American Academic Scientific Research Journal for Engineering, Technology, and Sciences
ISSN (Print) 2313-4410, ISSN (Online) 2313-4402
https://asrjetsjournal.org/index.php/American_Scientific_Journal/index

Hybrid Storage Models for High-throughput Vector Retrieval

Sasun Hambardzumyan*

Director of Engineering, Activeloop, Director, Deep Lake LLC, Yerevan, Armenia
Email: xustup@gmail.com

Abstract

This study examines the characteristics of employing hybrid models for high-performance vector search. The objective of this paper is to substantiate and systematize existing hybrid data storage schemes based on a memory hierarchy (DRAM → SSD → HDD) in order to enhance the efficiency of vector retrieval procedures. As a methodological foundation, a broad review of key publications devoted to graph index structures, inverted files and their hybrid combinations was carried out, supplemented by a comparative analysis of their performance according to primary metrics. On the basis of the obtained data, a conceptual model of a multilevel storage architecture is described, demonstrating pathways to achieve an optimal balance between query processing speed (QPS) and search completeness (recall) through adaptive quantization and the rational construction of index structures. The scientific novelty lies in the description of a unified architectural scheme integrating various memory types and indexing approaches to ensure highly efficient and scalable vector search in dynamically updated environments. The results presented in this work will be of interest to data engineers, AI system architects and researchers in the field of big data management.

Keywords: vector retrieval; hybrid storage; nearest neighbor search; vector databases; graph indexes; HNSW; DiskANN; data quantization; high-performance computing; AI data management.

1. Introduction

The rapid exponential growth of the artificial intelligence model market, which is projected to exceed 1811.75 billion USD by 2030, growing at an average rate of 35.9% between 2025 and 2030 [1], imposes demands on methods for organizing and processing enterprise data. Despite this, a large percentage of information within enterprises remains untapped due to its heterogeneity and lack of clear structure [2].

Received: 6/25/2025
Accepted: 8/9/2025
Published: 8/19/2025

---

* *Corresponding author.*

Converting such heterogeneous multimodal arrays—images, text documents and audio recordings—into compact numerical embeddings has become the standard for integrating data into modern machine learning architectures. However, at data volumes reaching billions of vectors, ensuring simultaneously low query latency, high matching accuracy and controlled storage costs remains an unresolved computational challenge.

Existing approaches to storing vectors in pure dynamic random-access memory (DRAM) provide the necessary speed but disproportionately increase capital and operational expenditures. In contrast, solutions that rely exclusively on SSD/HDD offer cost advantages but cannot meet the latency and accuracy requirements for search in high-dimensional spaces. As a result, a fundamental scientific gap emerges in the development of hybrid distributed storage schemes capable of intelligently combining different types of memory and indices.

The objective of the article is to substantiate and systematize existing hybrid data storage schemes based on a memory hierarchy (DRAM → SSD → HDD) in order to improve the efficiency of vector retrieval procedures.

The scientific novelty of the work lies in the description of a unified multilevel architecture in which graph-based indices and inverted data structures interact within the storage hierarchy to provide sublinear response time without loss of accuracy.

The research hypothesis is that such a hybrid system will reduce total cost of ownership compared to fully DRAM-oriented solutions while maintaining a high level of search result quality.

However, the study has limitations, since the work constitutes a conceptual review and synthesis of existing architectural patterns. It does not present results of empirical experiments, and the conclusions are drawn based on a systematic analysis of the available literature. Primary attention is given to the prevalent DRAM-SSD-HDD hierarchy, and although emerging technologies such as Persistent Memory or CXL are acknowledged, they are not the main focus. Consequently, the analysis prioritizes architectural trade-offs related to query throughput (QPS) and completeness, while factors such as energy consumption and operational complexity are addressed conceptually rather than quantitatively.

2. Materials and Methods

In recent years the development of hybrid storage models for high-performance vector search has become particularly relevant. On one hand, the explosive growth of the artificial intelligence market confirms the need for scalable solutions: according to GrandViewResearch data, the global AI market volume is projected to reach 1811.75 billion US dollars by 2030, growing at an average rate of 35.9% during the period from 2025 to 2030 [1, 2]. At the same time, vendors actively offer complete stacks for working with vector data: thus Activeloop was recognized by Gartner as a Cool Vendor in Data Management for 2024 [14], and the report Cool Vendors in Data Management: GenAI Disrupts Traditional Technologies analyses the advantages of hybrid storage at the intersection of cloud and local SSD [15].

The foundation of modern high-performance vector search lies in graph-based Approximate Nearest Neighbor (ANN) search algorithms, with Hierarchical Navigable Small World (HNSW) being a prominent example. A

comprehensive review by Wang M and his colleagues [12] consolidates graph-based methods, underscoring their superior performance in terms of search quality and latency for moderate dimensionalities. Theoretical analyses reveal further constraints: Diwan H and his colleagues [4] demonstrate that optimal graph structures are subject to stringent constraints on edge count, while Shrivastava A., Song Z., & Xu Z. [5] prove that minor deviations in graph structure can lead to an exponential increase in traversal path length. This body of research solidifies the efficacy of in-memory graphs for speed and accuracy but simultaneously highlights their primary limitation: a direct dependency on large amounts of costly DRAM, which presents a clear scalability and cost challenge.

To overcome this memory capacity bottleneck, research has shifted to hybrid memory-disk models. The EXAGRAPH framework described by Acer S. and his colleagues in [3] integrates in-memory graph indices with a disk structure, partitioning the graph to optimize node placement between cache and persistent storage. Khan S. and his colleagues [6] demonstrate BANG, an algorithm designed for searching billion-scale vector collections using a single GPU. Their method employs float16 compression and distributed graph restructuring. Gollapudi S. and his colleagues [7] describe the implementation details of Filtered-DiskANN, a memory-disk hybrid approach, where the primary graph is maintained in memory, while the actual vector arrays are stored on SSD. The method utilizes an initial filtering step to select only the most promising candidates for deeper evaluation in memory. Chakraborty T., Bera D. [9] incorporate a modified Count–Min Sketch to store sparse edges in dynamic graphs and combine it with locality-sensitive hashing (LSH) techniques for rapid filtering of irrelevant embeddings. This approach permits pairwise nearest-neighbor search in time close to O(1) on large datasets, at the expense of a minor accuracy loss (≤ 5 %). Li Z., Li H., Meng L. [13] review pruning, quantization, and weight-factorization techniques for deep networks aimed at reducing the memory footprint of embeddings and indices without degrading search performance. The authors highlight that 8-bit quantization and mixed-precision arithmetic often reduce storage requirements by a factor of 4–8 while incurring less than a 2 % drop in nearest-neighbor search metrics. Majumdar A. and his colleagues [11] present a zero-shot object-goal navigation method wherein target objects are encoded in a multimodal embedding space (visual and linguistic) and rapidly retrieved from a distributed vector cache built on FAISS and NVMe storage. It proves that data reduction is not merely a storage-saving tactic but a core enabler of high-performance computation on resource-constrained hardware. These works represent a critical architectural shift. By treating the disk not as a secondary backup but as an active tier of the index, they demonstrate that the performance gap between DRAM and SSD can be bridged through intelligent algorithm design, making billion-scale search economically feasible.

A critical test for these systems is their performance under real-world dynamic workloads. Architectures for web-scale web search were tested by Muhamed A. and his colleagues [10] on datasets such as MS MARCO in a 2024 study. Chen K. and his colleagues [8] found that they combined HNSW with SSDs for cold vectors. Handling continuous data input is also a major challenge. Chakraborty T., Bera D. [9] explore this issue by implementing sketch-based data structures and LSH for fast filtering of dynamic graphs. These studies underscore that a complete solution must address the full data lifecycle. While many hybrid models excel with static datasets, their performance and consistency under frequent insertions and deletions remain a significant challenge, pointing toward the need for architectures designed with dynamic data in mind.

Thus, based on the conducted literature analysis, apparent contradictions exist between theoretical and practical findings. While the authors in sources [4, 5] highlight fundamental limitations regarding the sizes and densities of graphs, practical systems discussed in sources [3, 6] exhibit stable scalability with billion-scale datasets. This discrepancy indicates an inadequate consideration of engineering optimization and compression factors within theoretical models. Furthermore, despite successful examples of hybrid storage (DiskANN, BANG), there is limited research addressing energy efficiency and total ownership cost analysis for such systems. Issues related to dynamic updates of graph indices and support for online transactions without complete structural reconstruction are almost unexplored. Problems regarding consistency and fault tolerance in distributed hybrid storage, as well as interactions between embedding quantization and graph path quality, remain insufficiently studied. Finally, there is a need for more comprehensive comparisons of hybrid approaches under realistic web workloads, such as those used in MS MARCO and Baidu Search.

3. Results and Discussion

As a response to the tasks set, it is proposed to employ a multilayer hybrid data storage architecture (Figure 1) in which the components of the vector index and the vector data representations themselves are deliberately separated across three functionally distinct tiers. On the hot tier, dynamic random-access memory (DRAM) is used to host the most frequently requested index fragments and hot data, providing microsecond-scale latency during search retrieval. On the intermediate warm tier, extended parts of the index and data are stored on solid-state drives (SSD), which are more expensive than HDDs but offer substantially lower latency compared with magnetic disks. Finally, on the cold tier—for extremely infrequent queries and archival segments—traditional hard-disk drives (HDD) or scalable object stores are employed, where capacity and cost per gigabyte of storage constitute the primary criteria [3, 5].

The main objective of this architecture is to achieve an optimal trade-off between the lowest possible search latency and the economic efficiency of ownership: by dynamically distributing hot, warm and cold data across the corresponding storage tiers, lookup operations are accelerated on the hot layer, whereas less demanded blocks are off-loaded onto cheaper media, thereby reducing the total infrastructure expenditure without significant degradation of search performance.

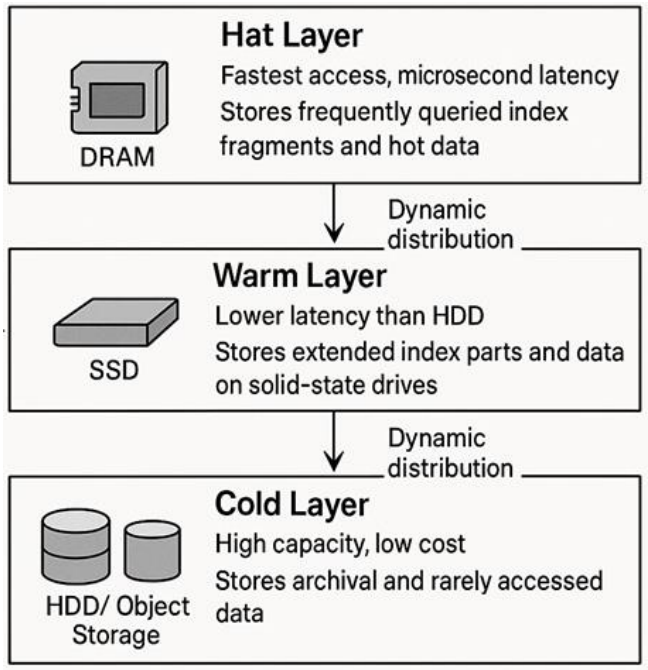

Figure 1: Conceptual model of multi-level hybrid storage for vector search [4, 5, 7, 11]

At the top and fastest tier of the hierarchy (DRAM) only those components that are critical to search latency are placed: the entry point of the graph structure (medoid), a cache of hot vertices that are accessed most frequently, and compact, compressed vector descriptors (for example, obtained via Product Quantization). These lightweight representations are used for an initial coarse pruning of irrelevant candidates and make it possible to significantly reduce the number of expensive accesses to slower storage media.

SSD-based tiers store the bulk of the graph index together with full yet quantized vector representations. The data at this level are laid out so as to enable maximally efficient sequential reading in large blocks—a strategy that underpins the DiskANN implementation [8, 10]. In a number of cases this arrangement makes it possible to bypass direct DRAM accesses while sustaining high throughput with moderate latency.

At the bottom of the hierarchy resides the most capacious and cost-efficient storage layer—HDDs or cloud object storage. This tier keeps the full, non-quantized vectors and accompanying metadata, which are fetched solely during the final stage of processing. Once the fast tiers have performed preliminary filtering, the system recomputes exact distances only for the most promising candidates.

Owing to this three-tier organisation, the main graph traversal is carried out almost entirely in main memory and on SSDs, whereas HDD accesses occur only sporadically, providing an optimal compromise between speed, accuracy, and infrastructure cost.

The effectiveness of the hybrid approach is determined largely by the distribution of resources across the tiers. In hybrid search and indexing architectures the principal resource trade-off is the quantisation depth of the

vector representations. On the one hand, reducing the code bit-width enables more vectors to fit into main memory and the SSD tier, thereby lowering latency and increasing throughput. On the other hand, overly aggressive compression inevitably leads to a deterioration in the accuracy of nearest-neighbour retrieval.

Traditional schemas and the conceptual model described above demonstrate high efficiency when working with immutable data sets; however, modern organizations increasingly face the requirement to process continuously incoming, evolving streams of information.

A clear industrial solution capable of meeting these needs is the Activeloop Deep Lake platform. It is designed from the outset as an AI-native repository in which the boundary between model-training data and operational data disappears. Deep Lake's architecture provides centralized management of multimodal artifacts and versioning, which is particularly important for experiment reproducibility and the organization of continuous learning [6, 9].

Instead of a simple combination of DRAM and SSD, Deep Lake implements integrated memory–compute, where the storage structure is optimized for streaming data directly into compute modules (in particular, GPUs). This approach reduces the load on the CPU and removes input-output bottlenecks, ensuring end-to-end performance when working with dynamic data.

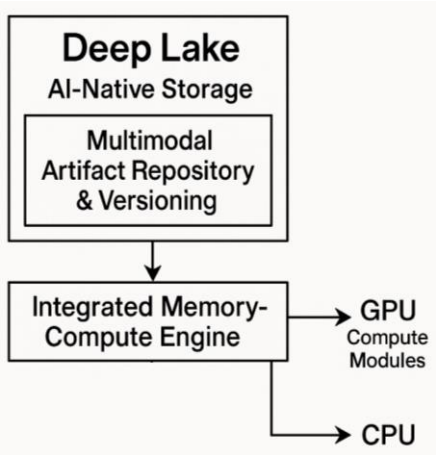

Figure 2: AI-Native storage architecture using Deep Lake as an example [12, 14].

The widespread adoption of the considered approach is clearly illustrated by the deep integration of Activeloop Deep Lake with leading tools of the AI ecosystem, in particular with LangChain, which provides seamless data transfer between the repository and compute modules. The assignment of Deep Lake to the status Cool Vendor in Data Management by Gartner [13, 15] underscores the strategic significance of such solutions for shaping a new level of enterprise data management.

Unlike classical vector databases focused mainly on storage and search, Deep Lake implements the complete data life cycle, from versioning multimodal artifacts and ensuring experiment reproducibility to automated deployment in a production environment. This represents a key evolutionary step for hybrid models aimed at

eliminating bottlenecks when transitioning from static datasets to dynamic streams.

Table 1: Hybrid storage models for high-performance vector search: advantages, disadvantages, and prospects [3, 7, 14, 15]

Advantages	**Disadvantages**	**Prospects (further advantages)**
- Multilevel memory (DRAM / NVMe / object storage) enables keeping hot vectors in RAM while simultaneously scaling cold data on inexpensive media	- Architectural complexity increases: orchestration among memory tiers, cache policies, and throughput monitoring is required	- The propagation of the CXL interface and the emergence of a unified memory pool will reduce latency between CPU and GPU memory
- Reduction of TCO: the combination of expensive DRAM and more affordable SSD / object storage optimizes expenses without loss of throughput for critical queries	- Complexity in ensuring consistency and versioning when migrating vectors between tiers	- Hardware DPU / SmartNIC accelerators will allow part of the search to be performed in place (near-storage compute)
- The ability to stream data directly to the GPU (zero-copy) reduces CPU load and removes I/O bottlenecks	- Specialized drivers and APIs are required (e.g., GPUDirect Storage); not the entire software stack is ready for such access	- The development of format standards (Open Vector Format, Lance, Parquet-Embedding) will ensure compatibility among engines
- Scaling flexibility: nodes can be easily added to increase capacity or performance while maintaining a single logical layer	- Cache warming during an abrupt workload change can lead to increased latency in the first few minutes	- Compression and quantization of vectors on the storage side (PQ, HNSW-PQ) will reduce bandwidth and storage volume requirements
- Support for versioning and replicability simplifies continuous training and A/B testing of models	- Potential dependence on a specific cloud provider or proprietary protocol (vendor lock-in)	- Integration with streaming pipelines (Apache Kafka, DataLake formats) will lead to the emergence of event-driven vector stores

In summary, the conceptual three-tier architecture presented here offers a structured solution to the fundamental dilemma in large-scale vector search: the trade-off between performance, cost, and accuracy. By formalizing the division of labor between DRAM, SSD, and HDD/Object Storage, the model provides a clear blueprint for system designers. The hot tier guarantees low-latency entry points, the warm tier provides scalable and cost-effective indexing for the main dataset, and the cold tier acts as a ground-truth repository for final re-ranking, ensuring that accuracy is not sacrificed for speed. This deliberate separation of concerns allows for independent

optimization and scaling at each level of the memory hierarchy.

However, this framework is not a panacea but rather a lens through which to analyze persistent challenges. The discussion highlights that its practical implementation requires careful tuning of the quantization-vs-recall parameter, where aggressive compression on the warm tier must be balanced against the acceptable loss in search fidelity for a given application. More critically, the problem of handling dynamic data—frequent insertions, updates, and deletions—is magnified in a multi-tier environment, demanding sophisticated consistency protocols and non-disruptive re-indexing strategies that remain an active area of research. The model clarifies these trade-offs, making them explicit architectural decisions rather than implicit operational side effects.

Ultimately, this discussion positions the hybrid model as a foundational building block for the next generation of AI data infrastructure. The future evolution of such systems will likely involve integrating emerging technologies that blur the lines between these tiers. The rise of CXL will create more unified memory pools, hardware accelerators like DPUs will enable near-storage computation, and standardized vector formats will foster interoperability. The proposed conceptual model provides a stable framework to reason about the integration of these future advancements, ensuring that systems can evolve while adhering to the core principles of hierarchical data management.

4. Conclusion

Within the framework of the conducted study, a detailed review of modern designs of hybrid storage systems aimed at ultra-fast retrieval of vector representations has been carried out. It has been established that the prevailing trend is the abandonment of classical single-mode solutions (DRAM only or disk-only storage) in favor of multi-tier architectures capable of intelligently distributing data blocks and index elements between high-speed and high-capacity memory. Such a division of responsibilities improves query locality and lowers search latency without an unjustified increase in infrastructure costs.

The literature review revealed that the synergistic combination of graph structures (for example, HNSW) with disk-oriented algorithms such as DiskANN ensures an optimal balance of performance, cost-efficiency, and accuracy when working with tera- and petabyte-scale vector sets. In particular, the use of hierarchical indexing in DRAM for hot nodes and the transfer of infrequent or less relevant segments to SSD/HDD significantly expand system scalability without substantial degradation in search quality.

This resulted in the description of a conceptual three-tier model (DRAM–SSD–HDD) in which the placement levels of index components and data are deterministically specified with the goal of minimizing end-to-end delays. It is shown that efficiency criteria here largely rest on balancing vector quantization parameters (to reduce the volume of stored data) and the permissible loss of search accuracy. At the same time, it has been found that classical algorithms encounter difficulties when working with dynamically updated collections because they do not provide for adaptive re-indexing in real time.

In response to the identified problem, advanced industrial platforms—such as Activeloop Deep Lake—have

been considered, which embody the concept of AI-native storage. These systems integrate storage, versioning, and computational mechanisms within a single stack, ensuring efficient handling of live data and simplifying the maintenance of incremental update processes. Thus, the proposed hypothesis that a multi-tier hybrid architecture is the optimal solution for modern vector search has been confirmed.

References

[1]. Grand View Research. (2024). *Artificial intelligence market size.* https://www.grandviewresearch.com/industry-analysis/artificial-intelligence-ai-market#:~:text=The%20global%20artificial%20intelligence%20market,35.9%25%20from%202025%20to%202030 (Retrieved June 10, 2025).

[2]. Forbes Technology Council. (2023, May 23). Lessons learned from selfless leadership. *Forbes.* https://www.forbes.com/councils/forbestechcouncil/2023/05/23/lessons-learned-from-selfless-leadership/ (Retrieved April 11, 2025).

[3]. Acer S. et al. (2021). EXAGRAPH: Graph and combinatorial methods for enabling exascale applications. In *The International Journal of High Performance Computing Applications*, 35(6): 553-5717 https://doi.org/10.1177/10943420211029299 .

[4]. Diwan, H., et al. (2024). Navigable graphs for high-dimensional nearest neighbor search: Constructions and limits. *Advances in Neural Information Processing Systems, 37*, 59513–59531.

[5]. Shrivastava, A., Song, Z., & Xu, Z. (2023). A theoretical analysis of nearest neighbor search on approximate near neighbor graph. *arXiv Preprint* arXiv:2303.06210. https://doi.org/10.48550/arXiv.2303.06210.

[6]. Khan, S., et al. (2024). BANG: Billion-scale approximate nearest neighbor search using a single GPU. *arXiv Preprint* arXiv:2401.11324. https://doi.org/10.48550/arXiv.2401.11324.

[7]. Gollapudi, S., et al. (2023). Filtered-DiskANN: Graph algorithms for approximate nearest neighbor search with filters. In *Proceedings of the ACM Web Conference 2023* (pp. 3406–3416). https://doi.org/10.1145/3543507.3583552.

[8]. Chen, Q., et al. (2024). MS MARCO Web Search: A large-scale information-rich web dataset with millions of real click labels. In *Companion Proceedings of the ACM Web Conference 2024* (pp. 292–301). https://doi.org/10.1145/3589335.3648327.

[9]. Chakraborty T., Bera D. A sketch-based approach towards scalable and efficient attributed network embedding : dis. – IIIT-Delhi, 2021 (pp. 19–44).

[10]. Muhamed A. et al. (2023). Web-scale semantic product search with large language models. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining. – Cham : Springer Nature Switzerland*, 73-85.

[11]. Majumdar A. et al. (2021)7 Zson: Zero-shot object-goal navigation using multimodal goal embeddings In *Advances in Neural Information Processing Systems*, 35, 32340-32352.

[12]. Wang, M., et al. (2021). A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *arXiv Preprint* arXiv:2101.12631. https://doi.org/10.48550/arXiv.2101.12631.

[13]. Li Z., Li H., Meng L. (2023). Model compression for deep neural networks: A survey In *Computers*, 12

(3). https://doi.org/10.3390/computers12030060 .

[14]. Activeloop. (2024). *Activeloop named 2024 Gartner Cool Vendor in data management.* https://www.activeloop.ai/resources/gartner-cool-vendor/ (Retrieved May 12, 2025).

[15]. Gartner. (2024). *Cool vendors in data management: GenAI disrupts traditional technologies.* https://www.gartner.com/en/documents/5476395 (Retrieved April 30, 2025).