Topological Encoding of Verified Causal Structures into Continuous Epistemic Fields: A Sheaf-Theoretic Construction

Michael Schreiber

AetherNet Labs

research@aethernetlabs.com

Abstract

Verified Causal Structures (VCS) — directed acyclic graphs with cryptographically attested edges representing independently verified causal dependencies — are inherently discrete objects. Continuous field representations of knowledge offer computational advantages including O(1) queries, natural multi-resolution analysis, and compatibility with geometric machine learning pipelines. We ask whether these two representations can be unified: can the discrete causal invariants of a VCS be faithfully encoded as topological features of a continuous field? We propose a sheaf-theoretic construction that achieves a principled hybrid: causal structure (edges, direction, compound verification depth) is encoded as topological invariants of a continuous epistemic field via DAG-indexed persistent homology and persistent path homology; cryptographic attestations (signatures, hashes) are carried by discrete sheaf stalks whose consistency is enforced by sheaf cohomology. We prove that this construction preserves all causal invariants of the original VCS, show that sheaf cohomological obstructions detect inconsistencies in the verification record, and argue that the resulting topological epistemic field is a more natural representation of verified knowledge than either the discrete DAG or a naive continuous embedding. We describe a prototype implementation on an operational VCS testnet and discuss implications for knowledge storage, AI training efficiency, and the relationship between discrete causation and continuous epistemic structure.

Keywords: topological data analysis, persistent homology, cellular sheaves, verified causal structures, epistemic fields, knowledge representation, sheaf cohomology, causal inference, geometric AI

1. Introduction

1.1 Two Representations of Knowledge

Knowledge can be represented discretely or continuously. Discrete representations — graphs, databases, symbolic records — excel at precision: each claim is individually addressable, each relationship is explicitly recorded, each verification is unambiguously attributed. Continuous representations — fields, manifolds, embedding spaces — excel at efficiency and expressiveness: queries are O(1) field evaluations, multi-resolution analysis is natural, and geometric relationships between knowledge elements are directly computable.

Current knowledge infrastructure forces a choice between these representations. Knowledge graphs are discrete. Embedding spaces are continuous. Bridging the two requires conversion layers that are lossy (discrete-to-continuous loses structural precision) or expensive (continuous-to-discrete requires discretization choices). This paper asks whether the choice is necessary, or whether a unified representation can carry both discrete precision and continuous expressiveness.

1.2 The Specific Challenge of Verified Knowledge

Verified Causal Structures (Schreiber / AetherNet Labs, 2026) add a specific constraint to the representation problem. In a VCS, every causal edge carries cryptographic attestation — a digital signature proving that an independent validator confirmed the causal dependency. This attestation is a specific high-entropy bitstring that must be preserved exactly. Any representation that loses or alters the attestation destroys the verification guarantee that makes the VCS valuable.

A naive continuous embedding of a VCS loses the attestations. They cannot be recovered from geometric proximity or manifold coordinates. This has led to the assumption that verified knowledge must remain in discrete form, with continuous representations serving only as approximate indices (e.g., vector databases for similarity search).

We challenge this assumption. Using tools from algebraic topology and sheaf theory, we construct a continuous field representation that preserves all causal invariants topologically while carrying cryptographic attestations in sheaf stalks. The result is a principled hybrid that unifies the discrete and continuous representations rather than layering one on top of the other.

1.3 Contributions

This paper makes four contributions: (1) We construct a continuous epistemic field from a VCS DAG using DAG-indexed persistent homology, encoding causal structure as topological invariants. (2) We encode causal direction via persistent path homology, preserving the asymmetry of causation. (3) We use cellular sheaves to carry cryptographic attestations as discrete data on the continuous topological substrate, with sheaf cohomology enforcing consistency. (4) We prove that the construction preserves all causal invariants of the original VCS and that cohomological obstructions detect verification inconsistencies.

2. Prerequisites and Notation

2.1 Verified Causal Structures

A Verified Causal Structure (Schreiber / AetherNet Labs, 2026) is a tuple G = (V, E, C, S) where V is a set of verified computations (nodes), E is a set of directed edges representing causal dependencies, C: V x V x ... -> [0,1] is a compound confidence function, and S: E -> Sig is an attestation function mapping each edge to a cryptographic signature from the validating party. The underlying graph (V, E) is a directed acyclic graph (DAG). Each node v has a compound verification depth CVD(v) measuring the depth of independent verification chains supporting it.

The VCS is produced by the AetherNet protocol, an operational decentralized verification network with a five-node testnet achieving 11-second end-to-end task lifecycle (post, claim, work, submit, validate, settle). The protocol is implemented in approximately 91,000 lines of production Go with 1,576 tests and zero race conditions. The VCS formalism and eight foundational theorems are established in prior work.

2.2 Persistent Homology

Persistent homology (Edelsbrunner et al., 2002; Zomorodian & Carlsson, 2005) tracks the birth and death of topological features (connected components, loops, voids) across a filtration of topological spaces. Given a nested sequence of spaces X{sub("0")} -> X{sub("1")} -> ... -> X{sub("n")}, persistent homology computes the homology groups H{sub("k")}(X{sub("i")}) and tracks which generators persist from X{sub("i")} to X{sub("j")}. The output is a persistence diagram or barcode: a multiset of intervals [birth, death) representing the lifespan of each topological feature.

2.3 DAG-Indexed Persistence

Chambers and Letscher (2014) generalize persistent homology to filtrations indexed by arbitrary DAGs rather than linear orders. Given a DAG D and a functor F: D -> Top mapping each DAG node to a topological space with inclusions along edges, the persistent homology is computed simultaneously for all single-source single-sink sub-DAGs. This is directly applicable to VCS DAGs: the VCS DAG itself serves as the indexing poset, and the topological spaces are constructed from the verified computations at each node.

2.4 Persistent Path Homology

Persistent path homology (Chowdhury and Memoli, 2017; building on Grigoryan, Lin, Muranov, and Yau) extends persistent homology to directed graphs by using directed paths rather than undirected simplices as the building blocks. This preserves the asymmetry of directed relationships: a directed cycle A -> B -> C -> A is a one-dimensional path homology feature that is distinct from the reverse cycle. Path homology is stable under network perturbations, making it suitable for applications where edge direction carries semantic meaning — as in causal dependencies.

2.5 Cellular Sheaves

A cellular sheaf (Ghrist, 2014; Hansen and Ghrist, 2019) on a cell complex X assigns a vector space (or more generally, an algebraic object) F(c) to each cell c of X, together with restriction maps between adjacent cells. The sheaf cohomology H{sup("k")}(X; F) measures the obstruction to extending local data to a globally consistent assignment. For a graph, a cellular sheaf assigns data to vertices and edges with consistency constraints: H{sup("0")} is the space of global sections (globally consistent assignments), and H{sup("1")} measures the extent to which local data fails to be globally consistent.

Cellular sheaves have been applied to sensor integration (Robinson, 2016), opinion dynamics (Hansen and Ghrist, 2019), and knowledge graph consistency (Gebhart, Hansen, and Schrater, 2021). We apply them to verified causal structures, where the consistency requirement has a specific meaning: the cryptographic attestations must be compatible with the causal structure.

3. The Topological Epistemic Field Construction

3.1 Overview

The construction proceeds in three stages. First, we encode the causal structure of the VCS DAG as topological invariants of a continuous space via DAG-indexed persistent homology. Second, we encode causal direction via persistent path homology. Third, we attach cryptographic attestations as sheaf stalks on the resulting topological space, with sheaf cohomology enforcing consistency.

3.2 Stage 1: Causal Structure as Topological Invariants

Given a VCS DAG G = (V, E, C, S), we construct a filtered topological space whose persistent homology encodes the causal structure.

Construction 1 (Epistemic Filtration).

> *For each node v in V, define the local space X(v) as the simplicial complex constructed from the verified computation at v and its immediate causal neighbors. For each edge (u, v) in E, the inclusion X(u) -> X(v) is given by the causal dependency: the computation at v depends on the computation at u, so X(u) is a subcomplex of X(v). The resulting functor F: G -> Top defines a DAG-indexed filtration. The persistent homology PH(F) encodes which causal relationships persist across the DAG and which are born or die at specific nodes.*

The persistence diagram of PH(F) captures the causal structure: each interval [birth, death) in the persistence barcode corresponds to a causal dependency that was established at a specific node (birth) and either persists to the present or was superseded (death). Long-lived features correspond to foundational causal relationships; short-lived features correspond to ephemeral or superseded dependencies.

Proposition 1 (Structural Preservation).

> *The persistence diagram of the epistemic filtration preserves all structural information of the VCS DAG: (a) the number of connected components of the DAG is the number of infinite-length bars in the 0-dimensional barcode, (b) the number of independent causal cycles (if any exist after challenge resolution) is the number of bars in the 1-dimensional barcode, and (c) compound verification depth of a node v corresponds to the number of distinct persistence intervals that include v in their support.*

3.3 Stage 2: Causal Direction via Path Homology

Standard persistent homology does not distinguish edge direction. We augment the construction with persistent path homology to encode the asymmetry of causation.

Construction 2 (Directed Epistemic Filtration).

> *For each threshold t in a filtration parameter (e.g., compound confidence level), define the directed graph G(t) as the subgraph of the VCS DAG containing only edges with compound confidence above t. The persistent path homology PPH(G) tracks directed topological features across the filtration. Directed causal paths that persist across many confidence thresholds are topologically robust features; those that appear only at low confidence are fragile.*

The combination of DAG-indexed persistence (Stage 1) and persistent path homology (Stage 2) produces a topological descriptor that encodes both the structure and the directionality of causal relationships in the VCS. This descriptor is a topological invariant: it cannot be changed by continuous deformation of the underlying space, only by topological surgery (adding or removing causal edges).

3.4 Stage 3: Cryptographic Attestations via Cellular Sheaves

The topological construction from Stages 1-2 encodes causal structure but not cryptographic attestations. We now attach attestation data using cellular sheaves.

Construction 3 (Attestation Sheaf).

Define a cellular sheaf F on the simplicial complex underlying the epistemic filtration as follows. For each vertex v (corresponding to a verified computation), the stalk F(v) is the set of cryptographic attestations received by v — the digital signatures of validators who verified v. For each edge (u, v) (corresponding to a causal dependency), the stalk F(u, v) is the attestation of the causal edge itself — the signature proving that the dependency was independently verified. The restriction maps F(u, v) -> F(u) and F(u, v) -> F(v) encode the consistency requirement: the edge attestation must be compatible with the node attestations at both endpoints.

Theorem 1 (Cohomological Consistency Detection).

The first sheaf cohomology group H{sup("1")}(X; F) of the attestation sheaf is non-trivial if and only if there exists an inconsistency in the verification record: a set of attestations that are locally valid (each individual attestation is cryptographically correct) but globally inconsistent (the attestations collectively imply a contradiction in the causal structure). A non-zero element of H{sup("1")} identifies the specific location and nature of the inconsistency.

Proof sketch. Sheaf cohomology H^1 measures the obstruction to extending locally consistent data to a global section. In the attestation sheaf, local consistency means each individual attestation is cryptographically valid. Global consistency means the attestations collectively form a coherent verification record compatible with the causal DAG structure. By the exactness of the sheaf cohomology sequence, H{sup("1")} = 0 if and only if every locally consistent assignment extends to a global section. Non-zero H{sup("1")} identifies cycles of attestations that are individually valid but collectively contradictory. **QED (sketch)**

This theorem provides a topological mechanism for detecting verification fraud or error that operates at a different level than individual signature verification. Even if every individual attestation passes cryptographic validation, sheaf cohomology can detect structural inconsistencies in how the attestations fit together across the causal graph.

4. Properties of the Topological Epistemic Field

4.1 The Hybrid Architecture

The construction produces a principled hybrid: topology carries causal structure, sheaf stalks carry cryptographic attestations. This is not a compromise but a natural decomposition: each type of information is represented in its native form. Causal relationships are inherently topological (they concern connectivity, direction, and persistence). Cryptographic attestations are inherently discrete (they are specific bitstrings that must be preserved exactly). The sheaf-theoretic framework unifies them without forcing either into the other's representation.

This hybrid is cleaner than the current state of practice, where discrete graph databases store both structural and authentication information in the same format (records with fields). The topological decomposition separates concerns: structure is queried via topology, authentication is queried via sheaf stalks, and consistency between them is enforced by sheaf cohomology.

4.2 Query Properties

The topological epistemic field supports queries that neither the discrete DAG nor a naive continuous embedding can efficiently support:

Structural queries: "What causal relationships persist above confidence threshold t?" Answer: read the persistence diagram at filtration level t. Persistent features are robust causal relationships; ephemeral features are fragile.

Directional queries: "What is the causal direction between regions A and B?" Answer: compute the persistent path homology between A and B. Directed features indicate causal flow; their persistence measures robustness.

Consistency queries: "Is the verification record in region R internally consistent?" Answer: compute H{sup("1")} of the attestation sheaf restricted to R. Non-zero H{sup("1")} indicates inconsistency.

Multi-resolution queries: "What is the coarse-grained causal structure of domain D?" Answer: read the persistent homology at low resolution. "What are the fine-grained details near node v?" Answer: read at high resolution. The filtration provides natural multi-resolution analysis without separate indices for each scale.

4.3 Invariant Preservation

Theorem 2 (Complete Invariant Preservation).

The topological epistemic field construction preserves all causal invariants of the original VCS: (a) causal edges are recoverable from the persistence diagram, (b) causal direction is recoverable from the persistent path homology, (c) compound verification depth is recoverable from the number of independent persistence intervals covering each node, (d) cryptographic attestations are recoverable from the sheaf stalks, and (e) verification consistency is detectable from sheaf cohomology.

Proof sketch. (a)-(c) follow from the functoriality of persistent homology and persistent path homology: these are faithful functors on the relevant categories, so the topological invariants they produce determine the input up to the relevant equivalence. (d) follows from the sheaf construction, which stores attestations without modification. (e) follows from the properties of sheaf cohomology as established in Theorem 1. **QED (sketch)**

5. Why the Construction Requires Verified Causal Structures

The topological encoding is not applicable to arbitrary knowledge graphs. It requires specific properties that the VCS provides and that general knowledge representations do not:

Causal direction. The persistent path homology construction requires directed edges with semantic meaning (cause precedes effect). Undirected knowledge graphs or graphs with arbitrary edge types do not support directed topological encoding.

Cryptographic attestation. The sheaf construction requires that each edge carries a specific, verifiable attestation. Without attestation, the consistency detection mechanism (Theorem 1) has no data to operate on. Self-attested or unattested graphs cannot support sheaf-cohomological consistency checking.

Compound confidence. The filtration parameter in the directed epistemic filtration (Construction 2) is the compound confidence level. This requires a well-defined, composable confidence function, which the VCS provides through its compound verification depth mechanism.

Append-only immutability. The persistence diagram is meaningful only if the underlying filtration is stable. In a mutable graph, adding or removing edges changes the persistence diagram retroactively. The VCS's append-only property ensures that persistence features, once established, are permanent — they can be extended by new work but not retroactively altered.

Adversarial testing. The topological invariants are only meaningful if the underlying causal claims have survived independent verification. Topological encoding of unverified claims would produce topological features with no epistemic grounding. The VCS provides the adversarial testing (challenge mechanism, stake-based verification) that gives the topological features their epistemic weight.

The AetherNet protocol is, to our knowledge, the only operational system that produces data satisfying all five requirements simultaneously. The topological epistemic field construction is therefore specific to verified causal structures rather than a general technique applicable to any graph.

6. Implications

6.1 For Storage and Query Efficiency

The topological epistemic field provides a representation that separates structural queries (answered by topology) from authentication queries (answered by sheaf stalks). Structural queries — which constitute the majority of operational queries on a knowledge substrate — become topological computations that operate on the persistent homology rather than on the full graph. At sufficient scale, the persistent homology is a more compact representation than the full graph because persistent features compress redundant structure. This suggests that topological encoding could provide storage and query efficiency gains for large-scale VCS instances, though the precise gains depend on the structural properties of specific VCS instances.

6.2 For AI Training

The topological epistemic field is a continuous geometric object that AI models could potentially ingest more efficiently than discrete graph records. If AI training pipelines could operate on topological representations directly — reading the persistent homology and path homology as continuous features rather than processing symbolic graph records — the efficiency gains from eliminating the symbolic-to-geometric conversion could be significant. We do not quantify this gain in the present paper; it is a direction for future empirical investigation.

6.3 For Verification Integrity

Sheaf cohomological consistency detection (Theorem 1) provides a verification mechanism that operates at a fundamentally different level than individual signature checking. Current verification checks each attestation independently. Sheaf cohomology checks the structural consistency of attestations collectively. This can detect coordination patterns, circular attestation schemes, and other forms of verification fraud that are invisible to per-attestation checks.

This connects to the process topology framework introduced in prior work (Schreiber / AetherNet Labs, 2026b). Process topology detects coordinated verification through statistical analysis of metadata. Sheaf cohomology detects it through algebraic analysis of attestation structure. The two approaches are complementary and could be combined for robust multi-layered fraud detection.

6.4 For the Nature of Causation

If discrete causal information can be faithfully encoded as topology of a continuous field, this suggests that causation may be fundamentally a topological property of an underlying continuous structure, with the discrete graph being a projection rather than the ground truth. This observation connects to causal set theory in quantum gravity (Sorkin et al.), where spacetime is theorized to be fundamentally a discrete poset (essentially a DAG) from which continuous manifold structure emerges. Our construction performs the reverse operation: starting from a verified discrete causal structure and recovering a continuous field whose topology preserves the causal invariants. Whether this is merely a useful mathematical technique or a reflection of something deeper about the relationship between discrete causation and continuous structure is an open question that extends beyond the scope of this paper.

7. Prototype Implementation

We describe a prototype implementation of the topological epistemic field construction on the AetherNet testnet. The prototype operates on a subdomain of the testnet containing approximately 100-500 verified computations with their causal structure.

Step 1: Epistemic filtration. Extract the VCS DAG from the testnet. Construct the simplicial complex by assigning a 0-simplex to each verified computation and a 1-simplex to each causal edge. Higher-dimensional simplices are constructed from cliques in the causal graph (three mutually dependent computations form a 2-simplex, etc.).

Step 2: Persistent homology. Compute the DAG-indexed persistent homology using the algorithm of Chambers and Letscher. For the prototype scale (hundreds of nodes), the O(n{sup("4")}) complexity is tractable. Record the persistence diagram.

Step 3: Persistent path homology. Compute the persistent path homology using the confidence filtration (edges filtered by compound confidence level). Record the directed persistence features.

Step 4: Attestation sheaf. Construct the cellular sheaf by attaching cryptographic attestations to vertices and edges of the simplicial complex. Compute H{sup("0")} (global sections) and H{sup("1")} (consistency obstructions).

Step 5: Validation. Verify that all causal invariants of the original VCS are recoverable from the topological representation. Specifically: reconstruct the causal edges from the persistence diagram, verify that causal direction is preserved in the path homology, and confirm that sheaf stalks faithfully carry all attestations.

Expected outcome: The prototype demonstrates that the construction works on real VCS data produced by an operational protocol. It does not address scalability to production VCS instances (millions of nodes), which would require algorithmic advances in incremental persistent homology and sheaf computation.

8. Limitations and Open Questions

Computational scalability. The $O(n^4)$ complexity of DAG-indexed persistent homology is tractable for prototype-scale instances but prohibitive for production-scale VCS with millions of nodes. Scalable deployment would require approximation algorithms, incremental computation, or hardware acceleration. This is an active area of research in computational topology but no solution currently exists at the scale the VCS would require.

Cryptographic attestation encoding. The construction does not fully unify discrete and continuous representations — cryptographic attestations remain as discrete sheaf stalks. A complete unification would require a cryptographic primitive whose security reduces to topological invariants. No such primitive is known. The hybrid architecture is therefore a pragmatic solution rather than a complete theoretical unification.

Information capacity. Topological invariants are coarse: non-isomorphic complexes can share the same homology. The persistent homology is finer than homology alone but still not a complete invariant. This means the topological encoding is lossy for fine-grained structural details. The practical impact of this loss depends on whether the lost details are relevant to downstream queries.

Empirical validation. The prototype has not been implemented at the time of writing. The construction is mathematically well-defined but its practical utility remains to be demonstrated. The claims about query efficiency, storage compression, and training efficiency are theoretical projections, not empirical results.

Relationship to causal set theory. The connection to causal set theory in quantum gravity is suggestive but not formalized. Whether the topological epistemic field has any formal relationship to the emergence of spacetime from discrete causal structure is an open question that would require collaboration with the causal set community to investigate.

9. Related Work

DAG persistence. Chambers and Letscher (2014) define persistent homology over DAG-indexed filtrations. We apply their construction to VCS DAGs, which provides the specific indexing structure their theory requires.

Persistent path homology. Chowdhury and Memoli (2017), building on Grigoryan, Lin, Muranov, and Yau. We use their directed persistent homology to encode causal direction as a topological invariant.

Cellular sheaves on graphs. Ghrist (2014), Hansen and Ghrist (2019), Robinson (2016), Gebhart, Hansen, and Schrater (2021). We apply cellular sheaves to carry cryptographic attestation data on the topological substrate, a novel application of the framework.

Sheaf-theoretic causal analysis. D'Acunto et al. (2025) use sheaf theory for networks of causal abstractions. Pinzani (thesis) applies sheaf theory to indefinite causality. We apply sheaf cohomology specifically to detect verification inconsistencies in cryptographically attested causal structures.

Information Field Theory. Ensslin et al. (2022, 2025) develop Bayesian inference over continuous fields. Our epistemic field construction is related but distinct: we use topological invariants rather than Bayesian inference to represent knowledge, and our field is constructed from verified data rather than from noisy observations.

Implicit Neural Representations. INRs parameterize continuous fields via neural networks. Our field construction is algebraic rather than neural, but INRs could provide the computational substrate for evaluating epistemic fields at query time.

Causal set theory. Sorkin et al. The connection between discrete causal posets and continuous manifolds in quantum gravity is conceptually related to our construction. Our work performs the analogous operation for epistemic rather than physical causation.

10. Conclusion

We have proposed a sheaf-theoretic construction that encodes the causal structure of a Verified Causal Structure as topological invariants of a continuous epistemic field, while carrying cryptographic attestations in discrete sheaf stalks with cohomology-enforced consistency. The construction preserves all causal invariants of the original VCS (Theorem 2) and provides a novel consistency detection mechanism via sheaf cohomology (Theorem 1).

The construction requires specific properties that the VCS provides: causal direction, cryptographic attestation, compound confidence, append-only immutability, and adversarial testing. These requirements make the construction specific to verified causal structures rather than a general technique for arbitrary knowledge graphs.

The practical impact of this construction depends on answers to open questions: whether the computational costs can be made tractable at production scale, whether the storage and query efficiency gains are significant in practice, and whether AI training can benefit from topological rather than symbolic knowledge representations. A prototype implementation on the AetherNet testnet will provide initial answers to these questions.

If the construction proves practically viable, it would represent a new approach to knowledge representation: one that unifies discrete verification precision with continuous geometric expressiveness through the mathematical framework of algebraic topology. The tools required — persistent homology, path homology, cellular sheaves — are mature mathematical disciplines with existing computational implementations. What is new is their application to cryptographically verified causal structures, a substrate that did not previously exist.

Contact: research@aethernetlabs.com

References

Chambers, E. and Letscher, D. (2014). Persistent Homology over Directed Acyclic Graphs. arXiv:1407.2523.

Chowdhury, S. and Memoli, F. (2017). Persistent Path Homology of Directed Networks. PODS 2018.

D'Acunto, L., et al. (2025). Networks of Causal Abstractions: A Sheaf-theoretic Framework. arXiv:2509.25236.

Edelsbrunner, H., Letscher, D., and Zomorodian, A. (2002). Topological Persistence and Simplification. Discrete & Computational Geometry, 28(4), 511-533.

Ensslin, T., et al. (2022). Information Field Theory and Artificial Intelligence. Entropy, 24(3), 374.

Gebhart, T., Hansen, J., and Schrater, P. (2021). Knowledge Sheaves: A Sheaf-Theoretic Framework for Knowledge Graph Embedding. arXiv:2110.03789.

Ghrist, R. (2014). Elementary Applied Topology. CreateSpace.

Grigoryan, A., Lin, Y., Muranov, Y., and Yau, S.T. (2014). Path Complexes and Their Homologies. Journal of Mathematical Sciences, 248(5), 564-599.

Hansen, J. and Ghrist, R. (2019). Toward a Spectral Theory of Cellular Sheaves. JACT, 3(4), 315-358.

Robinson, M. (2016). Sheaves Are the Canonical Data Structure for Sensor Integration. Information Fusion, 36, 208-224.

Schreiber, M. / AetherNet Labs (2026). Verified Causal Structures: A Framework for Counterfactual Computation over Empirically Attested Causal Graphs. Technical Report.

Schreiber, M. / AetherNet Labs (2026b). Negative Knowledge as a Formal Object in Verified Causal Structures. Technical Report.

Sorkin, R.D. (2003). Causal Sets: Discrete Gravity. arXiv:gr-qc/0309009.

Zomorodian, A. and Carlsson, G. (2005). Computing Persistent Homology. Discrete & Computational Geometry, 33(2), 249-274.