The Quest for Faster ANN Vector Search

Manos Chatzakis
Université Paris Cité
Paris, France
manos.chatzaki@gmail.com

Francesca Del Gaudio
Université Paris Cité
Paris, France
francesca.del-gaudio@etu.u-paris.fr

Sophia Sideri
Université Paris Cité
Paris, France
UoC, Greece
sophisid@csd.uoc.gr

Themis Palpanas
Université Paris Cité
Paris, France
themis@mi.parisdescartes.fr

Abstract

The increasing demand for faster and more scalable systems for Approximate Nearest Neighbor (ANN) vector search has led, in recent years, to the development of several novel methods that achieve significantly faster query execution times. These advances typically rely on parallel and distributed processing, support for streaming and evolving data, and early termination approaches, which have demonstrated state-of-the-art performance for ANN vector search tasks. In this tutorial, we provide a comprehensive review of recent advances in the field of vector search along these directions, and we identify challenges and open problems.

Keywords

Vector Search, Parallel & Distributed Processing, Streaming, Early Termination

1 Introduction

Approximate Nearest Neighbor (ANN) vector search is a fundamental component of a wide variety of modern tasks, including applications of Retrieval Augmented Generation (RAG) [31], multimedia databases [48], and recommendation systems [47], as it enables fast approximate retrieval of the top-k nearest vectors to a given query vector from a large collection of database vectors. In recent years, we have witnessed a rapidly growing demand for vector search, which has led to the development of purpose-built vector search systems [45, 52, 54] and the integration of vector search capabilities into non-relational [36, 53] and relational database systems [3, 13, 39].

To meet modern performance demands, vector search moves beyond single-threaded execution. Parallel processing exploits the capabilities of multi-core CPUs and GPUs by distributing the computational workload, accelerating both index construction and search response times [17, 18, 44]. In addition, distributed systems become essential to provide the necessary storage capacity, computational throughput, and overall system resilience. Cutting-edge solutions [16, 58, 64] implement advanced partitioning and data-locality strategies to maintain low latency across billions of vectors. In the meantime, workloads can be dynamic with continuous updates and evolving query distributions [24]. This challenges traditional ANN indexes that mostly operate on static data, updates are not immediately searchable, and changes in data distribution require effort to rebuild the index from scratch. Recent work handles vector search as a core streaming operation, enabling incremental indexing, progressive query answering, and adaptive query execution that enable fast search and high recall under continuous changes in the data distribution [34, 49, 55, 57, 61, 65]. Finally, several novel solutions have demonstrated that early termination is a promising direction for achieving significantly faster vector search [8, 10, 27, 32, 38, 61, 62], by controlling the termination points of each query search individually, without relying on predefined static search hyperparameters. Some recent studies are exploring solutions that combine ideas from all these three directions (i.e., parallel, streaming, and early termination) [38].

In this tutorial, we provide a comprehensive review of state-of-the-art advancements for faster vector search through parallel and distributed processing, novel search methods for streaming and evolving workloads, and early termination approaches, and identify directions for future research, open problems, and challenges.

Target Audience and Expected Background. This tutorial is designed to be suitable for both data management researchers and practitioners interested in the domain of vector search. During this tutorial, we cover the background information necessary to follow the entire presentation, along with the technical details of the presented approaches. Thus, both experts in the field and interested newcomers will be able to follow the material.

Relation to Previous Tutorials. In ANN vector search, existing tutorials have mainly focused on different types and taxonomies of indexing methods [7, 46], while [12] provided a comprehensive overview of filtered ANN vector search. This tutorial is complementary, presenting the most recent advancements in efficient ANN vector search, with a particular focus on parallel and distributed approaches, streaming support, and early termination methods, which have proven to be very promising in optimizing vector search performance. Finally, previous tutorials have presented overviews of exact nearest neighbor search for data series, a different type of high-dimensional vectors [19, 21].

2 Tutorial Scope

In this 1.5-hour tutorial, we cover recent advancements in data management that enable highly efficient and faster vector search. Below, we present a summary of the sections of the tutorial, along with a brief outline of the contents of each section.

(1) **Vector Search Basics** *(Themis Palpanas, 10 min)*
- Introduction and motivation for vector search.
- Background and preliminaries.
- Connections to data series similarity search solutions.

(2) **Parallel and Distributed Vector Search** *(Francesca Del Gaudio, 20 min)*
- Motivation and setup.
- Parallel processing approaches.
- Distributed processing approaches.

(3) **Streaming Vector Search** *(Sophia Sideri, 20 min)*
- Vector search challenges with evolving data.
- Approaches for streaming vector search.
- Open problems and future directions.

(4) **Early Termination** *(Manos Chatzakis, 20 min)*
- Motivation and gains of early termination.
- Challenges of hyperparameter tuning.
- Review of early termination methods.

EDBT '26, Tampere (Finland)

(5) **Concluding Remarks** *(Manos Chatzakis, 10 min)*
- Remarks on quality evaluation.
- Future directions and challenges.

2.1 Vector Search Basics

The tutorial introduces the problem of vector search and demonstrates the numerous applications in which vector search serves as a backbone element. Then, it presents the essential information related to indexing with different types of structures [18, 28, 35], as well as quantization techniques [22, 29]. Additionally, it introduces some key research advancements from the field of exact nearest neighbor data series search [40], and it connects the solutions the data series community developed for similar problems to vector search, including parallel and distributed processing [9, 42, 59], streaming [30], and early termination [20, 23].

2.2 Parallel and Distributed Vector Search

In this part, we motivate the need for parallel and distributed processing to accelerate vector search. We highlight the limitations of standalone approaches and provide an overview of existing parallel and distributed solutions, summarized in Table 1.

We begin by presenting methods for parallel query processing in vector search, exploring the spectrum of intra-query and inter-query parallelism, as well as approaches that combine both. These include graph-based ANN approaches that exploit intra-query parallelism during search [5, 44] and frameworks that focus on scalable parallel index construction and execution [17]. Furthermore, we discuss GPU-accelerated systems, which leverage massive hardware parallelism to improve query throughput, parallelizing distance computations and top-k selection [18, 33]. We also examine recent adaptive ANN approaches, which combine parallel processing with query-specific optimizations, by an indexing scheme that adjusts to dynamic query distribution [38].

We then present different distributed processing techniques for vector search, explain how they can be combined with parallel processing, and discuss how they generalize across different index types. These include approaches based on hierarchical data partitioning and routing [1, 16], work that explores balanced graph partitioning coupled with modular routing mechanisms [25], as well as systems that distribute ANN processing across nodes while combining local parallel search with coordination mechanisms, [14, 58]. We further discuss recent distributed approaches that integrate cooperative and adaptive search strategies to improve scalability and robustness under large-scale and heterogeneous workloads [64]. We also highlight how early termination is widely adopted in distributed vector search, and how approaches differ in the degree of communication and synchronization required, from global coordination to local pruning [56, 62].

We conclude this part by summarizing open problems and future directions in parallel and distributed processing, and by discussing open research challenges.

2.3 Streaming

In the following part, we shift our focus to vector search in dynamic and streaming environments, where data collections and query workloads evolve and change over time.

Table 2 summarizes the approaches of interest and their key features, including information about the index type, and the hardware used for query processing. We distinguish whether updates are served in batches or incrementally, and for incremental updates, whether the index is modified in-place [37, 51, 55, 57, 63] (i.e., the state of the index immediately reflects the new state), out-of-place [24] (i.e., the modifications are first marked in the index and materialized at a later point) or in batches [2, 34, 38, 49]. The approaches that partially rebuild the index [37, 38, 57] are indicated by an asterisk; the last column shows if the approach supports parallel or distributed execution.

In this part of the tutorial, we motivate the need for streaming-aware vector search by discussing how modern systems work with frequent insertions, deletions, and updates, as well as examining why traditional ANN indexes, designed for static data and offline index construction, struggle in such conditions [24, 34, 61]. We then present the main goals of a streaming vector search system, namely, supporting low-latency and high-recall queries, enabling updates on vectors to become immediately searchable, and avoiding expensive re-builds of the index [24, 63]. Based on these goals, we describe solutions that support incremental and in-place index updates [37, 51, 55, 57, 63]. We further explore extensions of streaming vector search, including dynamic and adaptive query execution techniques, indexing compression [2], and integration with stream processing [34, 38, 43].

Finally, we conclude by outlining open key challenges related to update efficiency, recall preservation, and consistency in streaming scenarios.

Table 1: Parallel and distributed approaches.

Method	Index	Storage	Hardware	Optimization Target	Execution
iQAN [44]	k-NN Graph	In-memory	CPU	Latency	Parallel
ParlayANN [17]	k-NN Graph	In-memory	CPU	Throughput	Parallel
Quake [38]	IVF	In-memory	CPU	Latency	Parallel
ELPIS [5]	Tree / k-NN Graph	In-memory	CPU	Recall/Throughput	Parallel
FAISS [18]	IVF	In-memory	GPU	Throughput	Parallel
Tagore [33]	k-NN Graph	Both	GPU	Index Construction	Parallel
DistributedANN [1]	k-NN Graph	Both	CPU	Latency + Throughput	Distributed
Pyramid [16]	k-NN Graph	In-memory	CPU	Throughput	Distributed
BatANN [14]	k-NN Graph	On-Disk	CPU	Throughput	Distributed
SPIRE [58]	k-NN Graph	Both	CPU	Throughput	Both
CoTra [64]	k-NN Graph	In-memory	CPU	Latency + Throughput	Distributed
Auncel [62]	IVF	In-memory	CPU	Latency	Distributed
Harmony [56]	IVF	In-memory	CPU	Throughput	Both
Graph Partitioning [25]	k-NN Graph	In-memory	CPU	Latency + Throughput	Both

Table 2: Streaming approaches; we list the characteristics of the Insert (I), Delete (D), and Update (U) operations.

Method	Index Type	Hardware	Incremental	Updates	Execution
FreshDiskANN [49]	k-NN Graph	CPU	I, D, U	batched	parallel
SPFresh* [57]	k-NN Graph	CPU	I, D, U	in-place	parallel
CleANN [63]	k-NN Graph	CPU	I, D, U	in-place	parallel
IP-DiskANN [55]	k-NN Graph	CPU	I, D, U	in-place	parallel
DEG [26]	K-NN Graph	CPU	I	batched	parallel
SVS [2]	k-NN Graph	CPU	-	batched	parallel
Ada-IVF* [37]	IVF	CPU	I, D, U	in-place	parallel
VStream [24]	k-NN Graph / IVF	CPU	I, D, U	out-of-place	both
Quake* [38]	IVF	CPU	I	batched	parallel
VectraFlow [34]	OPList	CPU	I	batched	parallel
RTAMS-GANNS [51]	IVF	GPU	I	in-place	parallel

Table 3: Early termination methods in vector search.

Method	Index	Dist. Meas.	Training	Tuning	Termination
LAET [32]	k-nn Graph / IVF	any	✓	✓	when neighbors found
PatienceEE [8]	IVF	any		✓	
SPANN [11]	IVF	any		✓	based on thresholds
ConANN [27]	IVF	any		✓	
Auncel [62]	IVF	any		✓	based on target recall
Quake [38]	IVF	any			
DARTH [10]	k-nn Graph / IVF	any	✓		
Ada-ef [61]	k-nn Graph	inner product			

2.4 Early Termination

Early termination approaches have emerged as a promising direction for performance optimization in ANN vector search. This family of methods achieves better performance than vector search approaches that focus on hyperparameter tuning to obtain good average performance for a query workload [15, 50, 60]. The main reason is that early termination is based on dynamic query signals, which improve the search of each query individually.

In this part of the tutorial, we cover the most recent advancements in early termination, presenting a wide variety of methods with different components and characteristics. We describe methods that early terminate the search once all the nearest neighbors of a query have been found (LAET [32], PatienceEE [8]), methods that terminate using predefined thresholds (SPANN [11], ConANN [27]), and methods that early terminate based on user-defined declarative recall targets (Auncel [62], Quake [38], Ada-ef [61], DARTH [10]). These methods, along with their main characteristics, are summarized in Table 3. Note that all the above methods are agnostic to the distance measure used by the index, except for Ada-ef [61], which only works for the inner product distance measure. We also discuss future promising research directions and open challenges for early termination.

2.5 Concluding Remarks

The final part of this tutorial provides directions for assessing the quality of vector search results. Current vector search benchmarks evaluate the average recall over a query workload [4, 6]. However, previous works [10, 41] demonstrated that fully assessing the result quality of a vector search algorithm should involve additional measures, such as the Ratio of Queries Under the recall Target (RQUT) [10], the Relative Distance Error (RDE) [41], recall distributions [10] and recall percentiles [10, 61].

We conclude the tutorial by summarizing recent research trends in ANN vector search, and summarizing the key challenges and open research directions.

3 Presenters

Manos Chatzakis is a PhD student at Université Paris Cité, working on vector data management. He holds an MSc from EPFL and a BSc from the University of Crete. He has worked at CEA (France), EPFL (Switzerland), and ICS-FORTH (Greece), as well as Google (USA), where he implemented vector search solutions in Google database products.

Francesca Del Gaudio is a PhD student at Université Paris Cité, working on efficient similarity search in high-dimensional vector spaces, with a particular focus on scalable and distributed systems. She holds an MSc and a BSc from the University of Calabria, where she also worked as a research intern collaborating with the Department of Biology and Genetics. She has been awarded a Marie Skłodowska-Curie Actions (MSCA) PhD scholarship and several Excellence Scholarships from the University of Calabria.

Sophia Sideri is a PhD student at Université Paris Cité and University of Crete, working on high-dimensional vector search over streaming data. She holds an MSc and a BSc from the University of Crete. She worked as a research intern at Huawei Technologies R&D (UK) and ICS-FORTH (Greece).

Themis Palpanas is an ACM Fellow, a Senior Fellow of the French University Insitute (IUF), a Distinguished Professor of Computer Science at Université Paris Cité, and has been the Founding Director of the Data Intelligence Institute of Paris (diiP). He has authored 15 patents, received 3 best paper awards and the IBM SUR award, has been Program Chair for VLDB 2025 and IEEE BigData 2023, General Chair for VLDB 2013, and has served as Editor in Chief for BDR. He has been working on high-dimensional vector management and analytics for more than 15 years, and has developed several of the state of the art methods. He has delivered 19 tutorials in top international conferences.

Acknowledgments

Work supported by EU Horizon projects ARMADA (101168951), TwinODIS (101160009), DataGEMS (101188416), ΥΠΑΙΘΑ & NextGenerationEU project HARSH (ΥΠ3ΤΑ – 0560901), and DEDALUS CREs (TA 5180519). Manos Chatzakis is supported with a PhD Scholarship from the Onassis Foundation.

References

[1] Philip Adams, Menghao Li, Shi Zhang, Li Tan, Mingqin Li, Qi Chen, Knut Magne Risvik, Jason (Zengzhong) Li, and Harsha Simhadri. 2025. DistributedANN: Efficient Scaling of a Single DiskANN Graph Across Thousands of Computers. In *The 1st Workshop on Vector Databases at the 2025 International Conference on Machine Learning*.

[2] Cecilia Aguerrebere, Ishwar Singh Bhati, Mark Hildebrand, Mariano Tepper, and Theodore L. Willke. 2023. Similarity search in the blink of an eye with compressed indices. *Proc. VLDB Endow.* 16, 11 (2023), 3433–3446. doi:10.14778/3611479.3611537

[3] Amazon Web Services. [n. d.]. Amazon Aurora PostgreSQL. https://aws.amazon.com/rds/aurora-postgresql/. Accessed: 2024-11-26.

[4] Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull. 2020. ANN-Benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. *Information Systems* 87 (2020).

[5] Ilias Azizi, Karima Echihabi, and Themis Palpanas. 2024. ELPIS: Graph-Based Similarity Search for Scalable Data Science. *Proceedings of the VLDB Endowment* 17, 12 (2024).

[6] Ilias Azizi, Karima Echihabi, and Themis Palpanas. 2025. Graph-based vector search: An experimental evaluation of the state-of-the-art. *Proceedings of the ACM on Management of Data* 3, 1 (2025).

[7] Sebastian Bruch. 2025. Advances in Vector Search *(WSDM '25)*. Association for Computing Machinery, New York, NY, USA. doi:10.1145/3701551.3703482

[8] Francesco Busolin, Claudio Lucchese, Franco Maria Nardini, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. 2024. Early exit strategies for approximate k-NN search in dense retrieval. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*.

[9] Manos Chatzakis, Panagiota Fatourou, Eleftherios Kosmas, Themis Palpanas, and Botao Peng. 2023. Odyssey: A Journey in the Land of Distributed Data Series Similarity Search. *Proc. VLDB Endow.* 16, 5 (2023), 14 pages. doi:10.14778/3579075.3579087

[10] Manos Chatzakis, Yannis Papakonstantinou, and Themis Palpanas. 2025. DARTH: Declarative Recall Through Early Termination for Approximate Nearest Neighbor Search. *Proceedings of the ACM on Management of Data* 3, 4 (2025).

[11] Qi Chen, Bing Zhao, Haidong Wang, Mingqin Li, Chuanjie Liu, Zengzhong Li, Mao Yang, and Jingdong Wang. 2021. Spann: Highly-efficient billion-scale approximate nearest neighborhood search. *Advances in Neural Information Processing Systems* 34 (2021).

[12] Yannis Chronis, Helena Caminal, Yannis Papakonstantinou, Fatma Özcan, and Anastasia Ailamaki. 2025. Filtered vector search: State-of-the-art and research opportunities. *Proceedings of the VLDB Endowment* 18, 12 (2025).

[13] Google Cloud. 2024. AlloyDB for PostgreSQL. https://cloud.google.com/alloydb/docs/overview Accessed: 2024-12-22.

[14] Nam Anh Dang and Ben Landrum. 2025. Passing the Baton: High Throughput Distributed Disk-Based Vector Search with BatANN. In *arXiv preprint arXiv:2512.09331*.

[15] Samuel Daulton, Maximilian Balandat, and Eytan Bakshy. 2020. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems* 33 (2020).

[16] Shiyuan Deng, Xiao Yan, Kelvin Kai Wing Ng, Chenyu Jiang, and James Cheng. 2019. Pyramid: A General Framework for Distributed Similarity Search on Large-scale Datasets. In *2019 IEEE International Conference on Big Data (IEEE BigData), Los Angeles, CA, USA, December 9-12, 2019*. IEEE. doi:10.1109/BIGDATA47090.2019.9006219

[17] Magdalen Dobson, Zheqi Shen, Yan Gu, Yihan Sun, and Laxman Dhulipala. 2024. ParlayANN: Scalable and Deterministic Parallel Graph-Based Approximate Nearest Neighbor Search. In *Proceedings of the 2024 International Conference on Management of Data (SIGMOD)*. ACM.

[18] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss Library. *IEEE Transactions on Big Data* (2025). doi:10.1109/TBDATA.2025.3618474

[19] Karima Echihabi and Themis Palpanas. 2022. Scalable Analytics on Large Sequence Collections. In *2022 23rd IEEE International Conference on Mobile*

Data Management (MDM). IEEE.
[20] Karima Echihabi, Theophanis Tsandilas, Anna Gogolou, Anastasia Bezerianos, and Themis Palpanas. [n. d.]. ProS: data series progressive k-NN similarity search and classification with probabilistic quality guarantees. *The VLDB Journal* 32, 4 ([n. d.]).
[21] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2021. New trends in high-d vector similarity search: ai-driven, progressive, and distributed. *Proceedings of the VLDB Endowment* 14, 12 (2021).
[22] Jianyang Gao and Cheng Long. 2024. RaBitQ: Quantizing High-Dimensional Vectors with a Theoretical Error Bound for Approximate Nearest Neighbor Search. *Proceedings of the ACM on Management of Data* 2, 3 (2024).
[23] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. 2019. Progressive similarity search on time series data. In *BigVis 2019-2nd International Workshop on Big Data Visual Exploration and Analytics*.
[24] Shenghao Gong, Haobo Sun, Ziquan Fang, Liu Liu, Lu Chen, and Yunjun Gao. 2025. VStream: A distributed streaming vector search system. *Proceedings of the VLDB Endowment* 18, 6 (2025).
[25] Lars Gottesbüren, Laxman Dhulipala, Rajesh Jayaram, and Jakub Lacki. 2025. Unleashing Graph Partitioning for Large-Scale Nearest Neighbor Search. *Proc. VLDB Endow.* 18, 6 (2025), 1649–1662. doi:10.14778/3725688.3725696
[26] Nico Hezel, Kai Uwe Barthel, Konstantin Schall, and Klaus Jung. 2023. Fast Approximate Nearest Neighbor Search with a Dynamic Exploration Graph using Continuous Refinement. *CoRR* abs/2307.10479 (2023). doi:10.48550/ARXIV.2307.10479 arXiv:2307.10479
[27] Sonia Horchidan, Fabian Zeiher, Henrik Boström, and Paris Carbone. 2025. ConANN: Conformal Approximate Nearest Neighbor Search. *Proceedings of the VLDB Endowment* 19, 1 (2025).
[28] Qiang Huang, Jianlin Feng, Yikai Zhang, Qiong Fang, and Wilfred Ng. 2015. Query-aware locality-sensitive hashing for approximate nearest neighbor search. *Proceedings of the VLDB Endowment* 9, 1 (2015).
[29] Herve Jegou, Matthijs Douze, and Cordelia Schmid. 2010. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence* 33, 1 (2010).
[30] Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, and Themis Palpanas. 2019. Coconut palm: Static and streaming data series exploration now in your palm. In *Proceedings of the 2019 International Conference on Management of Data*.
[31] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020).
[32] Conglong Li, Minjia Zhang, David G Andersen, and Yuxiong He. 2020. Improving approximate nearest neighbor search through learned adaptive early termination. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*.
[33] Zhonggen Li, Xiangyu Ke, Yifan Zhu, Bocheng Yu, Baihua Zheng, and Yunjun Gao. 2025. Scalable Graph Indexing using GPUs for Approximate Nearest Neighbor Search. *Proc. ACM Manag. Data* 3, 6 (2025). doi:10.1145/3769825
[34] Duo Lu, Siming Feng, Jonathan Zhou, Franco Solleza, Malte Schwarzkopf, and Uğur Çetintemel. 2025. VectraFlow: Integrating Vectors into Stream Processing. CIDR.
[35] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence* 42, 4 (2018).
[36] Microsoft Azure. [n. d.]. Azure Cosmos DB. https://learn.microsoft.com/en-us/azure/cosmos-db/vector-database. Accessed: 2024-12-19.
[37] Jason Mohoney, Anil Pacaci, Shihabur Rahman Chowdhury, Umar Farooq Minhas, Jeffrey Pound, Cédric Renggli, Nima Reyhani, Ihab F Ilyas, Theodoros Rekatsinas, and Shivaram Venkataraman. 2024. Incremental IVF Index Maintenance for Streaming Vector Search. *CoRR* (2024).
[38] Jason Mohoney, Devesh Sarda, Mengze Tang, Shihabur Rahman Chowdhury, Anil Pacaci, Ihab F. Ilyas, Theodoros Rekatsinas, and Shivaram Venkataraman. 2025. Quake: adaptive indexing for vector search. In *Proceedings of the 19th USENIX Conference on Operating Systems Design and Implementation* (Boston, MA, USA) *(OSDI '25)*. USENIX Association, USA, Article 9, 17 pages.
[39] Oracle Corporation. [n. d.]. Oracle AI Vector Search. https://www.oracle.com/database/ai-vector-search/. Accessed: 2024-11-26.
[40] Themis Palpanas. 2020. Evolution of a Data Series Index: The iSAX Family of Data Series Indexes: iSAX, iSAX2. 0, iSAX2+, ADS, ADS+, ADS-Full, ParIS, ParIS+, MESSI, DPiSAX, ULISSE, Coconut-Trie/Tree, Coconut-LSM. In *Information Search, Integration, and Personalization: 13th International Workshop, ISIP 2019, Heraklion, Greece, May 9–10, 2019, Revised Selected Papers 13*. Springer.
[41] Marco Patella and Paolo Ciaccia. 2008. The many facets of approximate similarity search. In *First International Workshop on Similarity Search and Applications (sisap 2008)*. IEEE.
[42] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2020. Messi: In-memory data series indexing. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE.
[43] Zhencan Peng, Miao Qiao, Wenchao Zhou, Feifei Li, and Dong Deng. 2025. Dynamic Range-Filtering Approximate Nearest Neighbor Search. *Proceedings of the VLDB Endowment* 18, 10 (2025).
[44] Zhen Peng, Minjia Zhang, Ruoming Jin, Kai Li, and Bin Ren. 2022. iQAN: Fast and Accurate Vector Search with Efficient Intra-Query Parallelism on Multi-Core Architectures. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP)*. ACM.
[45] Pinecone, Inc. [n. d.]. Pinecone. https://www.pinecone.io/. Accessed: 2024-12-19.
[46] Jianbin Qin, Wei Wang, Chuan Xiao, Ying Zhang, and Yaoshu Wang. 2021. High-Dimensional Similarity Query Processing for Data Science *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2 pages. doi:10.1145/3447548.3470811
[47] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023).
[48] Viktor Sanca, Manos Chatzakis, and Anastasia Ailamaki. 2024. Optimizing Context-Enhanced Relational Joins. (2024). doi:10.1109/ICDE60146.2024.00045
[49] Aditi Singh, Suhas Jayaram Subramanya, Ravishankar Krishnaswamy, and Harsha Vardhan Simhadri. 2021. FreshDiskANN: A Fast and Accurate Graph-Based ANN Index for Streaming Similarity Search. *CoRR* abs/2105.09613 (2021).
[50] Philip Sun, Ruiqi Guo, and Sanjiv Kumar. 2023. Automating Nearest Neighbor Search Configuration with Constrained Optimization. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
[51] Yiping Sun, Yang Shi, and Jiaolong Du. 2024. A Real-Time Adaptive Multi-Stream GPU System for Online Approximate Nearest Neighborhood Search. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*.
[52] SeMI Technologies. 2019. Weaviate: Open-Source Vector Search Engine. https://weaviate.io. Accessed: 2025-01-15.
[53] Nitish Upreti, Harsha Vardhan Simhadri, Hari Sudan Sundar, Krishnan Sundaram, Samer Boshra, Balachandar Perumalswamy, Shivam Atri, Martin Chisholm, Revti Raman Singh, Greg Yang, Tamara Hass, Nitesh Dudhey, Subramanyam Pattipaka, Mark Hildebrand, Magdalen Dobson, Jack Moffitt, Haiyang Xu, Naren Datha, Suryansh Gupta, Ravishankar Krishnaswamy, Prashant Gupta, Abhishek Sahu, Hemeswari Varada, Sudhanshu Barthwal, Ritika Mor, James Codella, Shaun Cooper, Kevin Pilch, Simon Moreno, Aayush Kataria, Santosh Kulkarni, Neil Deshpande, Amar Sagare, Dinesh Billa, Zishan Fu, and Vipul Vishal. 2025. Cost-Effective, Low Latency Vector Search with Azure Cosmos DB. *Proc. VLDB Endow.* 18, 12 (2025). doi:10.14778/3750601.3750635
[54] Jianguo Wang, Xiaomeng Yi, Rentong Guo, Hai Jin, Peng Xu, Shengjun Li, Xiangyu Wang, Xiangzhou Guo, Chengming Li, Xiaohai Xu, et al. 2021. Milvus: A purpose-built vector data management system. In *Proceedings of the 2021 International Conference on Management of Data*.
[55] Haike Xu, Magdalen Dobson Manohar, Philip A. Bernstein, Badrish Chandramouli, Richard Wen, and Harsha Vardhan Simhadri. 2025. In-Place Updates of a Graph Index for Streaming Approximate Nearest Neighbor Search. *CoRR* abs/2502.13826 (2025).
[56] Qian Xu, Feng Zhang, Chengxi Li, Lei Cao, Zheng Chen, Jidong Zhai, and Xiaoyong Du. 2025. Harmony: A scalable distributed vector database for high-throughput approximate nearest neighbor search. *Proceedings of the ACM on Management of Data* 3, 4 (2025), 1–28.
[57] Yuming Xu, Hengyu Liang, Jin Li, Shuotao Xu, Qi Chen, Qianxi Zhang, Cheng Li, Ziyue Yang, Fan Yang, Yuqing Yang, Peng Cheng, and Mao Yang. 2023. SPFresh: Incremental In-Place Update for Billion-Scale Vector Search. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*. ACM. doi:10.1145/3600006.3613166
[58] Yuming Xu, Qianxi Zhang, Qi Chen, Baotong Lu, Menghao Li, Philip Adams, Mingqin Li, Zengzhong Li, Jing Liu, Cheng Li, and Fan Yang. 2025. Scalable Distributed Vector Search via Accuracy Preserving Index Construction. *arXiv preprint arXiv:2512.17264* (2025).
[59] Djamel Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. 2017. DPiSAX: Massively Distributed Partitioned iSAX. In *IEEE International Conference on Data Mining, ICDM*. IEEE Computer Society.
[60] Tiannuo Yang, Wen Hu, Wangqi Peng, Yusen Li, Jianguo Li, Gang Wang, and Xiaoguang Liu. 2024. VDTuner: Automated Performance Tuning for Vector Data Management Systems. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*. IEEE. doi:10.1109/ICDE60146.2024.00332
[61] Chao Zhang and Renée J Miller. 2025. Distribution-Aware Exploration for Adaptive HNSW Search. *arXiv preprint arXiv:2512.06636* (2025).
[62] Zili Zhang, Chao Jin, Linpeng Tang, Xuanzhe Liu, and Xin Jin. 2023. Fast, approximate vector queries on very large unstructured datasets. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*.
[63] Ziyu Zhang, Yuanhao Wei, Joshua Engels, and Julian Shun. 2025. CleANN: Efficient Full Dynamism in Graph-based Approximate Nearest Neighbor Search. *arXiv preprint arXiv:2507.19802* (2025).
[64] Xiangyu Zhi, Baotong Lu, Meng Chen, Hui Li, and Qi Chen. 2025. Towards Efficient and Scalable Distributed Vector Search with RDMA. In *arXiv preprint arXiv:2507.06653*.
[65] Rui Zhu, Bin Wang, Xiaochun Yang, Baihua Zheng, and Guoren Wang. 2017. SAP: improving continuous top-k queries over streaming data. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017).