# Phaneron-Store: ID-less Graph Storage for Single-Layer Semantics at Trillion–Quadrillion Scale

Anonymous Preprint

November 13, 2025

**Abstract**

We present *Phaneron-Store*, an ID-less, versioned graph substrate that stores single-layer semantics at trillion–quadrillion scale. The design treats **Distinction** and **Connection** as the only primitives; all nuance—roles, $n$-ary relations, direction/time—is expressed via reusable *patterns* and a content-addressed state DAG. We combine pattern-first adjacency (edges as expansions), cross-time reference compression, container interning, hot→cold tiering (bitmaps→succinct), structural reordering, and intervalized edge presence. The result is a store whose cold majority approaches *single-digit bits per edge*, while hot regions sustain fast r-hop traversals and rewrites. We also sketch a reversible Lens+Codec that lets the substrate *absorb* files losslessly as meaning+surface streams, enabling query-while-compressed.

**Thesis.** IDs are scaffolding; semantics lives in structure. *Phaneron-Store* encodes only **Distinction** and **Connection** at the base; everything else is reusable *patterns*. The store is versioned, content-addressed, and engineered for near-linear passes with strong cold compression.

**Contributions.**

- **ID-less, content-addressed state DAG:** cross-time/node dedup of node states and delta shapes.
- **Pattern-first adjacency:** persist dictionary+placements; store only residual edges.
- **Adjacency reference compression:** copy-from + symmetric diff with global hash-cons of add/remove lists.
- **Two-tier adjacency:** Roaring-style hot tier; Elias–Fano/k$^2$-trees cold tier with tiny overlays.
- **Container interning & structural reordering:** dedup small containers; reorder addresses to shrink gaps and speed r-hop.
- **Replayable versioning:** append-only delta log with checkpoints, snapshots, and intervalized presence.
- **Lens+Codec (lossless):** reversible semantic absorption with meaning/surface streams; query without inflate.

## 1 Data Model and Principles

Undirected pseudograph; single layer (no labels/weights at base). Patterns reify relations/roles/time/direction. Versioning via delta events; merges and pattern-rewrites are first-class and replayable.

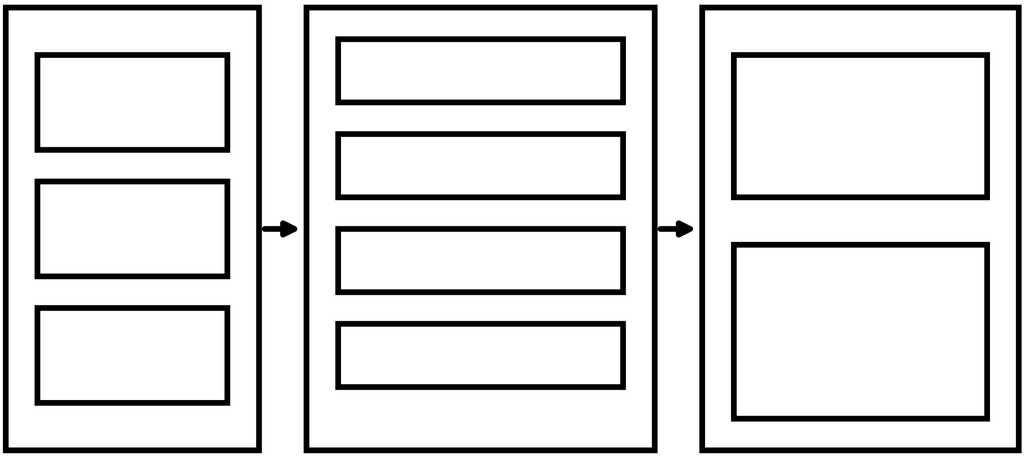

Figure 1: **Store memory layout.** Dictionary+patterns (left), placements/state-DAG (middle), adjacency tiers (right).

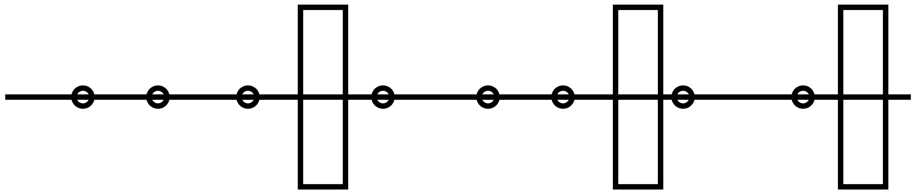

Figure 2: **Delta log and checkpoints.** Events (nodes) along a timeline; checkpoints bound replay; snapshots select content prior to a cut.

## 2 Layout and Addressing

Content-addressed *state DAG*: identical r-neighborhood states across nodes/time store once; transitions are edges. Pattern-first adjacency: edges implied by patterns are virtual (materialized on access); residual edges persist physically.

## 3 Versioning and Replay

Append-only delta log with periodic checkpoints; snapshots are cheap; long-lived edges become presence intervals.

## 4 Adjacency Encoding

### 4.1 Hot tier: bitmaps + overlays

Roaring/EWAH-style containers with *container interning*. Small overlays absorb edits; copy-on-write keeps references valid.

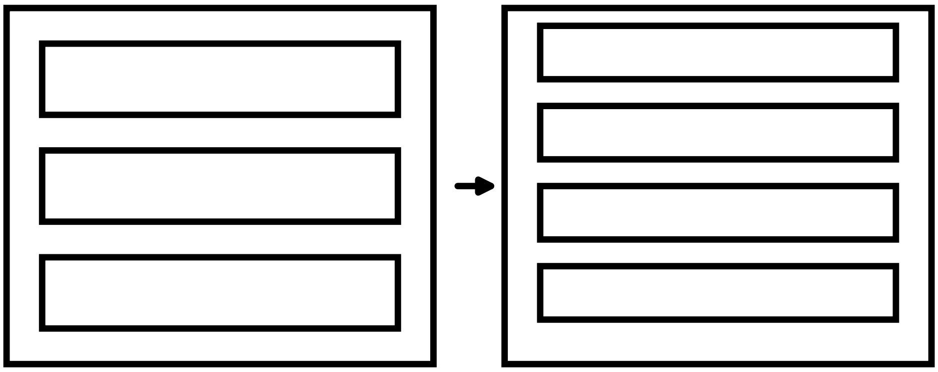

Figure 3: **Hot→cold tiering.** Actively-edited neighborhoods (left) migrate to succinct cold storage (right) with overlays.

### 4.2 Cold tier: succinct integers

Elias–Fano or $k^2$-trees with tiny overlays. Containers are shared globally; cold sets achieve single-digit bits per edge after reordering.

### 4.3 Reference compression (copy-from + diff)

At version $t$, a node's adjacency copies from a reference (often itself at $t$–1) plus small add/remove lists, hash-consed globally.

## 5 Reordering and Compaction

Periodic structural reordering (community/BFS/SlashBurn within shard) shrinks gaps; compaction rebuilds cold tiers; caches align with r-hop locality.

## 6 Core Operations

**r-hop expansion.** SIMD-friendly set ops; WL cache up to small $t$ with local invalidation on edits.

**Merge and pattern-replace.** Local $\Delta L$ (MDL) checks; provenance events; incremental index updates.

## 7 Sharding and Fault Tolerance

Structural hash with locality hints; replication within rack; cross-rack async replicas; local-repair erasure codes for cold tier.

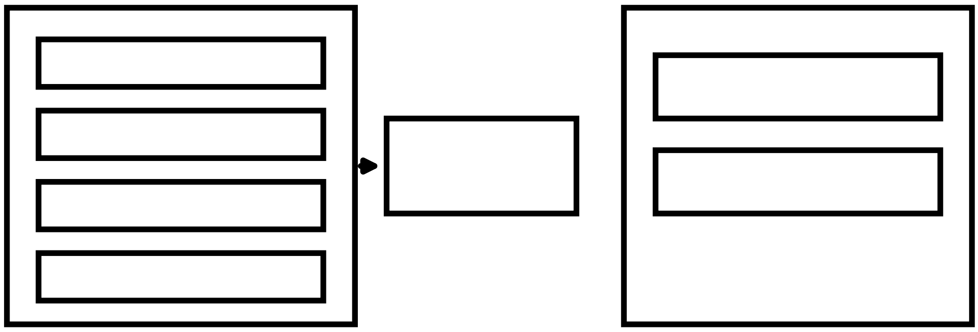

Figure 4: **Reference compression.** Full set (left) → copy-from block (middle) + tiny add/remove lists (right).

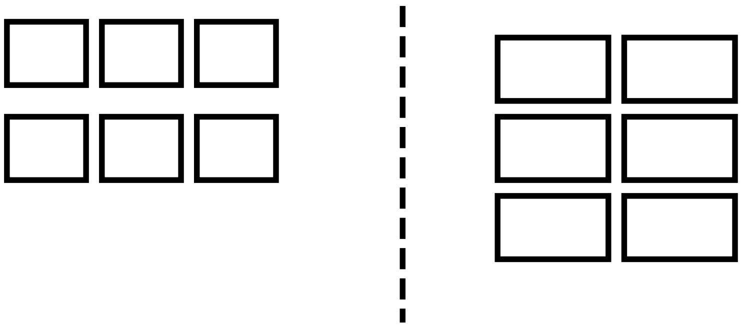

Figure 5: **Reordering and compaction.** Scattered blocks (left) reorganized into contiguous ranges (right).

# 8 Lens+Codec: Reversible Semantic Absorption

Absorb files into meaning+surface streams (lossless). Operate on meaning without inflate; reify exact bytes on demand.

# 9 Evaluation Plan

# 10 Ethics and Limits

Topology deanonymization risk; deletion semantics with provenance; compaction windows; GC pauses; hot-spot shards.

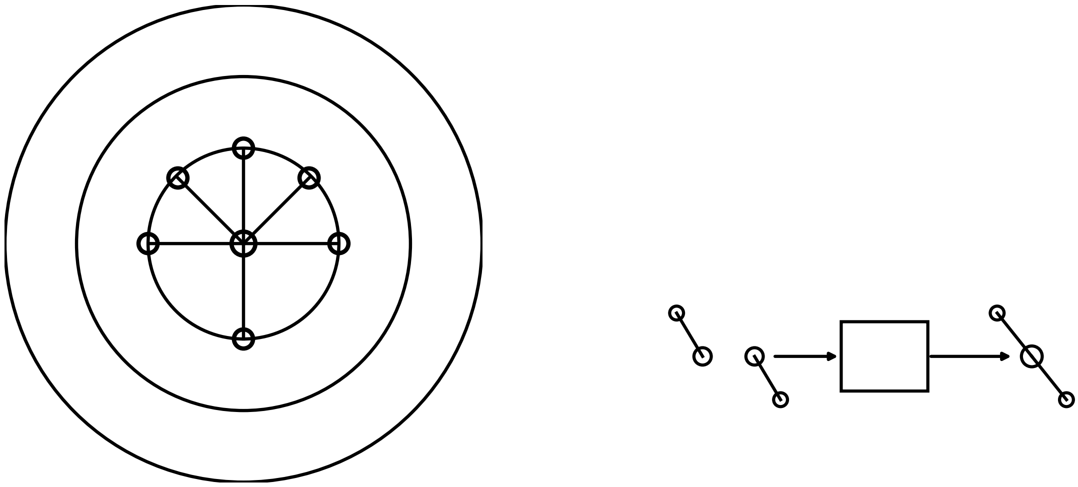

Figure 6: **Core ops.** Left: r-hop shells; Right: merge/replace flow with a rule box.

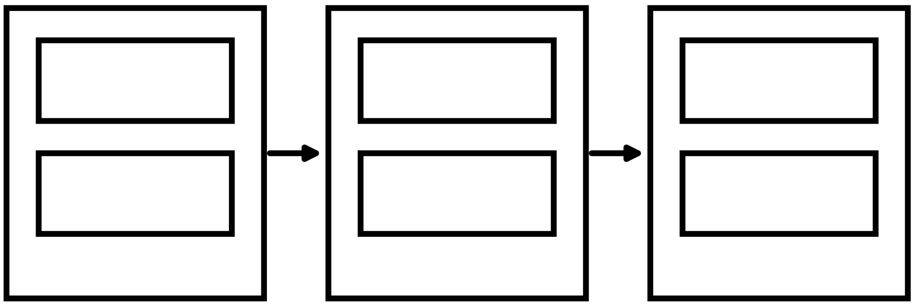

Figure 7: **Sharding and replication.** Shards with multiple replicas; arrows indicate replication flow.

## 11 Related Work (positioning sketch)

CSR/COO stores; bitmap families (Roaring/EWAH/WAH); succinct sets (Elias–Fano, $k^2$-trees); graph DB engines; differential graph stores; GraphBLAS; learned indexes; RLZ compression.

## 12 Conclusion

Phaneron-Store shows that single-layer semantics can ride an ID-less, content-addressed substrate to trillion–quadrillion scale with practical latency. The same substrate cleanly supports reversible absorption as a universal, queryable compression layer.

| Track | Datasets | Metrics | Bar (Pass Criteria) |
|---|---|---|---|
| State-DAG dedup | Temporal KG; code repos (commits); news streams | % state reuse; % delta-shape reuse | $\geq$30% cross-time state reuse; $\geq$50% delta-shape reuse |
| Adjacency ref-compress | Social/enterprise graphs; text-derived KG | Add/remove size vs full neighbors | $\geq 10\times$ median shrink; $\geq 100\times$ top quartile |
| Cold tier compression | Web graph; enterprise KG | Bits/edge (cold), blended bpe | Cold $\leq$3 bpe; blended $\leq$8 bpe |
| Pattern-first adjacency | Ontologies; org charts; code AST graphs | % edges covered by patterns | $\geq$60% edges materialized from patterns |
| Hot r-hop throughput | Hot shards under writes | P50/P99 per r-hop; cache hit rate | Competitive vs bitmap baselines, $<$2x overhead |
| Replay/versioning | Edit log under load | Snapshot latency; recovery time | Snapshot $<$ seconds; recovery bound by disk |
| Lens+Codec | Markdown/CSV/JSON corpora | Bits/byte vs zstd; round-trip fidelity | Parity on eclectic; win on coherent; 100% lossless |

Table 1: Evaluation plan: datasets, metrics, and acceptance bars for each contribution.