The Imagination Machine XXIV: Networked Completion, Event-Based Linearization, and Relational Structure

Mark Tracy

March 2026

Abstract

The Imagination Machine series develops a framework for embedded epistemic systems in which knowledge arises through iterative completion of partial structure under constraint. Earlier papers formalized compression and extension, graph-theoretic realizations of relational structure, and the interpretation of scientific inquiry as structured incompletion followed by completion.

The present paper extends this framework to networks of bounded systems. Individual units are modeled as genus-0 bodies embedded in $\mathbb{R}^3$, while nontrivial topology arises at the level of collective dynamics. We formalize networked completion, show how global relational structure emerges across such networks, and define distributed projection of that structure across nodes.

We further introduce an event-based formulation of neural computation, in which spike events impose local linear constraints on global network state. Under this view, neural systems approximate nonlinear dynamics through distributed, event-driven linearization.

This situates biological systems, neural computation, and scientific inference within a single operational principle: the distributed completion of partial structure under constraint across networks of bounded systems.

1 Introduction

The Imagination Machine framework begins from the condition of embeddedness. An epistemic system must construct a model of the world from within the world it models. Because such a system never has access to a complete description, it must operate on partial structure.

In early papers, this process was formalized as a cycle of *compression* and *extension*. Observations are compressed into relational representations, and these representations are extended to predict missing or future structure.

Subsequent papers showed that this process admits a graph-theoretic realization, in which knowledge is represented as a dynamically evolving relational graph. The extension step was further interpreted through analogical reasoning and simplicial horn-filling: partial structures are completed into coherent wholes.

More recent work reinterpreted scientific experimentation as a form of structured incompletion. By removing constraints from a hypothesized structure and observing how the system evolves, one probes the validity of the underlying relational model.

The present paper extends this line of thought by addressing the following question:

> How does completion operate not only within a single system, but across a network of bounded systems?

2 Locally Bounded Systems

Definition 1 (Locally Bounded System)**.** *A* locally bounded system *is a connected physical subsystem $B \subset \mathbb{R}^3$ whose boundary ∂B is a compact genus-0 surface.*

Remark 1. *This formalizes the observation that biological systems such as neurons and organisms are locally bounded bodies. The statement is topological: the boundary is equivalent to a sphere, though the geometry may be highly irregular.*

Definition 2 (Node)**.** *A* node *is a locally bounded system equipped with internal state and interaction channels:*

$$N_i = (B_i, S_i, U_i, Y_i).$$

Remark 2. *The distinction between physical topology and representational topology is essential. While nodes are genus-0 as physical objects, their collective dynamics may realize nontrivial manifolds.*

3 From Single-System Completion to Networked Completion

In earlier work, completion was defined for a single system: a partial relational structure is extended into a coherent whole.

We now generalize this to networks.

Definition 3 (Network)**.** *A network is a graph $G = (V, E)$ with nodes V and edges E representing interactions.*

Definition 4 (Partial Configuration)**.** *A* partial configuration *assigns incomplete relational structure to a subset of the system.*

Definition 5 (Networked Completion Operator)**.** *A mapping*

$$\Phi_G : \mathcal{P} \to \mathcal{C}$$

that assigns coherent global states to partial configurations.

Definition 6 (Distributed Completion)**.** *A completion is distributed if it cannot be determined by any single node, but emerges through network dynamics.*

Theorem 1 (Networked Completion)**.** *If node dynamics depend on relational inputs and network coupling propagates constraints, then coherent global completions emerge across the network.*

4 Emergence of Global Relational Structure

Definition 7 (Global Relational Structure)**.** *A relational structure that depends on interactions across nodes and cannot be reduced to independent node-level descriptions.*

Definition 8 (Network Fixed Point)**.** *A state x^* such that $T_G(x^*) = x^*$ for the network update operator T_G.*

Proposition 1. *Stable global structure corresponds to fixed points of distributed completion dynamics.*

Remark 3. *This extends earlier fixed-point interpretations of epistemic stability: the fixed point is no longer confined to a single system, but is distributed across the network.*

5 Distributed Projection and Relational Identity

Definition 9 (Node Projection). *A mapping* $\pi_i : X_G \to R_i$ *from global states to node-level representations.*

Definition 10 (Distributed Projection). *A network exhibits distributed projection if global structure is encoded across node projections collectively.*

Proposition 2. *The identity of a node is determined not solely by its local boundary, but by the stable completions in which it participates.*

Remark 4. *This formalizes the intuition that global structure is reflected across nodes, without being fully contained in any one node.*

6 Event-Based Linearization in Neural Systems

We now refine the model of neural nodes.

Definition 11 (Spike Event). *A spike event is a tuple*

$$e_i = (t_i, \phi_i, H_i, C_i),$$

encoding time, phase, internal history, and network context.

Remark 5. *This replaces a binary interpretation of spikes with an event-based dynamical one.*

Let $x(t)$ denote global state. Each neuron computes

$$s_i(t) = \sigma(w_i^\top x(t) + b_i + \Psi_i(H_i, t)).$$

Definition 12 (Threshold Crossing). *A spike occurs when* $s_i(t)$ *crosses a threshold.*

Proposition 3 (Event-Based Linearization). *Each spike imposes a local linear constraint on the global state.*

Theorem 2 (Distributed Linearization). *Network spikes collectively approximate nonlinear dynamics via distributed linear constraints.*

Remark 6. *Thus neural computation is an instance of distributed completion, where spikes provide the local constraints guiding completion.*

7 Manifold Structure and Coordinate Systems

Proposition 4. *Population activity lies on low-dimensional manifolds.*

Remark 7. *Toroidal and related structures arise from periodic variables and belong to the space of representation rather than physical embedding.*

Definition 13 (Coordinate Readout). *A mapping from relational structure to representational space.*

Proposition 5. *Allocentric and egocentric representations are coordinate transforms on a shared underlying structure.*

8 Synthesis with the Imagination Machine Framework

We now unify the preceding results with the broader framework.

Theorem 3 (Distributed Completion Principle)**.** *Scientific inference, neural computation, and distributed cognition instantiate the same process: the completion of partial structure under constraint across networks of bounded systems.*

Remark 8. *Compression corresponds to pruning of constraints, extension corresponds to completion, and networked interaction corresponds to distributed propagation of constraints.*

9 Conclusion

The Imagination Machine framework extends naturally from individual systems to networks. Locally bounded nodes participate in relational structures whose coherence emerges only through distributed completion. Neural computation implements this process through event-based dynamics, while scientific inquiry externalizes it.

The result is a unified picture in which structure is not contained locally but emerges globally, and is maintained through the consistent completion of partial information across a network of embedded systems.