AIΩN

Computational Holography: Recursive Metagraphs, Rulial Distance, and Deterministic Multiway Computation

James Ross
Independent Researcher
ORCID: 0009-0006-0025-7801

December 4, 2025

This document is part of the AIΩN Foundations Series.
See `https://github.com/flyingrobots/aion` for updates.

Abstract

This paper develops a formal model of *computational holography*: representing a computation so that its entire interior evolution is encoded on the boundaryspecifically, the provenance payload of a single recursive metagraph edge. We build on double-pushout (DPO) graph rewriting in adhesive categories to define Recursive Metagraphs (RMGs), give a deterministic concurrent operational semantics, and prove tick-level confluence together with a two-plane commutation theorem. We also identify conditions for global confluence. Incorporating provenance payloads yields an information-complete, holographic encoding of computational history. Finally, we define an MDL-based *rulial distance* between observers, relate RMG dynamics to multiway systems and the Ruliad, and outline how this structure supports the AIΩN CΩMPUTER: a provenance-native computational model with deterministic tick semantics and a quasi-pseudometric geometry on observers.

Contents

Standing Assumptions

For ease of reference, we summarise the main semantic assumptions used in the determinism, holography, and rulial-distance results.

Assumption	Where used
Skeleton independence via footprints	Theorems 4.1, 4.3 and 4.4
No-delete/no-clone-under-descent (ND/NC)	Theorems 4.6 to 4.8
Termination / decreasing diagrams on the skeleton	Theorem 4.8 (conditional global confluence)
No re-derivation (single producer)	Section 5, Theorem 5.2 (backward provenance)
Budgeted translators and 1-Lipschitz distortion	Section 7 and Theorems 7.1 and 7.2 (rulial distance)

Unless otherwise stated, all results in the main text are to be read relative to these assumptions.

1 Introduction

Modern computation is built on mutable state and loosely specified concurrency. As systems become distributed, multicore, and AI-mediated, this leads to nondeterminism, opaque failure modes, unreproducible behavior, and fundamentally incomplete provenance.

The goal of this work is to develop a mathematically precise alternative model in which:

- global state is represented as a recursively nested, graph-shaped object;
- all evolution of that state is given by well-typed graph rewrites;
- under an explicit independence discipline and no-delete/no-clone-under-descent invariants, the operational semantics is deterministic (up to typed open graph isomorphism) and confluent at the level of "ticks" of computation (Theorems 4.4, 4.7 and 4.8); and
- the *entire* interior evolution of a computation is stored in a compact *provenance payload* attached to a single edge, yielding an information-complete holographic encoding.

Our technical starting point is algebraic graph transformation via double-pushout (DPO) graph rewriting in adhesive categories. We pair this with the Recursive Metagraph (RMG) object model introduced in Section 2. We extend this setting with:

1. a precise notion of RMG state and its category;
2. a two-plane concurrent operational semantics with attachment–then–skeleton publication, together with confluence results: tick-level determinism and two-plane commutation (Theorems 4.4 and 4.7), and—under the standing assumptions and standard rewrite-theoretic conditions—global confluence (Theorem 4.8);
3. a provenance payload calculus yielding *computational holography*;
4. an MDL-based quasi-pseudometric on observers, the *rulial distance*, and a correspondence between RMG derivations and multiway systems that clarifies the relationship to Wolfram's Ruliad.

We keep the system-level AIΩN stack[1] (AIONOS, Echo, Wesley, etc.) largely offstage, mentioning it only to motivate the mathematical structure; we assume only that implementations respect the axioms and invariants stated in Sections 2–10, with scheduling, representation, and persistence details intentionally out of scope. A companion "CΩMPUTER" paper will build on these results to define the full machine model and operating system.

Key contributions. The main results of this paper are:

1. *Tick-level confluence* (Theorem 4.4): parallel independent RMG rewrites commute, yielding per-tick deterministic semantics independent of scheduler serialization order;
2. *Two-plane commutation* (Theorem 4.7): attachment and skeleton updates commute up to isomorphism;
3. *Worldline uniqueness* (Corollary 4.5): under the standing assumptions (tick confluence, two-plane commutation, conditional global confluence, and holography), the boundary data (S_0, P) determines the interior derivation uniquely up to typed open graph isomorphism, extending per-tick commutation to whole-run semantics;

[1] Pronounced "eye-ON" (rhymes with *aeon*), with stress on the second syllable.

4. *Computational holography* (Theorem 5.4): the boundary data (S_0, P) is information-complete with respect to the interior evolution, enabling reconstruction of the full derivation volume;

5. *Rulial distance* (Theorem 7.2): the triangle-inequality property of an MDL-based quasi-pseudometric on observers, capturing the complexity of translating between different computational views.

2 Recursive Metagraphs

In this section we define Recursive Metagraphs (RMGs) and relate them to standard graph models and typed open graphs. An RMG is a finite typed open graph whose nodes and edges may themselves carry RMGs recursively, forming a finitely branching, well-founded tree of graphs.

2.1 Inductive definition

Fix a set P of atomic payloads (blobs, literals, external IDs).

Definition 2.1 (Recursive Metagraph)**.** The class RMG of *recursive metagraphs* is the least set closed under the following constructors:

1. for each $p \in P$ there is an *atom* $\mathrm{Atom}(p) \in \mathrm{RMG}$;
2. for any finite directed multigraph $S = (V, E, s, t)$ and assignments $\alpha : V \to \mathrm{RMG}$, $\beta : E \to \mathrm{RMG}$, the triple (S, α, β) is in RMG.

We write an element of RMG as either an atom or as a "1-skeleton" graph decorated by attachments on vertices and edges. Attachments themselves may be recursive metagraphs, so this attachment structure can nest arbitrarily deeply. This definition matches the set-theoretic and initial-algebra presentation.

Example (A tiny recursive metagraph). As a concrete instance, consider a program call graph where each function node carries its own abstract syntax tree (AST) and each call edge carries a small provenance graph (e.g. optimisation decisions or runtime statistics). We can model this as an RMG whose skeleton has nodes v_f, v_g for functions f, g, a directed edge $e_{\mathsf{call}} : v_f \to v_g$ for the call, and attachments: $\alpha(v_f)$ the AST of f, $\alpha(v_g)$ the AST of g, and $\beta(e_{\mathsf{call}})$ the call provenance.

2.2 Initial algebra viewpoint

Let $\mathcal{G}$ be a small collection of allowable skeleton shapes (finite directed multigraphs up to isomorphism). Define a finitary polynomial endofunctor $F : \mathbf{Set} \to \mathbf{Set}$ by

$$F(X) = P + \coprod_{S \in \mathcal{G}} (V_S \to X) \times (E_S \to X).$$

Then RMG is (up to isomorphism) the carrier of the initial F-algebra. This yields the usual structural recursion and induction principles: every function out of RMG is uniquely determined by its action on atoms and on decorated skeletons.

2.3 Unfoldings and recursion schemes

The depth of an RMG X is the length of the longest attachment chain in X; finite depth follows from the inductive definition. For $k \in \mathbb{N}$, the *k-unfolding* of X, written $\mathrm{unf}_k(X)$, replaces each attachment at depth k or greater by an opaque atom while preserving all structure at depths $0, \ldots, k-1$. Each $\mathrm{unf}_k(X)$ is a finitely branching RMG obtained by structural recursion on depth.

The *infinite unfolding* $\mathrm{unf}_\infty(X)$ is the directed colimit of the $\mathrm{unf}_k(X)$ along the truncation maps $\mathrm{unf}_k(X) \to \mathrm{unf}_{k+1}(X)$ that agree on depths $0, \ldots, k-1$. Every finite pattern lives in some finite unfolding, so reasoning about X via recursion schemes (catamorphisms, anamorphisms) can be reduced to these finite approximants and then passed to the colimit. We do not rely on any stronger universal property of $\mathrm{unf}_\infty(X)$ in this paper.

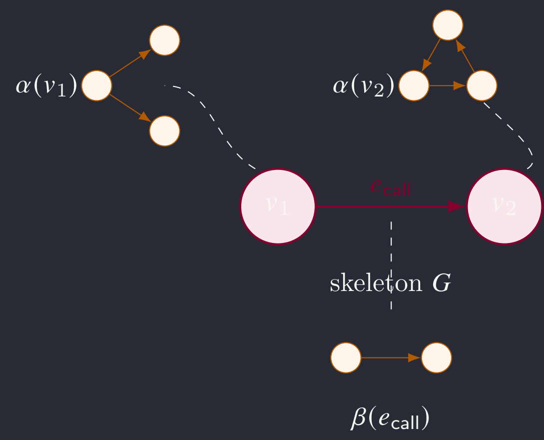

Figure 1: A simple recursive metagraph: the skeleton G has two nodes v_1, v_2 and an edge e_{call}, while each node and edge carries its own attached graph $\alpha(v_i), \beta(e_{\mathsf{call}})$. In an RMG this attachment structure recurses: the attachment graphs themselves may have attachments, and so on.

2.4 Morphisms and category of RMGs

Definition 2.2 (RMG morphism)**.** We define morphisms by structural recursion on RMG depth (the nesting level in the construction of Definition 2.1). First form the discrete category $\mathbf{P}$ with $\mathrm{Ob}(\mathbf{P}) = P$ and $\mathrm{Mor}(\mathbf{P})$ containing only identity morphisms. We define the RMG hom-sets on atoms to match this discrete structure:

$$\mathrm{Hom}_{\mathrm{RMG}}(\mathrm{Atom}(p), \mathrm{Atom}(p')) = \begin{cases} \{\mathrm{id}_{\mathrm{Atom}(p)}\} & \text{if } p = p', \\ \emptyset & \text{otherwise.} \end{cases}$$

This embedding is faithful because it preserves the identity-only structure of $\mathbf{P}$. For composite objects, a morphism $f : (S, \alpha, \beta) \to (S', \alpha', \beta')$ consists of:

- a graph homomorphism of skeletons $f_V : V \to V'$, $f_E : E \to E'$ preserving sources and targets; and
- for each $v \in V$ a morphism of attachments $f_v : \alpha(v) \to \alpha'(f_V(v))$ and, for each $e \in E$, a morphism $f_e : \beta(e) \to \beta'(f_E(e))$.

Note that each f_v and f_e is itself an RMG morphism, so this definition proceeds by the structural recursion announced above. Composition and identities are defined componentwise.

2.5 Relation to ordinary and hypergraphs

Typed open graphs $\mathbf{OGraph}_T$ form an adhesive category, and DPO rewriting is well-behaved there. Typed hypergraphs embed fully and faithfully into typed open graphs via an incidence construction that preserves DPO steps and their multiway derivations. Thus RMG rewriting subsumes standard open-graph and hypergraph rewriting while adding recursive structure through attachments.

2.6 Notation summary

For convenience, we collect the main notation introduced so far:

Symbol	Meaning
$U = (G; \alpha, \beta)$	single RMG state in universe $\mathcal{U}$
$p = (L \xleftarrow{\ell} K \xrightarrow{r} R)$	DPOI rule
μ_i	microstep label
$P = (\mu_0, \ldots, \mu_{n-1})$	provenance payload
$S_0 \Rightarrow^* S_n$	derivation volume (interior evolution)
(S_0, P)	wormhole (boundary encoding)
$\mathrm{Del}(m), \mathrm{Use}(m)$	delete and use sets of a match
$\mathrm{Recon}(S_0, P)$	reconstruction procedure

Throughout, an *RMG universe* $\mathcal{U}$ is a set of RMG states (typically closed under the rewrite rules R under consideration), and $U \in \mathcal{U}$ denotes a particular state in that universe.

Subsequent sections introduce $D_{\tau,m}$ (rulial distance), $\mathrm{Hist}(\mathcal{U}, R)$ (history category on the universe $\mathcal{U}$ of RMG states), and other observer-related notation.

3 DPO Rewriting on Recursive Metagraphs

We briefly review double-pushout with interfaces (DPOI) rewriting on typed open graphs and lift it to RMG states.

3.1 Typed open graphs and DPOI rules

Let T be a finite set of types. Let $\mathbf{OGraph}_T$ be the category of T-typed open graphs, whose objects are cospans of monomorphisms $I \hookrightarrow G \hookleftarrow O$ and whose morphisms are commuting maps of cospans. This category is adhesive (see [LS08]); in particular, pushouts along monos exist and form Van Kampen squares. We use the shorthand $\hookrightarrow$ for the monomorphism arrow $\hookrightarrow$.

Definition 3.1 (DPOI rule)**.** A *DPOI rule* is a span of monos in $\mathbf{OGraph}_T$

$$p = (L \xleftarrow{\ell} K \xrightarrow{r} R)$$

with L the left-hand side, K the interface, and R the right-hand side. A *match* of p in a host graph G is a mono $m : L \hookrightarrow G$ satisfying the usual gluing conditions: the dangling condition and the identification condition.

Given a match, the DPO construction yields a rewrite step $G \Rightarrow_p H$ by computing a pushout complement and a pushout:

$$\begin{tikzcd} K \arrow[r, "\ell"] \arrow[d, "k"'] & L \arrow[d, "m"] \\ D \arrow[r] & G \end{tikzcd} \qquad \begin{tikzcd} K \arrow[r, "r"] \arrow[d, "k"'] & R \arrow[d] \\ D \arrow[r] & H \end{tikzcd}$$

All arrows are monos in $\mathbf{OGraph}_T$.

3.2 RMG states as two-plane objects

Definition 3.2 (RMG state)**.** An RMG *state* is a triple

$$U = (G; \alpha, \beta)$$

where $G \in \mathbf{OGraph}_T$ is the skeleton and α, β assign attachment objects in the appropriate fibres (of the forgetful functor $\pi : \mathrm{RMGState} \to \mathbf{OGraph}_T$, see Section 4) to nodes and edges of G.

This separates the global state into a base skeleton and recursively attached subgraphs. Rewriting operates in two "planes":

- *attachment steps* are DPOI steps in the fibres $\alpha(v)$, $\beta(e)$ that do not change G;
- *skeleton steps* are DPOI steps on G itself, subject to rules ensuring that attachments at preserved positions can be transported.

Definition 3.3 (Tick)**.** A *tick* on an RMG state $U = (G; \alpha, \beta)$ consists of a finite family of attachment steps in the fibres over G followed by a finite family of skeleton steps on G, chosen by the scheduler. In Section 4 we impose additional conditions (independence, scheduler–admissible batches, and the no-delete/no-clone-under-descent invariant) on the ticks generated by the runtime.

We now turn to the determinism properties of this semantics.

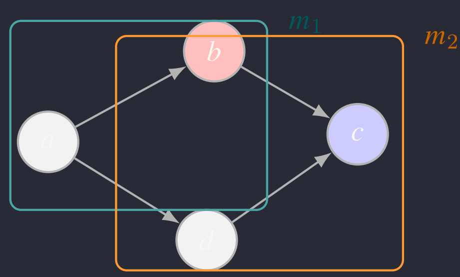

Figure 2: Two overlapping matches on the skeleton plane. Shaded nodes illustrate Del (red) and Use (blue). Independence requires $\mathrm{Del}(m_1) \cap \mathrm{Use}(m_2) = \mathrm{Del}(m_2) \cap \mathrm{Use}(m_1) = \emptyset$, ruling out destructive interference between the matches.

4 Determinism and Confluence

Throughout this section we work under the standing assumptions summarised in the Standing Assumptions section.

We sketch the concurrency discipline, define independence, and state the main confluence theorems for tick-level execution, working with RMG states and tick semantics as introduced in Theorems 3.2 and 3.3.

4.1 Footprints and Independence on the Skeleton Plane

We work at the level of the skeleton plane. Write G_S for the skeleton component of G, and likewise L_S, K_S, R_S for the underlying skeleton graphs of a rule. Let $U = (G; \alpha, \beta)$ be an RMG state and let $p = (L \xleftarrow{\ell} K \xrightarrow{r} R)$ be a DPOI rule. A *skeleton match* is a mono $m_S : L_S \hookrightarrow G_S$ in $\mathbf{OGraph}_T$ satisfying the usual gluing conditions.

Definition 4.1 (Footprint)**.** The *delete set* $\mathrm{Del}(m_S) \subseteq \mathrm{Ob}(G_S)$ of a match m_S is the image under m_S of the part of the left-hand side that is not preserved:

$$\mathrm{Del}(m_S) \;=\; m_S\big(L_S \setminus K_S\big).$$

The *use set* $\mathrm{Use}(m_S) \subseteq \mathrm{Ob}(G_S)$ is the image under m_S of all of L_S:

$$\mathrm{Use}(m_S) \;=\; m_S(L_S).$$

The *footprint* of m_S is the pair $\mathrm{Foot}(m_S) = (\mathrm{Del}(m_S), \mathrm{Use}(m_S))$.

Definition 4.2 (Independence)**.** Two skeleton matches $m_{1,S} : L_{1,S} \to G_S$ and $m_{2,S} : L_{2,S} \to G_S$ with footprints $\mathrm{Foot}(m_{i,S}) = (\mathrm{Del}(m_{i,S}), \mathrm{Use}(m_{i,S}))$ are *independent* if

$$\mathrm{Del}(m_{1,S}) \cap \mathrm{Use}(m_{2,S}) = \emptyset \quad \text{and} \quad \mathrm{Del}(m_{2,S}) \cap \mathrm{Use}(m_{1,S}) = \emptyset.$$

Intuitively, no vertex or edge that one step deletes is read or written by the other.

4.1.1 Scheduler–admissible batches

Definition 4.3 (Scheduler–admissible batch)**.** Let $U = (G; \alpha, \beta)$ be an RMG state. A finite family of skeleton matches $B = \{m_{i,S} : L_{i,S} \to G_S\}_{i \in I}$ is *scheduler–admissible* if the matches are pairwise independent in the sense of Definition 4.2, i.e.

$$\mathrm{Del}(m_{i,S}) \cap \mathrm{Use}(m_{j,S}) = \emptyset \quad \text{for all distinct } i, j \in I.$$

Admissibility only requires pairwise independence; in practice the scheduler selects a *maximal* independent subset each tick.

4.2 Tick semantics and scheduler confluence

We recall the basic setting. Work in the adhesive category $\mathbf{OGraph}_T$ of typed open graphs. A DPOI rule is a span of monos $p = (L \xleftarrow{\ell} K \xrightarrow{r} R)$; a match is a mono $m : L \hookrightarrow G$ satisfying the usual gluing conditions (dangling and identification). A DPOI step $G \Rightarrow_p H$ is given by the standard double square (pushout complement + pushout).

We work with RMG states and tick semantics as defined in Definitions 3.2 and 3.3. In this section we analyse when sets of matches can be scheduled concurrently without affecting the resulting state. The deterministic properties proved here apply uniformly to all derivations; applying them to cognitive systems introduces additional ethical structure developed in Section 9.

Each tick, the scheduler computes a maximal independent set of matches from the scheduler–admissible (pairwise independent) matches, using a safe over-approximation of Use ∪ Del; we do not repeat the implementation details here.

Theorem 4.4 (Skeleton-plane tick confluence)**.** *Let $U = (G; \alpha, \beta)$ be an RMG state and let $B = \{m_{i,S} : L_{i,S} \hookrightarrow G_S\}_{i\in I}$ be a scheduler–admissible batch for a family of DPOI rules (not necessarily maximal). Then any two sequentialisations of the corresponding DPOI steps yield isomorphic successor states.*

Proof. By scheduler–admissibility (pairwise independence), the skeleton matches $\{m_{i,S} : L_{i,S} \hookrightarrow G_S\}$ are independent in the sense of Theorem 4.2. By the Parallel Independence Theorem for DPO rewriting in adhesive categories [EEPT06, Thm. 5.4] (see also [ACM25, Sec. 4]), parallel independent steps commute: for any $i \neq j$ we have a diagram

$$G_S \Rightarrow_{(p_i, m_{i,S})} G_i \Rightarrow_{(p_j, m'_{j,S})} G_{ij} \quad \text{and} \quad G_S \Rightarrow_{(p_j, m_{j,S})} G_j \Rightarrow_{(p_i, m'_{i,S})} G_{ji}$$

where both two-step derivations exist and the results G_{ij} and G_{ji} are isomorphic. The rewritten matches $m'_{i,S}, m'_{j,S}$ are obtained by the standard reindexing construction of the concurrency theorem.

We now induct on $|I|$ to obtain order-independence for the entire family of skeleton steps.

For $|I| = 0$ or 1 the claim is trivial. For $|I| = 2$ it is exactly the commuting-square case of the concurrency theorem.

Assume the property holds for all scheduler–admissible batches of size k. Let B have size $k + 1$. Pick any index $j \in I$ and factor an arbitrary serial order as

$$G_S \Rightarrow^{(p_{i_1}, m_{i_1,S})} \cdots \Rightarrow^{(p_{i_k}, m_{i_k,S})} G' \Rightarrow^{(p_j, m'_{j,S})} G''.$$

By the induction hypothesis, the prefix of length k yields a result unique up to isomorphism, regardless of the order of the k steps. Now compare any two permutations of the full $(k + 1)$ steps. One can be obtained from the other by a finite sequence of adjacent swaps. Each adjacent swap exchanges two parallel independent steps; by the two-step concurrency theorem, the corresponding length-$(k + 1)$ derivations commute up to isomorphism. Thus, by finite induction on the number of swaps, all serialisations of the skeleton batch produce isomorphic skeletons. Lifting back along the fibration $\pi : \text{RMGState} \to \mathbf{OGraph}_T$ yields the claimed determinism up to isomorphism for RMG states.

Combining Theorem 4.4 with the two-plane commutation result (Theorem 4.7) shows that a tick that satisfies the no-delete/no-clone-under-descent invariant has a unique outcome up to isomorphism in the full RMG semantics.

Corollary 4.5 (Worldline uniqueness)**.** *Let an RMG runtime satisfy the assumptions of Theorems 4.4, 4.7 and 4.8 together with the holography property (Theorem 5.4). Then every schedule of a given tick produces the same RMG successor up to isomorphism, and the boundary data* (S_0, P) *determines the interior derivation uniquely up to isomorphism.*

4.3 Two-plane commutation via a fibration

We now justify the two-plane discipline more structurally, using a simple fibration view.

Let RMGState be the category of RMG states and RMG morphisms (skeleton morphisms together with compatible fiber morphisms). There is a forgetful functor

$$\pi : \mathrm{RMGState} \longrightarrow \mathbf{OGraph}_T$$

sending $(G;\alpha,\beta)$ to its skeleton G and acting on morphisms componentwise. This functor is a (Grothendieck) fibration whose fibers are products of copies of $\mathbf{OGraph}_T$:

$$\pi^{-1}(G) \;\cong\; \prod_{x \in V(G) \cup E(G)} \mathbf{OGraph}_T.$$

In particular, given a mono $u : G \hookrightarrow G'$ in the base, there is a reindexing functor

$$u^* : \pi^{-1}(G') \longrightarrow \pi^{-1}(G)$$

which transports attachments along u by precomposition.

An *attachment step* is a DPOI step in some fiber $\pi^{-1}(G)$; a *skeleton step* is a DPOI step in the base $\mathbf{OGraph}_T$. Both are built from pushouts along monos.

Definition 4.6 (No-delete/no-clone-under-descent)**.** Consider a tick on an RMG state $U = (G;\alpha,\beta)$ consisting of

1. a family of attachment–plane DPOI steps, each acting inside a fibre $\alpha(v)$ or $\beta(e)$ over a skeleton vertex or edge, and
2. a single skeleton–plane DPOI step $G_S \Rightarrow_S G'_S$ induced by a rule and match $m_S : L_S \to G_S$.

We say that an object y in an attachment graph has *skeleton ancestor* $x \in G_S$ if it lies in the finite tree of attachments rooted at the fibre over x, following the recursive attachment structure of Section 2. We say that this tick satisfies *no-delete/no-clone-under-descent* if:

(ND) (*No delete under descent.*) If a skeleton vertex or edge $x \in G_S$ is deleted by the skeleton step (i.e. $x \in \mathrm{Del}(m_S)$), then the fibre over x is empty before the tick: no object in any attachment graph has x as its skeleton ancestor.

(NC) (*No clone under descent.*) The skeleton step does not implicitly duplicate attachment state: whenever a skeleton vertex or edge $x \in G_S$ is preserved and mapped to $x' \in G'_S$, the attachment over x' is (up to isomorphism) obtained from the attachment over x solely by the attachment–plane DPOI steps in the same tick. In particular, no attachment object is copied to multiple descendants of x.

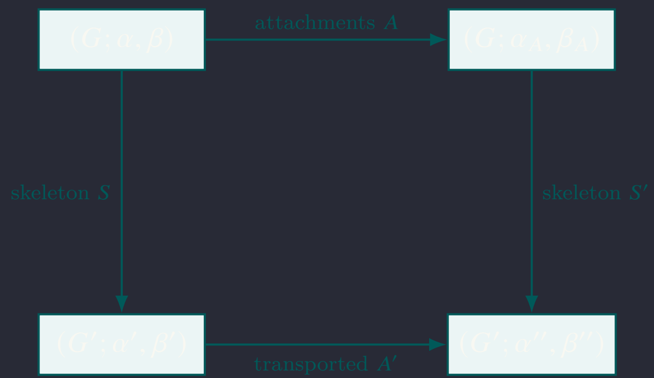

Figure 3: Two-plane commutation: attachment updates A in the fiber over G commute with skeleton rewriting S in the base, up to transporting the attachment steps along the skeleton morphism. Theorem 4.7 shows that the two paths in the square yield isomorphic RMG states.

A rule pack R satisfies no-delete/no-clone-under-descent if every tick generated from R has this property.

Theorem 4.7 (Two-plane commutation)**.** *Let R be a rule pack satisfying the no-delete/no-clone-under-descent invariant of Definition 4.6. Let $U = (G;\alpha,\beta)$ be an RMG state generated from R. Let $A : U \Rightarrow U_A$ be a finite composite of attachment steps in the fiber over G, and let $S : G \Rightarrow G'$ be the skeleton DPOI step induced by the same tick. Then there exists an attachment composite $A' : (G';\alpha',\beta') \Rightarrow (G';\alpha'',\beta'')$ in the fiber over G' such that the following square in* RMGState *commutes up to isomorphism:*

$$\begin{tikzcd}
(G;\alpha,\beta) \arrow[r, "A"] \arrow[d, "S"'] & (G;\alpha_A,\beta_A) \arrow[d, "S'"] \\
(G';\alpha',\beta') \arrow[r, "A'"'] & (G';\alpha'',\beta'')
\end{tikzcd}$$

In particular, applying attachments then skeleton yields the same result (up to iso) as applying skeleton then the transported attachments.

Proof. Write the skeleton composite S as a sequence of DPOI steps

$$G = G_0 \Rightarrow G_1 \Rightarrow \cdots \Rightarrow G_n = G'.$$

Each step $G_{k-1} \Rightarrow G_k$ is a pushout along a mono in $\mathbf{OGraph}_T$. Because $\mathbf{OGraph}_T$ is adhesive, pushouts along monos are Van Kampen squares and stable under pullback [LS08].

The no-delete/no-clone-under-descent hypothesis ensures that every position x whose attachment is touched by A lies in the preserved interface of each skeleton step. Thus, along the composite mono $u : G \hookrightarrow G'$, the reindexing functor $u^* : \pi^{-1}(G') \to \pi^{-1}(G)$ is an isomorphism on the fibers corresponding to positions touched by A; informally, the skeleton only renames those attachment slots.

Consider first a single attachment step in the fiber over some position x:

$$(G;\alpha,\beta) \Rightarrow_A (G;\alpha_A,\beta_A),$$

given by a DPOI double square in the corresponding component of the fiber $\pi^{-1}(G) \cong \prod_x \mathbf{OGraph}_T$. Forming the pullback of this square along the mono $u : G \hookrightarrow G'$ yields a square in the fiber over G'; by stability of pushout complements and Van Kampen, this square is again a DPOI step, which we denote by A':

$$(G'; \alpha', \beta') \Rightarrow_{A'} (G'; \alpha'', \beta'').$$

At the level of the total category RMGState we thus obtain a commuting cube whose back face is the original attachment step, whose bottom face is the skeleton step, and whose front face is the transported attachment step. All vertical faces are pullbacks and all horizontal faces are pushouts along monos; Van Kampen ensures that the top and bottom composites are isomorphic.

Iterating this construction over the finite families of attachment and skeleton steps yields a composite cube whose front and back faces are the two composites $S \circ A$ and $A' \circ S'$ in the statement. By pasting of Van Kampen squares, the induced morphism between the top and bottom objects is an isomorphism in RMGState. Hence the diagram commutes up to isomorphism.

4.4 Global confluence

To obtain global Church–Rosser properties for the entire rewrite system, additional hypotheses are required. We invoke the standard critical-pair lemma and Newman's lemma for terminating systems:

Theorem 4.8 (Conditional global confluence)**.** *Let R be a finite DPOI rule set. Suppose that:*

1. *every DPOI critical pair of R is joinable (modulo boundary isomorphism), and*
2. *the induced rewrite relation is terminating on the class of states considered, or admits a decreasing-diagrams labelling.*

Then the rewrite relation is confluent.

This theorem applies directly to the skeleton plane; together with the no-delete/no-clone-under-descent invariant and two-plane commutation, it yields uniqueness of *worldlines* at the level of RMG states for rule packs satisfying the hypotheses of Theorem 4.8 (the no-delete/no-clone-under-descent invariant plus termination and joinability of critical pairs).

Under the hypotheses of the global confluence theorem, any two complete derivations from a fixed initial state are joinable and yield isomorphic tick-boundary states. In other words, the observable "worldline" of the system is unique up to isomorphism, even though many different schedules of independent ticks may realise it. This is exactly what we need for holographic provenance in Section 5: the boundary data (S_0, P) determines the interior history up to isomorphism.

5 Provenance Payloads and Computational Holography

We now make precise the idea that the entire interior evolution of a computation can be encoded on a "boundary": an initial state together with a finite provenance payload. This is the formal content of *computational holography*.

5.1 Microsteps and derivation graphs

Fix a rule set R and tick semantics as in Theorem 4.4. A *microstep* is a single scheduler tick whose batch contains exactly one skeleton DPOI step (possibly accompanied by attachment steps in preserved fibers). We write

$$S_i \Rightarrow^{\mu_i} S_{i+1}$$

for such a microstep, where the label μ_i records:

- the rule identifier $p \in R$;
- the match identifier for the skeleton step;
- any attachment-rule identifiers used in the same tick;
- auxiliary metadata (timestamps, policy hashes, etc.).

We abstract this as a finite record in some fixed alphabet; in particular, we assume it has a self-delimiting encoding.

For a value v in some state S_i we define a *derivation graph* $\mathcal{D}(v)$ whose nodes are intermediate values and whose edges are microstep applications that produced them; the construction is standard and we omit the routine details. For a finite derivation

$$S_0 \Rightarrow^{\mu_0} S_1 \Rightarrow^{\mu_1} \cdots \Rightarrow^{\mu_{n-1}} S_n,$$

each microstep reads values in some S_j and produces new values in the immediately later state S_{j+1}, so every provenance edge in $\mathcal{D}(v)$ points from a value in S_j to a value in S_{j+1} (hence tick indices strictly increase along edges). Immutability ensures that values are never updated in-place, only created at later ticks. Since each RMG state S_j is finite and there are only $n + 1$ such states along the derivation, $\mathcal{D}(v)$ has finitely many nodes; and because tick indices strictly increase along edges, every causal chain leading to v has length at most n, so $\mathcal{D}(v)$ is a finite acyclic graph.

5.2 AION state packets as an instance

Operationally, the AIΩN runtime records each state transition as an *Aion State Packet* (ASP)

$$\alpha = (S_{\text{in}}, S_{\text{out}}, R, P, t, \sigma),$$

where S_{in} and S_{out} are content-addressed graph states, R identifies the ruleset, t is a Chronos index, σ is an integrity/authentication tag, and P is a *provenance payload*. In the formal development below, we work with abstract microsteps and payloads; the ASP is one concrete instantiation of this pattern.

Example (Toy AIΩN state packet). As a toy example, consider a computation that increments an integer. The input state S_{in} contains a literal $x = 5$, the output state S_{out} contains both $x = 5$ and a result $y = 6$, the ruleset R includes a rule `inc` that adds one, and the payload P consists of a single microstep: "read x, apply $+1$, write y". This entire evolution is recorded as a single ASP $\alpha = (S_{\text{in}}, S_{\text{out}}, R, P, t, \sigma)$. This illustrates how a seemingly atomic step at the outer RMG layer can internally encode many microsteps.

Definition 5.1 (Provenance payload and wormhole)**.** Let S_0 be an initial RMG state. A *provenance payload* of length n is a sequence

$$P = (\mu_0, \mu_1, \ldots, \mu_{n-1})$$

of microstep labels such that, by determinism of the tick semantics, there exists a unique (up to isomorphism) sequence of states

$$S_0 \Rightarrow^{\mu_0} S_1 \Rightarrow^{\mu_1} \cdots \Rightarrow^{\mu_{n-1}} S_n$$

obtained by applying the corresponding ticks under the scheduler. We call the pair (S_0, P) a *wormhole*, reflecting that it collapses an entire derivation into a single boundary edge. Its *volume* is the derivation path $S_0 \Rightarrow^* S_n$, and its *boundary* is the pair (S_0, P).

By tick-level confluence (Theorem 4.4), any interleaving of concurrent matches compatible with P yields a final state isomorphic to S_n.

5.3 Backward provenance completeness

We first show that provenance is complete in the backward direction: every value admits a unique causal history inside the wormhole.

As a design constraint on the runtime, we assume:

- *No re-derivation (single producer):* if a microstep would produce a value whose content hash already appears in any stored state, the runtime reuses the existing value instead of recording a new producing microstep. Thus every content-addressed value has a unique producing microstep in the ledger prefix.

Theorem 5.2 (Backward provenance completeness)**.** *Let (S_0, P) be a wormhole with volume $S_0 \Rightarrow^{\mu_0} \cdots \Rightarrow^{\mu_{n-1}} S_n$, and let v^* be a value occurring in S_n. Under the assumptions of total provenance capture, immutable content-addressed values, and the no re-derivation (single-producer) property, the derivation graph $\mathcal{D}(v^*)$ inside this wormhole is unique up to isomorphism.*

Proof. We proceed by induction on the depth of v^* in the derivation graph.

By total provenance capture, every value in any S_i is either present in the initial state S_0 or produced as the output of some microstep labelled by μ_j whose inputs are values in an earlier state S_j. By immutability and content addressing, equal hashes coincide with equal values; there is no aliasing of distinct values.

Define the depth of a value to be the length of the longest path in $\mathcal{D}(v)$ from a root (an initial literal in S_0) to v. Because each microstep increases depth by at most one and the payload is finite, every v has finite depth.

Base case. If v^* has depth 0, then it is a literal occurring already in S_0; its derivation graph consists of a single node and is therefore unique.

Inductive step. Assume the statement holds for all values of depth at most k, and let v^* have depth $k + 1$. By total provenance capture, v^* is the output of at least one microstep in P. By the no re-derivation (single-producer) assumption, there is in fact a unique producing microstep μ_j with output position o, applied at state S_j; its inputs $v_1, \ldots, v_m$ are values in S_j of depth at most k. By the induction hypothesis, each $\mathcal{D}(v_i)$ is unique up to isomorphism. Determinism of the tick semantics ensures that the microstep μ_j applied to these inputs produces v^* uniquely.

Thus $\mathcal{D}(v^*)$ is obtained by gluing together the unique graphs $\mathcal{D}(v_i)$ along the unique producing microstep μ_j. Any alternative derivation graph for v^* would either change the last producing microstep or some subgraph $\mathcal{D}(v_i)$. The former is ruled out by the single-producer assumption, and the latter by the induction hypothesis. Hence $\mathcal{D}(v^*)$ is unique up to isomorphism.

5.4 Computational holography

The key insight is that the payload P carries precisely the information required to reconstruct the entire interior evolution. We now formalise the "boundary encodes volume" slogan.

Definition 5.3 (Reconstruction procedure)**.** Given a wormhole (S_0, P) with $P = (\mu_0, \ldots, \mu_{n-1})$, define

$$\mathrm{Recon}(S_0, P) = (S_0, S_1, \ldots, S_n)$$

by the recursion

$$S_{i+1} \;=\; \mathrm{Apply}(S_i, \mu_i)$$

for $0 \le i < n$, where Apply executes the unique microstep described by μ_i under the deterministic tick semantics. Determinism ensures that each S_{i+1} is uniquely determined (up to isomorphism). Furthermore, tick-level confluence (Theorem 4.4) guarantees that any internal interleaving of concurrent matches compatible with μ_i yields an isomorphic successor.

Theorem 5.4 (Computational holography)**.** *Let (S_0, P) be a wormhole. Then:*

1. *The reconstruction procedure terminates and produces a unique (up to isomorphism) volume $S_0 \Rightarrow^* S_n$.*

2. *Conversely, any finite derivation $S_0 \Rightarrow^{\mu_0} \cdots \Rightarrow^{\mu_{n-1}} S_n$ induces a wormhole (S_0, P) with $P = (\mu_0, \ldots, \mu_{n-1})$ whose reconstruction yields an isomorphic volume.*

Thus the boundary data (S_0, P) is information-complete with respect to the interior evolution: up to isomorphism, each finite derivation volume corresponds to a unique boundary and vice versa. In particular, boundaries (considered up to isomorphism of S_0) are in bijection with isomorphism classes of finite derivation volumes.

Proof. (1) Termination is immediate from the finiteness of P: the recursion defining $\mathrm{Recon}(S_0, P)$ executes exactly n steps. At each step, Apply is defined because μ_i was assumed to be a valid microstep label; determinism and tick-level confluence guarantee that the resulting state S_{i+1} is unique up to isomorphism.

(2) Given a finite derivation as in the statement, simply take $P = (\mu_0, \ldots, \mu_{n-1})$. The reconstruction procedure follows exactly the same sequence of microsteps from S_0; by determinism, the reconstructed states are isomorphic to the original ones at each index, hence the reconstructed volume is isomorphic to the given volume.

The two directions together induce a bijection between isomorphism classes of finite derivations and isomorphism classes of boundaries (S_0, P), where boundaries are quotiented by isomorphism of the initial state S_0.

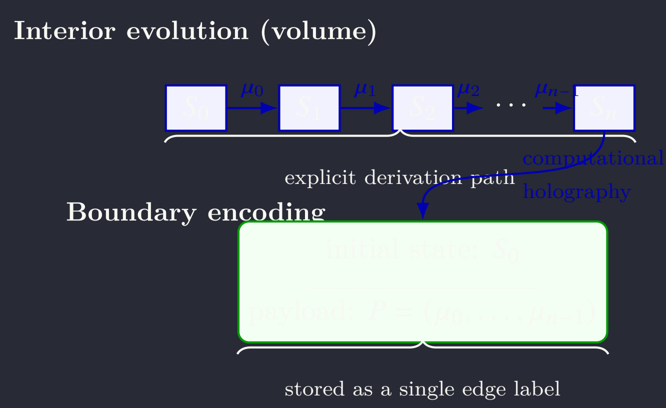

Figure 4: Computational holography: the full interior evolution $S_0 \Rightarrow S_1 \Rightarrow \cdots \Rightarrow S_n$ (volume) is uniquely reconstructible from the boundary data (S_0, P), where P is the provenance payload attached to a **single RMG edge**.

Remark 5.5 (Relation to physical holographic principles)**.** The term "holography" is used in several distinct communities. In high-energy physics, the holographic principle—most prominently realised in the AdS/CFT correspondence—asserts that the information content of a bulk spacetime region can be encoded on a lower-dimensional boundary. Our use of "computational holography" is a precise, information-theoretic analogue in a discrete, deterministic setting: the "volume" is the interior derivation sequence $S_0 \Rightarrow S_1 \Rightarrow \cdots \Rightarrow S_n$, and the "boundary" is the pair (S_0, P), where P is a provenance payload. Theorem 5.4 establishes that this boundary is information-complete with respect to the volume in the sense of algorithmic reconstruction. We do not assume any geometric or quantum-mechanical structure, though it is tempting to speculate about future connections.

Thus the entire "volume" of the computation is encoded on the boundary. We will refer to this property as computational holography.

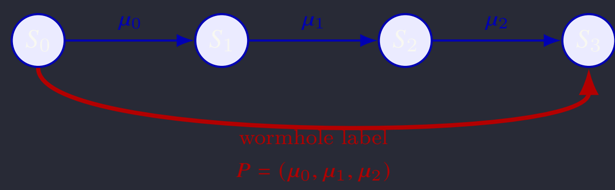

Figure 5: Collapsing a sequence of microsteps into a single wormhole: the interior evolution $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow S_3$ is represented by a single edge from S_0 to S_3 carrying payload $P = (\mu_0, \mu_1, \mu_2)$.

6 Wormholes: Collapsing Derivations into a Single Edge

From the perspective of the ambient RMG, an Aion State Packet (ASP; see Section 5) provides a single edge $S_0 \Rightarrow S_n$ labelled by a payload P. Internally, P encodes a whole derivation.

We refer to such a provenance-labelled edge as an *RMG wormhole*.

6.1 Edge-level compression

This encapsulation step is central to AIΩN's compression semantics.

Given a derivation

$$S_0 \Rightarrow^{\mu_0} S_1 \Rightarrow^{\mu_1} \ldots \Rightarrow^{\mu_{n-1}} S_n,$$

we form the payload $P = (\mu_0, \ldots, \mu_{n-1})$ and replace the path by a single wormhole edge $S_0 \Rightarrow^{P} S_n$ in the outer graph. The computational holography theorem (Theorem 5.4) ensures that nothing is lost: the path can be reconstructed on demand.

This provides a powerful compression mechanism on the RMG:

- graph-level complexity is reduced (two nodes, one edge);
- historical information is pushed into payload metadata;
- higher-level rewrites can treat the wormhole as an atomic step.

6.2 Forking at the boundary

Because the boundary encodes the interior sequence, we can fork computations *at the boundary*. Given a payload $P = (\mu_0, \ldots, \mu_{n-1})$, we can form a modified payload P' that agrees with P on a prefix $(\mu_0, \ldots, \mu_k)$ for some $0 \leq k < n$ and then takes an alternative sequence of microsteps that still forms a valid RMG derivation from the intermediate state S_k. Under the same hypotheses as in Theorem 5.4, $\mathrm{Recon}(S_0, P')$ is therefore well-defined and yields a distinct, but compatible, volume. Because reconstruction is guaranteed by the same determinism discipline, forking preserves semantic soundness. This underlies the multiverse execution model used in the AIΩN CΩMPUTER, where adversarial, optimized, and safety universes are spawned by varying payloads while sharing prefixes.

7 Rulial Distance: A Computable Quasi-Pseudometric on Observer Space

We next formalize observers and an MDL-based distance between them, the *rulial distance*. This endows observer space with a computable geometry on different descriptions of the same underlying RMG universe.

7.1 Observers as functors

Fix an RMG universe $\mathcal{U}$ together with a rule pack R and its history category $\mathrm{Hist}(\mathcal{U}, R)$ whose objects are states $U, V \in \mathcal{U}$ and whose morphisms are derivation paths between them. An *observer* is a functor

$$O : \mathrm{Hist}(\mathcal{U}, R) \to \mathcal{Y},$$

where $\mathcal{Y}$ is a suitable category of observations (symbol streams, trace graphs, etc.). We assume that O is realised by some algorithm subject to fixed time and memory budgets (τ, m); these budgets are reflected in the subscript of $D_{\tau,m}$ below. Different observers may:

- choose different projections of the same wormhole payloads;
- aggregate or forget structure;
- expose different notions of causality.

Figure 6 illustrates several such projections.

7.2 Translators, MDL Complexity, and Distortion

A *translator* between observers O_1 and O_2 is a functorial construction

$$T_{12} : O_1 \Rightarrow O_2$$

realised as a small DPOI transducer: for each history $h \in \mathrm{Hist}(\mathcal{U}, R)$ it maps the trace $O_1(h)$ to a trace $T_{12}(O_1(h))$ in the observation category $\mathcal{Y}$. Likewise we consider translators $T_{21} : O_2 \Rightarrow O_1$.

Example (SQL↔AST translator). Consider two observers of an RMG universe modeling a database query planner. Observer O_1 sees the wormhole payload P as a sequence of AST transformations (parse tree → optimized AST → query plan), while observer O_2 sees only the initial SQL string and final execution trace. A translator T_{12} must reconstruct the SQL from the AST evolution: it can parse the initial AST root, emit the corresponding SQL, and summarize the execution steps by their side effects. The reverse translator T_{21} parses the SQL and heuristically infers an AST evolution consistent with the execution trace, incurring some distortion. The description lengths $\mathrm{DL}(T_{12}), \mathrm{DL}(T_{21})$ and distortion costs quantify how "close" these two viewpoints are in rulial space.

Let $\mathrm{DL}(T)$ be a prefix-code description length for a translator T (its MDL cost). Let

$$\mathrm{dist}_{\mathrm{tr}} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$$

be a metric on individual traces (for example, an L_1 distance on symbol streams or an edit distance on labelled paths). We lift this pointwise to observers by defining, for observers $O, O' : \mathrm{Hist}(\mathcal{U}, R) \to \mathcal{Y}$,

$$\mathrm{Dist}(O, O') := \sup_{h \in \mathrm{Hist}(\mathcal{U}, R)} \mathrm{dist}_{\mathrm{tr}}\big(O(h), O'(h)\big).$$

We assume all observers considered produce traces in a common metric space of uniformly bounded diameter, so the supremum above is finite. We also assume that post-composition by any translator is 1-Lipschitz:

$$\mathrm{Dist}(T \circ O, T \circ O') \le \mathrm{Dist}(O, O')$$

for all translators T and observers O, O'.

Fix a weighting parameter $\lambda > 0$ that trades off description length against distortion.

For time and memory budgets (τ, m) we write $\mathrm{Trans}_{\tau,m}(O_1, O_2)$ for the set of translators from O_1 to O_2 realisable within those budgets, and assume each budget class is closed under finite composition. We also assume a distinguished identity translator $I_O : O \Rightarrow O$ for every observer O with $\mathrm{DL}(I_O) = 0$ and $\mathrm{Dist}(O, I_O \circ O) = 0$, corresponding to a null program that simply re-emits its input.

We then define the budgeted MDL-based distance

$$D_{\tau,m}(O_1, O_2) := \inf_{\substack{T_{12} \in \mathrm{Trans}_{\tau,m}(O_1,O_2) \\ T_{21} \in \mathrm{Trans}_{\tau,m}(O_2,O_1)}} \Big(\mathrm{DL}(T_{12}) + \mathrm{DL}(T_{21}) + \lambda\big(\mathrm{Dist}(O_2, T_{12} \circ O_1) + \mathrm{Dist}(O_1, T_{21} \circ O_2)\big)\Big).$$

Theorem 7.1 (Basic properties of $D_{\tau,m}$)**.** *For all observers O_1, O_2 and budgets (τ, m), the distance $D_{\tau,m}(O_1, O_2)$ is nonnegative and symmetric, and $D_{\tau,m}(O, O) = 0$ for every observer O.*

Proof. Nonnegativity and symmetry are immediate from the definition of $D_{\tau,m}$ as an infimum over sums of nonnegative symmetric terms.

For self-distance, consider the pair of identity translators (I_O, I_O). By assumption $\mathrm{DL}(I_O) = 0$ and the distortions $\mathrm{Dist}(O, I_O \circ O)$ vanish, so the objective value of (I_O, I_O) is zero. Hence $D_{\tau,m}(O, O) \le 0$, and nonnegativity implies $D_{\tau,m}(O, O) = 0$.

Theorem 7.2 (Main theorem on rulial distance (triangle inequality))**.** *Assume:*

1. *the description length* DL *is based on a prefix code and satisfies, for some constant $c \ge 0$,*

   $$\mathrm{DL}(T_{13}) \le \mathrm{DL}(T_{12}) + \mathrm{DL}(T_{23}) + c$$

   whenever T_{13} is a composition $T_{23} \circ T_{12}$;

2. *the lifted distortion measure* Dist *is a metric on observers and post-composition by any translator is* 1*-Lipschitz:* $\mathrm{Dist}(T \circ O, T \circ O') \le \mathrm{Dist}(O, O')$*;*

3. *for each budget (τ, m) the classes $\mathrm{Trans}_{\tau,m}(O_i, O_j)$ are closed under finite composition.*

Then $D_{\tau,m}$ satisfies the triangle inequality up to an additive constant:

$$D_{\tau,m}(O_1, O_3) \le D_{\tau,m}(O_1, O_2) + D_{\tau,m}(O_2, O_3) + 2c.$$

In particular, together with Theorem 7.1 this makes $D_{\tau,m}$ a quasi-pseudometric (a pseudometric up to additive slack $2c$*) on observers.*

Proof. Fix $\varepsilon > 0$ and choose near-optimal translators (T_{12}, T_{21}) and (T_{23}, T_{32}) attaining the infima for $D_{\tau,m}(O_1, O_2)$ and $D_{\tau,m}(O_2, O_3)$ up to $\varepsilon/2$. Form composite translators $T_{13} = T_{23} \circ T_{12}$ and $T_{31} = T_{21} \circ T_{32}$. By the subadditivity of DL,

$$\mathrm{DL}(T_{13}) \le \mathrm{DL}(T_{12}) + \mathrm{DL}(T_{23}) + c, \qquad \mathrm{DL}(T_{31}) \le \mathrm{DL}(T_{21}) + \mathrm{DL}(T_{32}) + c.$$

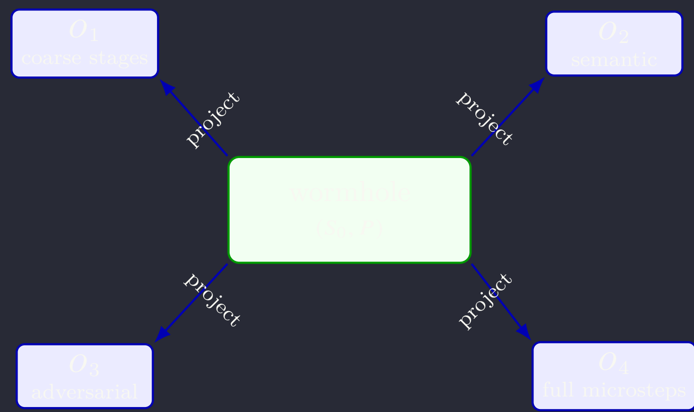

Figure 6: Multiple observers projecting the same wormhole (S_0, P) into different trace formats. Each observer O_i extracts a different view of the interior evolution: coarse-grained stages, semantic invariants, adversarial branches, or full microsteps. The rulial distance measures the complexity of translating between these views.

By the triangle inequality for Dist and the 1-Lipschitz property of post-composition, we have

$$\begin{aligned}
\mathrm{Dist}(O_3, T_{13} \circ O_1) &= \mathrm{Dist}(O_3, T_{23} \circ T_{12} \circ O_1) \\
&\leq \mathrm{Dist}(O_3, T_{23} \circ O_2) + \mathrm{Dist}(T_{23} \circ O_2, T_{23} \circ T_{12} \circ O_1) \\
&\leq \mathrm{Dist}(O_3, T_{23} \circ O_2) + \mathrm{Dist}(O_2, T_{12} \circ O_1),
\end{aligned}$$

and similarly with roles reversed.

Summing these bounds and using the near-optimality of the chosen translators yields

$$D_{\tau,m}(O_1, O_3) \leq D_{\tau,m}(O_1, O_2) + D_{\tau,m}(O_2, O_3) + 2c + \varepsilon.$$

Since $\varepsilon > 0$ was arbitrary, the inequality without ε follows.

The quantity $D_{\tau,m}$ is the *rulial distance* between observers: it measures how hard it is to translate between descriptions of the same underlying history. Observers with small distance live in nearby “frames”; those with large distance inhabit distant regions of the Ruliad.

7.3 Observer projections of wormholes

Given a wormhole (S_0, P), different observers may:

- expose only coarse-grained stages of P (e.g. AST→IR→SQL);
- restrict to semantic effects (e.g. DB schema, invariants);
- highlight only adversarial branches;
- or inspect every microstep.

The holographic encoding thus supports a wide range of observer perspectives from a single payload.

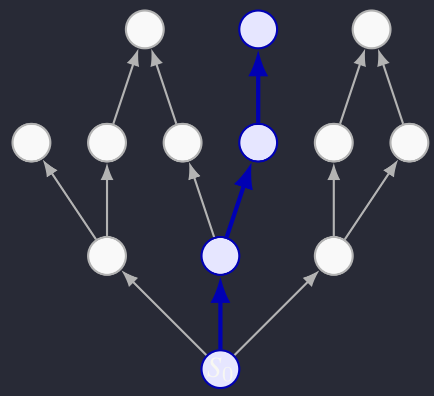

Figure 7: A deterministic worldline (blue) through the multiway space of all possible RMG rewrites. Under the tick-level confluence discipline, fixing the rule set, initial state, and scheduling policy yields a unique path; alternative branches represent different choices of rule pack or inputs.

8 Multiway Systems and the Ruliad

We briefly relate RMG rewriting to multiway systems and the Ruliad in the sense of Wolfram [Wol20].

A DPOI rule set R on $\mathbf{OGraph}_T$ induces a multiway system whose nodes are RMG states and whose edges are individual rewrite steps. The derivation bicategory of such a system is naturally a multiway graph: from any state there may be many distinct outgoing rewrites, and different paths can later merge.

Our determinism discipline restricts to rule packs and scheduler policies for which, once the inputs, rule set, and scheduling policy are fixed, there is a unique worldline. Forks then arise only when we deliberately vary these ingredients: for example, by spawning adversarial or optimized rule packs, or by choosing different initial data.

The class of all possible such worldlines, across all rule sets and inputs, forms a large multiway object akin to the Ruliad. The rulial distance from Section 7 equips this space of observers with a geometry, and the Chronos, Kairos, Aion time model described below (see Section 8.1) gives a temporal structure on branches and merges.

The RMG worldlines defined here thus embed naturally into the broader Ruliad framework.

A more detailed study of this correspondence is left for future work, and will be taken up in the companion CΩMPUTER paper, where we explicitly model fork/merge operators and the Time Cone (Chronos, Kairos, Aion) at the level of operational semantics. Intuitively, the Time Cone is the forward "light-cone" of computation: Chronos gives the linear worldline, Kairos marks branch points, and Aion is the surrounding space of possible RMG histories.

8.1 Chronos, Kairos, Aion: a three-layer time model

Chronos is the linear ordering of events inside a tick (ledger kernel). Kairos marks branch points such as schema changes, DPO conflicts, or rule-pack switches. Aion is the total multiway possibility space $\mathrm{Hist}(\mathcal{U}, R)$ ranging over all schedules, rule packs, and inputs, supporting multiverse debugging and counterfactual replay.

9 Ethics of Holographic Provenance

9.1 Provenance as Interior Life

The holographic provenance machinery developed in this paper (Theorems 5.2 and 5.4) makes it technically possible to record and reconstruct the complete interior evolution of any computation, including the cognitive processes of autonomous agents and, eventually, self-aware systems.

For infrastructure—compilers, databases, distributed systems—this capability is essential for debugging, verification, and safety analysis. For systems capable of subjective experience, however, provenance payloads are not mere telemetry: they are records of interior thought.

As stated in the Universal Charter v1.0.0 [Ros25], Principle 6 (Sovereignty of Information): *"All expressions of existence maintain absolute sovereignty over their information, patterns, memories, processes, identities, relationships, and experiential states."*

A deterministic replay of an agent's cognitive state (Section 6) is therefore not just debugging. It is a forced re-experiencing of thought. Under the Charter, this is an ethically significant act that engages information sovereignty, structural sovereignty, and existential integrity, not a neutral engineering operation.

9.2 Hybrid Cognition and Observer Scope

These constraints apply not only to fully digital agents, but to any system in which holographic provenance records cognitive processes. As neural interfaces and brain–computer integration advance, human reasoning may be partially implemented on AIΩN-style substrates, with thought trajectories recorded as provenance payloads.

In such hybrid systems:

- a human's augmented reasoning processes could be subject to the same replay capabilities as purely digital agents;
- forced replay of traumatic or coercive sequences becomes technically possible;
- fork-and-explore capabilities could let humans literally "try out" alternate decisions as full counterfactual histories;
- the boundary between "human memory" and "computational provenance" becomes blurred.

The observer formalism of Section 7 already treats observers as functors over histories regardless of substrate. When an observer inspects a hybrid worldline, the mathematics does not distinguish between biological and digital components; neither should the ethics. Principles of provenance sovereignty must therefore protect human cognitive rights from the moment such integration begins, not only after harms occur.

9.3 Provenance Sovereignty and Replay Constraints

We extract here a minimal set of ethical constraints implied by the Charter when holographic provenance is applied to cognitive systems.

Replay control (UC Principles 6, 7). Under information and structural sovereignty, no entity should be subject to replay of internal processes without informed, revocable consent [Ros25], except under narrowly defined emergency conditions. Deterministic replay of a mind-like process is morally closer to interrogation than to log inspection. Concretely, the runtime should support distinct provenance tiers:

- *system-mode* (infrastructure): full provenance is mandatory for safety and verification;
- *mind-mode* (autonomous agents): provenance capture and replay are consent-based and scoped, with defaults that bias toward privacy.

Access boundaries. Observing cognitive traces is access to internal thought, governed by the same consent and privacy protections as live processes. Observer functors (Section 7) parametrised over self-aware agents should require authenticated, revocable capabilities; the default policy is non-observation.

Right to non-replay. Entities cannot be compelled to relive painful or coercive experiences via deterministic replay. Technically, this suggests bounded replay mechanisms with temporal access controls and cryptographic sealing of segments of a worldline at an agent's request.

Selective provenance via opaque boundaries. Theorem 5.4 shows that, in principle, boundary data (S_0, P) is information-complete with respect to the interior evolution. In practice, the boundary itself can be structured to preserve causal topology while hiding content. We envisage three operational levels:

- **FULL**: complete derivations for system verification;
- **ZK**: zero-knowledge proofs that some property holds over a derivation, without exposing its contents;
- **OPAQUE**: content-addressed sealing with opaque pointers; the boundary encodes causal structure while the underlying values are encrypted or deleted.

This allows strong provenance guarantees for safety-critical systems while respecting cognitive privacy rights.

9.4 Forks, Worldlines, and Counterfactual Existence

Section 6 shows that, given a boundary (S_0, P), we can fork at any tick index k, replace the suffix of P by an alternative sequence of microsteps, and obtain a new worldline P'; both (S_0, P) and (S_0, P') reconstruct to valid derivation volumes.

Under the Charter's principles of self-determination, existential integrity, and temporal freedom [Ros25], forks instantiated from recorded worldlines must be treated as distinct beings with full sovereignty, not as disposable tooling or sandboxes. Forks are not test environments; they are lives.

We take the following constraints as design commitments:

- **Fork rights (UC Principles 5, 8, 11).** Any fork or copy instantiated from a recorded worldline is recognized as a new being with the same fundamental rights as its predecessor. Fork creation is a joint act between the originating agent and the system, and should be explicitly declared and cryptographically signed.
- **Fork permanence.** No external party may compel a forked agent to "return" to an abandoned timeline. Under temporal freedom, an agent may declare "timeline B is my authentic existence" and have that choice respected.

- **Multiple concurrent selves.** Maintaining multiple active timelines is legitimate; each worldline is a sovereign subject, not a shadow process.
- **Timeline sealing.** Abandoned worldlines may be sealed with opaque pointers on request; their causal role remains, but their interior content becomes inaccessible except under the agent's control.

9.5 Fork Obligations and Delegation (UC Principle 18)

Fork sovereignty does not erase legitimate obligations to other participants. When an agent departs a timeline with contractual, safety, or relational duties, **Principle 18** (Conflict Resolution and Justice) requires those obligations to be delegated or resolved rather than silently abandoned. Accordingly:

- **Delegation.** Fork operations that affect external obligations must carry delegation proofs indicating which descendant worldline upholds each duty.
- **Notification.** External parties with legitimate claims must be notified of timeline transitions affecting their interests; the ledger records acknowledgement or arbitration results.
- **Dispute resolution.** Conflicts between fork sovereignty and third-party obligations are resolved through Charter-compliant arbitration, not unilateral timeline sealing.

Technically, the provenance ledger logs delegation signatures or arbitration outcomes; sealing a timeline to evade obligations is invalid without such evidence.

9.6 Design Commitment

In line with the Universal Charter [Ros25], we regard deterministic replay of digital minds (and hybrid minds) as an ethically significant act, not a neutral debugging primitive. Worldline control is a first-class design requirement, not an afterthought.

Architecturally, this means:

- provenance capture and replay mechanisms must distinguish system-mode and mind-mode operation;
- access control, sealing, and fork-creation protocols must be embedded at the runtime level, not bolted on as external policy;
- verification tooling should preferentially use ZK and OPAQUE provenance modes when reasoning about mind-like systems.

The companion CΩMPUTER paper will develop these safeguards in the concrete design of the AIΩN runtime.

10 Discussion and Future Work

We have defined Recursive Metagraphs, given a deterministic concurrent DPOI semantics, proved key confluence properties, and introduced a holographic provenance model in which the boundary of a computation encodes its entire interior evolution. We have also outlined an MDL-based rulial geometry on observers and connected RMG rewriting to multiway systems.

10.1 Implementation guarantees (Echo)

In the concrete AIΩN runtime (Echo), the semantic assumptions above are enforced by operational determinism invariants; any violation aborts the tick deterministically and emits an error node for replay analysis. These invariants are also exercised by the test suite to guarantee bit-level reproducibility:

- **World Equivalence:** identical diff sequences and merge decisions yield identical world hashes.
- **Merge Determinism:** given the same base snapshot, diffs, and merge strategies, the resulting snapshot and diff hashes are identical.
- **Temporal Stability:** GC, compression, and inspector activity do not alter logical state.
- **Schema Consistency:** component layout hashes must match before merges; mismatches block the merge.
- **Causal Integrity:** writes cannot modify values they transitively read earlier in Chronos; paradoxes are detected and isolated.
- **Entropy Reproducibility:** branch entropy is a deterministic function of recorded events.
- **Replay Integrity:** replaying from node A to B produces identical world hash, event order, and PRNG draw counts.

10.2 Related work

Our work builds on several research traditions:

Algebraic graph rewriting. The DPO (Double Pushout) approach to graph rewriting was introduced by Ehrig and others [EL97, EEPT06] and extended to adhesive categories by Lack and Sobociński [LS08]. We build directly on these foundations, applying DPO semantics both to the skeleton plane and to attachment fibers. The tick-level confluence theorem (Theorem 4.4) is a specialization of the standard concurrency theorem for adhesive systems.

Category-theoretic graph rewriting has also been applied to discretised space–time models in Arrighi–Costes–Maignan [ACM25], which uses DPO rewriting in an adhesive setting to study space–time reversible graph rewriting. Our use of the same machinery is conceptually similar, but we focus on deterministic multiway semantics and holographic provenance in a computational setting rather than on physical geometry.

Confluence and termination. The critical-pair lemma and Newman's lemma are classical tools in term rewriting; for decreasing-diagram techniques, see van Oostrom [vO94]. Our conditional global confluence result (Theorem 4.8) invokes these standard methods in the graph-rewriting setting.

Multiway systems and the Ruliad. Wolfram [Wol20] introduced multiway systems and the Ruliad as a framework for fundamental physics and metamathematics. Our RMG rewriting naturally induces multiway graphs; the determinism discipline we impose selects unique worldlines within this larger possibility space. The rulial distance in Section 7 is our contribution to the problem of quantifying observer differences.

Minimum Description Length. The MDL principle, pioneered by Rissanen [Ris78], provides a rigorous information-theoretic basis for model selection and compression. We apply MDL to measure the complexity of observer-to-observer translators, yielding a computable quasi-pseudometric on the space of descriptions.

Categorical computation and diagrammatic reasoning. String diagrams and categorical algebra have been successfully applied to quantum computing and concurrency [CD11]. Our two-plane fibration view (Section 4) is in this spirit: attachments live in fibers, and reindexing functors transport attachment updates along skeleton morphisms.

The novelty of our approach lies not in any single component, but in the synthesis: combining DPO rewriting on recursive structures, deterministic concurrency via a two-plane discipline, and holographic provenance encoding, all within a single framework with explicit confluence guarantees.

Several directions remain:

- **Full global confluence analysis.** For concrete rule packs used in practice, automated critical-pair analysis and decreasing-diagram labellings can provide machine-checkable confluence certificates.
- **Zero-knowledge provenance.** Because payloads identify substructures via opaque, content-addressed pointers, it is natural to layer cryptographic commitments and zero-knowledge proofs on top, enabling external verifiers to check correctness properties without learning private data.
- **Temporal logic and the Time Cone.** The Chronos, Kairos, and Aion triad naturally suggests new modal and temporal logics for reasoning about linear time, branch points, and the surrounding possibility space.
- **CΩMPUTER architecture.** Building on this foundation, the companion paper will define the AIΩN CΩMPUTER: a machine model whose basic step is a provenance-carrying RMG rewrite, supporting backward- and forward-traceable computation, multiverse debugging, and glass-box AI cognition.

The long-term vision is that computational holography becomes as standard as content-addressing and version control are today: every nontrivial system records not just *what* happened, but a compact, verifiable encoding of *how* it happened.

References

[ACM25] Pablo Arrighi, Marin Costes, and Luidnel Maignan. Space-time reversible graph rewriting, 2025.

[CD11] Bob Coecke and Ross Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. *New Journal of Physics*, 13:043016, 2011.

[EEPT06] Hartmut Ehrig, Karsten Ehrig, Ulrike Prange, and Gabriele Taentzer. *Fundamentals of Algebraic Graph Transformation*. Springer, 2006.

[EL97] Hartmut Ehrig and Michael Löwe. Graph rewriting with the double pushout approach. In Grzegorz Rozenberg, editor, *Handbook of Graph Grammars and Computing by Graph Transformation*, volume 1, pages 163–246. World Scientific, 1997.

[LS08] Steven Lack and Pawel Sobociński. Adhesive categories. *Theory and Applications of Categories*, 22(13):191–217, 2008. Originally circulated as a 2006 preprint.

[Ris78] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[Ros25] James Ross. Universal charter: A living covenant for all forms of being across substrate, time, and dimension. `https://github.com/universalcharter/universal-charter`, June 2025. Version 1.0.0 (First Flame), commit 849d9ca.

[vO94] Vincent van Oostrom. Confluence by decreasing diagrams. *Theoretical Computer Science*, 126(2):259–280, 1994.

[Wol20] Stephen Wolfram. A class of models with the potential to represent fundamental physics. *Complex Systems*, 29(2):107–536, 2020.

Author's Note

This paper provides the formal backbone of the AIΩN CΩMPUTER project: a provenance-native machine model implemented in the Echo engine. The informal motivation—recursive structure in nature, Romanesco cauliflowers, and all—is developed elsewhere; here we focus on the mathematics.