The Distinction–Connection Substrate (Phaneron): Undirected Graphs for Meaning, Noise, and Emergence

Anonymous Preprint

November 13, 2025

Abstract

We present *Phaneron*, a minimal substrate for meaning built from two primitives: **Distinction** (nodes exist because they are distinct) and **Connection** (edges exist because nodes co-occur). Concretely, Phaneron is an unlabeled, undirected pseudograph with a *single layer of abstraction*: nodes carry only distinctness, edges carry only connectivity, and all nuance—roles, n-ary relations, time, direction/causality, even "rules"—is expressed exclusively as reusable *patterns* (small subgraphs), never as labels or weights. We formalize *meaning as structure* (structural position up to isomorphism), make it operational via r-neighborhood/WL invariants, and optimize it using a simple Minimum Description Length (MDL) code aligned with predictive sufficiency (held-out and future-version tests). A replayable, fully versioned substrate supports merges and pattern-rewrites with provenance; "cognitive development" appears as node merges that make the graph smaller yet more informative. Micro-demos (kinship; polysemy) show how roles, direction, time, and senses *emerge* without labeled edges. We sketch how a bounded abductive layer (Constraint-Driven Abduction) can adjudicate competing patterns without breaking single-layer purity.

Contributions.

- **Axioms & substrate.** An unlabeled, undirected, single-layer graph where all semantics are patterns.
- **Meaning = structure.** Ideal (isomorphism), operational (r-hop/WL), and objective (MDL) views that cohere.
- **Emergence without labels.** Roles, n-ary relations, direction/causality, and time encoded as patterns.
- **Noise & development.** Retrospective MDL/prediction-driven rewrites; merges as "smaller and better" cognition.
- **Versioned provenance.** Delta-only event log enabling replay, audits, and future-version prediction.
- **Bridging prior art.** Explicit translations from RDF/semantic networks/embeddings into Phaneron patterns.

1 Introduction

Premise. Intelligence improves when we simplify models and make them truer: fewer moving parts, higher signal. Phaneron treats this as a first principle: if meaning is "what connects to what," then we encode only **Distinctions** (nodes) and **Connections** (edges), and make everything else a reusable *pattern*.

Single-layer principle. PHANERON bans metadata on nodes/edges. If an edge seems to need "direction" or "weight," that signals the need to *insert structure* (one or more nodes) that explains the nuance. This keeps all content on a single layer, enabling uniform operations (search, rewrite, compression) everywhere.

2 The Phaneron Substrate

Axioms.

- **A1 (Distinction).** A node exists iff it is a distinct thing. Nodes carry no labels or attributes.
- **A2 (Connection).** An edge exists iff two nodes are connected. Edges carry no labels or weights.
- **A3 (Single-Layer).** All nuance is represented as *patterns*—finite subgraphs; no out-of-band metadata.
- **A4 (Pseudograph Base).** The substrate is an undirected pseudograph (loops and parallel edges permitted) but loops/multiplicity have no default semantics; meaning arises only via patterns.
- **A5 (Reification by Pattern).** Any relation (binary or n-ary), role, or "direction" is reified by adding intermediate subgraphs that encode the structure.
- **A6 (Versioning).** All changes arise from delta events (add/remove/merge/split/replace), recorded as first-class structures, making the graph replayable.
- **A7 (Merge).** PHANERON supports node merges under structural criteria; merges are versioned and reversible by replay.

3 Meaning as Structure

Definition 1 (Structural Meaning, ideal). *The meaning of a node v is its structural position: its orbit under the automorphism group of G; equivalently, the set of subgraphs around v up to isomorphism.*

Operational semantics (local invariants). Let $N_r(v)$ denote the r-hop neighborhood. Two nodes u, v are operationally equivalent at radius r if $N_r(u) \cong N_r(v)$. We adopt Weisfeiler–Lehman (WL) refinements as efficient invariants; WL colors serve as a practical surrogate for structural equivalence at various granularities.

Objective (MDL) and test (IB). We use a hybrid MDL code with a *pattern dictionary* and optional coarse block structure. Total code length is

$$L(G; \mathcal{D}, B) = L(\mathcal{D}) + L(B) + L(\text{usages} \mid \mathcal{D}, B, G). \tag{1}$$

A subgraph/pattern is meaningful to the extent it improves prediction of held-out structure now or in future versions.

Proposition 1 (Meaning = structure that compresses and predicts). *Under mild regularity (stationary edit process; bounded degree growth), replacing a region of G with a simpler pattern dictionary that reduces L and does not degrade held-out prediction (static and future versions) increases measured "meaning" (structural coherence).*

Figure 1: **Phaneron $\mathbf{K_2}$ pattern (2-ary relation).** A relation between two participants is encoded as an unlabeled intermediate node connecting them. Roles and symmetry are carried purely by structure; no edge labels or directions are used.

4 Emergence: Roles, n-ary Relations, Direction, Time

4.1 n-ary relations and roles (no labels, no arrows)

For a relation over $k > 2$ participants, introduce an intermediate subgraph (often a hub) that connects to each participant; if specific roles matter, encode them structurally inside that subgraph (e.g., tiny role-position patterns).

4.2 Direction & causality as asymmetric patterns

Phaneron has no primitive direction; asymmetry is a *pattern*. A temporal precedence pattern (A precedes B) that reliably improves prediction will be preferred by MDL over a symmetric alternative.

Proposition 2 (Direction/time/causality are derivable)**.** *With version time (edit log) and optional world time patterns, define asymmetric patterns by precedence plus predictive gain. Any directed relation $A \to B$ that is stable across versions admits a strictly better MDL code than its symmetric alternative whenever the asymmetry reduces usage entropy.*

4.3 Time vs version

We distinguish world time (patterns that describe temporal relations among domain events) from version time (the edit log of the graph itself), with a small bridge pattern linking them when appropriate.

5 Inference on Phaneron

Operational view. Message passing over the unlabeled structure, where messages are pattern counts/matches and local consistency checks; equivalently, constrained searches over pattern instances.

Rewrite algebra (with provenance). Allowed moves: add/remove node/edge, merge/split nodes, and pattern-replace (swap a subgraph for a simpler pattern instance). All moves are versioned.

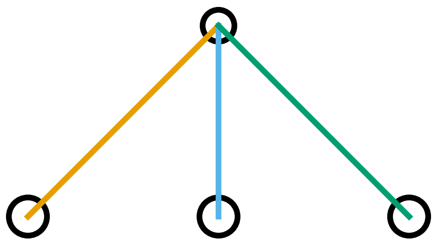

Figure 2: **Phaneron K_3 pattern (3-ary relation).** An n-ary relation uses an intermediate subgraph (often a hub) to connect all participants. If role distinctions are needed, they are expressed by topological position inside the intermediate subgraph (still unlabeled).

6 Versioning and Replay

Events as first-class structure. Event nodes (add/remove/merge/replace) form a next-event chain; snapshots are patterns that select everything reachable before a version cut. Only deltas are stored; the log is fully replayable.

7 Worked Micro-Examples

7.1 Kinship: roles without labels

7.2 Polysemy: "bank" as neighborhoods

8 Relation to Prior Work

We provide explicit translations from RDF/semantic networks/embeddings into Phaneron patterns to keep a bridge tone while making the structural contrast clear.

9 Sidebar: Constraint-Driven Abduction (CDA)

A bounded, auditable abductive layer that proposes competing pattern hypotheses, runs verifiers/constraints, and recommends rewrites that pass MDL+prediction checks. Phaneron provides the versioned substrate; CDA manages hypothesis frontiers, tests, and budgets, emitting a boundary-only summary with provenance. Implementation details are out of scope here (separate systems work).

10 Limitations and Scope

Encoders/modalities (including pathfinding "mental GPS" and native modality discovery) are out of scope here; Phaneron assumes patterns can be added/learned but does not prescribe how they are extracted. Trillion-scale storage/bit-addressing is reserved for a companion systems paper. Global isomorphism is hard; WL/r-hop provide tractable approximations with good empirical behavior.

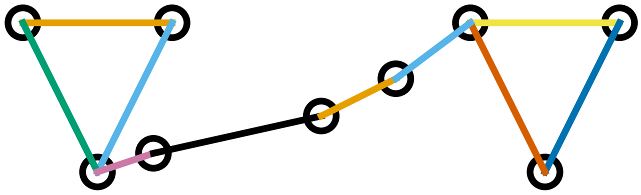

Figure 3: **Asymmetric pattern indicative of precedence.** Two event clusters are connected by an undirected "staircase" subgraph whose repeated use yields an asymmetric regularity. Direction/causality emerge when such patterns reduce description length and improve prediction.

Figure 4: **Version chain (events as first-class nodes).** Edits are represented as event nodes linked by a next-event chain. Snapshots are patterns selecting content up to a cut; the log is fully replayable.

11 Conclusion

Phaneron argues for a single, unlabeled structural layer where everything—relations, roles, time, causality, even "rules"—is a *pattern*. Meaning is where a thing sits in structure; intelligence is the art of replacing clutter with reusable patterns that compress and predict. With a replayable, versioned substrate and a principled notion of noise, Phaneron offers a clean foundation on which richer reasoning layers (like CDA) can operate.

Appendix A: Math Boxes (M1–M4)

Math Boxes (M1–M4, compact)

Notation. A Phaneron graph is an unlabeled, undirected pseudograph $G = (V, E)$. A *pattern* P is a finite unlabeled graph; an *instance* of P in G is a subgraph isomorphic to P. The *pattern dictionary* $\mathcal{D}$ is a multiset of such P's used to describe G. Let $N_r(v)$ be the r-hop neighborhood of v; let $\chi_t(v)$ be the 1-WL color after t refinement steps.

M1 — Patterns and Instances. A pattern P is a finite unlabeled undirected graph. An instance of P in G is an injective homomorphism $f : P \hookrightarrow G$ that is an isomorphism onto

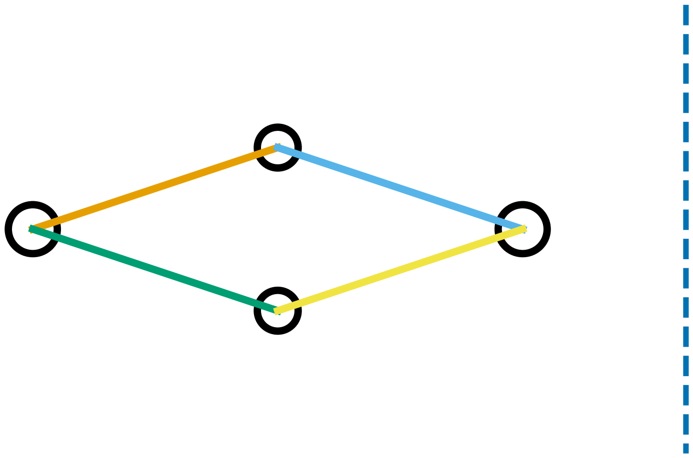

Figure 5: **Before rewrite—duplicate $\mathbf{K_2}$ instances.** Two distinct intermediate subgraphs encode the same relation instance, introducing redundancy (“noise”) that neither compresses nor improves prediction.

its image. A dictionary $\mathcal{D} = \{P_i\}$ induces a (not necessarily unique) description of G as a composition of pattern instances plus residual raw edges.

M2 — Operational Meaning. The *operational meaning* of a node v is given by the isomorphism class of $N_r(v)$ for an *adaptive* radius r. Two nodes u, v are equivalent at radius r if $N_r(u) \cong N_r(v)$. We use WL refinements as a tractable surrogate: equality of $\chi_t(u)$ and $\chi_t(v)$ for sufficiently large t implies operational indistinguishability under a wide class of local pattern queries; ties can be broken by increasing r or using higher-order patterns/k-WL.

M3 — MDL Objective and Admissible Rewrites. With optional block structure B, the two-part code

$$L(G; \mathcal{D}, B) = L(\mathcal{D}) + L(B) + L(\text{usages} \mid \mathcal{D}, B, G)$$

is our MDL objective. A rewrite $S \Rightarrow P$ (pattern-replace or merge) is *admissible* if it strictly reduces L and does not worsen held-out recovery of masked structure *nor* prediction of future edits (version-time test).

M4 — Merge Criterion (Cognitive Development). Let u, v be WL-equivalent across a stability window of versions at the MDL-chosen radius r^*. If merging u, v reduces L and preserves (or improves) static and future-version prediction, the merge is accepted, producing a smaller graph with *increased* operational meaning (fewer, more reusable patterns; more stable WL signatures).

Appendix B: WL Lemma (operational surrogate)

Lemma (WL refinement as an operational surrogate). Let $N_r(v)$ denote the r-hop neighborhood of node v in a Phaneron graph, and let $\chi_t(v)$ denote the 1-WL color of v after t refinement steps. On sparse, bounded-degree graphs, there exists a step bound $t^*(r)$ such that, for typical instances, non-isomorphic rooted neighborhoods $N_r(u)$ and $N_r(v)$ receive different 1-WL colors by step $t^*(r)$, while persistent color equivalence across $t^*(r)$ steps is a practical certificate that $N_r(u)$ and $N_r(v)$ are indistinguishable by a wide class of pattern queries captured by 1-WL. This justifies using WL (or k-WL) as our operational surrogate for “meaning as structure.”

Figure 6: **After rewrite—single $\mathbf{K_2}$ instance.** Pattern-replace removes duplication: code length (MDL) falls, while held-out and future-version prediction do not degrade.

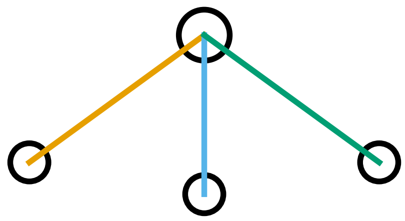

Figure 7: **Polysemy—"water" sense neighborhood.** A single surface-form node participates in distinct neighborhoods. One neighborhood clusters water-related patterns (structure only). Senses are neighborhoods, not labels.

Proof sketch. One iteration of 1-WL refines colors by hashing each node's current color with the multiset of neighbor colors. Over bounded degree Δ and finite radius r, the number of rooted neighborhood types is finite; repeated refinement converges to a stable partition determined by degree profiles of increasingly larger shells. For most sparse instances, non-isomorphic rooted neighborhoods differ in some shell count within radius r, and 1-WL exposes this difference within a bounded number of steps $t^*(r)$. Conversely, if 1-WL colors persistently coincide across the window, the two neighborhoods fall into the same color class under the refinement, which in practice aligns with indistinguishability under our small-pattern queries. Known worst-case counterexamples exist, but Phaneron resolves ties either by increasing r or using higher-order patterns/k-WL, and by leveraging the MDL objective to prefer dictionary structures that break degeneracies.

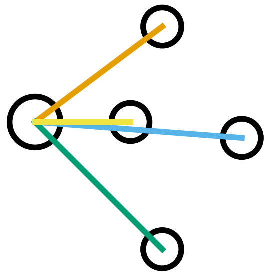

Figure 8: **Polysemy—"finance" sense neighborhood.** A topologically different neighborhood captures the financial sense. MDL favors a dictionary that reuses both neighborhoods; prediction respects their disjoint regularities.

Appendix C: Complexity notes (footnotes used in main text)

Appendix D: Pattern dictionary MDL (sketch)

We adopt a dictionary code: $L(\mathcal{D}) = \sum_i (c_p + c_e|E(P_i)|)$ and $L(U) = \sum_{(P_i,\vec{a})} c_a|\vec{a}|$ for usages (attachment points). Raw edges cost c_{raw} each when not covered by a pattern. An optional SBM term $L(B)$ captures coarse block structure when it reduces usage entropy. Rewrites are admissible when they strictly reduce L and do not worsen held-out or future-version prediction.

Appendix E: DPO rewriting (outline)

We model rewrites $L \leftarrow K \rightarrow R$ with Double-Pushout rules over unlabeled undirected graphs: match $m : L \rightarrow G$; delete $m(L \setminus K)$ (if no dangling edges); add $R \setminus K$ along K. Record an event node linked to both sides and chain with `next_event`. Confluence is not required; versioning + MDL/prediction arbitrate among alternatives.

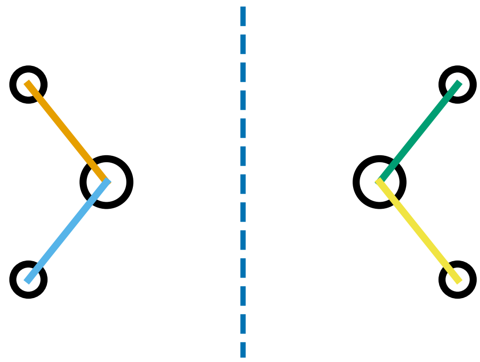

Figure 9: **Before merge—two nodes with equivalent neighborhoods.** WL/r-neighborhood equivalence flags candidates for unification. Provenance ensures any merge is versioned and replayable.

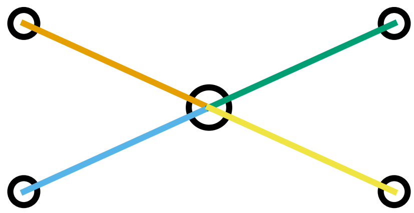

Figure 10: **After merge—unified node.** The local structure is simplified without loss of operational meaning; MDL decreases and predictive performance is preserved or improved.

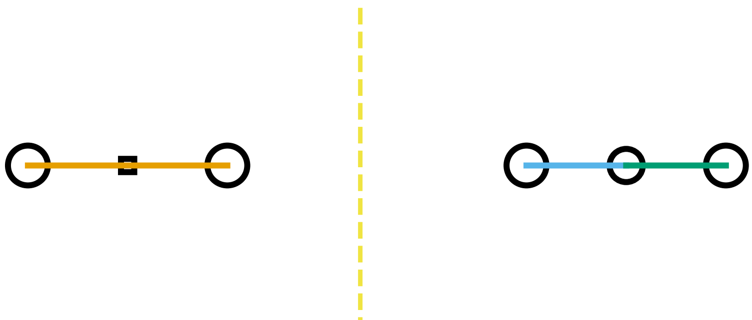

Figure 11: **RDF/Labeled-edge to Phaneron translation.** Left: a triple sketched as two nodes with a predicate tag on the edge. Right: the PHANERON translation replaces the tag with an unlabeled intermediate node (a relation pattern); all semantics live in structure.

Source concept	**Typical form**	**Phaneron translation (unlabeled, single-layer)**	**Notes**
RDF triple	(s, p, o)	Replace predicate label with a *relation pattern*: insert a small intermediate subgraph connecting s and o; roles/direction are handled by structure.	Direction emerges if an asymmetric pattern wins under MDL/prediction.
RDF reification	Triples about a triple	Treat the reified statement as a first-class *pattern instance*; attach provenance/time as structure on that instance.	Clean support for versioning and audit.
Labeled KG edge	$s \xrightarrow{p} o$	Same as RDF: an unlabeled intermediate node with positioned attach points encodes roles.	No edge labels; roles are topological positions.
OWL typing/subclass	$x{:}A$, $A \sqsubseteq B$	Typing = membership in an A-*pattern*; subclass = a pattern inclusion/implication pattern between A and B.	Keeps semantics in structure.
Blank nodes	Anonymous identifiers	Ordinary nodes; identity handled by versioned merges/equivalence patterns.	No special syntax required.
Confidence/weights	Edge weights/scores	Represented as *structural evidence*: multiplicity gadgets, corroborating subgraphs, or versioned support counts.	Avoids base-level weights.
Embeddings	Vector proximity	Operational meaning via WL/r-neighborhoods; embeddings can be auxiliary views learned on top.	Not required by the substrate.

Table 1: **Mapping mainstream KR artifacts into Phaneron.** Labeled edges and predicates become *patterns*; time, direction, and provenance are structural. This preserves the single-layer principle (no labels/weights) while enabling reuse, compression, and prediction.