(T)) \qquad \text{for all sufficiently large } t,$$

and this terminal frozen correction is given exactly by the finite sum

$$\Delta^p(X(T)) = \sum_{m=0}^{H-1} G^p_T(F^m_T(Q(T)))$$

for every integer H satisfying

$$F^H_T(Q(T)) = 0.$$

Proof. *By Proposition 7.24, there exists $H \le 2(M + 1)$ such that*

$$F^H_T(Q(T)) = 0.$$

By Lemma 7.19, the frozen trajectory is uniquely determined by the full frozen state. By Lemma 7.22, observer-symmetry is preserved throughout the frozen regime, so the observer increment at each frozen tick is exactly the queue-determined matrix

$$G^p_T(F^m_T(Q(T))).$$

Summing these per-tick corrections from $m = 0$ until the queue is empty yields

$$\Delta^p(X(T)) = \sum_{m=0}^{H-1} G^p_T(F^m_T(Q(T))).$$

Once the queue has emptied, no further observer change is possible, so this $\Delta^p(X(T))$ is the unique eventual increment for all sufficiently large t.

If $H' \geq H$ is another integer satisfying $F_T^{H'}(Q(T)) = 0$, then all additional summands vanish because $F_T^m(Q(T)) = 0$ for every $m \geq H$. Hence the displayed finite sum is independent of the particular valid choice of H. □

Definition 7.28 (Recovery compatibility)**.** *Let $X(T)$ be a full frozen state satisfying the hypotheses of Proposition 7.27, so that the exact terminal frozen correction*

$$\Delta^p(X(T))$$

is defined for every observer $p \in P$. Then $X(T)$ is said to be ***recovery-compatible*** *if, for every observer $p \in P$,*

$$\Delta^p(X(T)) = D^p(T).$$

Equivalently,

$$O^p(T) + \Delta^p(X(T)) = L^* \qquad \text{for all } p \in P.$$

Remark 7.29 (Interpretive status)**.** *The main theorem below is a conditional recovery theorem. What has been strengthened is the status of the terminal correction: for observer-symmetric full frozen states, it is not introduced abstractly as an eventual increment, but as an exact finite functional of the full frozen state $X(T)$. The remaining open issue is whether recovery compatibility of the full frozen state can itself be derived from a deeper invariant of the pre-freeze local-update bookkeeping.*

Theorem 7.30 (Conditional exact recovery after freeze)**.** *Assume the hypotheses of Proposition 7.27. Then the following are equivalent:*

(i) there exists a finite time $T_s \geq T$ such that

$$O^p(t) = L^* \qquad \text{for all } p \in P \text{ and all } t \geq T_s;$$

(ii) the full frozen state $X(T)$ is recovery-compatible.

Proof. By Proposition 7.27, each observer matrix has the eventual value

$$O^p(T) + \Delta^p(X(T)).$$

If $X(T)$ is recovery-compatible, then

$$O^p(T) + \Delta^p(X(T)) = L^*$$

for every p, so every observer recovers the frozen latent matrix after finite settling.

Conversely, if late-time recovery holds, then the eventual observer matrix equals L^*. Hence

$$O^p(T) + \Delta^p(X(T)) = L^*,$$

which rearranges to

$$\Delta^p(X(T)) = L^* - O^p(T) = D^p(T).$$

Thus the full frozen state is recovery-compatible. □

Corollary 7.31 (Observer agreement after recovery). *Under the assumptions of Theorem 7.30, if the full frozen state $X(T)$ is recovery-compatible, then*

$$O^p(t) = O^q(t) = L^* \qquad \text{for all } p, q \in P \text{ and all sufficiently large } t.$$

Proof. By Theorem 7.30, every observer matrix equals L^* after finite settling. Hence all observers agree with one another and with the frozen latent matrix. □

Corollary 7.32 (Late-time dependence only on the full frozen state). *Under the assumptions of Proposition 7.27, the terminal corrections*

$$\Delta^p(X(T))$$

and hence the eventual observer matrices depend only on the full frozen state $X(T)$.

Proof. By Proposition 7.27, the terminal correction at each observer is an explicit finite sum built from the frozen queue state $Q(T)$, the fixed latent matrix L^*, and the deterministic operator F_T. These are all components of the full frozen state $X(T)$. Hence the eventual observer matrices are determined by $X(T)$ alone. □

7.8 A worked frozen-transport example

Purpose. The frozen Kernel B theorem package is algebraically simple but can look abstract on first reading. The present example shows, in a finite three-particle case, how the frozen queue evolves, how the exact terminal frozen correction is accumulated, and how observer recovery can occur after a finite number of frozen transport ticks.

Example 7.33 (Three-particle frozen recovery). *Consider three point particles $1, 2, 3$ with frozen latent matrix*

$$L^* = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

Assume the frozen observer matrices are

$$O^1(T) = O^2(T) = O^3(T) = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Thus each observer is initially missing the pair $(1,3)$ and its opposite orientation $(3,1)$.

Assume further that the frozen queue state has exactly two nonzero face arrays,

$$A^{1\leftarrow 3}(T) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad A^{3\leftarrow 1}(T) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

with all other face arrays equal to zero. Since $L^*_{13} = 1$, each corresponding receiver-indexed face array has exactly two rows: the receiver-adjacent row and the far-end row.

***Tick** $T \to T+1$. At observer 1, the row $e_1 = (1,0,0)$ is absorbed from $A^{1\leftarrow 3}$ into column 3. Hence*

$$O^1_{\bullet,3}(T+1) = O^1_{\bullet,3}(T) + e_1.$$

The absorbed radial scalar is

$$\rho_{1\leftarrow 3}(T) = 1,$$

because the absorbed vector has value 1 *in the receiver-indexed radial slot. The frozen transport rule therefore rebroadcasts this scalar toward* 2*, preserving the origin label* 3*. Thus one new staged face array is*

$$A^{2\leftarrow 1}(T+1) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Observer-local radial symmetry at 1 *then enforces*

$$O^1_{3,1}(T+1) = O^1_{1,3}(T+1) = 1,$$

so

$$O^1(T+1) = L^*.$$

At observer 3*, the row* $e_3 = (0, 0, 1)$ *is absorbed from* $A^{3\leftarrow 1}$ *into column* 1*. Hence*

$$O^3_{\bullet,1}(T+1) = O^3_{\bullet,1}(T) + e_3.$$

The absorbed radial scalar is

$$\rho_{3\leftarrow 1}(T) = 1,$$

so the frozen transport rule rebroadcasts this scalar toward 2*, preserving the origin label* 1*. Thus the second new staged face array is*

$$A^{2\leftarrow 3}(T+1) = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Observer-local radial symmetry at 3 *then enforces*

$$O^3_{1,3}(T+1) = O^3_{3,1}(T+1) = 1,$$

so

$$O^3(T+1) = L^*.$$

Tick $T+1 \to T+2$***.*** *At time* $T+1$*, neither staged rebroadcast entry is yet adjacent to receiver* 2*. Hence no observer update occurs at this tick. After one frozen shift, the two relevant arrays become*

$$A^{2\leftarrow 1}(T+2) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad A^{2\leftarrow 3}(T+2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Tick $T+2 \to T+3$***.*** *At observer* 2*, the arriving row from* $A^{2\leftarrow 1}$ *is* $e_3 = (0, 0, 1)$*, so*

$$O^2_{\bullet,1}(T+3) = O^2_{\bullet,1}(T+2) + e_3.$$

This contributes the entry

$$O^2_{3,1}(T+3) = 1.$$

Also at observer 2*, the arriving row from* $A^{2\leftarrow 3}$ *is* $e_1 = (1, 0, 0)$*, so*

$$O^2_{\bullet,3}(T+3) = O^2_{\bullet,3}(T+2) + e_1.$$

This contributes the entry

$$O^2_{1,3}(T+3) = 1.$$

Therefore

$$O^2(T+3) = L^*.$$

Both arrivals at observer 2 are non-radial, because their carried labels are 3 and 1, not 2. Hence

$$\rho_{2\leftarrow 1}(T+2) = 0, \qquad \rho_{2\leftarrow 3}(T+2) = 0,$$

so no further rebroadcast is generated. The observer-local radial symmetry step at 2 affects only entries involving observer index 2; it does not create the pair $(1,3)$ from a single non-radial arrival. In the present example, observer 2 recovers because it receives both opposite non-radial orientations from the two opposite radial histories.

Terminal frozen correction. *In this example the exact terminal frozen corrections are*

$$\Delta^1(X(T)) = \Delta^2(X(T)) = \Delta^3(X(T)) = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

These agree exactly with the frozen defects

$$D^1(T) = L^* - O^1(T), \qquad D^2(T) = L^* - O^2(T), \qquad D^3(T) = L^* - O^3(T).$$

Hence this full frozen state is recovery-compatible on the nose, and the theorem package yields finite-time observer recovery:

$$O^1(T+1) = L^*, \qquad O^2(T+3) = L^*, \qquad O^3(T+1) = L^*.$$

This finite example displays the whole dependency chain in concrete form:

(i) a full frozen state is fixed;

(ii) the frozen queue evolves deterministically;

(iii) each direct radial arrival generates exactly one rebroadcast;

(iv) each rebroadcast is non-radial at its next receiver and therefore terminates;

(v) the terminal frozen correction is finite and explicitly computable;

(vi) recovery compatibility then yields finite-time observer recovery.

7.9 Bridge from Kernel A to a structurally quiescent constant-point-particle regime

Purpose of the bridge. The active split-enabled regime of Kernel A and the constant-point-particle metric regime of Kernel B use different state descriptions. The purpose of this subsection is therefore not to prove that active Kernel A dynamics reduces to Kernel B dynamics. It is to isolate a late structurally quiescent regime of Kernel A and state the explicit hypotheses under which that regime can be identified with a Kernel B frozen-state description.

Definition 7.34 (Structural quiescence time)**.** *A Kernel A trajectory is said to reach **structural quiescence** at time T_q if, for all $t \geq T_\mathrm{q}$:*

(i) no further point-particle split occurs;

(ii) no active structural marker remains;

(iii) no later local move reactivates the structural-marker sector.

Definition 7.35 (Metric-compatible quiescent regime)**.** *A structurally quiescent Kernel A trajectory is said to be in a **metric-compatible quiescent regime** from time T_q onward if, for all $t \geq T_\mathrm{q}$:*

(i) the point-particle set is constant;

(ii) no active structural marker is present;

(iii) the only remaining local update moves are space-element duplication, space-element reduction, and, if the scheduler family permits it, localized compact-bond reopening moves that preserve the point-particle set and only reopen an already fused boundary by inserting a nonzero filament length;

(iv) no remaining admissible move can create a new structural marker or change the point-particle set.

Proposition 7.36 (Eventual entry into a metric-compatible quiescent regime)**.** *Suppose a Kernel A scheduler suppresses splits after a finite split horizon. Suppose every structural marker created before that horizon has finite propagation lifetime. Suppose further that, after some finite time, every move capable of creating a new structural marker or changing the point-particle set is suppressed, except possibly localized compact-bond reopening moves that preserve the point-particle set. Then the trajectory enters a metric-compatible quiescent regime after some finite time T_q.*

Proof. Once the split horizon has passed, no later point-particle split occurs. By hypothesis, every structural marker created before that horizon is absorbed or extinguished after finitely many propagation steps. Hence, after some finite time, no active structural marker remains.

The additional suppression hypothesis prevents any later move from creating a new structural marker or changing the point-particle set. The only optional reopening moves still permitted are localized compact-bond reopenings that preserve the point-particle set and modify only filament-length data on already existing particle pairs.

Therefore, after some finite time T_q, the trajectory has a constant point-particle set, no active structural markers, no structural reactivation, and only the metric-side local moves listed in Definition 7.35. This is precisely entry into a metric-compatible quiescent regime. □

Definition 7.37 (Kernel A-to-B identification)**.** *A metric-compatible quiescent Kernel A regime is said to admit a **Kernel B identification** if there exists a map*

$$\Pi$$

from the quiescent Kernel A states to theorem-level Kernel B states such that:

(i) Π *preserves the fixed point-particle set;*

(ii) Π *sends each latent pairwise filament length in the quiescent Kernel A state to the corresponding entry of the Kernel B latent separation matrix;*

(iii) Π *sends the remaining metric-side local update moves in the quiescent Kernel A regime to the corresponding Kernel B local-update moves;*

(iv) once the identified trajectory freezes, its post-freeze dynamics is exactly the paper-faithful frozen transport rule of Definition 7.13;

(v) the identified full frozen state is observer-symmetric in the sense of Definition 7.12.

Theorem 7.38 (Conditional Kernel A-to-B bridge theorem)**.** *Let a Kernel A trajectory enter a metric-compatible quiescent regime at time* T_{q}*, and suppose this regime admits a Kernel B identification* Π*. Suppose further that the identified Kernel B trajectory freezes at time* $T \geq T_{\mathrm{q}}$*. Then:*

(i) the identified post-freeze trajectory evolves deterministically and settles in finite time;

(ii) for each observer p*, the exact terminal frozen correction* $\Delta^p(X(T))$ *is well defined and is determined by the identified full frozen state* $X(T)$*;*

(iii) late-time observer recovery and observer agreement hold exactly under the recovery-compatibility condition for the identified full frozen state.

Proof. By Definition 7.37, the identified post-freeze trajectory is governed exactly by the paper-faithful frozen Kernel B transport rule, and the identified full frozen state is observer-symmetric. Lemma 7.19 gives deterministic post-freeze evolution. Proposition 7.24 gives finite settling, proving (i)*. Proposition 7.27 gives the exact terminal frozen correction as a finite functional of the identified full frozen state, proving* (ii)*. Finally, Theorem 7.30 and Corollary 7.31 give the recovery and agreement statements precisely under recovery compatibility of that full frozen state, proving* (iii)*.* □

Remark 7.39 (What the bridge theorem does and does not say)**.** *The bridge theorem does not claim that the active structural-growth phase of Kernel A is a Kernel B execution. It applies only after structural quiescence has been reached, and only after an explicit Kernel B identification of the remaining metric-side regime has been supplied. Quiescence diagnostics in simulations may indicate that the hypotheses are plausible in a trajectory, but they do not by themselves establish the identification required by the theorem.*

8 Corollaries for computational observables

This section explains how the theorem package of Sec. 7 constrains the main classes of computational observables used in companion numerical work. Its purpose is not to turn those observables into new theorems, but to distinguish three cases clearly: observables that directly track proved late-time behaviour, observables whose interpretation depends on the Kernel A-to-B bridge step, and observables that remain empirical in the present paper.

8.1 Observable-status classes

Definition 8.1 (Observable-status classes)**.** *Every computational observable discussed in relation to the present theorem package belongs to one of the following classes:*

(i) ***Theorem-target observable:*** *an observable whose late-time value or qualitative behaviour is a direct consequence of a proved theorem under the stated hypotheses;*

(ii) ***Bridge-sensitive observable:*** *an observable whose interpretation depends on structural quiescence, a bridge hypothesis, or an identification of a quiescent regime with a Kernel B state;*

(iii) ***Empirical-only observable:*** *an observable used for numerical exploration or phenomenological description that is not claimed here as a theorem consequence.*

An observable may be important computationally without being theorem-covered, and a theorem-target observable may still be noisy or implementation-sensitive in practice. The point of the present section is simply to keep those questions separate.

8.2 Kernel B theorem-target observables

Let a Kernel B trajectory freeze at time T, and let

$$L^* = L(T)$$

be its frozen latent metric state. In this regime the theorem-target observables are those that measure:

(i) stabilization after freeze;

(ii) one-generation rebroadcast and finite settling of residual metric records;

(iii) the exact terminal frozen correction accumulated during post-freeze drain for observer-symmetric full frozen states;

(iv) conditional late-time observer recovery relative to L^*, under recovery compatibility of the full frozen state;

(v) conditional late-time agreement between observer matrices, again under recovery compatibility of the full frozen state.

Any observable that measures whether the residual frozen queue has emptied, or equivalently whether the observer matrices have ceased to change, is a theorem-target observable for Proposition 7.24 and Corollary 7.25. In particular, diagnostics of the form

$$\|O^p(t+1) - O^p(t)\|$$

for any reasonable matrix norm must eventually vanish under the hypotheses of the theorem package.

Similarly, observables measuring the cumulative post-freeze correction

$$\|O^p(t) - O^p(T)\|$$

or comparing that correction with queue-derived or absorbed-mass bookkeeping are natural numerical probes of Proposition 7.27. Such diagnostics are not proofs, but they do track the exact object appearing in the theorem statement.

The observer-recovery theorem is conditional. Accordingly, observables of the form

$$\|O^p(t) - L^*\|$$

or

$$\|O^p(t) - O^q(t)\|$$

are theorem-target observables only under the hypotheses of Theorem 7.30 and Corollary 7.31. Under those hypotheses they must vanish after finite settling.

A central point of the theorem package is that late-time behaviour depends on the full frozen state

$$X(T) = (L(T), A(T), O(T)),$$

not on the frozen latent matrix L^* alone. Consequently, post-freeze agreement diagnostics are informative only relative to the queue and observer content from which deterministic drain begins.

8.3 Bridge-sensitive observables for Kernel A

The bridge theorem of Sec. 7 concerns entry into a structurally quiescent constant-point-particle latent regime. Observables used to diagnose:

(i) cessation of point-particle splits,

(ii) extinction of active structural carriers,

(iii) stabilization of the point-particle count,

are therefore best classified as bridge-sensitive.

Typical examples include:

(i) the number of active structural carriers as a function of time;

(ii) the time after which the point-particle count remains constant;

(iii) the disappearance of split-generated structural activity in the latent state.

These observables are useful because they indicate when a Kernel A trajectory may have entered the class of latent regimes for which Proposition 7.36 and Theorem 7.38 become relevant. They do not, by themselves, imply Kernel B observer recovery, because the bridge theorem still requires an explicit identification of the quiescent regime with a Kernel B state.

8.4 Implementation-level agreement diagnostics

The companion computational paper uses explicit matrix-based and queue-sensitive diagnostics to study the relation between observer states and latent structure in the constant-point-particle sector. In the language of the present paper, these diagnostics should be read as numerical probes of three narrower questions:

(i) has the frozen queue effectively settled;

(ii) does the observed post-freeze correction behave like the exact terminal frozen correction;

(iii) does the full frozen state appear to be recovery-compatible in the implemented run.

Their role is therefore evidential and interpretive, not substitutive. They may function as theorem-target observables when the hypotheses of the frozen Kernel B theorem package are intended to hold, and they may function as empirical probes of whether recovery compatibility is plausibly realized in an implementation. They do not replace proof.

Two diagnostic classes are especially relevant. First, settle-sensitive diagnostics measure whether the post-freeze transport behaves like a draining queue system; these are the most direct numerical probes of Proposition 7.24 and Proposition 7.27. Second, recovery-sensitive diagnostics compare observer matrices to the frozen latent state or to one another after settling; these are informative because Theorem 7.30 and Corollary 7.31 state exactly what such diagnostics must do under recovery compatibility.

8.5 Empirical-only observables

Any observable that characterizes the detailed active-phase dynamics of Kernel A or Kernel B before freeze is empirical-only unless explicitly covered by a theorem. This includes, for example, branching patterns, transient filament-growth histories, and trajectory-specific timing effects.

Likewise, observables used to describe gravity-like attraction, Yukawa-like screening, condensation, CPPN-rich phases, or broader emergent phenomenology remain empirical-only in the present paper. These may be important numerically and conceptually, but they are not consequences of the theorem package proved here.

Any observable defined through an external embedding or visualization pipeline is also empirical-only unless separately formalized. The present theorem package is stated on the latent state and on observer-derived metric summaries in the restricted kernels; it does not prove claims about embedding geometry, visual clustering, or presentation-level layout.

8.6 Summary of the corollarial relation

The theorem-covered numerical narratives of the paper are narrow but clear. In the constant-point-particle sector, deterministic evolution after freeze, one-generation rebroadcast, finite settling, stabilization after freeze, exact terminal frozen correction for observer-symmetric full frozen states, and conditional observer recovery and observer agreement under recovery compatibility of the full frozen state all have direct observable forms once the hypotheses of the theorem package are satisfied.

In the split-enabled structural sector, the theorem-covered numerical narratives concern only the approach to structural quiescence and the conditional applicability of the Kernel B theorem package after an identification step. Observables in this sector are therefore bridge-sensitive rather than directly theorem-targeted.

Much of the wider phenomenology of the SERD programme remains empirical from the standpoint of the present paper. The value of the current theorem package is not that it closes the whole interpretive gap, but that it fixes the meaning of a precise subset of late-time numerical diagnostics by explicit theorem.

9 Implementation correspondence

The purpose of this section is to record how the formal objects defined in the previous sections correspond to the reference computational implementations used in companion numerical work. This section is not part of the theorem package. The theorem-bearing claims of the paper are governed by the mathematical definitions, hypotheses, and proofs stated in the main text. The reference implementations are treated as executable realizations of restricted kernels and as correspondence evidence for the companion computational work; they do not define or enlarge the theorem envelope.

9.1 Formal priority and implementation correspondence

Formal priority. The parent SERD framework, the distinction between microscopic law and scheduler families, the restricted kernels, and the theorem package are stated mathematically before their computational counterparts are discussed. In this paper, formal definitions and proofs are primary for theorem-bearing claims. The role of code is to instantiate selected restricted kernels, expose diagnostics, and support correspondence with companion computational work.

Why correspondence is still necessary. Although implementation details do not replace mathematical definitions, correspondence matters for two reasons:

(i) the companion computational paper studies explicit trajectories of restricted kernels rather than of the unrestricted parent framework;

(ii) several numerical diagnostics are meaningful only if the reader can see how the formal objects are represented computationally.

This section therefore identifies the computational counterparts of the formal kernels without allowing implementation choices, plotting routines, or scheduler controls to become additional theorem assumptions.

Scope of the correspondence. The correspondence recorded here is limited to three questions:

(i) which formal kernel is instantiated by which reference implementation;

(ii) which implementation controls belong to the scheduler family rather than to the microscopic law;

(iii) which implementation diagnostics correspond to theorem-target observables, bridge-sensitive observables, or empirical-only diagnostics.

9.2 Kernel-to-code map

Formal object	Formal role	Reference implementation	Status in this paper
Parent SERD framework	Typed latent state, admissible move family, deterministic propagation architecture	Not identified with any single code file	Formal reference layer
Kernel A	Split-enabled structural restriction	Core logic: `serd_causal_invariance_splits.py` Experiment driver: `02a_causal_invariance_splits.py` Configuration: `02a_causal_invariance_splits.yaml`	Restricted executable kernel
Kernel B	Constant-point-particle metric restriction with freeze-and-settle transport	Core logic: `serd_causal_invariance_constpp.py` Experiment driver: `02b_causal_invariance_constpp.py` Configuration: `02b_causal_invariance_constpp.yaml`	Primary theorem-bearing implementation
Scheduler family	Selection rule for admissible update families, suppression horizons, freeze controls	Driver- and configuration-level execution controls	Execution structure only
Observer-side metric summaries	Observer-indexed matrices used in late-time recovery and agreement diagnostics	Observer matrices and matrix-based agreement diagnostics in the constant-point-particle implementation	Theorem-target observable layer

Table 1: *Formal-to-implementation correspondence used in the present paper. The implementations instantiate restricted kernels and expose diagnostics relevant to the theorem package, but they do not define the mathematics.*

The table distinguishes kernel logic from drivers and configuration files for the same reason the paper distinguishes microscopic law from scheduler family. The kernel modules carry the restricted-kernel state representation, local move semantics, and propagation rules. The experiment drivers and YAML files add scheduler-level controls, aggregation routines, output structure, and figure-generation settings.

9.3 Scheduler parameters versus microscopic law

Implementation-level probabilities, stopping times, and execution controls are interpreted as scheduler parameters unless explicitly stated otherwise. They do not modify the parent SERD move family or the local semantics of admissible moves; they modify only which admissible trajectories are sampled or when a trajectory is forced into a restricted regime such as split suppression or freeze.

Within the present computational setting, the following are therefore treated as scheduler-level controls:

(i) split proposal probabilities and split horizons in the split-enabled structural kernel;

(ii) duplication and reduction proposal probabilities or selection weights;

(iii) finite freeze times in the constant-point-particle metric kernel;

(iv) optional suppression or activation of localized fused-boundary reopening channels in executable variants.

These controls affect which trajectory is executed, but not what the local moves mean.

9.4 Code-level realization of the restricted kernels

Kernel A realization. In the split-enabled structural implementation, the latent state is represented by explicit dictionaries for point particles, space elements, information gaps, and structural carriers. The update phase samples split, duplication, and reduction proposals and applies them to the typed latent state. The propagation phase then advances the active structural carriers deterministically. Optional localized boundary-opening moves may also be present in executable variants. This is the computational counterpart of the formal split-enabled structural kernel, but it is richer than the theorem-bearing core whenever such optional extension channels are active.

Kernel B realization. In the constant-point-particle implementation, the latent state is represented by a fixed point-particle set, integer latent filament lengths on each pairwise connection, receiver-indexed face arrays carrying transported metric records, and observer-indexed matrices. The update phase acts on filament refinement and coarsening. After freeze, the remaining evolution is carried entirely by the transport phase. This is the computational counterpart of the formal Kernel B specialization studied in Sec. 7.

The correspondence recorded here is intentionally structural. It identifies which code architecture instantiates which restricted kernel. It does not claim that every software detail—plotting routines, output formatting, visualization settings, aggregation helpers, or embedding controls—has a formal analogue in the mathematics.

9.5 Theorem-target and bridge-sensitive diagnostics

Kernel B diagnostics. For Kernel B, the diagnostics of most direct relevance are:

(i) agreement between observer matrices and the frozen latent metric state;

(ii) pairwise agreement between observer matrices;

(iii) freeze markers identifying entry into the settle regime;

(iv) residual transport diagnostics tracking the remaining in-flight metric records after freeze.

These diagnostics are not themselves theorems. Their role is to provide numerical summaries of the conclusions of the Kernel B theorem package, especially deterministic frozen evolution, one-generation rebroadcast, finite settling, exact terminal frozen correction for observer-symmetric full frozen states, and conditional observer recovery and observer agreement under recovery compatibility of the full frozen state. In particular, agreement diagnostics must be interpreted relative to the full frozen state and the recovery-compatibility condition of Sec. 7; they are not numerical substitutes for the theorem hypotheses.

Kernel A diagnostics. For Kernel A, the diagnostics most relevant to the present paper are those that witness the hypotheses of the bridge theorem:

(i) suppression of further split activity;

(ii) disappearance of residual structural carriers;

(iii) stabilization of the point-particle count;

(iv) onset of a structurally quiescent constant-point-particle latent regime.

These diagnostics are bridge-sensitive rather than theorem-bearing. They do not prove the bridge theorem, but they indicate when a computational trajectory appears to satisfy the hypotheses under which the bridge theorem becomes relevant. In particular, they do not by themselves establish the identification step required before the Kernel B theorem package can be invoked.

9.6 Limits of correspondence

The existence of a working implementation does not by itself establish any theorem stated in this paper. The implementations are relevant because they instantiate the restricted kernels and expose theorem-relevant diagnostics, not because they replace formal proof.

Likewise, not every numerical pattern observed in the companion computational work lies inside the present theorem envelope. The formal role of implementation correspondence is only to identify which parts of the computational narrative are theorem-target observables, which are bridge-sensitive observables, and which remain empirical.

Finally, the appearance of a numerical effect in a restricted executable kernel does not imply that the same effect holds across the full parent SERD move family. The present paper proves results only for explicit restricted kernels under explicit hypotheses.

Remark 9.1 (Function of this section)**.** *The parent* SERD *framework supplies the ontology and move family; the restricted kernels supply theorem-bearing sectors; the theorem package establishes rigorous results within those sectors; and the reference implementations provide executable realizations together with diagnostics that can be interpreted relative to those results.*

10 Limitations and open problems

The present paper establishes a controlled theorem package for two restricted kernels of the SERD framework and clarifies how those kernels relate to the broader programme and to the reference implementations. That is already a substantial step, but the scope of the current results remains deliberately limited. The broader SERD programme still contains conceptual, formal, and computational problems that remain open.

10.1 The theorem envelope is intentionally restricted

Restricted-kernel scope. The results of this paper are proved only for explicit restricted kernels. They do not amount to a theorem-level treatment of the full parent SERD move family. In particular, the paper does not prove general results for unrestricted combinations of split-, merge-, fusion-, and fission-enabled dynamics, nor does it establish a single global theorem covering every admissible sector of the broader framework.

Why this matters. The parent SERD framework is intentionally broader than the current theorem package. That breadth is a strength at the programme level, but it also means that tractable kernel restrictions are needed if rigorous proofs are to be obtained at the present stage. The restricted-kernel strategy adopted here should therefore be understood as a controlled theorem-bearing entry point into the wider model family, not as a final closure of the formal programme.

10.2 Limits of the Kernel B observer-recovery theorem

What is now exact and what remains conditional. The strongest result in the constant-point-particle sector is now more structured than a bare recovery characterization. The paper proves deterministic post-freeze transport, one-generation rebroadcast, finite settling of the residual frozen queue, and, for observer-symmetric frozen states, an exact terminal frozen correction theorem expressing the eventual observer-side correction as a finite functional of the full frozen state. What remains conditional is the final step from that exact correction to exact late-time recovery of the frozen latent metric.

Why recovery is still conditional. The observer-recovery theorem still depends on recovery compatibility of the full frozen state. This is no longer because the post-freeze evolution itself is unclear: the frozen transport algebra has now been formalized exactly. Rather, it is because the paper does not yet derive, from a deeper pre-freeze invariant, that the exact terminal frozen correction must equal the frozen observer defect.

What remains open. The main open problem in this sector is therefore sharper and more focused than before. One would like to replace the present recovery-compatibility condition by sufficient conditions stated directly in terms of the local bookkeeping and transport algebra that generate the frozen queue. More concretely, one would like to derive recovery compatibility, or useful sufficient criteria for it, from a finer analysis of:

(i) the generation of metric corrections by duplication, reduction, and compact-bond reopening;

(ii) the transport and one-generation rebroadcast algebra on frozen latent states;

(iii) the relation between the exact terminal frozen correction and the frozen observer defect.

Why this is the right next step. A proof of this kind would not replace the exact terminal frozen correction theorem; it would complete the recovery side of it. The present paper already identifies, for observer-symmetric full frozen states, the exact post-freeze correction determined by the full frozen state. The missing step is to show that the update-generated queued correction is precisely the outstanding observer defect at freeze. That would convert the current conditional observer-recovery characterization into a stronger exact recovery theorem.

Initialization dependence. A related issue is the role of observer initialization. The theorem layer has been formulated to match the constant-point-particle implementation used in the companion computational work, where observer matrices are initialized by the latent metric at the start of the run. This makes late-time recovery the natural theorem-level target. A more general theory of arbitrary observer initialization remains open.

10.3 Limits of the Kernel A bridge theorem

Structural quiescence is not full active-phase control. The bridge theorem proved in this paper concerns the onset of a structurally quiescent constant-point-particle latent regime. It does *not* provide a complete theorem-level treatment of the active split phase itself. In particular, the paper does not prove a general causal analysis of split-enabled structural growth or a full classification of all possible structural-carrier interactions during the active phase.

The identification step remains external. Even after structural quiescence has been reached, the bridge theorem still requires an explicit identification of the quiescent latent regime with a Kernel B state. That identification is stated carefully in the theorem section, but it is not yet derived here as a universal consequence of the parent SERD framework. This leaves an important open problem: to characterize, in a kernel-independent way, exactly when and how a structurally quiescent latent regime can be represented by a constant-point-particle metric kernel.

Optional extension channels. The present bridge result also excludes localized fusion/fission-style extension channels unless they are explicitly suppressed or explicitly built into the hypotheses. This is appropriate for the current restricted-kernel analysis, but it leaves open the broader question of how such executable extension sectors should be incorporated into a larger theorem package.

10.4 Scheduler dependence outside the frozen regime

What the paper does not prove about scheduler dependence. The present results clarify the distinction between microscopic law and scheduler family, and they prove deterministic evolution after freeze in the constant-point-particle metric sector. They do not, however, eliminate scheduler dependence in general. Many trajectory-level questions remain sensitive to scheduler choice, including:

(i) the detailed history of active structural growth in Kernel A;

(ii) the timing of entry into structural quiescence;

(iii) the precise pre-freeze latent history of Kernel B;

(iv) the quantitative shape of empirical observables in the companion computational paper.

Open direction. A deeper formal treatment of scheduler dependence would require more than the present paper provides. In particular, it would require either:

(i) stronger schedule-invariance theorems for suitable restricted move algebras; or

(ii) a classification of scheduler families that lead to equivalent late-time behaviour.

At present, the paper proves deterministic settle-regime behaviour under explicit conditions, not general scheduler-independence for the full active dynamics.

10.5 Relation to computational and phenomenological claims

Not every numerical pattern is theorem-covered. The companion computational paper studies observables that are wider in scope than the present theorem package. This includes diagnostics connected to attraction-like behaviour, screening-like behaviour, condensation, CPPN-rich phases, and broader phenomenology of the evolving latent network. These may be important for the development of the wider programme, but they remain empirical from the standpoint of the present GFJ paper unless explicitly tied to a proved theorem.

No final physical identification is claimed. For the same reason, the paper does not claim a final physical identification of the restricted kernels with established continuum theories. Any comparison to gravity-like, Yukawa-like, field-theoretic, or condensed-phase behaviour should therefore be read as part of the broader research programme rather than as a theorem-level conclusion of the present article.

Embedding and visualization remain downstream. Embedding-based plots, animations, and visual latent-state summaries may be useful for interpretation and communication, but they do not play a formal role in the theorem package. The mathematical results proved here are stated on the latent state and on observer-derived metric summaries of the restricted kernels, not on visualization-dependent representations.

10.6 Open formal directions

The most important open formal directions suggested by the present paper are the following:

(i) derive recovery compatibility, or workable sufficient conditions for it, from more primitive invariants of the frozen transport algebra;

(ii) strengthen the bridge from structurally quiescent Kernel A regimes to Kernel B by deriving the identification step from intrinsic latent conditions rather than postulating it separately;

(iii) extend the restricted-kernel theorem package to include additional executable extension channels, especially localized fusion/fission sectors;

(iv) develop stronger results on schedule dependence or schedule robustness beyond the settle regime;

(v) clarify how the present restricted-kernel results sit inside a larger theorem-level treatment of the parent SERD framework.

10.7 What the present paper nevertheless establishes

These limitations do not weaken the central contribution of the paper. What has been established here is a disciplined formal middle layer for the current SERD programme:

(i) a parent parameter-free framework with a clear law/scheduler distinction;

(ii) two explicit restricted kernels derived from that framework;

(iii) a rigorous theorem package for deterministic frozen evolution, one-generation rebroadcast, finite settling, exact terminal frozen correction for observer-symmetric full frozen states, and conditional observer recovery and observer agreement under recovery compatibility in the constant-point-particle sector;

(iv) a conditional bridge result connecting the split-enabled structural sector to that theorem-bearing metric sector under explicit hypotheses;

(v) a clarified relation between theorem-target observables, bridge-sensitive observables, and empirical-only diagnostics.

The open problems listed above should therefore be read not as failures of the present paper, but as the next formal steps required if the wider SERD programme is to be developed into a more comprehensive theory.

11 Conclusion

This paper has developed the formal middle layer of the current SERD programme. Its central contribution has been to place the programme on a clearer mathematical footing by separating the parent SERD framework from scheduler families, defining two implementation-faithful restricted kernels derived from that framework, and proving a bounded but nontrivial theorem package within those kernels.

Main mathematical result. The strongest rigorous results are obtained for the frozen regime of Kernel B. In that setting, the paper establishes deterministic evolution after freeze, one-generation rebroadcast, finite settling of the paper-faithful receiver-indexed transport rule, and, for observer-symmetric frozen states, an exact terminal frozen correction theorem expressing the eventual observer-side correction as a finite functional of the full frozen state. On that basis, it gives a conditional characterization of late-time observer recovery and observer agreement in terms of recovery compatibility of the full frozen state. This is the clearest theorem-bearing core of the paper.

Kernel A and the bridge theorem. For Kernel A, the paper proves a narrower result. Under finite split stopping, finite structural-marker lifetime, and the stated suppression assumptions needed for structural quiescence, a split-enabled structural trajectory can enter a metric-compatible quiescent regime. The paper then states the explicit Kernel B identification under which such a regime may be brought into the constant-point-particle frozen-state language. The bridge theorem is therefore not a reduction of active Kernel A dynamics to Kernel B. It is a conditional route by which a structurally richer late regime may enter the frozen metric sector where the strongest theorem package applies.

What has and has not been achieved. The contribution of the paper is substantial but deliberately bounded. It does not provide complete theorem-level closure of the full SERD move family. It does not prove unrestricted active-dynamical observer recovery, and it does not claim a final continuum identification with any established physical theory. What it does provide is a disciplined formal hierarchy in which the relation between local rewrites, transported records, frozen correction, conditional recovery, executable kernels, and theorem-target observables can be stated clearly.

Interpretive significance. Although the exact theorem package is formulated for a literal freeze protocol, that regime may also be read heuristically as an idealized approximation to a quasi-static or metastable latent sector. In such a sector, the characteristic timescale for residual transport would be short relative to the timescale on which the local latent substrate undergoes further structural reconfiguration. On a bounded causal domain, the frozen theorem package can therefore be viewed as a controlled model of local observer-side settling on an effectively fixed latent background. The paper does not prove such an approximation principle, but it clarifies why the frozen regime may still be conceptually relevant to the broader SERD programme.

Next formal steps. The main open formal problems are now sharper. On the Kernel B side, the next task is to derive useful sufficient conditions for recovery compatibility of the full frozen state directly from the local update bookkeeping and frozen transport algebra, rather than leaving recovery as a conditional classification. More specifically, one wants to show that the exact terminal frozen correction identified here for observer-symmetric full frozen states equals the outstanding observer defect generated before freeze. On the Kernel A side, the next task is to strengthen the bridge theorem by deriving the A-to-B identification from intrinsic latent conditions rather than postulating it separately. More broadly, the parent SERD framework still requires a larger theorem-level treatment that incorporates additional move sectors and deepens the analysis of active dynamics outside the frozen regime. **Final perspective.** The present paper should therefore be read as a formalization and restricted-kernel theorem paper, not as a claim of full SERD closure. Its main achievement is to isolate a theorem-bearing frozen transport sector in which exact post-freeze correction can be described cleanly and conditional observer recovery can be stated precisely. If the wider SERD programme is to develop into a more comprehensive theory of emergent discrete relational structure, this kind of bounded formal layer is necessary. The present paper is intended as that layer for the current stage of the work.

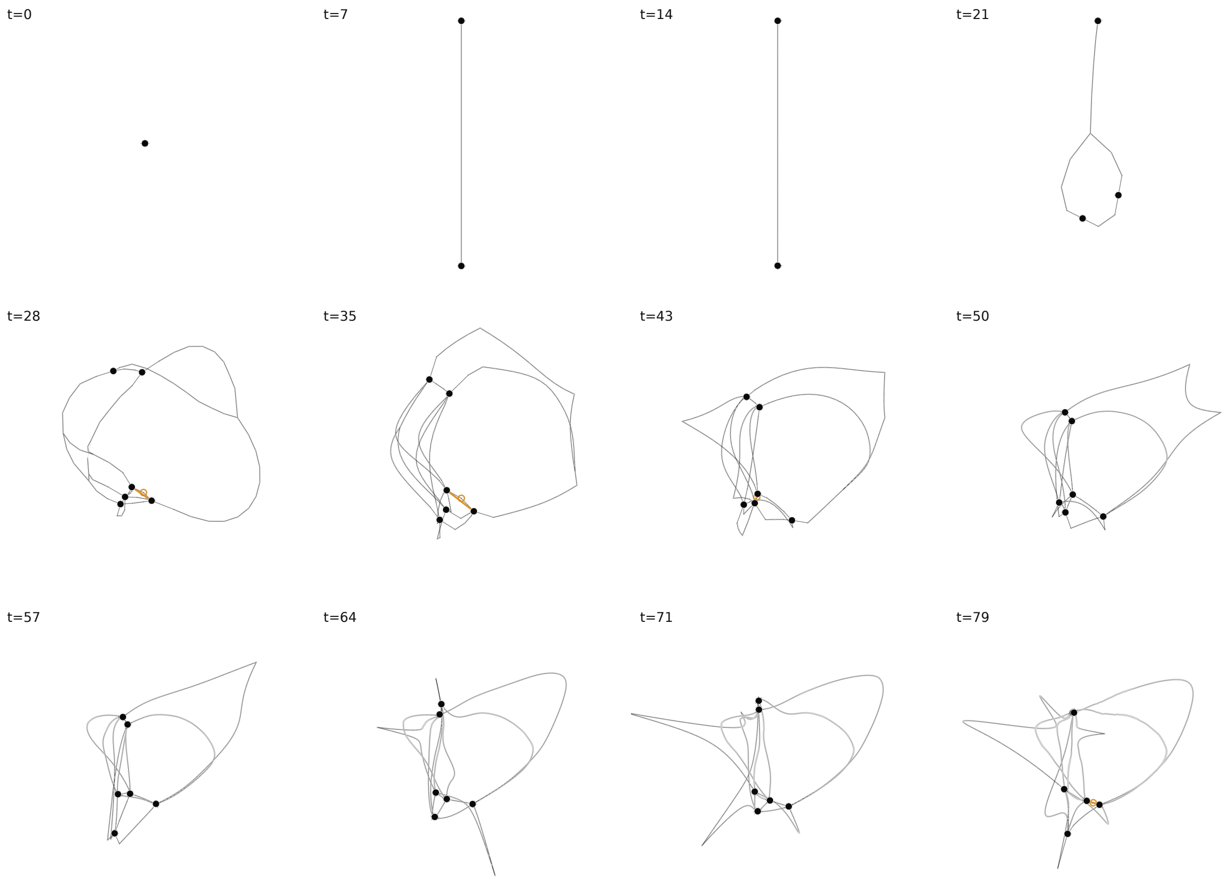

Figure 5: **Kernel A latent-state evolution.** Representative single trajectory of the restricted split-enabled SERD kernel. Black disks denote point particles (PPs), light grey points denote information gaps (IGs), grey line segments denote embedded space-element (SE) filaments, and thin dark segments indicate local PP–IG incidences. Starting from a two-PP filament, PP splitting together with local SE duplication and reduction generates a branched latent substrate with multiple PP neighbourhoods and dynamically reconfigured filament adjacencies. The sequence provides visual intuition for the Kernel A–to–Kernel B bridge theorem: only after the explicit bridge hypotheses are imposed—including finite stopping of PP splits, finite settling of transported structural records, and identification of the remaining PP–filament data with a fixed-PP metric state—may the late Kernel A configuration be interpreted in the constant-PP language used by Kernel B. The planar embedding visualizes latent topology only and is not used as theorem-level evidence.

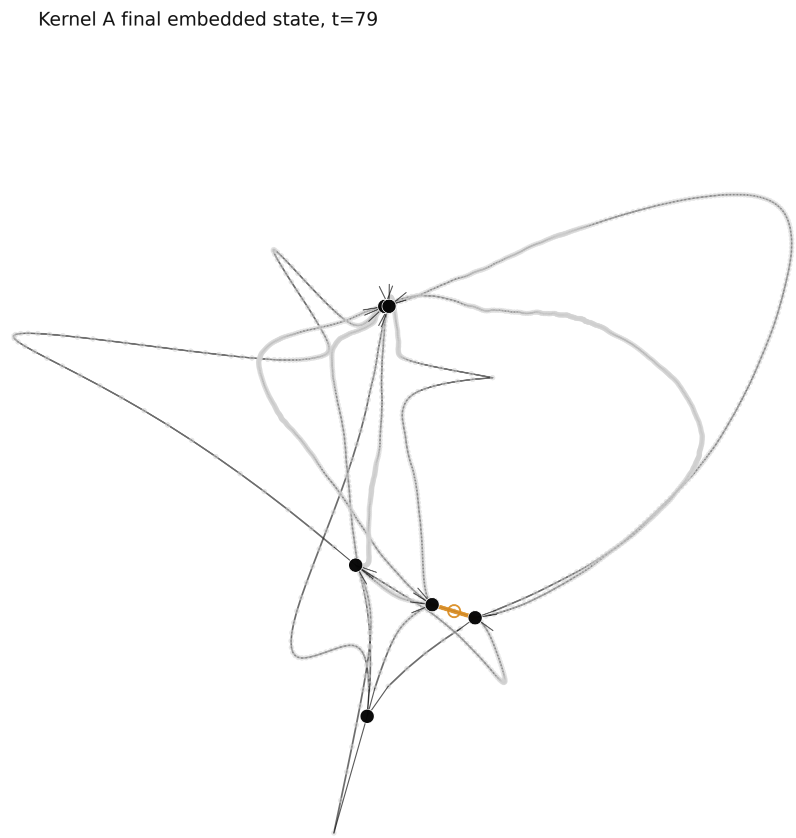

Figure 6: **Final embedded Kernel A state.** Enlarged view of the final sampled state from the representative Kernel A trajectory shown in Fig. 5. Black disks denote PPs, light grey points denote IGs, grey line segments show the embedded SE filament structure, and thin dark segments show PP–IG incidences. Where present, orange highlights mark compact or fused local contacts. The panel makes the late relational substrate visually explicit, showing several PP neighbourhoods linked by dynamically generated filaments and local incidences. As in Fig. 5, this embedding is included only as a visual diagnostic of the latent topology and not as a proof object.

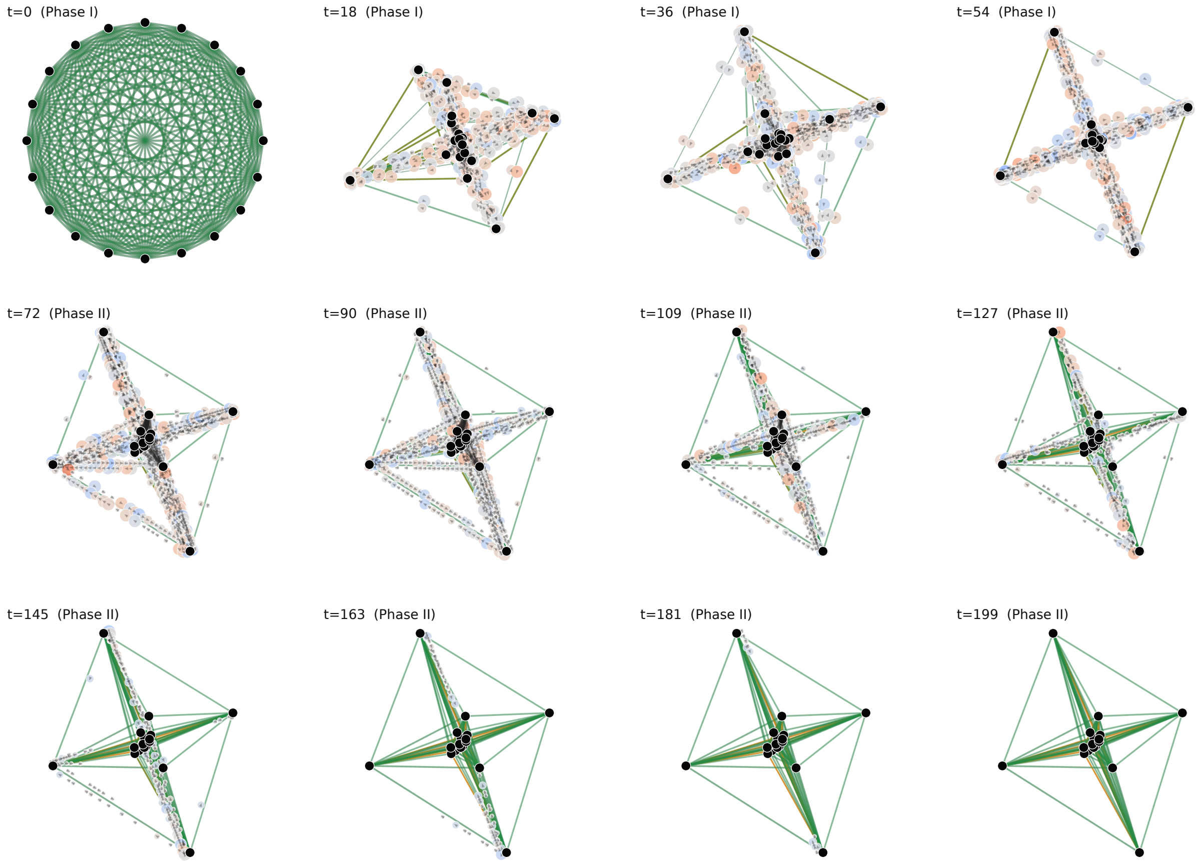

Figure 7: **Kernel B latent metric, transported information, and observed-state overlay.** Representative single run of the constant-PP metric kernel. Black disks denote PPs; grey line segments show the latent PP–PP filaments determined by the current filament-length data; pale grey sites indicate IG rows along those filaments; and green line segments show the observer-side reconstructed metric overlay. Blue/red markers denote nonzero signed transported metric-record content (called TIP content in the visualization layer) on directed filament faces, with marker area proportional to unsigned packet mass. Orange highlights indicate compact contacts, corresponding to off-diagonal latent lengths $L_{ij} = 0$, when present. During Phase I, local duplication, reduction, and compact-bond reopening updates modify the latent metric and generate transported records. After the freeze time, local updates are disabled, and Phase II illustrates the deterministic frozen-regime dynamics: transport, one-generation-only rebroadcast, and finite settling of the remaining metric-record content. The figure visualizes the setting of the frozen-regime theorem package and should be read as an illustrative diagnostic rather than as an additional proof.

Declarations

Acknowledgments

The author gratefully acknowledges colleagues and collaborators for valuable discussion, feedback, and support during the development of this work. In particular, the author thanks Prof. Andrew Adamatzky, primary supervisor and Director of the Unconventional Computing Laboratory at the University of the West of England, for his continued support of the wider research programme, and Prof. Larry Bull, secondary supervisor, for helpful discussions, feedback, and encouragement throughout the development of the project.

AI Contributions Statement

Large language models were used during the preparation of this manuscript as structured research-assistance tools. Their use included theorem exploration, proof-map generation, adversarial consistency checking, cross-comparison of candidate formalizations, code–paper correspondence analysis for the restricted kernels, editorial restructuring, drafting support for selected sections, final minor-revision consistency checks, figure-caption drafting, and provenance-package preparation.

AI outputs were not treated as authoritative. Proposed definitions, proof strategies, algebraic interpretations, and explanatory prose were reviewed critically by the author and were accepted only after comparison with the manuscript's formal framework, the implementation-faithful kernel semantics, and direct mathematical inspection. The author retained final responsibility for all definitions, theorem statements, proofs, interpretations, editorial decisions, and any errors.

A separate AI worklog and integrity package accompanies this article. It documents the model-use categories, source-of-truth hierarchy, claim-verification table, artifact manifest, editor-facing notes, and normalized thread self-report summaries used to reconstruct and bound the AI-assisted workflow.

Artifact Availability

The manuscript source, bibliography, final figures, and AI worklog / integrity package accompany the publication materials. The worklog package includes its README, integrity report, source-of-truth and verification note, claim-verification table, model-usage log, artifact manifest, editor cover note, and normalized thread self-report summaries. Reference implementations of the restricted kernels are part of companion computational work and will be made available through the corresponding project repository when released.

Energy Disclosure

The principal results of this article did not require large-scale numerical experimentation. Compute use associated with manuscript preparation and AI-assisted drafting was limited to standard interactive research workflows.

Conflicts of Interest

The author declares no competing interests.

Funding

Research has been funded by the *University of the West of England Open Project PhD studentship*.

Ethics

No human-subjects research, animal research, or clinical data collection was conducted for this article.

References

[1] Alfred North Whitehead. *Process and Reality*. Macmillan, New York, 1929.

[2] James Ladyman and Don Ross. *Every Thing Must Go: Metaphysics Naturalized*. Oxford University Press, Oxford, 2007.

[3] Carlo Rovelli. Relational quantum mechanics. *International Journal of Theoretical Physics*, 35(8):1637–1678, 1996.

[4] Luca Bombelli, Joohan Lee, David Meyer, and Rafael D. Sorkin. Space-time as a causal set. *Physical Review Letters*, 59(5):521–524, 1987.

[5] Sumati Surya. The causal set approach to quantum gravity. *Living Reviews in Relativity*, 22(1):5, 2019.

[6] Fay Dowker. Causal sets as discrete spacetime. *Living Reviews in Relativity*, 25(1):5, 2022.

[7] Renate Loll. Quantum gravity from causal dynamical triangulations: a review. *Classical and Quantum Gravity*, 37(1):013002, dec 2019.

[8] Tomasz Konopka, Fotini Markopoulou, and Simone Severini. Quantum graphity: A model of emergent locality. *Physical Review D*, 77(10):104029, May 2008.

[9] Carlo Rovelli. *Quantum Gravity*. Cambridge University Press, 2004.

[10] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, 2002.

[11] Stephen Wolfram. *A Project to Find the Fundamental Theory of Physics*. Wolfram Media, 2020.

[12] Jonathan Gorard. Some relativistic and gravitational properties of the wolfram model. *arXiv preprint arXiv:2004.14810*, 2020.

[13] Xerxes D. Arsiwalla and Jonathan Gorard. Pregeometric spaces from wolfram model rewriting systems as homotopy types, 2021.

[14] John Archibald Wheeler. Information, physics, quantum: The search for links. In Wojciech H. Zurek, editor, *Complexity, Entropy, and the Physics of Information*, pages 3–28. Addison-Wesley, 1990.

[15] Rolf Landauer. Irreversibility and heat generation in the computing process. *IBM Journal of Research and Development*, 5(3):183–191, 1961.

[16] Seth Lloyd. Ultimate physical limits to computation. *Nature*, 406:1047–1054, 2000.

[17] Mark Van Raamsdonk. Building up space-time with quantum entanglement. *General Relativity and Gravitation*, 42:2323–2329, 2010.

[18] Thanu Padmanabhan. Surface density of spacetime degrees of freedom from equipartition law in theories of gravity. *Physical Review D*, 81(12):124040, 2010.

[19] Erik Verlinde. On the origin of gravity and the laws of newton. *Journal of High Energy Physics*, 2011(4):29, 2011.

[20] Erik Verlinde. Emergent gravity and the dark universe. *SciPost Physics*, 2(3):016, 2017.

[21] Eugenio Bianchi. Entropy of non-extremal black holes from loop gravity. *Physical Review Letters*, 111(4):041301, 2012.

[22] Toshiyuki Nakagaki, Hiroyasu Yamada, and Akiko Toth. Intelligent behavior of a Physarum plasmodium foraging for food. *Biophysical Chemistry*, 84(1):195–202, 2000.

[23] Mark Fricker, Lynne Boddy, and Daniel Bebber. Network organisation of mycelial fungi. *Biology of the Fungal Cell*, pages 309–330, 2007.

[24] Luke Heaton, Boguslaw Obara, Vincente Grau, Nick Jones, Toshiyuki Nakagaki, Lynne Boddy, and Mark D. Fricker. Analysis of fungal networks. *Fungal Biology Reviews*, 26(1):12–29, 2012.

[25] Graeme P. Boswell and Fordyce A. Davidson. Modelling hyphal networks. *Fungal Biology Reviews*, 26(1):30–38, 2012.

[26] Tommy Wood, Tuomas Sorakivi, Phil Ayres, and Andrew Adamatzky. Exploring discrete space–time models for information transfer: Analogies from mycelial networks to the cosmic web. *BioSystems*, 243:105278, 2024.

[27] Mohammad Retaful Islam, Gregory Tudryn, Ronald Bucinell, Linda Schadler, and R. Catalin Picu. Morphology and mechanics of fungal mycelium. *Scientific Reports*, 7(1):13070, 2017.

[28] Lia Papadopoulos, Pablo Blinder, Henrik Ronellenfitsch, Florian Klimm, Eleni Katifori, David Kleinfeld, and Danielle S. Bassett. Comparing two classes of biological distribution systems using network analysis. *PLoS Computational Biology*, 14(9):e1006428, 2018.

[29] Thomas L. Wood. Emergence of massive equilibrium states from fully connected stochastic substitution systems. *Journal of Cellular Automata*, 12(3-4):189–208, 2017.

[30] Thomas L. Wood. Nonlocal and light cone dynamics emergent from information-propagating complete graph. *Complex Systems*, 27(2), 2018.

[31] Thomas L. Wood. Space element reduction duplication (SERD) model produces photon-like information packets and light-like cosmological horizons. *The Journal of Methodology*, 2022.

[32] Tommy Wood. Emergent mechanics from self-generating topological information network. *International Journal of Unconventional Computing*, 19, 2024.