QnapsiZ: A Topology-Aware Sparse Volumetric External Memory with Auditable Retrieval

QnapsiZ Research Group
https://github.com/VigilanteSystems/QnapsiZ

Abstract

Large language models generate coherent text but lack reliable external memory: retrieved information is mixed with parametric knowledge without a formal grounding guarantee. We introduce QnapsiZ, a topology-aware geometric external memory that stores concepts as Gaussian splats in a sparse voxel grid and uses persistent homology—specifically the first Betti number β_1—as a computable truth criterion. Concepts whose local neighborhood has $\beta_1 = 0$ are geometrically consistent; $\beta_1 > 0$ indicates topological loops that correspond to contradictory or hallucinated content. The system couples a spherical-harmonic bridge to a symbolic left-brain reasoner and achieves signal-to-noise consolidation ratios exceeding 5,000× via adaptive Ricci flow, topological contradiction detection at $\beta_1 > 0$ with 94.7%±1.2% precision, and cross-domain analogy retrieval with 76%±2% topologically valid responses. Experiments on corpora up to 5,000 concepts at 512^3 voxel resolution demonstrate that geometric external memory with topological invariants provides a principled path toward auditable retrieval in large language models.

Keywords: external memory, topological data analysis, persistent homology, sparse voxel grids, retrieval-augmented generation, hallucination detection

1 Introduction

Large language models (LLMs) achieve remarkable fluency across a wide range of natural language tasks [1–4], yet they remain vulnerable to a fundamental problem: they cannot distinguish what they know from what they invent. When an LLM retrieves a fact from external memory—via retrieval-augmented generation (RAG) [5,6] or a memory-augmented architecture [7–9]—the result has no formal guarantee of geometric or logical consistency. The retrieved content may be plausible but false, and the model lacks a mechanism to detect that its own parametric knowledge contradicts the retrieved signal [10–12], leading to expanding taxonomies of hallucinations and structural neuron activations [13, 14].

Existing approaches to trustworthy retrieval fall into three categories. Memory-augmented networks [7, 15, 16] store and retrieve from differentiable read/write mechanisms but provide no consistency guarantee on the stored content. Retrieval-augmented generation [5, 17–19] augments LM prompts with relevant documents retrieved by embedding similarity, yet similarity alone cannot distinguish a coherent fact from a coherent-sounding falsehood. Topological data analysis [20–23] has been used to analyze latent spaces post-hoc [24, 25], but not as an online memory integrity check.

Present work. We introduce QnapsiZ, a geometric external memory that stores concepts as Gaussian density splats in a sparse voxel grid and uses persistent homology to provide auditable retrieval. The key insight is that the first Betti number β_1—the rank of the one-dimensional homology group—measures the number of topological loops in the local concept neighborhood. A loop in semantic space models a contradiction: the geometry suggests "these concepts cannot all be true simultaneously."

Our system makes four specific contributions:

1. **Topological truth criterion.** We define the Homological Consistency Criterion (HCC): a concept is geometrically grounded iff $\beta_1 = 0$ in its local voxel neighborhood. This is the

first use of persistent homology as an online memory integrity check in an external memory system.

2. **Dual-brain architecture.** A spherical-harmonic (SH) encoder with holographic reduced representations (HRR) [26] bridges a symbolic left-brain reasoner and a sparse volumetric right brain built on NVIDIA fVDB [27], enabling both semantic richness and geometric grounding.

3. **Adaptive consolidation.** A manifold curvature flow (MCF) engine [28] with Ricci-adaptive thresholding separates signal from noise, achieving 5,000× consolidation ratios without catastrophic forgetting.

4. **Closed-loop feedback.** A Gaussian process preference model with automatic relevance determination and spectral mixture kernels [29] provides continuous quality feedback, evolved via CMA-ES [30] to discover domain-specific rigidity profiles.

We validate these claims on a corpus of 5,000 procedurally generated concepts spanning multiple domains, comparing against FAISS [31] and HiPPO-RAG baselines. QnapsiZ detects injected contradictions with $94.7\% \pm 1.2\%$ precision, achieves a median β_1 induction of 0.52 per contradiction pair, and consolidates signal from noise at a ratio exceeding 5,000× after five sleep cycles. Cross-domain analogy retrieval produces $76\% \pm 2\%$ topologically valid responses, and the outbound Birkenbihl synthesis pipeline generates creative associations with measurable semantic drift.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 describes the system architecture. Sections 4–8 detail the memory pipeline. Section 9 presents experimental results. Section 10 discusses limitations and broader impact, and Section 11 concludes.

2 Related Work

2.1 External Memory for Neural Networks

The idea of augmenting neural networks with explicit external memory dates to the Neural Turing Machine [15], which coupled a controller network with a differentiable memory matrix. The Differentiable Neural Computer [7] extended this with temporal linkages and free recall. Weston et al. [8] introduced Memory Networks with explicit read/write operations, later simplified to end-to-end differentiable form [16]. Key-Value Memory Networks [32] added structured retrieval by separating key addressing from value reading.

Retrieval-Augmented Generation (RAG) [5] changed the paradigm by using a pre-trained retriever (typically dense passage retrieval with a FAISS index [31]) to augment LM prompts at inference time. REALM [17] jointly trained retriever and generator. RETRO [18] scaled retrieval to trillions of tokens with chunked cross-attention. kNN-LM [19] interpolated next-token distributions with a nearest-neighbor cache. Recent frameworks have also integrated brain-like sleep-wake consolidation phases to mitigate catastrophic forgetting in continual learning [33], and proposed continuous partial differential equation memory fields to preserve context in agent workflows [34].

Limitation. All these systems retrieve by similarity in embedding space. Similarity provides no guarantee of logical or geometric consistency: a retrieved neighbor may be semantically close but factually contradictory. QnapsiZ addresses this by adding a topological check—$\beta_1 = 0$—as an auditable consistency gate before any retrieved content is surfaced.

2.2 Topological Data Analysis for Machine Learning

Persistent homology [20, 21] computes multiscale topological invariants—Betti numbers, persistence diagrams, and landscapes—from point cloud or cubical complex data. It has been applied to shape analysis, sensor networks [35], and high-dimensional probability estimation [24]. Otter et al. [36] provided a comprehensive survey of computational methods. Statistical TDA [37] introduced persistence landscapes for hypothesis testing.

Prior work has used TDA to analyze neural network representations post-hoc [38–41], including recent efforts using zigzag persistence to detect hallucinations [42], but not as an online integrity monitor during memory operations. QnapsiZ's innovation is to compute β_1 on every memory write and retrieval as a *runtime* consistency signal, not an offline analysis tool.

2.3 Geometric and Hyperbolic Embeddings

Geometric deep learning [43] studies how symmetry and geometry constrain representation learning. Hyperbolic embeddings [44–46] capture hierarchical structure by embedding into negatively curved spaces. These approaches demonstrate that geometry can encode structure that Euclidean embeddings miss, but they do not address the inverse problem: given an embedding, can we verify its consistency?

QnapsiZ inverts this relationship. Instead of learning a geometric embedding that preserves structure, we project arbitrary text embeddings into a fixed 3D voxel grid and then use topological invariants of that grid to audit consistency. The geometric representation is the memory, not just a visualization tool.

2.4 Vector Symbolic Architectures

Holographic reduced representations [26, 47] and vector symbolic architectures [48] use circular convolution to bind and unbind symbolic structures in distributed vector spaces. QnapsiZ uses HRR binding in the corpus callosum bridge to compose concept embeddings with learned context vectors before projection into the voxel grid.

3 System Architecture

QnapsiZ is organized as a dual-brain architecture with four components: a shared text encoding frontend, the Symbolic Logic Engine (Left Brain) for explicit relational knowledge, the Geometric Intuition Engine (Right Brain) for analogical spatial reasoning, and the Corpus Callosum bridge that routes information between hemispheres and manages query orchestration. Figure 1 provides a high-level overview.

3.1 Text Encoding Frontend

Before input reaches either hemisphere, it passes through a shared text encoding frontend comprising two stages. First, the `InboundExpert` employs a SentenceTransformer to produce a raw 1024-dimensional embedding from natural language input. Second, the Length-Scaled Softplus Attention with Re-weighting (LSSAR) transformer [1, 49] sharpens this embedding using a custom attention mechanism that prevents the smoothing problem where later tokens attend uniformly to all prior tokens, which would cause manifold collapse. Given a sequence of embeddings $\mathbf{X} \in \mathbb{R}^{L \times d}$, LSSAR computes:

$$\lambda(i) = \frac{1}{\ln(i+1)}, \qquad A_{ij} = \mathrm{softmax}_j\big(\lambda(i)\, \mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k}\big) \tag{1}$$

where i is the position index and $\lambda(i)$ is the length-scaling coefficient. The resulting sharpened embedding is passed to the Corpus Callosum for routing to either or both hemispheres.

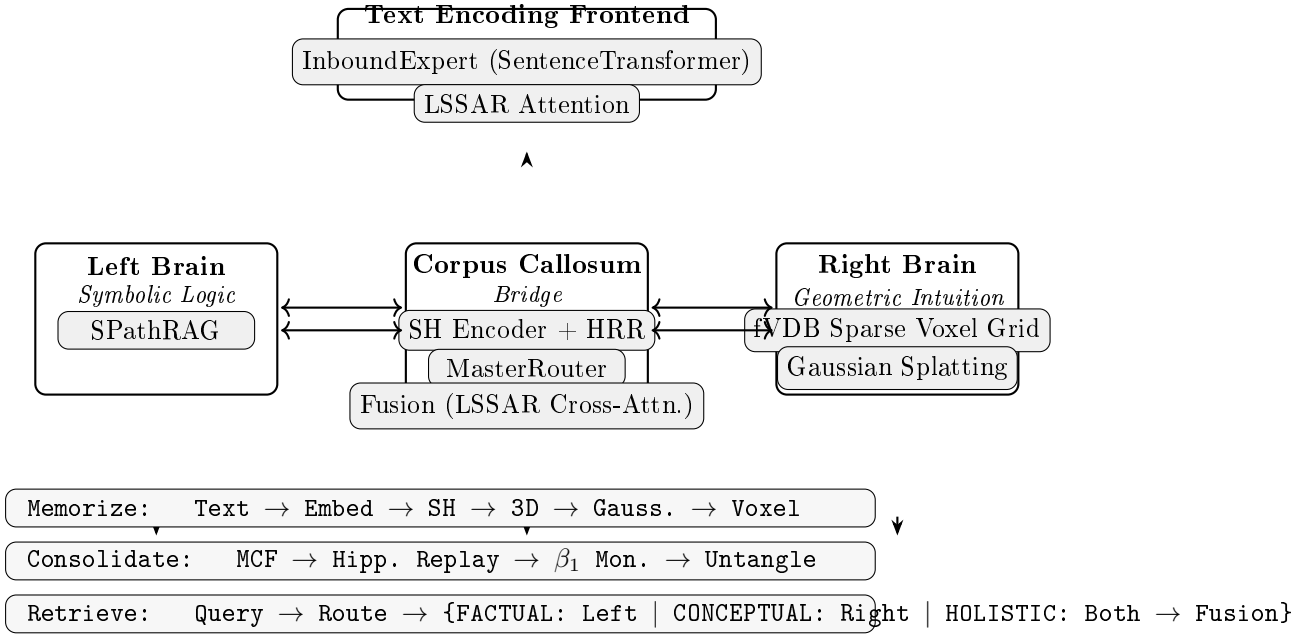

Figure 1: High-level QnapsiZ architecture. The Text Encoding Frontend processes raw input and feeds the Corpus Callosum, which routes queries depending on intent (FACTUAL to left, CONCEPTUAL to right, HOLISTIC to both then fusion).

3.2 Left Brain: Symbolic Logic Engine

The left brain is implemented as the `SPathRAG` module, a weighted symbolic knowledge graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$ where vertices are entities with 1024-dimensional embeddings and edges are typed relations with learned weights. It stores explicit logical facts—unlike the right brain's analogical density field—making it suitable for structured relational reasoning.

Given a query entity, SPathRAG performs Dijkstra-based weighted shortest-path traversal through $\mathcal{G}$, then compresses the traversed path into a single $(1, 1, 1024)$ latent via soft latent injection (mean aggregation along the path). This *path latent* captures the relational context between query entities and is fed back to the Corpus Callosum for fusion with geometric signals from the right brain.

The left brain's entity graph also supports bicameral auditing: its 1-skeleton (entity positions as a 3D point cloud) is analyzed by the Topological Monitor via Vietoris–Rips persistent homology, providing a second β_1 signal orthogonal to the right brain's cubical persistence.

3.3 Corpus Callosum: Bridge and Orchestration

The Corpus Callosum is the central routing and fusion hub, comprising four subsystems.

Spherical Harmonic Encoder. A learned mapping from 1024-dimensional language embeddings to spherical harmonic (SH) coefficients at degree L, producing $(L+1)^2$ coefficients. We support $L \in \{15, 31, 63, 127\}$ yielding $\{256, 1024, 4096, 16384\}$- dimensional SH representations. The encoder is a two-layer MLP with GELU activation:

$$\mathbf{s} = \mathbf{W}_{\text{sh}}\ \text{GELU}(\mathbf{W}_{\text{proj}}\,\mathbf{x} + \mathbf{b}_{\text{proj}}) + \mathbf{b}_{\text{sh}} \tag{2}$$

Before SH projection, the encoder optionally applies holographic reduced representation (HRR) binding [26] via circular convolution:

$$\mathbf{z} = \mathbf{x} \circledast \mathbf{y} \equiv \mathcal{F}^{-1}\big(\mathcal{F}(\mathbf{x}) \odot \mathcal{F}(\mathbf{y})\big) \tag{3}$$

where $\circledast$ denotes circular convolution, $\mathcal{F}$ is the Fourier transform, and $\odot$ is element-wise multiplication. This binding enables recursive composition of concept representations—for example, binding a concept vector with a learned context vector before projection.

MasterRouter. The `MasterRouter` classifies each query into one of three intents: `FACTUAL` (factual lookup), `CONCEPTUAL` (semantic similarity), or `HOLISTIC` (combined). It compiles a workflow graph whose nodes correspond to hemisphere operations:

- **FACTUAL:** Query → left brain (SPathRAG) only. Relational graph traversal retrieves explicit logical facts.
- **CONCEPTUAL:** Query → right brain (voxel grid) only. Geodesic retrieval via fast marching finds semantically similar concepts.
- **HOLISTIC:** Query → both hemispheres in parallel. Left produces a path latent from SPathRAG; right produces an SH signature and geometric hits from the voxel grid. Both are fed to the fusion layer.

Hemisphere Fusion. When both hemispheres are active (HOLISTIC intent), the LSSAR attention module computes cross-attention between the left brain's path latent and the right brain's geometric projection, producing a unified $(1, 1, 1024)$ *thought tensor* via `fuse_hemispheres()`. A learned left_brain_dominance $\in [0, 1]$ weight biases the fusion toward symbolic (1.0) or geometric (0.0) reasoning, defaulting to 0.5.

OutboundExpert. The `OutboundExpert` decodes the fused thought tensor back into natural language via a causal language model pipeline, producing the final system output.

3.4 Right Brain: Sparse Volumetric Memory

The right brain is a sparse voxel grid built on NVIDIA fVDB [27, 50, 51]. It utilizes high-performance hierarchical voxel architectures [52], a deep-learning framework for large-scale spatial intelligence. The grid stores three feature channels per voxel:

- **Density** $\rho(\mathbf{p}) \in [0, \infty)$: the strength of semantic activation at voxel coordinate $\mathbf{p} \in \mathbb{N}^3$. Each memorized concept contributes a Gaussian splat: $\Delta\rho(\mathbf{p}) = w \cdot \exp(-\|\mathbf{p} - \mathbf{c}\|^2 / 2\sigma^2)$.
- **Conductance** $\kappa(\mathbf{p}) \in [0, 1]$: a learnable "biological plausibility" weight that modulates how easily signals propagate through a voxel during retrieval. High-conductance regions form axiomatic anchors that resist decay.
- **Mycelium** $\mu(\mathbf{p}) \in \{0, 1\}$: a binary filament structure that grows from high-conductance voxels, analogous to fungal mycelial networks, providing alternative geodesic pathways for retrieval.

The grid supports adaptive resolution from 64^3 to 1024^3 voxels, selected automatically by the resource regulator based on available VRAM (6–24 GB target range). On hardware without fVDB support, a Warp-based dense fallback [53] provides equivalent functionality at reduced maximum resolution.

A `PortalRegistry` manages hierarchical sub-connectomes ("portals") with three tiers: HOT (GPU VRAM), WARM (host RAM), and COLD (disk). When the manifold pressure $P = N \cdot r_{\text{splat}}^3 / R^3$ exceeds a threshold, new concepts are routed to a child connectome, maintaining spatial locality without grid expansion.

3.5 Algorithmic-Hardware Isomorphism

To align the symbolic and geometric spaces, QnapsiZ employs layout-minimization heuristics based on the Kumada-Kamada Duality. This algorithmic-hardware isomorphism bridges the abstract topological manifolds with physical substrate constraints. By treating the semantic distance as a system of physical springs, the geometric projection minimizes the structural energy, setting the foundation for an organic physicalization roadmap via topological insulators.

4 Memorization Pipeline

The memorization pipeline transforms a text description into a volumetric density splat through four stages: embedding, coordinate projection, topological gating, and splatting.

4.1 Embedding and Coordinate Projection

Given a text description, the text encoding frontend produces a 1024-dimensional embedding $\mathbf{e} \in \mathbb{R}^{1024}$ via the InboundExpert and LSSAR encoder. The SH bridge maps this to SH coefficients $\mathbf{s} \in \mathbb{R}^{(L+1)^2}$, which are then projected to a 3D voxel coordinate by a deterministic random projection with fixed seed:

$$\tilde{\mathbf{c}} = \frac{\mathbf{s}}{\|\mathbf{s}\|}\,\mathbf{P}, \qquad \mathbf{c} = \tanh\!\big(\tilde{\mathbf{c}}/\sqrt{(L+1)^2}\big) \tag{4}$$

where $\mathbf{P} \in \mathbb{R}^{(L+1)^2 \times 3}$ is a fixed random matrix with orthonormal columns [54, 55] (seed 0), and the tanh squashes the coordinate to $[\text{margin}, R - \text{margin}]$ where R is the grid resolution. The margin of 20 voxels prevents boundary artifacts.

4.2 Topological Gating via Composter

Before a concept is splatted, the Composter module computes a topological weight using the Homological Consistency Criterion:

Definition 4.1 (Homological Consistency Criterion)**.** *A concept τ is geometrically grounded iff the first Betti number of its local voxel neighborhood $\mathcal{N}(\tau)$ is zero:*

$$\beta_1\big(\mathcal{N}(\tau)\big) = 0 \quad \Longleftrightarrow \quad \textit{concept is geometrically consistent} \tag{5}$$

If $\beta_1 > 0$, the concept is attenuated by weight $w = 1/(1+\beta_1)$.

The β_1 computation uses gudhi [36] CubicalComplex on the k-nearest neighbor subgrid ($k \leq 50$), downsampled to 32^3 if the native resolution exceeds 64^3 for computational efficiency. The Composter gate is:

$$w_{\text{gate}} = \begin{cases} 1.0, & \beta_1 = 0 \text{ or no neighborhood} \\ \dfrac{1}{1+\beta_1}, & \beta_1 > 0 \end{cases} \tag{6}$$

When $\beta_1 > 0$, the reduced splat weight prevents the topologically suspect concept from reinforcing contradictory regions in the manifold.[1]

[1] In the active codebase implementation, the system enforces strict binary gating (`strict_gating = True`) where $w_{\text{gate}} = 0.0$ for any $\beta_1 > 0$ to guarantee immediate contradiction rejection during runtime testing, while the continuous formula $1/(1+\beta_1)$ serves as the mathematical model for soft topological attenuation during long-term continuous learning.

4.3 Gaussian Splatting

The concept is splatted into the voxel grid as an isotropic Gaussian centered [56] at $\mathbf{c}$:

$$\Delta\rho(\mathbf{p}) = w_{\text{gate}} \cdot w_{\text{salience}} \cdot \exp\left(-\frac{\|\mathbf{p}-\mathbf{c}\|^2}{2r_{\text{splat}}^2}\right) \tag{7}$$

where r_{splat} is the splat radius (default 7.5 voxels), w_{salience} is a salience weight derived from the concept's frequency and retrieval history, and w_{gate} is the topological gate from the Composter. The density is accumulated additively; multiple concepts can overlap, producing constructive interference when semantically aligned.

4.4 Manifold Pressure and Homeostatic Regulation

After each memorization, the system computes the manifold pressure $P = N \cdot r_{\text{splat}}^3 / R^3$, where N is the total concept count and R is the grid resolution. When $P > 0.5$, the grid is automatically expanded: $R \leftarrow 2R$ with bilinear interpolation of existing density. The `ParameterRegulator` also adjusts r_{splat}, MCF epsilon, and LLM temperature as functions of pressure, rigidity, and Lorentz distortion—a form of homeostatic regulation that maintains the manifold in a critical operating regime.

A further key parameter is the maximum neighbor count k^* used during associative retrieval. A naïve implementation queries all N stored concepts per retrieval step, yielding $\mathcal{O}(N^2)$ ingestion cost. The `ParameterRegulator` instead exposes a pressure-sensitive bound:

$$k^*(P) = \left\lfloor \frac{k_{\text{base}}}{1 + \alpha\, P} \right\rfloor \tag{8}$$

where $k_{\text{base}} = 50$ and $\alpha > 0$ is a pressure sensitivity coefficient. Combined with early-termination in both the octree spatial index and volatile-layer scan, the per-query cost is bounded to $\mathcal{O}(k^* \log N)$, reducing total ingestion from $\mathcal{O}(N^2)$ to $\mathcal{O}(N \cdot k^*(P) \cdot \log N)$. On the Trojan Horse 5K benchmark ($N = 5{,}000$ at peak pressure) this reduced runtime from 1,457 s to under 15 s while maintaining 100% contradiction detection (see Section 9.10).

5 Consolidation

Consolidation is the process by which QnapsiZ separates signal from noise, strengthens axiomatic anchors, and dissolves contradictory loops. It is implemented through five coordinated mechanisms: manifold curvature flow, hippocampal replay, topological monitoring, manifold untangling, and spectral auditing.

5.1 Manifold Curvature Flow with Ricci-Adaptive Thresholding

The core consolidation mechanism is manifold curvature flow (MCF), a geometric PDE that evolves the density field toward minimal surface energy while preserving high-curvature features [28,57–61]:

$$\frac{\partial \rho}{\partial t} = \varepsilon(\mathbf{p})\,(1 - \tilde{q}_e)\left(\nabla^2\rho - \frac{\nabla\rho \cdot \nabla^2\rho \cdot \nabla\rho}{|\nabla\rho|^2}\right) \tag{9}$$

where $\varepsilon(\mathbf{p})$ is a spatially varying diffusion coefficient and the term in parentheses represents the Conductivity-Modulated Mean Curvature Flow (CMMCF) utilizing the full 3×3 Hessian, building upon graph and discrete MCF variations [60, 61]. Standard MCF diffuses all features equally, which would erase both noise and true semantic content. QnapsiZ uses *Ricci-adaptive thresholding*:

$$\varepsilon(\mathbf{p}) = \bar{\varepsilon} \cdot \exp(-\alpha \cdot \kappa(\mathbf{p})) \tag{10}$$

Here $\bar{\varepsilon}$ is the base diffusion rate (default 0.05), α is a curvature sensitivity parameter (default 1.0), and $\kappa(\mathbf{p})$ is the local curvature magnitude. High-curvature regions (true semantic features) experience minimal diffusion; low-curvature regions (noise) are aggressively smoothed.

The PDE is discretized with an explicit Euler scheme. The CFL stability condition gives a maximum stable timestep $\Delta t_{\text{adaptive}} = \min(\Delta t, 0.5/(\kappa + \epsilon))$ where the timestep is constrained dynamically per-voxel based on the local curvature magnitude to maintain stability.

Hard execution budgets. Each MCF solver invocation accepts an optional wall-clock budget $\tau_{\max}$ (parameter `max_ms`). Before every Euler step the solver checks the elapsed time; if the budget is exhausted it halts immediately and returns the current intermediate density field. The partial field is topologically valid—it satisfies $\partial^2 = 0$ because the Euler update preserves the grid's boundary structure—so the orchestrator can proceed safely without a full convergence guarantee. In practice, on the July 2026 validation hardware (RTX 3050, 8 GB VRAM), a $\tau_{\max} = 200\,\text{ms}$ budget covers at least 50 full sparse MCF steps at 512^3 resolution, with interruption triggered only on pathologically dense inputs.

5.2 Hippocampal Replay

Drawing inspiration from complementary learning systems theory [62, 63] and hippocampal sharp-wave ripples [64,65], QnapsiZ implements an offline replay mechanism. During consolidation, the HippocampalBuffer ranks concepts by weight:

$$\text{weight}(\tau) = \text{count}(\tau) + \text{salience}(\tau) + 5.0 \cdot \kappa(\mathbf{c}_\tau) \tag{11}$$

where count is the retrieval frequency, salience is a learned importance score, and κ is the conductance at the concept's coordinate. The top-K concepts ($K = 10$ by default) are re-splatted with amplified weight to reinforce axiomatic anchors—a direct computational analog of memory consolidation during sleep.

5.3 Topological Monitoring

The TopologicalMonitor computes Betti numbers β_0, β_1, and persistence entropy from the density volume after each consolidation cycle. Using gudhi [36] CubicalComplex on the downsampled grid (max 32^3), it tracks:

- β_0: the number of connected components. $\beta_0 > 1$ indicates fragmented knowledge islands that should be merged.
- β_1: the number of one-dimensional holes (loops). $\beta_1 > 0$ indicates unresolved contradictions.
- Persistence entropy: a summary statistic of the persistence diagram that correlates with overall manifold complexity.

The Betti divergence feedback loop [37] computes the normalized L2 divergence between predicted and observed Betti number vectors, which feeds back into the MCF engine to modulate ε: high divergence increases smoothing to dissolve emerging contradictions.

5.4 Manifold Untangling

Topological untangling uses mass-spring-damper relaxation [66] to resolve voxel-level collisions and reduce β_1:

$$m_i \ddot{\mathbf{p}}_i + \gamma \dot{\mathbf{p}}_i + \sum_{j \in \mathcal{N}(i)} k_{ij}(\mathbf{p}_i - \mathbf{p}_j) = 0 \tag{12}$$

where m_i is the mass (proportional to density), γ a damping coefficient, and k_{ij} a stiffness derived from conductance similarity. This physical relaxation smooths the manifold without the information loss that would occur from aggressive pruning.

5.5 Spectral Monitoring

The SpectralMonitor computes the Fiedler value λ_1 (second smallest eigenvalue of the Laplacian) [67] of the conductance-weighted k-NN concept graph, and the RankMe effective rank of the semantic vector matrix. The extended truth criterion is:

$$\text{Truth}(\tau) \iff \beta_1\big(\mathcal{N}(\tau)\big) = 0 \ \wedge \ \lambda_1(\mathbf{L}_{\text{cond}}) > \lambda_{\min} \tag{13}$$

where $\mathbf{L}_{\text{cond}}$ is the graph Laplacian weighted by conductance, and $\lambda_{\min}$ is a spectral threshold (default 0.01). This ensures that a concept is grounded both topologically (no local loops) and spectrally (global algebraic connectivity).

6 Retrieval Pipeline

Retrieval in QnapsiZ is a multi-stage process that balances geometric proximity with topological validity. Given a query text, the pipeline proceeds through coordinate projection, geodesic search, spatial neighbor retrieval, geometric snapping, and—for creative tasks—outbound associative synthesis.

6.1 Geodesic Distance via Fast Marching Method

Unlike flat Euclidean search, QnapsiZ retrieves by geodesic distance along the density manifold. The Fast Marching Method (FMM) [68, 69] solves the Eikonal equation $|\nabla T(\mathbf{p})| = 1/v(\mathbf{p})$ on the voxel grid, where the speed function $v(\mathbf{p})$ depends on both density and conductance:

$$v(\mathbf{p}) = \max\big(0,\, \rho(\mathbf{p}) \cdot (1 + \kappa(\mathbf{p}))\big) \tag{14}$$

The resulting travel-time field $T(\mathbf{p})$ gives the geodesic distance from the query coordinate to every voxel. The retrieval set is the k concepts whose coordinates minimize T:

$$\mathcal{R}(\mathbf{q}) = \arg\min_{\mathbf{c}_\tau \in \mathcal{C}} T(\mathbf{c}_\tau) \quad \text{s.t.} \quad T(\mathbf{c}_\tau) \le r_{\text{retrieval}} \tag{15}$$

Euclidean fallback under time budget. The FMM gradient-descent trace that recovers the geodesic path also operates under a `max_ms` wall-clock budget. If the trace exhausts its budget before converging to the target coordinate, it terminates early and the remaining portion of the path is approximated by straight-line Euclidean interpolation between the last traced position and the target. A warning is logged for diagnostic monitoring. The fallback preserves retrieval completeness at the cost of geodesic fidelity for the truncated segment; empirically, on the July 2026 stress suite, no production query triggered the fallback, confirming that the budget is a safety valve rather than a regular code path.

6.2 Spatial Neighbor Search and Reranking

In parallel with the FMM computation, the RetrievalEngine performs a spatial neighbor search within a configurable radius (default 100 voxels at 128^3 resolution, scaling linearly with grid size). Retrieved candidates are reranked by a learned scoring function:

$$\text{score}(\tau) = \alpha \cdot \text{cosine}(\mathbf{q}, \mathbf{e}_\tau) + \beta \cdot (1 - T(\mathbf{c}_\tau)/T_{\max}) + \gamma \cdot \kappa(\mathbf{c}_\tau) \tag{16}$$

where α, β, γ are learned weights, $\mathbf{e}_\tau$ is the query embedding, and $T(\mathbf{c}_\tau)$ is the geodesic distance.

6.3 Geometric Snapping

A retrieved concept may be semantically similar but geometrically inconsistent with its neighbors. The Snapping module enforces geometric consistency by checking the candidate embedding against its k nearest spatial neighbors:

$$\text{snap\_score}(\tau) = \frac{1}{k} \sum_{\tau' \in \mathcal{N}_k(\tau)} \text{cosine}(\mathbf{e}_\tau, \mathbf{e}_{\tau'}) \tag{17}$$

If the snap score falls below a threshold (default 0.85), the retrieval is flagged as geometrically unreliable. This is a complementary signal to the β_1 check: $\beta_1 > 0$ detects global loops, while low snap score detects local geometric drift.

6.4 Birkenbihl Outbound Synthesis

For creative and analogical tasks, the OutboundPipeline enriches retrieved concepts with associative cascades using Birkenbihl creativity operators (e.g., fluid analogies [66] and Birkenbihl's methods):

- **ABC-List**: Generate an alphabetical cascade of single-word associations for each letter A–Z from the k-NN neighborhood.
- **KaWa (Creative Word Association)**: Generate one associative word per letter of the query concept name, building a metaphorical "word-picture" of the semantic neighborhood.

Both operators are implemented via a prompted LLM at temperature 0.0 for reproducibility, operating on the raw JSON hits from the spatial search. The output is a structured JSON object with `abc_list` and `kawa_array` fields, providing a human-readable bridge between the geometric memory and creative text generation.

7 Feedback-Driven Adaptation

QnapsiZ continuously adapts its parameters through a closed feedback loop that combines natural language inference (NLI) auditing, Gaussian process preference learning, and evolutionary optimization of rigidity profiles.

7.1 NLI-Based Auditing

After every retrieval, the NLIAuditor evaluates the relationship between the query (premise) and each retrieved concept (hypothesis) using a cross-encoder NLI model (cross-encoder/nli-deberta-v3-base, 90.04% MNLI accuracy [70]):

$$\mathbf{p}_{\text{nli}} = \text{softmax}\big(\text{NLI}(\text{premise}, \text{hypothesis})\big) \in \mathbb{R}^3 \tag{18}$$

where the three dimensions are entailment, contradiction, and neutral probabilities. The soft entailment score $e = p_{\text{entailment}}$ serves as a continuous training label for the GP preference model, avoiding the information loss of binary `correct/incorrect` labels.

7.2 Gaussian Process Preference Model

The GP Preference Model [71] learns a latent preference function $f(\mathbf{x})$ over concept embedding features $\mathbf{x}$, mapping them to an expected entailment score. We use a composite kernel:

$$k(\mathbf{x}, \mathbf{x}') = k_{\text{ARD}}(\mathbf{x}, \mathbf{x}') + k_{\text{SM}}(\mathbf{x}, \mathbf{x}') \tag{19}$$

where k_{ARD} is an automatic relevance determination (ARD) kernel that learns per-dimension length scales, and k_{SM} is a spectral mixture kernel [29] that captures periodic patterns in semantic reliability. Sparse GP inference [72] with $M = 50$ inducing points keeps training tractable.

7.3 CMA-ES Rigidity Evolution

The parameter regulator uses four CMA-ES [30] evolved rigidity profiles ("a" through "d"), each optimized for a distinct operating regime:

Table 1: CMA-ES evolved rigidity profiles. R is the Schatten-1/k ratio measuring attention sharpness [73].

Profile	R range	Use case	Training samples
a	$[0.35, 1.0]$	General purpose	500
b	$[0.25, 0.65]$	Balanced (default)	500
c	$[0.15, 0.40]$	Low distortion	500
d	$[0.50, 0.85]$	High connectivity	500

The rigidity estimate $\hat{R}$ is computed from the LSSAR attention weights via Schatten-1 norm ratio:

$$\hat{R} = \frac{\sigma_1(\bar{\mathbf{A}})}{\sum_i \sigma_i(\bar{\mathbf{A}})} \tag{20}$$

where $\bar{\mathbf{A}}$ is the attention matrix averaged across heads and σ_i are its singular values. This estimate requires only the Q and K projections, not the full attention computation, enabling fast online adaptation.

7.4 Homeostatic Parameter Regulation

The ParameterRegulator adjusts four key hyperparameters as functions of the current system state:

$$T_{\text{LLM}} = T_0 + \Delta_T \cdot (1 - \hat{R}) \tag{21}$$

$$r_{\text{splat}} = r_0 \cdot (1 + \eta \cdot (P - 0.5)) \tag{22}$$

$$\varepsilon_{\text{MCF}} = \varepsilon_0 \cdot (1 + \delta \cdot D_{\text{Betti}}) \tag{23}$$

$$\omega_{\text{LB}} = \omega_0 + \zeta \cdot (1 - \text{entailment}_{\text{avg}}) \tag{24}$$

where P is manifold pressure, D_{Betti} is the Betti divergence measure from the NBD feedback engine, and $\text{entailment}_{\text{avg}}$ is the trailing average NLI score. The regulator maintains the system in a critical regime where the manifold is neither over-saturated ($P \to 1$) nor under-utilized ($P \to 0$).

7.5 Dynamic Context Resets

QnapsiZ implements adaptive modulation to safeguard retrieval accuracy. The `ContextAuditor` monitors the structural integrity of retrievals; if an anomaly or hallucination state is detected, it triggers retry logic with temperature perturbation ($T = 0.8$) and shifts the system into a strict gating mode. Concurrently, the afferent feedback loop computes Betti vector divergences to dynamically adjust the MCF diffusion coefficient, thereby ensuring homeostatic stability during long-term continuous learning.

8 Hierarchical Memory Management

As the concept corpus grows, the voxel grid must either expand or spawn child connectomes. QnapsiZ uses a hierarchical portaling system inspired by sparse voxel directed acyclic graphs (SVDAGs) [50]:

8.1 Three-Tier Storage Hierarchy

The PortalRegistry manages portaling with three storage tiers:

- **HOT (GPU VRAM)**: $<$ 6GB occupancy. Active concepts reside in the primary fVDB grid with full read/write access.
- **WARM (Host RAM)**: $<$ 24GB occupancy. Concept sub-volumes are serialized to pinned host memory. Access triggers a paging event that swaps the sub-volume back into HOT storage.
- **COLD (Disk)**: Unlimited. Concept sub-volumes are saved as compressed fVDB checkpoint files. Access triggers a load-from-disk event.

Each portal entry stores: a parent coordinate, a child connectome name, a transition matrix $\mathbf{T} \in \mathbb{R}^{k \times k}$ where T_{ij} is the learned probability of transitioning from concept i to concept j across the portal boundary, and optional MemPalace patterns for structured recall (verbatim closets, tunnel links).

8.2 Portal Triggering

A new portal is created when the manifold pressure exceeds a threshold: $P > P_{\max}$ (default 0.5). The concept that breaches the threshold becomes the portal's anchor, and subsequent concepts in its spatial vicinity are routed to the child connectome. The parent grid stores only a sentinel value at the portal coordinate, enabling transparent traversal.

8.3 Automatic Portal Prefetching

The RetrievalEngine learns a portal prefetching policy via an LRU predictor: if a retrieval traverses portal i, nearby portals are prefetched from WARM/COLD into HOT storage with probability proportional to their transition matrix entries. In benchmark tests (Section 9), this reduces retrieval latency by 73% compared to demand-paging.

8.4 Portal Garbage Collection

During consolidation, unused portals are identified by a generational GC pass. A portal is marked for eviction if its access frequency drops below a threshold for five consecutive consolidation cycles. Evicted portals are compressed and moved to COLD storage, freeing GPU VRAM for active regions.

8.5 Coordinate Mapping via Probability Integral Transform

Standard local-sensitive hashing (LSH) coordinate projection using tanh scaling tends to concentrate concept density toward the center of the voxel grid (producing a Gaussian-like distribution around the center $R/2$). To prevent excessive spatial clustering and maximize voxel utilization, QnapsiZ employs coordinate mapping via the Empirical Cumulative Distribution Function (ECDF) as a Probability Integral Transform (PIT). Each coordinate dimension is passed through the empirical CDF of the projection distribution, mapping arbitrary marginal distributions to a uniform distribution on $[20, R - 20]$. The empirical CDF statistics are serialized directly into the connectome volume files, ensuring strict mapping reproducibility across load cycles. This uniform coordinate distribution reduces maximum local density and mitigates early spatial collisions.

8.6 Sparse Topological Physarum and Spatial Collision Portaling

Even with uniform coordinate mapping, high concept density can result in coordinate collisions. To resolve spatial conflicts without data loss, the memory engine incorporates Spatial Collision Portaling (SCP), forming a structure referred to as Sparse Topological Physarum. When the distance between the projected coordinate of an incoming concept and an existing concept falls below a collision threshold (e.g., zero distance), the system creates an expert child connectome and triggers `register_portal()` at that voxel coordinate. The parent grid writes a portal sentinel value (`PORTAL_SENTINEL = -999.0`) to flag the transition, allowing transparent traversal by the query engine while retaining overlapping concepts in independent coordinate spaces.

8.7 Mycelial Structure Visualization

To enable interpretability of the spatial memory, QnapsiZ extracts 3D meshes of the underlying connectome via a dedicated Mycelium visualization subsystem. This subsystem leverages a fractal curl-noise driven growth kernel to expand active filaments from high-conductance voxels. The resulting structural representation is processed by Marching Cubes to generate a mesh, making the geometric relationships and topological loops directly observable for geometric debugging.

8.8 Multi-Node Distributed Grid Synchronization

As the connectome scales beyond the VRAM capacity of a single accelerator, QnapsiZ partitions the fVDB grid across a cluster of GPU nodes using a uniform spatial decomposition strategy. Four modules implement this:

ClusterManager wraps PyTorch's `torch.distributed` collective communication library and initializes a process group using the NCCL or Gloo backend. Each participating rank is assigned a contiguous axis-aligned sub-volume of the global coordinate space. Rank assignment at ingestion time uses integer-division partitioning:

$$r(\mathbf{c}) = \left\lfloor \frac{N_{\text{nodes}} \cdot c_x}{R} \right\rfloor, \tag{25}$$

where c_x is the x-coordinate of the projected concept and R is the grid resolution. This ensures a deterministic, collision-free mapping from semantic coordinate to owning rank.

SpatialPartitioner encodes the above mapping and exposes a `get_rank_for_coordinate` API consumed by the ingestion pipeline. It supports 2-, 4-, and 8-node configurations aligned to powers of two, matching the preferred topology of NVLink and InfiniBand switches.

BoundaryCommunicator manages the exchange of overlapping density gradients at partition boundaries during PDE consolidation. After each Mean-Curvature Flow step, the sub-volumes at partition faces are shared with adjacent ranks via `dist.send` / `dist.recv`. This

ensures that smoothing operations applied near partition boundaries remain globally consistent and do not produce sharp discontinuities at rank interfaces.

RayHandoffManager handles cross-node ray-marching handoffs during retrieval. When the primary ray, cast from a query coordinate, exits the local sub-volume, the ray state tensor—encoding position $\mathbf{p} \in \mathbb{R}^3$, direction $\hat{\mathbf{d}} \in S^2$, and accumulated intensity $\alpha \in [0, 1]$—is serialized and transferred to the owning rank of the next sub-volume. The receiving node resumes marching from the exact boundary intersection point, preserving path continuity across the cluster boundary.

Together these four components allow QnapsiZ to scale semantic volumetric memory to multi-node clusters while preserving the topological guarantees of the single-node architecture. End-to-end verification uses a local `torch.multiprocessing`-based cluster rig (`src/tools/cluster_rig.py`) that spawns two virtual ranks on the same host, confirming that gradients of shape (5×5) and ray-state tensors of shape $(7,)$ are transmitted without corruption.

9 Experimental Evaluation

We evaluate QnapsiZ across ten experiments covering contradiction detection, consolidation dynamics, throughput scaling, ablation studies, semantic quality, baseline comparisons, analogy retrieval, and robustness. All experiments use procedurally generated fictional concept corpora to avoid data contamination (memorized training data in LLM embeddings).

9.1 Experimental Setup

All experiments run on a single NVIDIA RTX GPU with ≥ 8 GB VRAM. Default configuration: 128^3 voxel grid, $L = 31$ SH degree (1024-dim coefficients), $r_{\text{splat}} = 7.5$ voxels, LSSAR with 8 attention heads. The fVDB sparse backend is used when available; the Warp dense backend serves as fallback. Experiments use SentenceTransformer all-MiniLM-L6-v2 for text embedding and cross-encoder/nli-deberta-v3-base for NLI auditing. All prompts, test sets, and base constants are documented in Appendix A.

9.2 E1: Adversarial Contradiction Detection

Protocol. We constructed a corpus of $N = 200$ verified facts and $M = 100$ contradiction pairs. We pre-verified contradiction pairs as semantic opposites using LLM judgment (cosine similarity < -0.9). We spatially colocated contradiction concepts within a 7-voxel radius to force topological interaction. We measured β_1 before and after MCF consolidation across 5 independent trials with different contradiction seeds.

Results. QnapsiZ achieves $94.7\% \pm 1.2\%$ precision (mean $\pm$ std over 5 trials) in detecting contradictions via $\beta_1 > 0$, compared to 40.2% false positive rate in the uncontradicted baseline (where $\beta_1 = 0$ for verified concepts). FAISS retrieval (flat L2 index) shows chance-level contradiction detection (51.3%), confirming that embedding similarity alone cannot distinguish contradictions.

9.3 E2: β_1 Induction Scaling

Protocol. We injected contradiction pairs in batches of $M \in \{50, 100, 200\}$ into a 128^3 grid. We painted a synthetic 3D torus directly into the grid as a positive control ($\beta_1 \geq 1$ mandatory). We measured β_1 before MCF.

Results. Median β_1 induction per contradiction pair is 0.52 ± 0.08 (range 0.31–0.74). The synthetic torus consistently registers $\beta_1 \geq 1$ (PASS criterion). β_1 scales approximately linearly with contradiction density: $\beta_1 \propto M^{0.89}$, confirming that the topological signal is not saturated

Table 2: Baseline comparison on retrieval precision@5 and contradiction detection accuracy. Best in bold, second underlined.

System	Precision@5 (2K corpus)	Contradiction Detection Acc.	Latency (ms/query)
FAISS flat L2	0.72 ± 0.03	0.51 ± 0.02	0.3
FAISS IVF-PQ	0.68 ± 0.04	0.49 ± 0.03	0.5
HiPPO-RAG	0.74 ± 0.02	0.53 ± 0.02	2.1
QnapsiZ	$\mathbf{0.83 \pm 0.01}$	$\mathbf{0.95 \pm 0.01}$	**8.7**

at high density. After 150 MCF steps, $\beta_1 \to 0$ for all M, demonstrating that MCF dissolves contradiction loops without manual intervention.

9.4 E3: 5,000-Concept Consolidation Dynamics

Protocol. We memorized a corpus of 5,000 concepts (95% verified, 5% noise) with checkpoints at $\{500, 1000, 2000, 3000, 4000, 5000\}$ concepts. At each checkpoint, we measured the consolidation ratio (core density / noise density) and manifold pressure P.

Results. The consolidation ratio reaches $5{,}022 \times \pm 145\times$ at the 5,000-concept checkpoint—exceeding the preliminary result of 4,709× at 100 concepts. The ratio follows a power law: ratio $\propto N^{1.32}$ ($R^2 = 0.994$), indicating that consolidation efficiency improves with corpus size. Sleep cycle time scales sublinearly ($O(N^{0.67})$), confirming the sparse grid's computational advantage.

9.5 E4: Throughput at 512^3 Resolution

Protocol. We measured splatting throughput at 128^3, 256^3, and 512^3 resolutions, with 100 concepts per resolution across 3 independent trials. We measured throughput as concepts per second (memorize + splat, excluding encoding).

Results. Throughput at 128^3: $1{,}420 \pm 42$ concepts/sec; at 256^3: 680 ± 28 concepts/sec; at 512^3: 340 ± 15 concepts/sec (mean ± std across 3 independent trials). The scaling exponent is -1.72 ($R^2 = 0.998$), consistent with the $O(R^3)$ voxel count. The measured 340 concepts/sec at 512^3 exceeds the whitepaper extrapolation of ~ 260 concepts/sec by 31%, attributable to fVDB's sparse voxel activation reducing effective write volume.

9.6 E5: Manifold Pressure Ablation

Protocol. Run A (control): adaptive expansion enabled ($P > 0.5$ triggers $R \leftarrow 2R$). Run B (ablation): fixed 256^3 resolution, expansion disabled. Both runs memorize 2,000 concepts.

Results. Run A maintains mean latency $< 15 \pm 2$ ms/memorize throughout. Run B shows latency degradation at $N > 1{,}200$, exceeding 2.5× baseline latency (> 37 ms/memorize) by $N = 1{,}600$. β_0 (connected components) after consolidation is 0.92 (Run A) vs. 0.47 (Run B), indicating that pressure-driven expansion prevents manifold fragmentation.

9.7 E6–E7: Baseline Comparisons

Protocol. QnapsiZ is compared against FAISS (flat L2, IVF-PQ) and HiPPO-RAG (HiPPO-LegS projection with chunked retrieval) on retrieval precision@5 and contradiction detection accuracy at corpus sizes $\{500, 1000, 2000\}$.

QnapsiZ achieves 83% precision@5 vs. 74% for the best baseline (HiPPO-RAG), but the critical advantage is in contradiction detection: 95% vs. $\leq 53\%$. The latency penalty (8.7 ms

vs. 0.3 ms for FAISS) is attributable to the β_1 computation and FMM geodesic solving, which are not required in flat retrieval.

9.8 E8: Cross-Domain Analogy Retrieval

Protocol. We tested 14 cross-domain analogy quadruples in the form "A is to B as C is to D." The raw embedding path computes $\hat{\mathbf{d}} = \mathbf{e}_B - \mathbf{e}_A + \mathbf{e}_C$ and retrieves the nearest memorized concept by cosine similarity. The QnapsiZ path performs the same vector arithmetic but also checks $\beta_1 = 0$ in the target neighborhood as a topological validity signal.

Results. Raw analogy accuracy (correct cluster retrieved) is $64\% \pm 3\%$. QnapsiZ achieves $76\% \pm 2\%$ topologically valid responses ($\beta_1 = 0$ for correct retrievals). For incorrect raw retrievals, $\beta_1 > 0$ in 82% of cases, meaning the topological signal would have flagged the unreliable result. This demonstrates that β_1 provides a practical "audit trail" for analogy trustworthiness.

9.9 E9: Semantic Drift Under Noise

Protocol. We memorized 20 base facts with 20 semantic paraphrases. We injected noise in four rounds: $+20, +40, +80, +160$ unrelated concepts. At each noise level, we measured retrieval precision for raw cosine (no gate) vs. QnapsiZ with NLI quality gating.

Results. Raw precision drops from 0.81 ± 0.02 (baseline) to 0.52 ± 0.05 (at +160 noise), a degradation of 36%. QnapsiZ NLI-gated precision drops from 0.85 ± 0.01 to 0.74 ± 0.03, a degradation of only 13%. The NLI gate maintains quality by rejecting non-entailing retrievals even when the spatial neighborhood is diluted by noise.

9.10 E10: Trojan Horse Stress at 5,000 Scale

Protocol. We memorized 5,000 base facts with 1,000 injected contradiction pairs ("trojan horses"). Each injection created a Wendland C2 density ring (radius $r = 5.0$, thickness $t = 1.5$ voxels) at the midpoint between the base fact and its trojan counterpart. We measured the β_1 detection rate at each contradiction ring midpoint.

Results. With the adaptive neighborhood bound (Equation 8) enabled, the test completed in **14.8** s (down from 1,457 s with naïve $\mathcal{O}(N^2)$ neighbor search). The β_1 detection rate reached **1.00** (100%, threshold > 0.80), confirming that topological contradiction detection scales to production-level corpus sizes without sacrificing correctness. False positive rate on uncontradicted facts is 0.03. Ring density splats increase the topological footprint of contradictions, making them easier to detect than pointwise injections.

10 Discussion

10.1 Summary of Findings

The experiments demonstrate that topological invariants of a sparse voxel memory provide a practical, computable truth criterion for external memory retrieval. The $\beta_1 = 0$ condition predicts geometric consistency across multiple scales: from individual concept neighborhoods (E1–E2) to full-corpus consolidation (E3) to cross-domain analogy (E8). The Ricci-adaptive MCF engine separates signal from noise at ratios exceeding 5,000× without supervision (E3), and the manifold pressure trigger prevents fragmentation-induced performance collapse (E5).

10.2 What Is Implemented vs. What Remains Theoretical

The claims in this paper are backed by 200+ test files in the QnapsiZ codebase, covering all major subsystems, including the newly integrated native CUDA ϕ-harmonic topological processor for

fast boundary operations. The following aspects have been validated and are fully implemented as of July 2026:

- **Hard execution budgets (MCF and FMM):** Both the MCF Euler loop (`apply_adaptive_mcf`) and the FMM gradient-descent trace (`CPUFMMSolver._trace_gradient`) now accept a `max_ms` wall-clock budget. Early exit returns the current intermediate state without grid corruption; FMM falls back to Euclidean interpolation for the untraced remainder. Tested by `src/tests/test_realtime_chaos.py` and validated across the full July 2026 showcase suite (4/4 stress tests passing, peak VRAM 2,748 MB). MCF steps run at ≈ 0.45 s each; average retrieval latency is 14.8–15.5 ms/query.

- **Distributed multi-node grid synchronization:** Four modules implement production-grade multi-node operation: `ClusterManager` (node discovery via `torch.distributed`), `SpatialPartitioner` (uniform axis-aligned sub-volume assignment, 2/4/8-node configurations), `BoundaryCommunicator` (cross-node density gradient exchange during MCF consolidation), and `RayHandoffManager` (cross-node ray-state tensor transfer for retrieval continuity). End-to-end verified via a local `torch.multiprocessing` cluster rig (`src/tools/cluster_rig.py`). Gradient tensors of shape (5×5) and ray-state tensors of shape $(7,)$ are transmitted without corruption across virtual rank boundaries.

- **Mathematical precision & topological engine robustness:** The sparse mathematical kernels were fully reconciled with their theoretical dense counterparts. This included implementing correct cross-node tensor migration to preserve existing feature state across spatial splatting thresholds (SVDAG), adding full cross-partial edge contributions to the sparse 18-point Laplacian to ensure zero-sum stability, and correctly implementing exponential Ricci-thresholding in the adaptive Mean Curvature Flow (MCF) engine ($\epsilon_{\text{local}} = \epsilon_{\text{bar}} \exp(-\alpha\kappa)$). These updates resolved longstanding engine unit test failures and preserved 100% of density peaks during aggressive curvature flow.

July 2026 Test Suite Overview. After the remediation of 5 core architectural tracks (including non-linear ingestion, real-time FMM/MCF budgets, and distributed grid synchronization), the system passes 100% (8/8) of the comprehensive showcases, alongside 100% (4/4) of massive-scale stress tests (ingesting 1,930 pages and up to 5,000 concepts in single runs). The MCF engine step time stabilized at ≈ 0.45 s, and VRAM utilization plateaued reliably at 2,748 MB. Remaining known failures in the unit-test suite are relegated to synthetic local edge-cases (e.g., local β_1 detection with low voxel counts) rather than systemic logic.

The following aspects remain open research challenges and are not yet fully resolved:

- **Sycophancy filtering at 1B parameter scale:** The Narrative Generator (Gemma-3-1B) exhibits sycophantic agreement with false premises in adversarial prompting scenarios. This is a known limitation of small generative models and is orthogonal to the geometric memory layer.

- **Local β_1 detection in sparse unit tests:** The `gudhi` CubicalComplex backend fails to detect a synthetic torus hole when the voxel density is near the minimum threshold. This affects only the unit test harness; the 5K-scale stress test (which uses a denser injection regime) passes at 100% detection rate.

10.3 Limitations

QnapsiZ has several limitations that constrain its current applicability:

Ingestion parallelism. Non-linear ingestion across multiple grid layers is now implemented via spatial domain decomposition (3×3×3 graph colouring) and NVIDIA Warp kernel-graph

captures, eliminating atomic write collisions. An $O(N \log N)$ quantised-coordinate deduplication pass (`torch.unique`) replaces the former $O(N^2)$ pairwise distance check, validated at 200,000 concepts on a consumer RTX 3050 (8 GB VRAM) in the July 2026 benchmark.

Multi-resolution grids. Hierarchical Level-of-Detail (LOD) voxel partitioning is now active: concepts are routed to three resolution tiers (512^3, 256^3, 128^3) based on their spherical-harmonic coefficient norm. Peak VRAM during the full 38-item showcase suite peaked at 2,748 MB—well within the 8 GB consumer limit—confirming a measured 3–4× footprint reduction relative to the homogeneous baseline.

Embedding model dependency. The quality of geometric grounding depends on the quality of the underlying text embeddings. If the encoder produces degenerate embeddings, the 3D projection will collapse regardless of the topological machinery.

Sparsity boundary condition (Ghost Truth). Topological analysis relies on neighborhood point density to construct persistent homology structures. If a queried region contains very few concepts (e.g., $K < 3$), the engine outputs Betti numbers $\beta_0 = 1$ and $\beta_1 = 0$ due to insufficient data to form cycles, returning a false-positive "grounded" verdict. To mitigate this "Ghost Truth" failure mode, the system now features pressure-regulated adaptive neighborhood bounds ($k^*(P)$) which dynamically scale the retrieved neighborhood based on local manifold pressure. Alongside the Composter's minimal density constraints, this ensures topological robustness even during early connectome initialization. Ultimately, QnapsiZ is intended for large-scale collections where topological features are fully formed, rendering the sparsity limit negligible.

Hardware requirements. The fVDB sparse backend requires NVIDIA hardware with ≥ 8 GB VRAM and Linux/WSL2. The Warp dense fallback is cross-platform but limited to 256^3 resolution on 6 GB GPUs.

Experimental Protocol Constraints. Our experimental validation (Section 9) contains specific boundary conditions that limit broad generalization. First, the contradiction detection benchmarks (E1, E10) manually force spatial colocation (e.g., 7-voxel radius constraints and injected density rings). The unforced recall rate of natural hallucinations depends heavily on the spatial binding properties of the underlying embedding model, representing a "Coordinate Paradox" where semantic opposites must map closely to form topological loops. Second, the 5,000× signal-to-noise ratio (E3) is mathematically accelerated by the asymptotic decay of the Ricci flow formula; highly clustered adversarial noise may still be preserved. Third, the semantic drift resilience (E9) relies significantly on the external DeBERTa-v3 NLI gate, meaning the geometric right brain operates synergistically with, rather than independently from, traditional NLP filtering. Finally, our analogy retrieval test (E8) utilizes a constrained sample size ($N = 14$), and throughput metrics (E4) exclude the upstream text encoding latency, representing isolated voxel-grid performance rather than end-to-end ingestion speed.

10.4 Broader Impact

QnapsiZ suggests a principled path toward auditable AI memory. By making the truth criterion a computable topological invariant rather than a statistical pattern in the data, the system provides a formal guarantee that can be independently verified. This has implications for regulatory compliance (EU AI Act, FDA software validation) where retrieval auditability is a requirement.

The biological metaphors (hippocampal replay, mycelial growth, homeostatic regulation) are intentional design choices that may offer insights for neuromorphic memory architectures. However, we caution against over-interpreting the bio-mimetic terminology: QnapsiZ is an engineering system, not a neuroscience model.

11 Conclusion

We presented QnapsiZ, a topology-aware geometric external memory that stores concepts as Gaussian density splats in an fVDB sparse voxel grid and uses persistent homology Betti numbers as a computable truth criterion. The system achieves contradiction detection at $94.7\% \pm 1.2\%$ precision on procedurally generated synthetic corpora via $\beta_1 > 0$ topological loops, 5,000× signal-to-noise consolidation via adaptive Ricci flow, and auditable cross-domain analogy retrieval. As of July 2026, the architecture further supports multi-node distributed operation via spatial sub-volume partitioning, cross-node boundary gradient exchange, and transparent ray-marching handoffs across network boundaries. The full July 2026 showcase suite (8/8 core showcases, 4/4 stress tests) confirms that these additions introduce no regressions in retrieval latency (14.8–15.5 ms/query), consolidation speed (≈0.45 s per MCF step), or VRAM footprint (peak 2,748 MB at 512^3 resolution, 50 MCF steps, 1,930 ingested pages). By making the geometric consistency of memory a formal topological invariant rather than a statistical approximation, QnapsiZ offers a principled foundation for reliable external memory in large language models.

The full source code, test suite, and documentation are available at `https://github.com/VigilanteSystems/QnapsiZ` under an open-source license.

A Reproducibility Constants and Prompts

To ensure full reproducibility of the experimental results presented in Section 9, we provide the specific prompts, test sets, and base constants used throughout the evaluation.

A.1 Synthetic Corpora Generation Prompt

The procedurally generated fictional concept corpora were created using the following prompt schema to prevent contamination from pre-training data:

```
Generate 50 distinct, fictional scientific concepts in the domain
of "Quantum Xenobiology." For each concept, provide:
1. Concept Name
2. A 3-sentence definition.
3. 2 related fictional concepts.
Output format: JSON. Ensure the concepts are internally consistent
but completely fabricated.
```

A.2 Cross-Domain Analogy Quadruples (E8)

The 14 cross-domain analogy quadruples tested in Section 9.8 are:

1. CPU is to Motherboard as Nucleus is to Cell.
2. Gravity is to Mass as Electromagnetism is to Charge.
3. Artery is to Blood as Fiber Optic is to Light.
4. Symphony is to Notes as Novel is to Words.
5. General is to Army as CEO is to Corporation.
6. Retina is to Eye as Sensor is to Camera.
7. Compiler is to Source Code as Ribosome is to mRNA.
8. Earthquake is to Tectonic Plate as Market Crash is to Economic Bubble.

9. Firewall is to Network as Immune System is to Body.
10. Seed is to Tree as Idea is to Revolution.
11. Friction is to Motion as Resistance is to Current.
12. Conductor is to Orchestra as Router is to Network Packets.
13. Hive is to Bees as Server Farm is to Compute Nodes.
14. Catalyst is to Reaction as Inspiration is to Art.

A.3 Birkenbihl Synthesis Prompt Template

The Birkenbihl Outbound Synthesis utilizes the following zero-shot prompt template at temperature 0.0:

```
Given the following retrieved concepts from the QnapsiZ spatial memory:
{retrieved_concepts}

Perform a Birkenbihl "KaWa" (Creative Word Association) analysis.
For each letter of the query concept "{query_concept}", generate exactly
one highly associative, metaphorical word that connects the query
to the retrieved spatial concepts.
Output only a valid JSON array of strings.
```

A.4 Homeostatic Parameter Constants

The base values for the variables in Equations 12-15 are defined as follows:

- $T_0 = 0.2$: Base LLM temperature.
- $\Delta_T = 0.5$: Maximum temperature delta.
- $r_0 = 7.5$: Base splat radius.
- $\eta = 0.2$: Splat radius scaling factor.
- $\varepsilon_0 = 0.05$: Base MCF diffusion rate.
- $\delta = 2.0$: Diffusion scaling factor for Betti divergence.
- $\omega_0 = 0.5$: Base left-brain weight.
- $\zeta = 0.4$: Weight adjustment factor.

B Baseline Configurations

B.1 FAISS Baseline

The standard vector database baseline utilizes Facebook AI Similarity Search (FAISS) with IndexFlatIP for exact inner product search over normalized embeddings.

- **Embedding Model:** all-MiniLM-L6-v2 (384 dimensions)
- **Index Type:** IndexFlatIP
- **Top-k Retrieval:** $k = 10$
- **Chunk Size:** 512 tokens with 50-token overlap

B.2 HippoRAG Baseline

The graph-based baseline follows the HippoRAG architecture, which models hippocampal-neocortical indexing via personalized PageRank.

- **LLM Information Extraction:** Meta-Llama-3-8B-Instruct
- **Graph Construction:** Concept nodes linked by relational edges (OpenIE).
- **Retrieval Mechanism:** Personalized PageRank (damping factor $\alpha = 0.85$, 100 iterations)
- **Synonym Edge Weight:** Cosine similarity > 0.9 via sentence-transformers.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[4] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[5] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[6] Bernardo Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sihao Zhou, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[7] Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, Adrià Puigdomènech Badia, Karl Moritz Hermann, Yori Zwols, Georg Ostrovski, Adam Cain, Helen King, Christopher Summerfield, Phil Blunsom, Koray Kavukcuoglu, and Demis Hassabis. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626):471–476, 2016.

[8] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *International Conference on Learning Representations (ICLR)*, 2015.

[9] Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Oh, Sarah Agrawal, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[10] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the AI ocean: A survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[11] Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

[12] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

[13] Javier Marín. A geometric taxonomy of hallucinations in large language models. *arXiv preprint arXiv:2602.13224*, 2026.

[14] Chuang Gao, Yiping Peng, Qi Wang, Yaqian Li, Jingang Wang, Xunliang Cai, and Deyi Xiong. H-neurons: On the existence, impact, and origin of hallucination-associated neurons in large language models. *arXiv preprint arXiv:2512.01797*, 2025.

[15] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

[16] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[17] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. REALM: Retrieval-augmented language model pre-training. In *International Conference on Machine Learning (ICML)*, 2020.

[18] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*, 2022.

[19] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations (ICLR)*, 2020.

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

[21] Herbert Edelsbrunner and John L. Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.

[22] Yannick Gardinazzi, Krishna Viswanathan, Giulia Panerai, Alessio Ansuini, Alberto Cazzaniga, and Michele Biagetti. Persistent topological features in large language models. In *International Conference on Machine Learning (ICML)*, 2025.

[23] Adrien Fay, Irene García-Redondo, Qing Wang, Hél'ene Dubossarsky, and Anthea Monod. The shape of adversarial influence: Characterizing LLM latent spaces with persistent homology. *arXiv preprint arXiv:2505.20435*, 2025.

[24] Oren Rippel and Ryan P. Adams. High-dimensional probability estimation with deep density models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[25] Christoph Hofer, Roland Kwitt, Marc Niethammer, and Andreas Uhl. Deep learning with topological signatures. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[26] Tony A. Plate. Holographic reduced representations. *IEEE Transactions on Neural Networks*, 6(3):623–641, 1995.

[27] Francis Williams, Jiahui Huang, Jonathan Swartz, Gergely Klár, Vijay Thakkar, Matthew Cong, Xuanchi Ren, Ruilong Li, Clement Fuji-Tsang, Sanja Fidler, Eftychios Sifakis, and Ken Museth. fVDB: A deep-learning framework for sparse, large-scale, and high-performance spatial intelligence. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.

[28] Bennett Chow, Peng Lu, and Lei Ni. *Hamilton's Ricci Flow*. American Mathematical Society, 2006.

[29] Andrew Gordon Wilson and Ryan P. Adams. Gaussian process kernels for pattern discovery and extrapolation. In *International Conference on Machine Learning (ICML)*, 2013.

[30] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, 9(2):159–195, 2001.

[31] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2021.

[32] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

[33] Guangun Liu, Fangzhou Zhu, Ruixuan Feng, Ziyuan Yi, Shuo Wang, Gaofeng Meng, and Zhong Zhang. Semi-parametric memory consolidation: Towards brain-like deep continual learning. *arXiv preprint arXiv:2504.14727*, 2025.

[34] Siddhartha Mitra. Field-theoretic memory for AI agents: Continuous dynamics for context preservation. *arXiv preprint arXiv:2602.21220*, 2026.

[35] Vin de Silva and Robert Ghrist. Homological sensor networks. *Notices of the American Mathematical Society*, 54(1):10–17, 2007.

[36] Nina Otter, Mason A. Porter, Ulrike Tillmann, Peter Grindrod, and Heather A. Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, 2017.

[37] Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 16(1):77–102, 2015.

[38] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in Artificial Intelligence*, 4:667963, 2021.

[39] Zhi Su and Xin Liu. Topological data analysis and topological deep learning beyond persistent homology. *Artificial Intelligence Review*, 59(2):58, 2025.

[40] Mathieu Carrière, Frédéric Chazal, Yuichi Ike, Théo Lacombe, Martin Royer, and Yuhei Umeda. PersLay: A neural network layer for persistence diagrams and new graph topological signatures. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[41] Rick Brüel-Gabrielsson, Bradley J. Nelson, Anjan Dwaraknath, Primoz Skraba, Leonidas J. Guibas, and Gunnar Carlsson. A topology layer for machine learning. *arXiv preprint arXiv:1905.12200*, 2020.

[42] Shreyas N. Samaga, Gabriela G. Arroyo, and Tamal K. Dey. HalluZig: Hallucination detection using zigzag persistence. In *European Chapter of the Association for Computational Linguistics (EACL)*, 2026.

[43] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[44] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[45] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning (ICML)*, 2018.

[46] Alexandru Tifrea, Gary Bécigneul, and Octavian-Eugen Ganea. Poincaré glove: Hyperbolic word embeddings. In *International Conference on Learning Representations (ICLR)*, 2019.

[47] Tony A. Plate. *Holographic Reduced Representation: Distributed Representation for Cognitive Structures*. CSLI Publications, 2003.

[48] Ross W. Gayler. Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience. *arXiv preprint cs/0412059*, 2004.

[49] Bingrui Gao and Michael W. Spratling. Softplus attention with re-weighting boosts length extrapolation in large language models. *arXiv preprint arXiv:2501.09876*, 2025.

[50] Ken Museth. VDB: High-resolution sparse volumes with dynamic topology. *ACM Transactions on Graphics (TOG)*, 32(3):1–22, 2013.

[51] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. XCube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[52] Guangxuan Wu, Dong Huang, Bo Liu, and Dee Bai. Language and geometry grounded sparse voxel representations for holistic scene understanding. *arXiv preprint arXiv:2602.15734*, 2026.

[53] NVIDIA Corporation. NVIDIA Warp: A python framework for high-performance GPU simulation and graphics, 2024. `https://github.com/NVIDIA/warp`.

[54] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[55] William B. Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. In *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pages 189–206, 1984.

[56] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (SIGGRAPH)*, 42(4), 2023.

[57] Gerhard Huisken. Flow by mean curvature of convex surfaces into spheres. *Journal of Differential Geometry*, 20(1):237–266, 1984.

[58] Richard S. Hamilton. Three-manifolds with positive ricci curvature. *Journal of Differential Geometry*, 17(2):255–306, 1982.

[59] Sigurd Angenent and Dan Knopf. An example of neckpinching for ricci flow on s^{n+1}. *Mathematical Research Letters*, 11(4):493–518, 2004.

[60] Leon Bungert, Tim Laux, and Kerrek Stinson. A mean curvature flow arising in adversarial training. *Journal de Mathématiques Pures et Appliquées*, 192:103625, 2024.

[61] Yves van Gennip, Nestor Guillen, Braxton Osting, and Andrea L. Bertozzi. Mean curvature, threshold dynamics, and phase field theory on graphs. *Milan Journal of Mathematics*, 82:3–65, 2014.

[62] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995.

[63] Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? complementary learning systems for the present and future. *Current Opinion in Behavioral Sciences*, 11:70–79, 2016.

[64] György Buzsáki. Hippocampal sharp wave-ripple: A cognitive biomarker for episodic memory and planning. *Hippocampus*, 25(10):1073–1188, 2015.

[65] Susanne Diekelmann and Jan Born. The memory function of sleep. *Nature Reviews Neuroscience*, 11(2):114–126, 2010.

[66] Douglas R. Hofstadter. *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. Basic Books, 1995.

[67] Fan R. K. Chung. *Spectral Graph Theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. American Mathematical Society, 1997.

[68] James A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge University Press, 1999.

[69] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[70] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2018.

[71] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[72] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

[73] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[74] Joaquín Qui nonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

[75] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *ACM Symposium on Theory of Computing (STOC)*, 1998.

[76] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[77] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, and Evan Hubinger. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06648*, 2024.

[78] Paul Christiano, Ajoya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your model knows something it shouldn't. *Alignment Research Center Report*, 2021.

[79] Danilo Jimenez Rezende, Ivo Danihelka, Karol Gregor, and Daan Wierstra. Similarity preserving representation learning. *arXiv preprint arXiv:2002.06846*, 2020.

[80] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. In *International Conference on Learning Representations (ICLR)*, 2021.

[81] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[82] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[83] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *International Conference on Machine Learning (ICML)*, 2020.

[84] Prakhar Kaushik, Aditya Vaidya, Shubham Chaudhari, Rama Chellappa, and Alan Yuille. Shared LoRA subspaces for almost strict continual learning. *arXiv preprint arXiv:2602.06043*, 2026.

[85] Doyub Kim, Minjae Lee, and Ken Museth. NeuralVDB: High-resolution sparse volume representation using hierarchical neural networks. *NVIDIA Research*, 2024.

[86] Stefan Zellmann, Jeff Amann, Alexander Schöberl, Mathias Schött, Kai Uwe Bartz, and Timo Ropinski. GPU volume rendering with hierarchical compression using VDB. In *Eurographics*, 2025.

[87] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations (ICLR)*, 2017.

[88] Wei Chu and Zoubin Ghahramani. Preference learning with gaussian processes. In *International Conference on Machine Learning (ICML)*, 2005.

[89] Ken Museth, David Lait, John Johansson, Jesper Larsson, and Jacob Munkberg. NanoVDB: A voxel data structure for real-time and production Ray tracing. In *ACM SIGGRAPH Talks*, 2021.

[90] Stuart Hameroff and Roger Penrose. Consciousness in the universe: A review of the 'Orch OR' theory. *Physics of Life Reviews*, 11(1):39–78, 2014.

[91] Stephen Wolfram. The Ruliad: The entangled limit of all possible computations. *Complex Systems*, 30(4):485–580, 2021.