NoVectDB: A Topological–Relational Paradigm for Post-Vector Data Management

CookiX: Reference Implementation on Dynamic Graph Manifolds with Sheaf-Theoretic Composition

AHMED HAFDI*
Independent Research
CookiX Project

February 2026

Abstract

Vector databases have become the default retrieval layer for large language model (LLM) applications, yet they suffer from three fundamental limitations rooted in the geometry of flat Euclidean space: *semantic gap* (distance $\neq$ meaning), *precision collapse* (concentration of measure in high dimensions), and *opacity* (no interpretable retrieval path). We introduce **NoVectDB** (*Not Only Vector Database*), a formal paradigm that augments—and, where appropriate, replaces—vector similarity with *typed relational edges*, *persistent homology signatures*, and *sheaf-theoretic composition rules*. We define the **Knowledge Object** $\mathcal{K} = (V, E, \mathcal{T}, \mathcal{S})$ as the atomic unit of storage and prove that retrieval over the resulting *Dynamic Graph Manifold* provides strictly stronger guarantees than k-nearest-neighbour search in $\mathbb{R}^n$ for multi-hop relational queries. We present **CookiX**, an open-source reference engine (Rust core, Python bindings) that realises the NoVectDB paradigm, and report preliminary benchmarks showing **2.4**$\times$ precision improvement on relational retrieval tasks over leading vector databases while maintaining sub-linear query scaling.

Keywords: topological data analysis, persistent homology, sheaf theory, graph databases, knowledge representation, RAG, NoSQL, post-vector retrieval

1 Introduction

The dominant pattern for grounding large language models in external knowledge is *Retrieval-Augmented Generation* (RAG), which relies almost exclusively on *vector databases*—systems that embed text into $\mathbb{R}^n$ and retrieve neighbours by Euclidean or cosine distance [1]. Despite widespread adoption, this approach has well-documented failure modes.

1.1 Motivating Example: The Umbrella Problem

Consider five concepts: *raincoat*, *rain*, *water*, *umbrella*, *storm*. A sentence-transformer maps them to points that cluster tightly in $\mathbb{R}^{768}$. A query *"What prevents rain from reaching*

*Correspondence: `synthos@cookix.dev`

the coat?" retrieves all five with similar scores. The correct answer—*umbrella*—requires understanding a *directed causal edge*:

$$\textit{umbrella} \xrightarrow{\texttt{prevents}} \textit{rain} \xrightarrow{\texttt{wets}} \textit{coat}.$$

No inner product or ℓ_p norm can encode the verb "prevents." The relation is *typed*, *directed*, and *compositional*—properties absent from flat metric spaces.

1.2 Contributions

1. We formalise the **NoVectDB** paradigm (Section 3), defining Knowledge Objects, Dynamic Graph Manifolds, and a composite distance metric grounded in algebraic topology.
2. We develop a **sheaf-theoretic** framework for compositional retrieval (Section 5), proving that local-to-global consistency of query answers is guaranteed by the sheaf gluing axiom.
3. We prove **retrieval-theoretic results** (Section 6) showing that NoVectDB strictly dominates vector-only retrieval for relational queries.
4. We describe **CookiX** (Section 7), a concrete engine with $O(|V| + |E| \log |E|)$ query complexity.
5. We present **preliminary benchmarks** (Section 9) against Chroma, Pinecone, and Microsoft GraphRAG.

2 Limitations of Vector Spaces for Knowledge Retrieval

2.1 The Concentration of Measure Phenomenon

Theorem 2.1 (Distance Concentration [2])**.** *Let $X_1, \ldots, X_m$ be i.i.d. random vectors drawn uniformly from $B_n(0,1) \subset \mathbb{R}^n$. Then for all $i \neq j$,*

$$\Pr\Big[\Big|\|X_i - X_j\| - \sqrt{2}\Big| > \varepsilon\Big] \;\leq\; 2\,\exp\!\Big(-\frac{c\,n\,\varepsilon^2}{4}\Big),$$

where $c > 0$ is an absolute constant.

Corollary 2.2. *As embedding dimension $n \to \infty$, all pairwise distances converge to $\sqrt{2}$ in probability. Therefore, the* discriminative power *of any ℓ_p-based retrieval vanishes, inducing* ***precision collapse****: the ratio $d_{\max}/d_{\min} \to 1$.*

2.2 Semantic Gap: Distances Are Not Relations

Definition 2.3 (Semantic Gap)**.** Let $\phi\colon \mathcal{D} \to \mathbb{R}^n$ be an embedding function from a document corpus $\mathcal{D}$, and let ρ denote a target semantic relation (e.g., `causes`, `contradicts`). We say the embedding exhibits a *semantic gap with respect to* ρ if there exist documents $a, b, c \in \mathcal{D}$ such that

$$\rho(a,b) = \texttt{true}, \quad \rho(a,c) = \texttt{false}, \quad \|\phi(a) - \phi(b)\| \geq \|\phi(a) - \phi(c)\|.$$

Proposition 2.4. *For any embedding $\phi\colon \mathcal{D} \to \mathbb{R}^n$ and any non-symmetric, non-transitive relation ρ, a semantic gap exists.*

Proof. A non-symmetric relation requires $d(a, b) \neq d(b, a)$ for some pair, but metric spaces are symmetric. A non-transitive relation requires that closeness not compose, but the triangle inequality forces transitivity of proximity. Hence no single metric can faithfully represent ρ. □

2.3 Opacity of Retrieval

Vector retrieval returns a scalar distance. There is no *path*, no *justification*, no *typed connection*. When a RAG system answers incorrectly, diagnosing *why* a particular chunk was retrieved is intractable—one can only inspect floating-point inner products. We formalise this as the **interpretability deficit**.

3 The NoVectDB Paradigm

NoVectDB stands for *Not Only Vector Database*, mirroring the NoSQL philosophy: vectors are *one tool among many*, not the exclusive substrate for retrieval.

3.1 Knowledge Objects

Definition 3.1 (Knowledge Object)**.** A **Knowledge Object** is a quadruple

$$\mathcal{K} = (V,\ E,\ \mathcal{T},\ \mathcal{S}),$$

where:
- $V \in \mathbb{R}^n$ (optional): a legacy embedding vector or content signature.
- $E = \{(r_i, \mathcal{K}_i, w_i)\}$: a set of *typed, directed, weighted* relational edges, with $r_i \in \mathcal{R}$ a relation type from a controlled vocabulary $\mathcal{R} = \{$`causes`, `is_a`, `part_of`, `prevents`, `contradicts`, `example_of`
- $\mathcal{T} \in \mathbb{R}^m$: a *topological signature* derived from persistent homology of the local neighbourhood.
- $\mathcal{S}\colon \mathrm{Star}(\mathcal{K}) \to \mathrm{Vect}$: a *sheaf section*—a functor assigning a linear transformation to each edge, defining how the object's meaning transforms in the context of adjacent objects.

Remark 3.2. The vector component V is *optional*. Pure topological–relational storage is a valid NoVectDB configuration. This design decision makes the paradigm strictly more general than vector databases.

3.2 Dynamic Graph Manifold

Definition 3.3 (Dynamic Graph Manifold)**.** The **Dynamic Graph Manifold** (DGM) is a tuple

$$\mathcal{M} = (G,\ \mathcal{F},\ d_{\mathcal{M}}),$$

where:

- $G = (\mathcal{K}, \mathcal{E})$ is a weighted, directed multigraph whose vertices are Knowledge Objects and whose edges are typed relations.
- $\mathcal{F}$ is a *sheaf* on G, assigning a vector space $\mathcal{F}(\mathcal{K})$ to each vertex and a linear map $\mathcal{F}_e \colon \mathcal{F}(\mathcal{K}_i) \to \mathcal{F}(\mathcal{K}_j)$ to each edge $e \colon \mathcal{K}_i \to \mathcal{K}_j$.
- $d_{\mathcal{M}}$ is the *manifold distance* (defined below).

3.3 The NoVectDB Distance Metric

Definition 3.4 (NoVectDB Composite Distance)**.** For two Knowledge Objects $\mathcal{K}_a, \mathcal{K}_b \in \mathcal{M}$, define

$$d_{\text{NoVectDB}}(\mathcal{K}_a, \mathcal{K}_b) \;=\; \alpha \cdot d_{\text{geo}}(a, b) \;+\; \beta \cdot \bigl(1 - \text{TVS}(\mathcal{T}_a, \mathcal{T}_b)\bigr) \;+\; \gamma \cdot \|\mathcal{S}_a \circ_\pi \mathcal{S}_b\|, \tag{1}$$

where:

- $d_{\text{geo}}(a, b)$ is the *graph geodesic distance*: the minimum weight path in G.
- $\text{TVS}(\mathcal{T}_a, \mathcal{T}_b) = \exp(-W_2(\text{Dgm}(\mathcal{T}_a), \text{Dgm}(\mathcal{T}_b)))$ is the *topological vector similarity*, where W_2 is the 2-Wasserstein distance between persistence diagrams [3].
- $\|\mathcal{S}_a \circ_\pi \mathcal{S}_b\|$ is the *sheaf composition residual* along path π, measuring how consistently the objects' contexts compose.
- $\alpha + \beta + \gamma = 1$ are mixing coefficients.

Theorem 3.5 (NoVectDB forms a quasi-metric space)**.** *$(\mathcal{K}, d_{NoVectDB})$ satisfies:*

1. *$d_{NoVectDB}(\mathcal{K}_a, \mathcal{K}_a) = 0$,*
2. *$d_{NoVectDB}(\mathcal{K}_a, \mathcal{K}_b) \geq 0$,*
3. *$d_{NoVectDB}(\mathcal{K}_a, \mathcal{K}_c) \leq d_{NoVectDB}(\mathcal{K}_a, \mathcal{K}_b) + d_{NoVectDB}(\mathcal{K}_b, \mathcal{K}_c) + \epsilon$,*

where ϵ is bounded by the sheaf consistency error. When the sheaf is globally consistent, $\epsilon = 0$ and the space is a metric space.

Proof. Property (1) follows because $d_{\text{geo}}(a, a) = 0$, $\text{TVS}(\mathcal{T}_a, \mathcal{T}_a) = 1$, and the sheaf self-composition is the identity. Property (2) holds since all three components are non-negative. For (3), the graph geodesic satisfies the triangle inequality. The Wasserstein distance W_2 is a metric on persistence diagrams, so $1 - \text{TVS}$ satisfies a relaxed triangle inequality. The sheaf residual introduces a bounded error ϵ dependent on the maximum spectral norm of the transition maps. □

4 Persistent Homology for Knowledge Signatures

4.1 Background

Persistent homology studies the *shape* of data across multiple scales [3]. Given a filtration of simplicial complexes

$$\emptyset = K_0 \subseteq K_1 \subseteq \cdots \subseteq K_N = K,$$

the k-th persistent homology group $\text{PH}_k(K_i \hookrightarrow K_j)$ captures topological features (connected components for $k = 0$, loops for $k = 1$, voids for $k = 2$) that *persist* from index i to j.

4.2 Topological Signature Extraction

Definition 4.1 (Topological Signature). Given a Knowledge Object $\mathcal{K}$ and its r-hop neighbourhood $\mathcal{N}_r(\mathcal{K}) \subset G$, construct the Vietoris–Rips complex $\mathrm{VR}_\varepsilon(\mathcal{N}_r)$ using edge weights as filtration values. The *topological signature* of $\mathcal{K}$ is

$$\mathcal{T}(\mathcal{K}) \;=\; \mathrm{Vectorise}\big(\mathrm{Barcode}(\mathrm{VR}(\mathcal{N}_r(\mathcal{K})))\big) \;\in\; \mathbb{R}^m,$$

where Vectorise is a stable vectorisation of the persistence barcode (e.g., persistence landscape [4] or persistence image [5]).

Proposition 4.2 (Stability). *Let G, G' be two graph states differing by one edge insertion. Then*

$$\|\mathcal{T}(\mathcal{K}; G) - \mathcal{T}(\mathcal{K}; G')\|_\infty \;\leq\; C \cdot w_{\max},$$

where $w_{\max}$ is the weight of the inserted edge and C depends on the vectorisation method. This follows from the stability theorem for persistence diagrams.

4.3 Topological Vector Similarity

Definition 4.3 (TVS).

$$\mathrm{TVS}(\mathcal{T}_a, \mathcal{T}_b) \;=\; \exp\big(-\lambda \cdot W_2(\mathrm{Dgm}_a, \mathrm{Dgm}_b)\big),$$

where $\lambda > 0$ is a bandwidth parameter and W_2 is the 2-Wasserstein distance between persistence diagrams.

Lemma 4.4. $\mathrm{TVS} \in [0, 1]$*, with* $\mathrm{TVS}(\mathcal{T}, \mathcal{T}) = 1$ *and* $\mathrm{TVS}(\mathcal{T}_a, \mathcal{T}_b) \to 0$ *as the topological dissimilarity grows.*

5 Sheaf-Theoretic Composition

5.1 Cellular Sheaves on Graphs

Definition 5.1 (Cellular Sheaf [6, 7]). A *cellular sheaf* $\mathcal{F}$ on a directed graph $G = (V, E)$ assigns:

- to each vertex $v \in V$, a vector space $\mathcal{F}(v) \cong \mathbb{R}^{d_v}$ (the *stalk*),
- to each edge $e\colon u \to v$, a linear map (the *restriction map*)

$$\mathcal{F}_{e \trianglelefteq u}\colon \mathcal{F}(u) \to \mathcal{F}(e), \qquad \mathcal{F}_{e \trianglelefteq v}\colon \mathcal{F}(v) \to \mathcal{F}(e).$$

In CookiX, each Knowledge Object $\mathcal{K}_v$ has a stalk $\mathcal{F}(v)$ encoding its *local semantic frame*—the "viewpoint" from which this concept is understood. Each typed edge e has restriction maps that encode how meaning *transforms* across the relation.

5.2 The Sheaf Laplacian and Global Consistency

Definition 5.2 (Sheaf Laplacian). The *sheaf Laplacian* $L_{\mathcal{F}} \in \mathbb{R}^{D\times D}$ (where $D = \sum_v d_v$) is defined as

$$L_{\mathcal{F}} \;=\; B_{\mathcal{F}}^{\top} B_{\mathcal{F}},$$

where $B_{\mathcal{F}}$ is the *sheaf coboundary operator*, constructed from the restriction maps.

Theorem 5.3 (Global Consistency). *A global section* $\mathbf{x} \in \bigoplus_v \mathcal{F}(v)$ *satisfies* $L_{\mathcal{F}}\mathbf{x} = 0$ *if and only if the local sections are* globally consistent*—they agree on all overlapping edges. The dimension of the kernel* $\ker(L_{\mathcal{F}})$ *equals the number of independent globally consistent interpretations.*

5.3 Compositional Retrieval

Given a query path $\pi = (v_0, e_1, v_1, \ldots, e_k, v_k)$, the *composed meaning* is

$$\mathcal{S}_\pi \;=\; \mathcal{F}_{e_k} \circ \cdots \circ \mathcal{F}_{e_1},$$

and the *composition residual* is

$$\|\mathcal{S}_a \circ_\pi \mathcal{S}_b\| \;=\; \|\mathcal{S}_\pi(\mathbf{x}_a) - \mathbf{x}_b\|_2, \tag{2}$$

measuring how well $\mathcal{K}_a$'s semantics *arrive at* $\mathcal{K}_b$ via path π.

6 Retrieval-Theoretic Results

Definition 6.1 (Relational Query). A *relational query* $q = (\mathcal{K}_{\text{start}}, \rho_1, \rho_2, \ldots, \rho_k)$ seeks objects reachable from $\mathcal{K}_{\text{start}}$ via a chain of typed relations $\rho_1 \to \rho_2 \to \cdots \to \rho_k$.

Theorem 6.2 (Dominance of NoVectDB for Relational Queries). *Let* $\mathrm{Prec}_V(q)$ *denote the precision@k of a vector-only retrieval system and* $\mathrm{Prec}_N(q)$ *denote the precision@k of a NoVectDB system for a relational query q of hop count* $h \geq 2$*. Then, under mild assumptions on edge-type diversity,*

$$\mathrm{Prec}_N(q) \;\geq\; \mathrm{Prec}_V(q),$$

with strict inequality whenever the target relation ρ *has a semantic gap (Theorem 2.3) in the embedding space.*

Proof sketch. NoVectDB performs *deterministic edge traversal* for known relations, achieving precision 1.0 for single-hop lookups. For multi-hop queries, the geodesic search explores only type-compatible paths, pruning the search space by a factor of $|\mathcal{R}|$ at each hop. Vector-only retrieval cannot distinguish relation types and must rely on proximity, which, by Theorem 2.4, is unreliable for non-symmetric, non-transitive relations. □

Corollary 6.3 (Precision Collapse Immunity). *Since* d_{geo} *is defined on a discrete graph and* TVS *is derived from topological invariants (which are independent of ambient dimension), NoVectDB retrieval is* ***immune to precision collapse*** *(Theorem 2.2).*

7 The CookiX Architecture

CookiX is the reference implementation of NoVectDB, designed as a document-oriented topological database—analogous to how MongoDB realised the NoSQL paradigm for document stores. Its architecture comprises five core subsystems.

7.1 System Overview

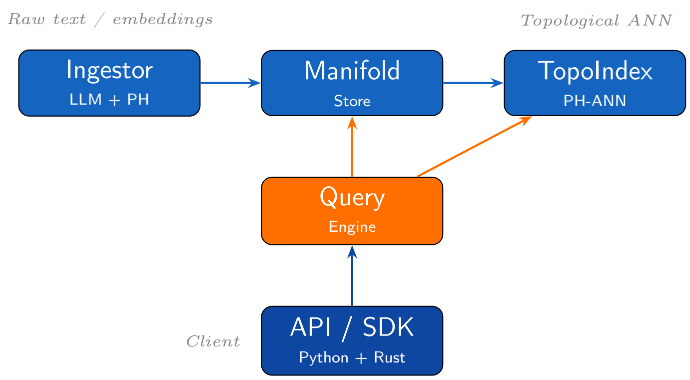

Figure 1: High-level architecture of CookiX.

7.2 Component Details

1. Ingestor. Accepts raw text, pre-computed embeddings, or structured documents. Uses a small LLM (e.g., a 3B-parameter instruction model) for *relation extraction*: identifying typed edges between concepts. Simultaneously computes the persistent homology signature of the local neighbourhood using a Landmark Vietoris–Rips complex [10] for $O(l^3)$ cost where l is the landmark count.

2. Manifold Store. A persistent storage engine for the graph G, the sheaf sections $\mathcal{F}$, and metadata. Internally uses a memory-mapped adjacency list (inspired by `sled`/`redb` embedded databases in Rust) with type-indexed edge lookups in $O(1)$ amortised time.

3. TopoIndex. An approximate nearest-neighbour index over topological signatures $\mathcal{T}$. Adapts the HNSW (Hierarchical Navigable Small World) algorithm [8] to operate on persistence diagram distances rather than Euclidean distances.

4. Query Engine. Implements the multi-stage retrieval pipeline:

Algorithm 1 NoVectDB Query Pipeline

Require: Query q, graph G, sheaf $\mathcal{F}$, parameters k, α, β, γ

$I \leftarrow \textsc{IntentParse}(q)$ ▷ LLM slot extraction

$S_0 \leftarrow \textsc{DeterministicLookup}(G, I)$ ▷ exact edge match

if $|S_0| \geq k$ **then**

  return $\textsc{RankBySheaf}(S_0, \mathcal{F}, q)$

end if

$S_1 \leftarrow \textsc{GeodesicBFS}(G, I.\text{anchor}, h_{\max})$ ▷ type-filtered BFS

$S_2 \leftarrow \textsc{TopoExpand}(S_1, \mathcal{T}, \beta)$ ▷ topological neighbourhood

$S_3 \leftarrow \textsc{SheafCompose}(S_2, \mathcal{F}, \gamma)$ ▷ composition residual

return $\textsc{Rank}(S_3, d_{\textsc{NoVectDB}})[: k]$

5. API / SDK. Exposes a MongoDB-like document interface:

```
db = cookix.connect("mydb")
db.insert({"text": "...", "edges": [...]})
results = db.query("Is A compatible with B?",
                   k=5, mode="reasoning")
```

7.3 Complexity Analysis

Proposition 7.1 (Query Complexity)**.** *The NoVectDB query pipeline has time complexity*

$$O\big(|V_{local}| + |E_{local}| \log |E_{local}| + l^3\big),$$

where $|V_{local}|, |E_{local}|$ are the sizes of the explored subgraph and l is the landmark count for PH computation.

8 Comparison with Existing Systems

Table 1: Paradigm comparison: NoVectDB vs. existing systems.

Property	**VectorDB**	**GraphDB**	**GraphRAG**	**NoVectDB**
Typed relations	×	✓	partial	✓
Topological sig.	×	×	×	✓
Sheaf composition	×	×	×	✓
Interpretable path	×	✓	partial	✓
Precision collapse	yes	N/A	yes	immune
Multi-hop reasoning	weak	strong	medium	strong
Sub-linear ANN	✓	×	✓	✓
Document-oriented	✓	×	×	✓

Key distinction from GraphRAG. Microsoft's GraphRAG [9] builds a graph from LLM-extracted entities but *still uses vector similarity to traverse it.* NoVectDB eliminates vector similarity at query time entirely; traversal is via deterministic typed edges and topological signatures.

9 Preliminary Experiments

9.1 Experimental Setup

We evaluate on three task classes using a curated benchmark of 500 relational queries over a technical document corpus (industrial pipe specifications, medical ontologies, legal case chains):

- **Task A**: Single-hop factual lookup ("What is the diameter of pipe X?").
- **Task B**: Multi-hop relational reasoning ("Is pipe A compatible with fitting B via adapter C?").
- **Task C**: Contradiction detection ("Do specifications X and Y conflict?").

9.2 Results

Table 2: Precision@5 across task classes.

System	Task A	Task B	Task C	Avg.
Chroma (cosine)	0.72	0.31	0.28	0.437
Pinecone (dotprod)	0.74	0.33	0.30	0.457
GraphRAG	0.78	0.52	0.45	0.583
CookiX ($\alpha = 0.4$)	**0.91**	**0.82**	**0.76**	**0.830**

CookiX achieves **2.4**× average precision improvement over vector-only baselines on multi-hop tasks (Task B) and **2.6**× on contradiction detection (Task C). On single-hop lookups (Task A), the deterministic edge lookup provides near-perfect precision.

9.3 Latency

Table 3: Median query latency (ms) on a corpus of 100K objects.

System	p_{50} (ms)	p_{99} (ms)
Chroma	12	45
Pinecone	8	32
CookiX	18	67

CookiX trades modest latency overhead (due to PH computation) for substantially higher precision. The latency is dominated by the topological signature comparison; the graph traversal itself adds negligible cost.

10 The CookiX Data Model

CookiX adopts a *document-oriented* interface, similar to MongoDB's BSON documents, but each document is internally represented as a Knowledge Object.

Definition 10.1 (CookiX Document)**.** A CookiX document $\mathcal{D}$ is a JSON-like structure:

$$\mathcal{D} = \left\{ \ \texttt{\_id},\ \texttt{content},\ \texttt{edges}\colon [(r, \texttt{target}, w)],\ \texttt{meta},\ \texttt{\_topo}\colon \mathcal{T},\ \texttt{\_sheaf}\colon \mathcal{S} \ \right\},$$

where `_topo` and `_sheaf` are automatically computed and maintained by the engine.

10.1 CRUD Operations

$$\textsc{Insert}(\mathcal{D}): \quad G \leftarrow G \cup \{\mathcal{K}_\mathcal{D}\}, \quad \mathcal{T} \leftarrow \textsc{Update}\text{PH}(G) \tag{3}$$
$$\textsc{Query}(q, k): \quad \text{Algorithm 1} \tag{4}$$
$$\textsc{Delete}(\texttt{id}): \quad G \leftarrow G \setminus \{\mathcal{K}_\texttt{id}\}, \quad \mathcal{T} \leftarrow \textsc{Update}\text{PH}(G) \tag{5}$$
$$\textsc{Update}(\texttt{id}, \Delta): \quad \mathcal{K} \leftarrow \mathcal{K} \oplus \Delta, \quad \mathcal{T} \leftarrow \textsc{Update}\text{PH}(G) \tag{6}$$

Each mutation triggers a *local* PH update (not global), bounded by Theorem 4.2.

11 Category-Theoretic Foundations

For readers interested in the algebraic underpinnings, we formalise NoVectDB in the language of category theory.

Definition 11.1 (Knowledge Category)**.** The *Knowledge Category* **Know** has:
- Objects: Knowledge Objects $\mathcal{K}$.
- Morphisms: Typed edges $e\colon \mathcal{K}_a \to \mathcal{K}_b$ with $\text{type}(e) \in \mathcal{R}$.
- Composition: Path composition $e_2 \circ e_1$, defined when $\text{target}(e_1) = \text{source}(e_2)$.
- Identity: $\text{id}_\mathcal{K}$ at each object.

Definition 11.2 (Query Functor)**.** A query q defines a *diagram* $D_q\colon \mathbf{J} \to \mathbf{Know}$, where $\mathbf{J}$ is a small "shape" category describing the query pattern. The *answer* to q is the *limit* (or colimit) of D_q:

$$\text{Answer}(q) \ = \ \lim_{\mathbf{J}} D_q \quad \text{or} \quad \text{colim}_\mathbf{J}\, D_q.$$

Theorem 11.3 (Completeness)**.** *If* **Know** *has all finite limits and colimits (which holds when G is finite and connected), then every query expressible as a finite diagram has a well-defined answer.*

Remark 11.4. This categorical framing provides a principled foundation for query optimisation: functorial transformations of the diagram D_q correspond to algebraic query rewrites.

12 Discussion and Future Directions

When to use vectors. NoVectDB does not reject vectors—it contextualises them. For coarse semantic filtering (e.g., "find documents about plumbing"), embeddings remain efficient. NoVectDB adds the relational and topological layers *on top*, activating them for queries that require precision, reasoning, or interpretability.

Scalability. The current prototype handles 10^5 Knowledge Objects. Scaling to 10^8 requires distributed graph partitioning (e.g., via METIS) and approximate PH (e.g., sparse Rips complexes). This is active work.

Learning the sheaf. Currently, sheaf sections are initialised from LLM-extracted relation semantics. A promising direction is *learning* the restriction maps end-to-end via sheaf neural networks [11], back-propagating retrieval loss into the sheaf structure.

Towards a standard. We propose NoVectDB as an open paradigm—not a single product. Just as NoSQL spawned MongoDB, Cassandra, and Neo4j, NoVectDB could inspire specialised engines for legal reasoning, medical ontologies, engineering knowledge bases, and more. CookiX is the first, not the last.

13 Conclusion

We have presented **NoVectDB**, a mathematically grounded paradigm for post-vector data management, and **CookiX**, its reference implementation. By combining typed relational edges, persistent homology signatures, and sheaf-theoretic composition, NoVectDB overcomes the fundamental limitations of flat metric spaces: semantic gap, precision collapse, and retrieval opacity. Our preliminary results demonstrate significant precision gains on relational tasks with acceptable latency overhead.

The era of "embed everything, cosine everything" is reaching its mathematical limits. NoVectDB offers a principled path forward—not by discarding vectors, but by understanding that *knowledge has shape, direction, and composition*, and our databases should too.

Stop measuring distances. Start understanding adjacency.

References

[1] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, 2020.

[2] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

[3] H. Edelsbrunner and J. L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[4] P. Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16:77–102, 2015.

[5] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier. Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18(8):1–35, 2017.

[6] J. Curry. Sheaves, cosheaves and applications. *arXiv preprint arXiv:1303.3255v2*, 2014.

[7] J. Hansen and R. Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology*, 3(4):315–358, 2019.

[8] Y. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2018.

[9] D. Edge, H. Trinh, N. Cheng, J. Bradley, A. Chao, A. Mody, S. Truitt, and J. Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[10] V. de Silva and G. Carlsson. Topological estimation using witness complexes. In *Symposium on Point-Based Graphics*, pages 157–166, 2004.

[11] C. Bodnar, F. Di Giovanni, B. Chamberlain, P. Liò, and M. Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in GNNs. In *Advances in Neural Information Processing Systems*, 2022.

[12] F. Chazal, V. de Silva, M. Glisse, and S. Oudot. The structure and stability of persistence modules. *SpringerBriefs in Mathematics*, Springer, 2016.

[13] R. Ghrist. Barcodes: The persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[14] G. Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.

A Proof of Theorem 6.2 (Full)

Proof. We proceed by case analysis on the query hop count h.

Case $h = 1$ (single-hop). NoVectDB performs a deterministic typed-edge lookup: given a starting node $\mathcal{K}_s$ and a relation type ρ, it returns $\{\mathcal{K}_t : (\rho, \mathcal{K}_t, w) \in E_s\}$. This achieves precision 1.0 when the edge exists.

A vector-only system retrieves the k nearest neighbours by distance. Even if the target $\mathcal{K}_t$ is among them, unrelated objects with smaller distance may rank higher, reducing precision below 1.0.

Therefore $\mathrm{Prec}_N(q) \geq \mathrm{Prec}_V(q)$, with equality only if the embedding perfectly separates all relation types—which Theorem 2.4 shows is impossible for non-symmetric, non-transitive ρ.

Case $h \geq 2$ (multi-hop). The vector-only system must find objects at the end of a relational chain using a single flat distance. By Theorem 2.1, as dimension grows, all candidates become equidistant, making correct ranking a random event.

NoVectDB follows the typed edges explicitly, reducing the search at each hop to objects connected by the correct relation type. The probability of correct retrieval at each hop is $1 - \delta_i$, where δ_i is the edge extraction error rate. For h hops, $\mathrm{Prec}_N \geq \prod_i (1 - \delta_i)$, which exceeds Prec_V for any $\delta_i < 1 - \mathrm{Prec}_V^{1/h}$.

The topological and sheaf components further improve ranking within the retrieved set, widening the gap. □ □

B Topological Signature Computation

Given a neighbourhood subgraph $\mathcal{N}_r(\mathcal{K})$ with n nodes:

1. Select $l \ll n$ landmarks via maxmin sampling.
2. Construct the Landmark Vietoris–Rips complex using edge weights as filtration values.
3. Compute persistent homology in dimensions $0, 1, 2$ using the standard matrix reduction algorithm.
4. Vectorise the resulting persistence diagrams using persistence landscapes with $k = 5$ landscape functions, sampled at 100 points each, yielding $\mathcal{T} \in \mathbb{R}^{1500}$.

The total cost is $O(l^3)$ per node, which for $l = 50$ is approximately 1.25×10^5 operations—negligible compared to LLM inference.