Vector Databases and Their Role in Modern Data Management and Retrieval: A Survey

www.surveyx.cn

Abstract

Vector databases represent a transformative advancement in data management, engineered to efficiently handle high-dimensional vector data essential for applications ranging from natural language processing to geoinformation science. This survey paper explores the pivotal role of vector databases in overcoming traditional keyword-based search limitations, enabling enhanced semantic understanding and rapid similarity searches. The integration of machine learning techniques, such as deep neural networks and learned index structures, optimizes indexing and querying processes, facilitating robust solutions for managing rich, unstructured data. The survey identifies key challenges, including scalability, efficiency, and integration with traditional systems, while proposing future research directions to enhance vector database functionality. Applications in recommendation systems, domain-specific tasks, and machine learning underscore the versatility and impact of vector databases in modern data-driven environments. By leveraging advanced algorithms and architectural innovations, vector databases are positioned as indispensable tools in contemporary data management strategies, offering efficient and accurate information retrieval across diverse domains. This paper contributes to the understanding and development of vector databases, highlighting their significance in enhancing data management and retrieval practices in an increasingly data-centric world.

1 Introduction

1.1 Introduction to Vector Databases

Vector databases represent a transformative approach in the realm of modern data management, engineered to efficiently handle high-dimensional vector data, which is crucial for applications ranging from multilingual natural language processing to geoinformation science. These databases play a crucial role in addressing the shortcomings of traditional keyword-based search methods by facilitating vector search, which employs neural networks to transform both user queries and documents into dense vector embeddings. This innovative approach not only enhances the comprehension of user intent and semantic relationships but also significantly improves retrieval accuracy and user satisfaction across various domains, including e-commerce, healthcare, and legal research. By implementing advanced indexing strategies and multi-vector search operations, vector search technologies effectively bridge semantic gaps and optimize search performance, marking a transformative shift in information retrieval practices. The transition from token-based search to vector search marks a significant evolution in information retrieval, as vector databases facilitate the encoding and classification of complex data structures, such as medical texts and spatial data. The growing significance of these databases is underscored by their ability to manage the increasing complexity and volume of data generated in today's digital landscape.

In addition to improving semantic search through techniques like Multi-Categorization Semantic Analysis (MCSA), vector databases integrate personalization technology, thereby optimizing search

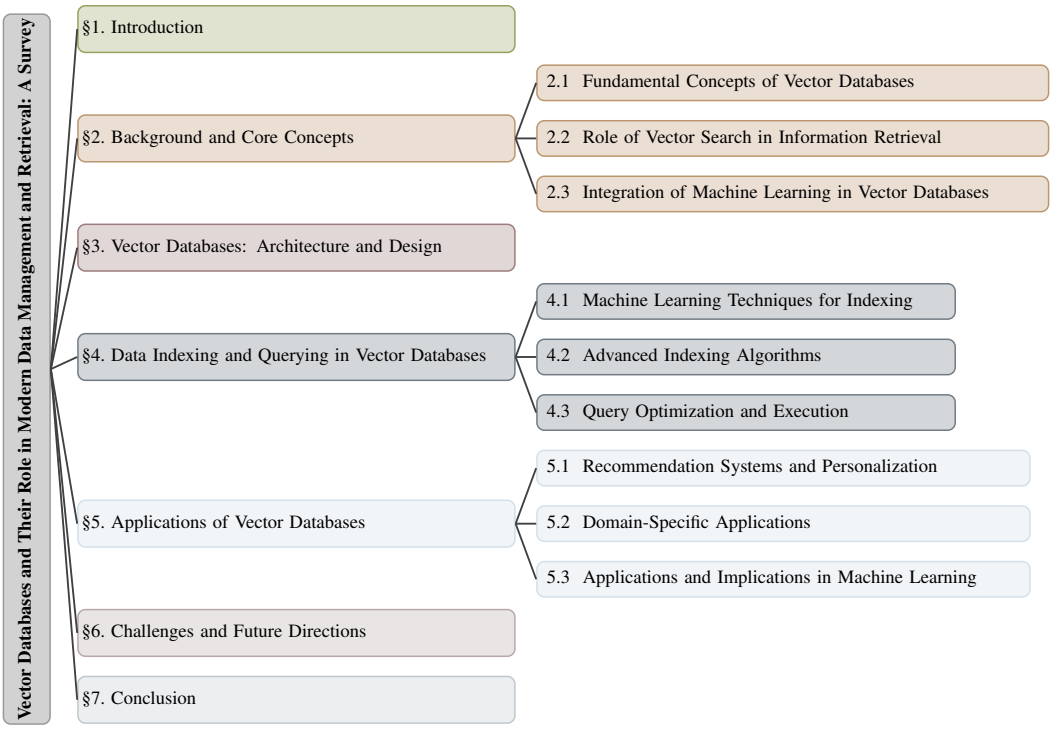


Figure 1: chapter structure

results and enhancing user experience. The use of deep neural networks in vector databases further challenges the misconception that dedicated vector stores are necessary, demonstrating their versatility and efficiency in implementing vector search. This capability is essential for managing and retrieving rich, unstructured data represented as numerical vectors, highlighting the growing need for vector database management systems (VDBMS). The proliferation of applications relying on vector search, such as recommendation systems and personalized content delivery, reinforces the necessity for robust vector database solutions that can adapt to various data types and user requirements. As these technologies evolve, they continue to redefine the standards of data retrieval and management, pushing the boundaries of traditional methods.

Vector databases are designed to address inefficiencies in traditional data management systems, as exemplified by VDMS, which integrates visual data management with metadata handling to facilitate efficient access and analytics. Furthermore, automated performance tuning in vector data management systems (VDMSs) underscores their importance in large-scale information retrieval and machine learning applications. These advancements position vector databases as indispensable tools in contemporary data management strategies, offering robust solutions for handling high-dimensional data and enhancing machine learning applications. The ability to seamlessly integrate with existing infrastructures while providing superior performance metrics is a critical aspect of their appeal. As organizations increasingly rely on data-driven decision-making, the role of vector databases in optimizing data accessibility and usability becomes paramount.

In the context of integrating vector search systems with relational databases, vector databases address the complexities of handling high-dimensional vector indices, enabling the execution of complex queries that were previously challenging in traditional systems. Additionally, estimating set similarity in large datasets, a critical aspect of information retrieval and data management, is effectively managed through vector databases, which play a significant role in fields like social networks and computational advertising. These capabilities underscore the importance of vector databases in addressing data sparsity challenges and improving the accuracy and efficiency of information retrieval processes across various domains. The ongoing research into enhancing the algorithms and structures that underpin these databases is crucial for future advancements, ensuring they remain at the forefront of data management technology.

AI-generated, for reference only.

1.2 Motivation for the Survey

The motivation for this survey arises from the pressing need to bridge the gap between machine learning models and data management systems, which often lack a common conceptual language, as highlighted by Rosner. Traditional keyword-based search methods have proven inefficient, frequently overlooking user semantics and preferences, thereby necessitating a shift towards more sophisticated vector-based approaches. In the context of medical data, challenges in accurately classifying sparse datasets underscore the importance of integrating AI technologies for improved data management. The necessity for enhanced methods is further accentuated by the growing volume of unstructured data, which traditional systems struggle to process effectively. This survey aims to explore these challenges and provide insights into how vector databases can address them.

The inefficiency in managing and retrieving visual data, particularly in fields like medical imaging, further emphasizes the need for advanced vector database systems. Moreover, the complexity and interdependence of tuning parameters in vector data management systems (VDMSs) present significant challenges that this survey aims to address. Traditional search systems, which rely heavily on exact token matching, often fail to capture the nuances of human language and user intent, highlighting the necessity for more comprehensive solutions. By focusing on the integration of AI and machine learning techniques, this survey seeks to identify the gaps in current methodologies and propose innovative approaches that can enhance the effectiveness of data retrieval systems. The intersection of these fields is crucial for developing technologies that meet the demands of modern data environments.

The survey also seeks to evaluate the effectiveness of integrating vector search capabilities within existing infrastructures, such as the Lucene ecosystem, without the need for entirely new components. As digital text data continues to grow exponentially, there is an urgent need for robust methodologies that can convert unstructured data into meaningful feature vectors, thereby enhancing document retrieval processes. Furthermore, the stringent requirements of high accuracy, low latency, and minimal memory footprint in vector search applications drive the demand for innovative approaches. These factors underscore the importance of this survey in highlighting the current state of vector databases and their potential to revolutionize data management practices across various sectors. By addressing these challenges, the survey aims to contribute to the ongoing discourse on the future of data retrieval technologies.

This survey seeks to investigate the intricate challenges associated with vector database management systems, which have become essential for efficient data management and retrieval in today's data-driven landscape. By addressing issues such as high dimensionality, sparsity, and the integration of vector embeddings into various applications—including recommender systems, healthcare, and legal research—this research aims to facilitate the development of advanced vector database solutions that enhance the performance of modern information retrieval technologies. The exploration of these themes is expected to yield valuable insights that can inform both academic research and practical applications, ultimately contributing to the evolution of data management systems in an increasingly complex digital world.

1.3 Structure of the Survey

This survey is meticulously structured to provide a comprehensive exploration of vector databases, beginning with an introduction that establishes the foundational understanding of these systems and their significance in modern data management. Following the introduction, Section 2 provides an in-depth exploration of the foundational concepts and background of vector databases, highlighting their pivotal role in modern information retrieval systems. It discusses how these databases utilize neural networks to transform user queries and documents into dense vector embeddings that capture semantic relationships, thereby enhancing retrieval accuracy. Additionally, the section examines the integration of machine learning techniques to address the unique challenges posed by high-dimensional and sparse vectorized data, ultimately demonstrating how these advancements improve the functionality and efficiency of vector databases across various applications, including e-commerce, healthcare, and legal research. This foundational knowledge is essential for understanding the subsequent discussions on architectural and operational aspects of vector databases.

Section 3 focuses on the architecture and design of vector databases, highlighting the unique challenges and solutions associated with their development, as well as the design principles tailored for

specific applications. It delves into how various architectural choices impact the performance and scalability of vector databases, providing insights into best practices for system design. This section underscores the importance of aligning database architecture with application requirements, ensuring that vector databases can effectively support a wide range of use cases. The discussion will also cover emerging trends in database architecture that leverage advancements in hardware and software technologies, further enriching the reader's understanding of the field.

Section 4 focuses on an in-depth analysis of the techniques employed for data indexing and querying in vector databases. It specifically highlights the integration of machine learning methodologies and sophisticated algorithms designed to enhance the efficiency and effectiveness of these processes. The section also discusses the transition from traditional token-based search to vector search, emphasizing how advanced indexing strategies and semantic embeddings improve information retrieval across various domains such as e-commerce, healthcare, and legal research. Furthermore, it addresses the challenges associated with high-dimensional vector data and presents innovative solutions that optimize retrieval accuracy and user satisfaction. This analysis is critical for understanding the operational dynamics of vector databases and their impact on user experience and system performance.

The survey then transitions to Section 5, which discusses the diverse applications of vector databases across various domains, including their impact on recommendation systems, domain-specific implementations, and implications in machine learning. By examining real-world use cases, this section illustrates the practical benefits of vector databases and highlights successful implementations that have leveraged their capabilities. The exploration of applications serves to contextualize the theoretical discussions in previous sections, demonstrating the tangible outcomes that arise from adopting vector database technologies. This practical perspective is essential for stakeholders looking to implement vector databases in their organizations.

In Section 6, the survey comprehensively examines the current challenges faced by vector databases, including issues related to scalability, efficiency, and integration with traditional systems. It highlights the unique characteristics of vectorized data, such as high dimensionality and sparsity, which necessitate specialized solutions for effective storage and retrieval. Additionally, the section outlines potential future research directions aimed at addressing these challenges, including advancements in indexing strategies, optimization techniques, and AI-driven approaches to enhance the performance of vector search across various applications, from e-commerce to healthcare. This critical examination of challenges and opportunities provides a roadmap for future research and innovation in the field.

Finally, the conclusion in Section 7 summarizes the key findings and reiterates the importance of vector databases in enhancing data management and retrieval practices in today's data-centric landscape. By synthesizing the insights gained from the survey, this section aims to reinforce the significance of vector databases and their potential to drive future advancements in data management technologies. The conclusion will also emphasize the need for continued research and development in this area, encouraging scholars and practitioners to explore new avenues for enhancing the capabilities and applications of vector databases.The following sections are organized as shown in Figure 1.

2 Background and Core Concepts

2.1 Fundamental Concepts of Vector Databases

Vector databases are crucial for handling high-dimensional data essential in applications like recommendation systems and machine learning [1]. With increasing data complexity, traditional methods struggle to manage large datasets, prompting the need for learned indexes that leverage machine learning to improve indexing efficiency [2]. The vector search problem, which involves finding the K closest vectors in a D-dimensional space, is computationally intensive, particularly in real-time applications like online recommendations [3]. Approximate nearest neighbor (ANN) search is vital for rapid similarity searches [4], with tools like the Faiss library providing efficient solutions [5]. The HNSW algorithm, influenced by intrinsic dimensionality and data insertion order, further refines search techniques [6].

Current technologies can handle vector search effectively without complex architectures [7]. However, challenges like integrating vector and raster data for 3D modeling persist, leading to data duplication [8]. Evaluating range search performance is crucial for applications like image retrieval, where a threshold distance determines matches [9]. Efficient computation of set intersections is essential in

AI-generated, for reference only.

applications like duplicate document detection [10]. KNN-SEQ facilitates rapid k-nearest-neighbor machine translation, enhancing data retrieval [11]. The inability of high-dimensional vector indices to exhibit monotonicity poses a challenge for efficient queries [12].

By integrating advanced search algorithms and learned indexing techniques, vector databases offer robust solutions for efficient data management [13]. This integration enhances performance and underscores the need for continuous innovation in data management.

2.2 Role of Vector Search in Information Retrieval

Vector search transforms information retrieval by capturing semantic similarities overlooked by traditional methods. This is crucial in applications requiring contextual understanding, like conversational agents managing complex queries [14]. Systems like Thistle leverage implicit knowledge representations to overcome explicit knowledge limitations [15]. Vector search enhances RAG-based Summarization AI by condensing content and providing references, beneficial in domains like medical texts [16]. The MCSA-P approach personalizes search results by capturing multiple relevant categories, improving user engagement [17].

Vector search surpasses traditional token matching by capturing synonyms and context-dependent meanings through deep learning models [13]. VDMS treats visual data as first-class entities, outperforming traditional systems in handling complex queries [18]. Challenges remain in scalability and tool-equipped agent effectiveness [19], and predicting optimal K values for queries is inefficient [12]. These issues necessitate ongoing research to refine methodologies and enhance applicability across sectors.

Vector search continues to offer robust solutions for efficient information retrieval, bridging conceptual gaps in big data analytics [20]. Its evolving role is central to developing intelligent systems capable of processing complex information.

2.3 Integration of Machine Learning in Vector Databases

Machine learning integration in vector databases optimizes data management, enhancing query handling and data diversity. The Faiss library exemplifies this by balancing accuracy, speed, and memory usage, improving high-dimensional data processing [1]. Faiss utilizes models like Contriever to generate text embeddings, showcasing practical machine learning applications [5]. TorchOpera enhances functionality by improving language model prompts through contextual grounding [21]. Vec2Vec enhances accessibility and interoperability by translating embeddings across models [22]. The VectorSearch framework employs advanced language models to refine document retrieval [23].

VDTuner uses multi-objective Bayesian optimization for automated performance tuning in VDMS [24]. The eCIL-MU framework modifies data representations for Class Incremental Learning with Machine Unlearning [25]. Jaseci streamlines operations by focusing on higher-level abstractions [26]. Establishing theoretical lower bounds on model size for accuracy provides insights into optimizing machine learning integration [27].

These advancements highlight machine learning's role in transforming vector databases into sophisticated systems managing complex data environments. Neural networks enhance search performance by transforming queries and documents into dense vector embeddings, improving retrieval accuracy across sectors like e-commerce and healthcare. Advanced indexing strategies address high dimensionality and sparsity challenges, consistently improving user satisfaction compared to traditional methods [13, 23, 28]. The ongoing integration of machine learning continues to expand vector databases' capabilities, meeting modern data management challenges with innovative solutions.

3 Vector Databases: Architecture and Design

Exploring vector databases necessitates an understanding of the architectural challenges these systems face, particularly in managing high-dimensional data efficiently. This section examines these challenges and the innovative solutions that address them, setting the stage for a detailed exploration of design principles tailored to specific applications.

3.1 Architectural Challenges and Solutions

Vector databases encounter significant architectural challenges, primarily due to the complexities of high-dimensional data management and the necessity for rapid retrieval and processing. The curse of dimensionality exacerbates computational difficulties and data sparsity, critical in complex information retrieval tasks [23]. Systems like VB ASE illustrate the integration of high-dimensional vector indices into relational databases, essential for executing complex queries involving both vector and scalar data, thereby necessitating robust architectural solutions [12].

Recent surveys emphasize the importance of semantic understanding through vector embeddings, crucial for categorizing vector search across various domains [13]. This involves developing frameworks that distinguish vector database management systems (VDBMSs) from traditional databases, focusing on their unique functionalities and capabilities [28]. Innovative solutions like dynamically generating user-defined functions (UDFs) enhance query execution and optimize data processing [29].

Theoretical frameworks establish the relationship between model size, accuracy, and dataset characteristics, providing necessary conditions for reliable performance in vector databases [27]. Understanding these relationships allows vector databases to implement advanced indexing strategies and efficient data processing techniques, overcoming traditional architectural limitations. These solutions collectively enhance the performance and scalability of vector databases, enabling them to meet modern data management tasks effectively.

3.2 Design Principles for Specific Applications

Vector databases are designed with specific applications in mind, requiring tailored principles to address each domain's unique requirements. In recommendation systems, efficiently managing and retrieving high-dimensional data is crucial, necessitating approximate nearest neighbor (ANN) search algorithms like those in the FAISS library, which balance speed, accuracy, and memory usage [1]. These algorithms are essential for personalized content and recommendations by efficiently processing user data and preferences.

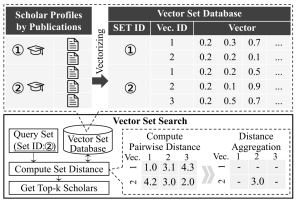
In geoinformation systems, vector databases handle spatial data effectively, integrating vector and raster data to facilitate 3D modeling and visualization [8]. This integration requires overcoming challenges related to data duplication and unorganized storage, necessitating specialized indexing techniques and data structures optimizing spatial queries and retrieval.

For machine learning applications, vector databases support seamless integration of neural network models, allowing translation of embeddings across different models, as demonstrated by the Vec2Vec method [22]. This interoperability enhances accessibility and ensures vector databases adapt to the evolving landscape of machine learning technologies.

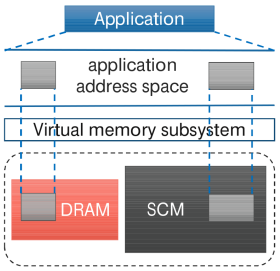
In medical data management, vector databases must accommodate sparse dataset classification, integrating AI technologies to improve retrieval and analysis [16]. This involves developing robust data management strategies to handle the complexities of medical texts and provide accurate and efficient retrieval of relevant information.

Overall, design principles for specific applications of vector databases emphasize tailored solutions addressing unique challenges and requirements of each domain. By employing sophisticated indexing algorithms, integrating advanced machine learning models, and enhancing data management practices, vector databases deliver robust and efficient solutions across diverse applications such as recommender systems, similarity search, and chatbots. These databases effectively manage high-dimensional and sparse vectorized data, essential for accurately capturing and processing rich information types like text, images, and video. The transition to vector search technology revolutionizes information retrieval by enabling systems to understand and process user queries semantically, improving discovery accuracy and user satisfaction in various domains, including e-commerce, healthcare, and legal research [28, 23, 13].

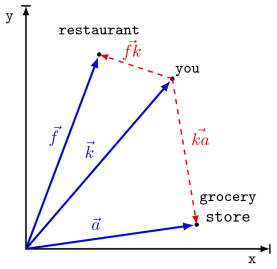
As shown in Figure 2, exploring vector databases through architecture and design highlights the nuanced application of design principles tailored to specific needs. The example of scholar profiles by publications demonstrates how vector databases efficiently pinpoint top-k scholars using a structured approach that includes a vector set database and a query set. This method underscores the importance of organized data representation and retrieval in academic contexts. The virtual memory subsystem



(a) Vector Set Search for Scholar Profiles by Publications[30]



(b) Virtual Memory Subsystem Architecture[31]



(c) The image shows a vector diagram illustrating the displacement of a person from a restaurant to a grocery store.[28]

Figure 2: Examples of Design Principles for Specific Applications

architecture offers insight into memory management within computer systems, showcasing how virtual memory subsystems optimize application address space for improved performance. The vector diagram illustrating displacement from a restaurant to a grocery store serves as a practical example of vector application in everyday scenarios, emphasizing vectors' role in mapping spatial relationships and movements. Together, these examples provide a comprehensive view of how vector databases are designed and implemented across diverse applications, each with unique requirements and challenges [30, 31, 28].

4 Data Indexing and Querying in Vector Databases

Category	Feature	Method
Machine Learning Techniques for Indexing	Similarity Estimation Methods Dimensionality Reduction Strategies Vector-Based Techniques	DESSERT[32] ODCR[33] eCIL-MU[25], KNN-SEQ[11]
Advanced Indexing Algorithms	Hardware Optimization Multi-Representation Techniques	STMI[34], NDP[35] Z[3]
Query Optimization and Execution	Dynamic Execution Strategies Unified Data Management	Tao[4], ARTF[19] MC-VRF[8]

Table 1: This table presents a comprehensive overview of various methodologies utilized in vector databases for data indexing and query optimization. It categorizes the techniques into three primary areas: machine learning techniques for indexing, advanced indexing algorithms, and query optimization and execution strategies. Each category highlights specific methods and their contributions to enhancing the efficiency and accuracy of data management in high-dimensional vector databases.

The integration of advanced methodologies in data indexing and querying is essential for managing high-dimensional data complexities in vector databases. As the demand for efficient data retrieval grows, specific techniques that enhance indexing processes have become crucial. Figure 3 illustrates the hierarchical structure of data indexing and querying techniques in vector databases, highlighting machine learning methods, advanced indexing algorithms, and query optimization strategies. Each category encompasses specific methodologies that contribute to the efficiency and accuracy of data management in high-dimensional vector databases. Table 1 provides a detailed examination of the methodologies employed in vector databases for improving data indexing and query execution processes. Additionally, Table 4 presents a comprehensive comparison of various methodologies used in vector databases to enhance data indexing and query execution, emphasizing the roles of machine learning, advanced algorithms, and optimization strategies. The following subsection explores machine learning techniques for indexing, emphasizing their role in optimizing data management and retrieval systems. Understanding these methodologies highlights current advancements and sets the stage for future innovations in vector database technology, vital for applications across various domains.

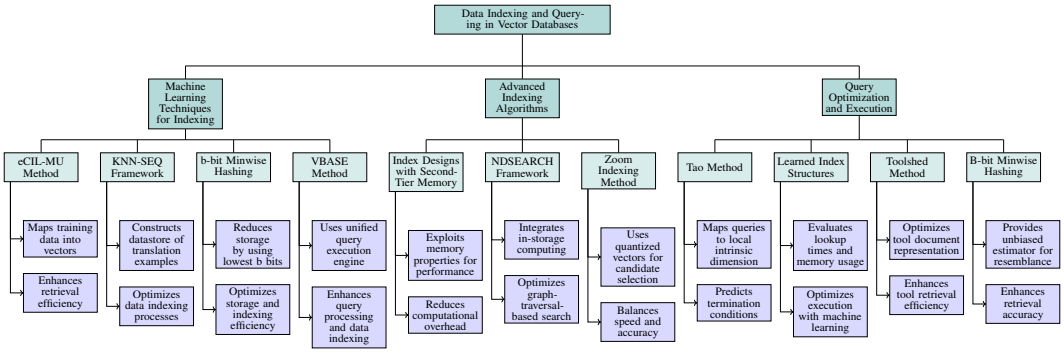


Figure 3: This figure illustrates the hierarchical structure of data indexing and querying techniques in vector databases, highlighting machine learning methods, advanced indexing algorithms, and query optimization strategies. Each category encompasses specific methodologies that contribute to the efficiency and accuracy of data management in high-dimensional vector databases.

Method Name	Indexing Efficiency	Data Embeddings	Scalability Solutions
eCIL-MU[25]	Significantly Improving Speed	Embedding Techniques	Non-destructive Unlearning
KNN-SEQ[11]	Efficient Indexing Methods	Key Vectors Computation	Large Datastores Scalability
DESSERT[32]	Significantly Improving Speed	Hash Values	Rapid Similarity Computations
ODCR[33]	Improve Context Retrieval	Embedding Vectors	Increasing Data Volume

Table 2: Overview of machine learning methods for enhancing indexing efficiency, data embeddings, and scalability solutions in vector databases. The table highlights four distinct approaches: eCIL-MU, KNN-SEQ, DESSERT, and ODCR, each contributing unique techniques such as embedding strategies and non-destructive unlearning to improve data management and retrieval processes.

4.1 Machine Learning Techniques for Indexing

Machine learning techniques significantly advance data indexing within vector databases, offering scalable solutions for high-dimensional data management across diverse applications. The eCIL-MU method exemplifies this by mapping training data into vectors and utilizing vector matching during inference to enhance retrieval efficiency [25]. This approach transforms raw data into structured embeddings, facilitating efficient data management and retrieval, crucial in today's data-driven landscape where data volume and variety are increasing.

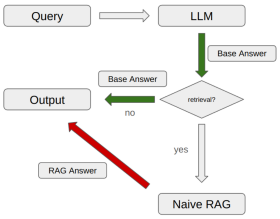
The KNN-SEQ framework illustrates efficient indexing methods by constructing a datastore of translation examples and retrieving relevant instances during translation, optimizing data indexing processes [11]. Such methodologies refine indexing techniques, ensuring rapid data access and retrieval. The innovative use of nearest neighbor search streamlines indexing and allows real-time updates, essential in dynamic environments with evolving data.

Incorporating techniques like b-bit minwise hashing reduces storage by using only the lowest b bits of each hashed value, showcasing machine learning's role in optimizing storage and indexing efficiency [10]. This optimization is crucial for managing high-dimensional data complexities and ensuring scalable data processing within vector databases, directly impacting querying operations' performance.

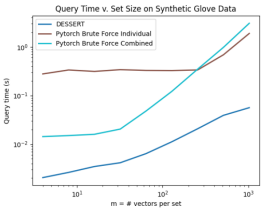
The VBASE method employs a unified query execution engine based on relaxed monotonicity, allowing flexible query processing and efficient data indexing [12]. This adaptability addresses high-dimensional data indexing challenges and optimizes query execution in vector databases. By adjusting to varying query requirements, VBASE enhances user experience by ensuring fast and accurate data retrieval, vital for applications requiring timely information.

These advancements underscore machine learning's transformative impact on vector database indexing. By converting user queries and documents into dense vector embeddings, machine learning enhances semantic data understanding, crucial for modern applications in domains like e-commerce, healthcare, and legal research. This evolution enables robust, accurate, and scalable data retrieval solutions, improving discovery accuracy and user satisfaction while addressing vector search implementation challenges, including optimized indexing strategies and advanced algorithms [13, 23].

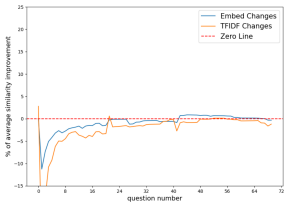
AI-generated, for reference only.



(a) Query-LLM-Base Answer-RAG Answer[36]



(b) Query Time v. Set Size on Synthetic Glove Data[32]



(c) The graph shows the percentage of average similarity improvement over time for three different methods: Embed Changes, TFIDF Changes, and the Zero Line.[33]

Figure 4: Examples of Machine Learning Techniques for Indexing

Exploring data indexing and querying within vector databases reveals machine learning techniques' pivotal role in enhancing efficiency and accuracy, as illustrated in Figure 4. The examples showcase methodologies and their impact on the process. "Query-LLM-Base Answer-RAG Answer" details integrating language models and retrieval-based answer generators to formulate query responses, highlighting generated versus database-retrieved answers. "Query Time v. Set Size on Synthetic Glove Data" examines query time and vector set size relationships across methods like DESSERT and Pytorch Brute Force, offering insights into scalability and performance. The graph on "The percentage of average similarity improvement over time" contrasts methods like Embed Changes and TFIDF Changes, enhancing similarity measures over successive queries. These examples underscore diverse strategies optimizing data indexing and querying, emphasizing machine learning's critical role in advancing vector databases [36, 32, 33]. Additionally, Table 2 presents a comparative analysis of various machine learning techniques employed to optimize indexing processes within vector databases, emphasizing their efficiency, data embedding strategies, and scalability solutions.

4.2 Advanced Indexing Algorithms

Advanced indexing algorithms optimize vector databases' performance and efficiency, crucial for handling high-dimensional data and accelerating similarity search processes. Innovative index designs exploiting second-tier memory properties achieve optimal performance with lower index amplification than traditional SSD-based methods [34]. Leveraging hardware-specific characteristics enhances indexing efficiency and reduces computational overhead, essential for real-time data processing applications.

The NDSEARCH framework integrates in-storage computing to accelerate graph-traversal-based approximate nearest neighbor search, optimizing data access patterns for improved search speed and accuracy [35]. Architectural innovations refine indexing algorithms and boost system performance, ensuring query speed and result accuracy, crucial for precision-dependent applications.

The Zoom indexing method employs a multi-view approach using quantized vectors for initial candidate selection and full-length vectors for accurate reranking [3]. This technique balances speed and accuracy, ensuring rapid initial filtering of potential matches while maintaining high precision in final results. A layered indexing approach enhances scalability and robustness in vector search systems, beneficial in large data environments for efficient processing without sacrificing outcome quality.

These advanced indexing algorithms highlight cutting-edge techniques and hardware optimizations' significance in vector databases. By tackling high-dimensional data management complexities and enhancing search efficiency, these algorithms evolve resilient and scalable vector database systems. These systems handle contemporary data-driven applications' increasing demands, such as recommendation engines, similarity searches, and natural language processing, utilizing numerical vector representations for efficient storage and retrieval. Implementing sophisticated indexing strategies and quantization techniques, as seen in projects like Quantixar, further optimizes performance and reduces computational costs, addressing high-dimensional and sparse data challenges [28, 37, 13].

AI-generated, for reference only.

4.3 Query Optimization and Execution

Method Name	Optimization Techniques	Data Management	Query Execution
Tao[4]	Adaptive Approaches	Complex Data Types	Structured Planning
ARTF[19]	Advanced Rag Techniques	Toolshed Knowledge Base	Query Planning Transformation
MC-VRF[8]	Adaptive Approaches	Data Storage Efficiency	Structured Planning

Table 3: Comparison of various query optimization and execution methods in vector databases, highlighting their distinct optimization techniques, data management strategies, and query execution plans. The table summarizes the adaptive approaches and structured planning methods utilized by each method, providing insights into their efficiency in handling high-dimensional data.

Query optimization and execution are critical in vector databases, particularly for efficient high-dimensional data management. Innovative methods streamline data retrieval and improve performance. Table 3 presents a comprehensive comparison of innovative query optimization and execution methods used in vector databases, elucidating their unique strategies for enhancing data retrieval and performance. Tao maps queries to a local intrinsic dimension number, predicting termination conditions to optimize execution by adapting to each query's characteristics [4]. This adaptive approach ensures efficient query processing, reducing retrieval time.

Learned index structures, as explored by Kraska et al., emphasize evaluating lookup times, memory usage, and performance against baseline structures, optimizing execution through machine learning techniques [2]. Leveraging historical performance data refines query plans, reducing execution time and resource consumption, significantly improving query performance for larger datasets.

The Toolshed method enhances tool retrieval by optimizing tool document representation and employing structured query planning and transformation, improving execution efficiency in rapid tool access environments [19]. Structuring queries ensures swift, accurate tool retrieval, enhancing system performance, useful in automated systems or real-time applications.

Integrating vector and raster data in a single database, demonstrated by Huang et al., reduces redundancy and improves organization, enabling simultaneous storage and efficient execution [8]. Handling complex data types ensures efficient execution without duplication, improving management efficiency and query response speed.

B-bit minwise hashing provides an unbiased estimator for resemblance, optimizing execution by reducing storage and enhancing retrieval accuracy [10]. Managing high-dimensional data efficiently ensures swift, accurate query processing, contributing to robust querying processes.

Recent vector database advancements highlight integrating machine learning techniques, structured query planning, and robust data management strategies to enhance execution efficiency. Driven by transitioning from token-based to vector-based approaches using neural networks for dense vector embeddings, these strategies improve discovery accuracy and user satisfaction in information retrieval systems across domains like e-commerce, healthcare, and legal research. Addressing high-dimensional and sparse vectorized data challenges, such as specialized indexing and optimization techniques, positions vector databases at the forefront of data management technology, paving the way for efficient and effective retrieval solutions [13, 23, 28, 38].

Feature	Machine Learning Techniques for Indexing	Advanced Indexing Algorithms	Query Optimization and Execution
Optimization Approach	Vector Embeddings	Hardware-specific	Adaptive Execution
Performance Focus	Retrieval Efficiency	Search Speed	Execution Efficiency
Scalability Solution	Real-time Updates	Layered Indexing	Structured Query Planning

Table 4: This table provides a comparative analysis of methodologies employed in vector databases, focusing on machine learning techniques for indexing, advanced indexing algorithms, and query optimization and execution. It highlights the optimization approaches, performance focus, and scalability solutions associated with each method, demonstrating their impact on data retrieval efficiency and execution effectiveness in high-dimensional environments.

5 Applications of Vector Databases

Vector databases are increasingly pivotal across sectors such as e-commerce, healthcare, academic research, and legal studies, significantly enhancing data management and retrieval. Transitioning from

traditional token-based methods to vector search allows these systems to leverage neural networks for converting queries and documents into dense vector embeddings, capturing semantic relationships and improving discovery accuracy and user satisfaction [13, 23]. This shift addresses technical challenges like indexing strategies and optimization techniques, paving the way for future applications. A key area of impact is in recommendation systems and personalization, where vector databases capture complex user preferences and item characteristics through high-dimensional vectors, fostering tailored user experiences and driving engagement.

5.1 Recommendation Systems and Personalization

Vector databases enhance recommendation systems and personalized experiences by efficiently managing high-dimensional vector data, capturing intricate user preferences and item characteristics. Applications such as image search and chatbots benefit from these capabilities, providing recommendations better aligned with user interests, thereby boosting engagement and satisfaction [28, 13]. Multimodal vector search systems, using models like CLIP, further enhance user experience by integrating diverse data types, crucial for generating accurate recommendations. The Faiss library exemplifies efficient similarity search capabilities essential for large-scale data handling, further augmented by innovations like DESSERT, which enhance search speed for low-latency applications [5].

In domains like astronomy, vector databases support efficient data querying, demonstrating adaptability across fields [39]. Their versatility extends to online marketplaces, where embedding fusion methods recommend auto parts, showcasing the applicability in diverse scenarios. Distance-sensitive hashing addresses the challenge of nuanced similarity assessments in recommendation systems, enhanced by strategies prioritizing retrieval performance through hard negatives and controlled diversity, reducing false negatives and improving response quality from Large Language Models (LLMs) [33, 40].

The Pyramid framework, evaluated with datasets like MovieLens, highlights vector databases in personalized movie recommendations, enhancing user satisfaction by leveraging dense vector embeddings for improved discovery accuracy. This framework addresses scaling challenges, including indexing and optimization techniques, contributing to refined recommendation systems [13, 28]. These advancements underscore the transformative impact of vector databases on recommendation systems, providing personalized experiences by efficiently managing high-dimensional data.

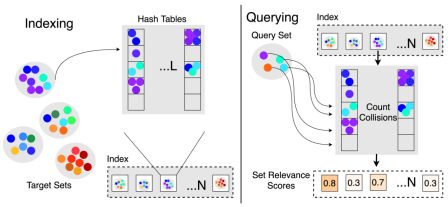
5.2 Domain-Specific Applications

Vector databases have significant applications across domains, efficiently managing and retrieving high-dimensional data tailored to specific needs. In air traffic control, language model-based agents utilize vector databases for human-like reasoning and decision-making [41], enhancing real-time data processing in critical domains. The BREATHE framework showcases versatility in fields like transistor modeling, cosmology, and neural architecture search, adapting vector databases for complex data structures and advanced analytics [42].

In spatial data management, vector databases handle multi-dimensional points for range queries and k-nearest neighbor searches, essential for geographic information retrieval [43]. The VOCAL-UDF framework optimizes video querying in video data management systems, highlighting adaptability in managing domain-specific data types [29]. The R UN O F T HE M ILL dataset, derived from YFCC100M, serves as a benchmark for vector search performance in image retrieval, demonstrating capabilities in image-intensive domains [9].

These examples illustrate vector databases' transformative impact in domain-specific applications, handling complex data environments and delivering tailored solutions across fields. Advanced indexing techniques and machine learning models enable efficient management of high-dimensional and sparse data, supporting similarity-based searches in e-commerce, healthcare, and natural language processing [28, 37, 44, 13].

As shown in Figure 5, domain-specific applications of vector databases harness data indexing and querying to address unique challenges. The first example, "Indexing and Querying for Set-Based Data," involves managing set-based data via a multi-level hash table, optimizing data retrieval in prevalent scenarios. The second example, "Thought Analysis Step-by-Step," emphasizes structured



(a) Indexing and Querying for Set-Based Data[32]

```
Thought: Let me analyze step-by-step:
1. Current Status:
   - Progress: [what's been tried]
   - Findings: [what's been found]
   - Missing: [what's still needed]

2. Strategy:
   - Next tool: [selected tool]
   - Expected outcome: [what we hope to find]
   - Confidence: [current confidence score 0-1]
   - Reasoning: [why this choice]
```

(b) Thought Analysis Step-by-Step[45]

Figure 5: Examples of Domain-Specific Applications

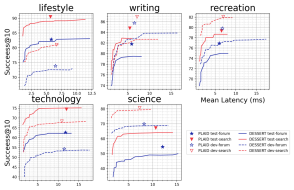
data representation in thought analysis, highlighting the importance of categorization. These examples demonstrate vector databases' adaptability and effectiveness in addressing specialized data challenges [32, 45].

5.3 Applications and Implications in Machine Learning

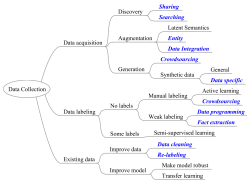
Vector databases are crucial in advancing machine learning applications, managing high-dimensional data to enhance model performance and computational efficiency. The eCIL-MU framework accelerates model training and enables non-destructive unlearning, impacting machine learning processes by developing adaptive models [25]. The KNN-SEQ framework illustrates scalable solutions for large datastores, optimizing data retrieval and processing for effective model handling [11].

VB ASE improves query performance and accuracy, essential for precise data retrieval in machine learning applications [12]. The b-bit minwise hashing method simplifies set similarity estimation, reducing storage while maintaining accuracy, beneficial for large-scale data processing [10]. These advancements highlight vector databases' impact on machine learning, offering robust solutions for modern applications across domains.

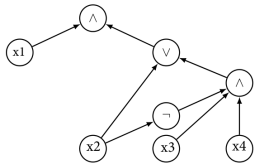
By integrating sophisticated indexing methods and machine learning models, vector databases enhance innovation in machine learning, managing high-dimensional and sparse data crucial for recommender systems, similarity search, and chatbots. Transitioning to vector search revolutionizes information retrieval by semantically processing user queries, improving discovery accuracy and user satisfaction in sectors like e-commerce, healthcare, and legal research. Techniques like Retrieval-Augmented Generation (RAG) and hybrid approaches combining knowledge graphs with vector retrieval advance data processing and analysis, particularly in specialized domains [13, 28, 46, 23, 44].



(a) Performance and Latency Comparison of PLiAD and DESSERT in Different Tasks and Settings[32]



(b) Data Collection and Data Collection Techniques[47]



(c) A Directed Acyclic Graph (DAG) with nodes labeled x1, x2, x3, and x4, and edges connecting them[48]

Figure 6: Examples of Applications and Implications in Machine Learning

As shown in Figure 6, vector databases significantly impact machine learning by optimizing real-time data processing and decision-making, as demonstrated by performance comparisons of PLiAD and DESSERT chatbots. Efficient data retrieval and processing speeds are critical for immediate response applications. Data collection mind maps emphasize vector databases' role in integrating diverse data sources for robust machine learning models, including crowdsourcing and synthetic data generation. The directed acyclic graph (DAG) visualizes dependencies and processes, illustrating how vector databases streamline complex data relationships, ensuring efficient data flow and reliability [32, 47, 48].

AI-generated, for reference only.

6 Challenges and Future Directions

Understanding the challenges faced by vector databases is crucial for enhancing their scalability and efficiency, particularly as the demand for managing large datasets intensifies. This section outlines the key obstacles and explores future research directions to improve vector databases' performance and applicability across various domains, focusing on scalability, integration with traditional systems, and prospective advancements.

6.1 Scalability and Efficiency Challenges

Vector databases encounter significant scalability and efficiency issues, especially in managing large datasets while ensuring fast query processing. The complexity of algorithms and the computational demands of neural network measures are major hurdles [23]. Limitations in Lucene's vector dimensions further underscore the need for enhancements in scalability and efficiency [7]. The rapid data growth necessitates that vector databases efficiently handle larger data volumes without compromising performance.

Integrating vector and raster data offers improved compression and organization but requires strategies to manage complexity and maintain data integrity [8]. The maturity of vector database management systems (VDBMSs) is hindered by stability and feature completeness issues, necessitating ongoing research to enhance scalability [28]. As real-time analytics demand grows, VDBMSs must scale efficiently while maintaining performance.

Zoom's indexing method, despite its benefits, poses implementation challenges and potential overheads in dynamic data scenarios [3]. Innovative indexing techniques, such as adaptive and hybrid approaches, could mitigate these issues by improving retrieval speed and accuracy.

Future research should focus on advanced indexing strategies, memory optimization, and algorithmic efficiency to address scalability and performance challenges. Enhancing prompt context quality in question-answer systems and implementing vector search techniques that leverage semantic embeddings can improve retrieval accuracy and user satisfaction. Outlier detection methods to filter irrelevant documents will be crucial for reducing processing complexity and enhancing large language models' performance [13, 33, 23].

6.2 Integration with Traditional Systems

Integrating vector databases with traditional systems involves overcoming challenges to ensure seamless operation and enhanced functionality. The quality of the entity database impacts applications like automatic speech recognition (ASR), necessitating efficient strategies for data processing [49]. Effective integration requires leveraging both traditional relational databases and modern vector databases.

Traditional systems, reliant on relational structures, may struggle with high-dimensional vector data. Hybrid systems, like HybridRAG, combine Knowledge Graph-based Retrieval Augmented Generation (GraphRAG) with VectorRAG techniques, enhancing question-answer systems and improving retrieval accuracy in complex environments [13, 28, 50, 46, 44]. These systems manage scalar and vector data, execute complex queries, and maintain data integrity.

Addressing data duplication and unorganized storage, particularly when combining vector and raster data, requires advanced indexing techniques and optimized data structures [8]. Standardized protocols for data interchange can facilitate smoother integration, ensuring seamless data flow and accessibility.

The stability and maturity of VDBMSs are critical for successful integration with traditional systems. Enhancements in feature completeness, reliability, and scalability are necessary to support modern applications [28]. Continuous evolution of VDBMSs must align with data-centric demands to complement traditional systems effectively.

Developing robust integration strategies will enable vector databases to complement traditional systems, enhancing data management capabilities and providing powerful solutions for complex environments. This integration maximizes vector databases' capabilities across applications like e-commerce, healthcare, and legal research, enhancing discovery accuracy and user satisfaction [13, 23, 28].

6.3 Future Directions in Vector Database Research

Future research in vector databases aims to enhance scalability, efficiency, and applicability across diverse domains. Optimizing query planning and execution by integrating additional indices could improve data retrieval performance and accuracy [12]. New indexing methodologies, such as learned indices, may address large-scale data processing challenges.

Real-time data processing and visualization represent promising research areas. Optimizing methods for larger datasets can expand vector databases' utility in dynamic environments, facilitating rapid analysis and visualization [8]. Integrating vector search capabilities into existing systems and developing managed services could streamline operations and enhance accessibility [7].

Enhancing semantic query understanding through attention mechanisms could improve search result accuracy and relevance [23]. Refining theoretical bounds on model size and accuracy, and exploring their implications for various data distributions, could improve vector databases' reliability in complex environments [28].

Future research should also focus on improving VDBMS algorithms, enhancing high-dimensional data readability, and addressing retraining costs. These efforts aim to make vector databases more accessible and efficient [28]. Enhancing the eCIL-MU framework for complex learning scenarios could advance vector databases' capabilities in machine learning [25].

Advancements in vector databases, including integration with neural networks for semantic understanding, are expected to improve accuracy and user satisfaction across sectors like e-commerce, healthcare, and legal research. These developments will facilitate efficient data retrieval and processing, providing robust tools for applications in recommender systems, chatbots, and environmental management [13, 46, 28].

7 Conclusion

This survey highlights the profound impact of vector databases on modern data management, illustrating their ability to significantly improve search accuracy and user satisfaction. Vector databases address the limitations of traditional keyword-based search methods by employing advanced algorithms that capture semantic relationships, thus enhancing information retrieval processes. This capability not only results in more relevant search outcomes but also elevates the user experience, establishing vector databases as vital components in the current data landscape.

As technologies such as the Internet of Things (IoT) and blockchain continue to advance, along with the increasing demand for secure 5G networks, the role of vector databases becomes even more crucial. These databases are adept at managing the complex and voluminous data generated by such technologies, offering real-time analytics and supporting seamless integration. Their inherent ability to handle high-dimensional data ensures that organizations can maintain competitiveness in a data-centric environment, suggesting a promising trajectory for vector databases in future data management strategies.

Moreover, the survey emphasizes the importance of optimizing model size within learned database operations to maintain accuracy across diverse datasets. This optimization is essential for ensuring the reliability and performance of vector databases, as their efficiency depends on processing and retrieving information accurately from extensive data sets. Therefore, ongoing research must focus on refining models to balance computational efficiency with retrieval accuracy, thereby enhancing the reliability and applicability of vector databases across various domains.

By integrating advanced indexing algorithms, machine learning techniques, and robust architectural solutions, vector databases are well-equipped to tackle the dynamic challenges of contemporary data management. These innovations provide powerful solutions that significantly improve the efficacy and accessibility of information retrieval systems. As organizations increasingly depend on data-driven decision-making, vector databases will play a pivotal role in facilitating efficient data retrieval and management. It is imperative for researchers and practitioners to collaborate in developing new methodologies to further enhance vector databases, ensuring they remain at the forefront of data management solutions.

References

[1] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library, 2025.

[2] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures, 2018.

[3] Minjia Zhang and Yuxiong He. Zoom: Ssd-based vector search for optimizing accuracy, latency and memory, 2018.

[4] Kaixiang Yang, Hongya Wang, Bo Xu, Wei Wang, Yingyuan Xiao, Ming Du, and Junfeng Zhou. Tao: A learning framework for adaptive nearest neighbor search using static features only, 2021.

[5] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. The faiss library. *arXiv preprint arXiv:2401.08281*, 2024.

[6] Owen Pendrigh Elliott and Jesse Clark. The impacts of data, ordering, and intrinsic dimensionality on recall in hierarchical navigable small worlds, 2024.

[7] Jimmy Lin, Ronak Pradeep, Tommaso Teofili, and Jasper Xian. Vector search with openai embeddings: Lucene is all you need, 2023.

[8] YS Huang, GQ Zhou, Tao Yue, HB Yan, WX Zhang, Xin Bao, QY Pan, and JS Ni. Vector and raster data layered fusion and 3d visualization. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:1127–1134, 2020.

[9] Gergely Szilvasy, Pierre-Emmanuel Mazaré, and Matthijs Douze. Vector search with small radiuses, 2024.

[10] Ping Li and Arnd Christian Konig. b-bit minwise hashing, 2009.

[11] Hiroyuki Deguchi, Hayate Hirano, Tomoki Hoshino, Yuto Nishida, Justin Vasselli, and Taro Watanabe. knn-seq: Efficient, extensible knn-mt framework, 2023.

[12] Qianxi Zhang, Shuotao Xu, Qi Chen, Guoxin Sui, Jiadong Xie, Zhizhen Cai, Yaoqi Chen, Yinxuan He, Yuqing Yang, Fan Yang, et al. {VBASE}: Unifying online vector similarity search and relational queries via relaxed monotonicity. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 377–395, 2023.

[13] Siddharth Pratap Singh. Vector search in the era of semantic understanding: A comprehensive review of applications and implementations. *INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING AND TECHNOLOGY (IJCET)*, 15(6):1794–1805, 2024.

[14] Nick Alonso, Tomás Figliolia, Anthony Ndirango, and Beren Millidge. Toward conversational agents with context and time sensitive long-term memory, 2024.

[15] Brad Windsor and Kevin Choi. Thistle: A vector database in rust, 2023.

[16] Rishabh Goel. Using text embedding models as text classifiers with medical data, 2024.

[17] Yinglong Ma and Moyi Shi. Using multi-categorization semantic analysis and personalization for semantic search, 2014.

[18] Luis Remis, Vishakha Gupta-Cledat, Christina Strong, and Ragaad Altarawneh. Vdms: Efficient big-visual-data access for machine learning workloads, 2018.

[19] Elias Lumer, Vamse Kumar Subbiah, James A. Burke, Pradeep Honaganahalli Basavaraju, and Austin Huber. Toolshed: Scale tool-equipped agents with advanced rag-tool fusion and tool knowledge bases, 2024.

[20] Frank Rosner and Alexander Hinneburg. Translating bayesian networks into entity relationship models, extended version, 2016.

AI-generated, for reference only.

[21] Shanshan Han, Zijian Hu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Dimitris Stripelis, Zhaozhuo Xu, and Chaoyang He. Torchopera: A compound ai system for llm safety, 2024.

[22] Andrew Kean Gao. Vec2vec: A compact neural network approach for transforming text embeddings with high fidelity, 2023.

[23] Solmaz Seyed Monir, Irene Lau, Shubing Yang, and Dongfang Zhao. Vectorsearch: Enhancing document retrieval with semantic embeddings and optimized search, 2024.

[24] Tiannuo Yang, Wen Hu, Wangqi Peng, Yusen Li, Jianguo Li, Gang Wang, and Xiaoguang Liu. Vdtuner: Automated performance tuning for vector data management systems, 2024.

[25] Zhiwei Zuo, Zhuo Tang, Bin Wang, Kenli Li, and Anwitaman Datta. ecil-mu: Embedding based class incremental learning and machine unlearning, 2024.

[26] Jason Mars, Yiping Kang, Roland Daynauth, Baichuan Li, Ashish Mahendra, Krisztian Flautner, and Lingjia Tang. The jaseci programming paradigm and runtime stack: Building scale-out production applications easy and fast, 2023.

[27] Sepanta Zeighami and Cyrus Shahabi. Towards establishing guaranteed error for learned database operations, 2024.

[28] Toni Taipalus. Vector database management systems: Fundamental concepts, use-cases, and current challenges, 2024.

[29] Enhao Zhang, Nicole Sullivan, Brandon Haynes, Ranjay Krishna, and Magdalena Balazinska. Self-enhancing video data management system for compositional events with large language models [technical report], 2025.

[30] Yiqi Li, Sheng Wang, Zhiyu Chen, Shangfeng Chen, and Zhiyong Peng. Approximate vector set search: A bio-inspired approach for high-dimensional spaces, 2024.

[31] Ismail Oukid and Lucas Lersch. On the diversity of memory and storage technologies, 2019.

[32] Joshua Engels, Benjamin Coleman, Vihan Lakshman, and Anshumali Shrivastava. Dessert: An efficient algorithm for vector set search with vector set queries. *Advances in Neural Information Processing Systems*, 36:67972–67992, 2023.

[33] Vitaly Bulgakov. Optimization of retrieval-augmented generation context with outlier detection, 2024.

[34] Rongxin Cheng, Yifan Peng, Xingda Wei, Hongrui Xie, Rong Chen, Sijie Shen, and Haibo Chen. Characterizing the dilemma of performance and index size in billion-scale vector search and breaking it with second-tier memory, 2024.

[35] Yitu Wang, Shiyu Li, Qilin Zheng, Linghao Song, Zongwang Li, Andrew Chang, Hai "Helen" Li, and Yiran Chen. Ndsearch: Accelerating graph-traversal-based approximate nearest neighbor search through near data processing, 2024.

[36] Tristan Kenneweg, Philip Kenneweg, and Barbara Hammer. Retrieval augmented generation systems: Automatic dataset creation, evaluation and boolean agent setup, 2024.

[37] Gulshan Yadav, RahulKumar Yadav, Mansi Viramgama, Mayank Viramgama, and Apeksha Mohite. Quantixar: High-performance vector data management system, 2024.

[38] Viktor Sanca and Anastasia Ailamaki. Efficient data access paths for mixed vector-relational search, 2024.

[39] Alexander S. Szalay, Peter Kunszt, Ani Thakar, and Jim Gray. Designing and mining multi-terabyte astronomy archives: The sloan digital sky survey, 1999.

[40] Timo Kats, Peter van der Putten, and Jan Scholtes. Relevance feedback strategies for recall-oriented neural information retrieval, 2023.

[41] Justas Andriuškevičius and Junzi Sun. Automatic control with human-like reasoning: Exploring language model embodied air traffic agents, 2024.

[42] Shikhar Tuli and Niraj K. Jha. Breathe: Second-order gradients and heteroscedastic emulation based design space exploration, 2023.

[43] Qiyu Liu, Maocheng Li, Yuxiang Zeng, Yanyan Shen, and Lei Chen. How good are multi-dimensional learned indices? an experimental survey, 2024.

[44] Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 608–616, 2024.

[45] Ioannis Papadimitriou, Ilias Gialampoukidis, Stefanos Vrochidis, Ioannis, and Kompatsiaris. Rag playground: A framework for systematic evaluation of retrieval strategies and prompt engineering in rag systems, 2024.

[46] Hang Yang, Jing Guo, Jianchuan Qi, Jinliang Xie, Si Zhang, Siqi Yang, Nan Li, and Ming Xu. A method for parsing and vectorization of semi-structured data used in retrieval augmented generation, 2024.

[47] Yuji Roh, Geon Heo, and Steven Euijong Whang. A survey on data collection for machine learning: a big data – ai integration perspective, 2019.

[48] Leopoldo Bertossi. Attribution-scores in data management and explainable machine learning, 2023.

[49] Ernest Pusateri, Anmol Walia, Anirudh Kashi, Bortik Bandyopadhyay, Nadia Hyder, Sayantan Mahinder, Raviteja Anantha, Daben Liu, and Sashank Gondala. Retrieval augmented correction of named entity speech recognition errors, 2024.

[50] Bhaskarjit Sarmah, Benika Hall, Rohan Rao, Sunil Patel, Stefano Pasquali, and Dhagash Mehta. Hybridrag: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction, 2024.

Disclaimer:

SurveyX is an AI-powered system designed to automate the generation of surveys. While it aims to produce high-quality, coherent, and comprehensive surveys with accurate citations, the final output is derived from the AI's synthesis of pre-processed materials, which may contain limitations or inaccuracies. As such, the generated content should not be used for academic publication or formal submissions and must be independently reviewed and verified. The developers of SurveyX do not assume responsibility for any errors or consequences arising from the use of the generated surveys.

AI-generated, for reference only.