Research Project Directions for Vector Databases

CS 594: Vector Databases

Fall 2026, Abolfazl Asudeh

Literature checked through September 2026, using GPT

Contents

HNSW, LSH, IVF, product quantization, standard tree indexes, DiskANN, and basic filtered search are course material and should be treated as baselines or components, not as sufficient research projects by themselves. Every bullet below is instead tied to at least two papers that students can reproduce, compare, or extend.

The list prioritizes recent papers at SIGMOD, VLDB, USENIX ATC, NeurIPS, ICML, ICLR, CVPR, ICCV, KDD, WWW, ACL, SOSP, OSDI, FAST, and USENIX Security. A small number of especially timely accepted papers, workshop papers, or preprints are marked as [frontier]. These are useful for finding open problems, but students should anchor evaluation in the accompanying peer-reviewed work. A sound semester project should normally (i) reproduce a strong baseline, (ii) isolate a concrete limitation, and (iii) test one focused extension under a realistic workload.

We use the tags ALGORITHMIC, SYSTEMS, THEORY, ML/IR, HPC, SECURITY, and EMPIRICAL.

1 Advanced Vector Search Algorithms and Query Models

1.1 Advanced ANN Index Structures

Literature assessment: Very active. The promising work is no longer "implement another HNSW," but data-dependent construction, query-dependent search, distribution shift, and compression/search co-design.

- **Hybrid and data-dependent proximity graphs.** Study when a graph should combine sparse long-range links, local geometric structure, and learned or probabilistic routing. CSPG and probabilistic routing provide two concrete, reproducible starting points [1, 2]. Promising extensions include workload-aware edge budgets, update-aware construction, and filtered or SSD-resident variants.

- **Query-adaptive routing and search effort.** Replace one global search-width setting with a per-query choice of entry points, routing policy, or stopping effort. ANNiE predicts the effort needed for a recall target, while probabilistic routing learns how to traverse the graph [2, 3]. A good project would test robustness under workload drift or transfer the policy across indexes.

- **Out-of-distribution and cross-modal query distributions.** Index and query vectors often come from different encoders or modalities. OOD-DiskANN and RoarGraph both redesign graph search for this setting [4, 5]. Useful extensions include drift detection, mixed in-distribution/OOD traffic, and dynamic repair when the query distribution changes.

- **Quantization and distance-pruning co-design.** Go beyond applying standard PQ by exploiting certified quantization error or data-aware lower bounds during candidate pruning. RaBitQ and DADE expose complementary design points [6, 7]. Students could study adaptive bit allocation, non-Euclidean similarity, or integration with graph and filtered search.

Tags: ALGORITHMIC, THEORY, SYSTEMS.

1.2 Filtered and Constrained ANN Search

Literature assessment: Very active, with sustained SIGMOD, PVLDB, WWW, ICML, and NeurIPS activity. The main opportunity is robust behavior across predicates, selectivities, correlations, and updates, not another single-filter benchmark.

- **Joint vector–attribute graph indexes.** NHQ integrates attributes into graph construction, while ACORN develops a predicate-agnostic graph strategy [8, 9]. A promising project can compare structural augmentation against compact auxiliary indexes under multiple attributes, skew, and updates.

- **Range, interval, and window filters.** UNIFY and the ICML work on window-filtered ANN give modern algorithmic and theoretical baselines [10, 11]. Extensions could target temporal windows, several range attributes, or learned layouts that remain stable as ranges drift.

- **Arbitrary Boolean and multi-predicate filters.** ACORN supports broad predicates, while CGIF combines proximity graphs and inverted files for arbitrary predicates [9, 12]. Students can explore conjunction/disjunction planning, compact predicate summaries, or shared traversal across related predicates.

- **Selectivity- and correlation-robust execution.** Filtered-DiskANN demonstrates the systems difficulty of label filtering, and CGIF shows why richer constraints require new index/search coordination [12, 13]. Particularly useful projects would predict when to prefilter, traverse jointly, or postfilter, including adversarially correlated filters.

Tags: Algorithmic, Systems, Theory, Empirical.

1.3 Multi-Vector and Late-Interaction Search

Literature assessment: Very active and growing. ColBERT-style MaxSim retrieval is a distinct indexing and query-processing problem, and it is sufficiently mature to support both systems and theoretical projects.

- **Late-interaction indexing and execution.** ColBERTv2 establishes the retrieval model and compression baseline, while PLAID redesigns the engine for efficient centroid interaction and pruning [14, 15]. Extensions can target filtered retrieval, SSD execution, or mixed document lengths.

- **Candidate generation and token-level pruning.** DESSERT treats the problem as vector-set search, whereas XTR trains token retrieval so that expensive gathering and rescoring can be reduced [16, 17]. Students can study safe bounds, query-adaptive token budgets, or tail-latency behavior.

- **Fixed-size proxies and compressed multi-vector representations.** MUVERA maps multi-vector objects to fixed-dimensional encodings, while ColBERTv2 uses residual compression and denoised supervision [14, 18]. Useful extensions include learned error bounds, asymmetric query/document compression, and update-friendly encodings.

- **Query decomposition and native multi-vector operators.** POQD optimizes query decomposition for multi-vector retrieval; MV-IVF explores clustering multiple vectors as a native index [19, 20]. The latter is a workshop paper[frontier], so an especially good project would turn its idea into a rigorous systems comparison against PLAID, DESSERT, and fixed-dimensional encodings.

Tags: Algorithmic, Systems, Theory, ML/IR.

1.4 Rich Vector Query Models

Literature assessment: Active but heterogeneous. The strongest projects should choose one precise query contract and evaluate it against a realistic workload rather than bundle unrelated semantics.

- **Range/threshold retrieval and large-k search.** Recent work studies graph-based range retrieval and large-k result collection [21, 22]. Both are frontier-stage results[frontier]; a strong project should add a careful comparison with exact range search and standard top-k graph traversal, emphasizing output sensitivity and stopping conditions.

- **Dense–sparse and multi-score retrieval.** HYRR uses hybrid retrieval to train robust rerankers, while PromptReps obtains sparse and dense representations from prompted language models [23, 24]. Database-oriented extensions include score normalization, joint candidate generation, and query-adaptive fusion under latency constraints.

- **Diversity-aware vector retrieval.** Approximate diverse kNN supplies a current database baseline, and VRSD analyzes the combined similarity/diversity problem [25, 26]. VRSD is a preprint[frontier]; projects should therefore emphasize reproducible indexes, approximation quality, and comparisons with reranking-only baselines.

- **Declarative recall and quality requirements.** DARTH and DARTH+ make user-provided recall targets part of execution rather than an offline tuning exercise [27, 28]. DARTH+ is a workshop paper[frontier]. Promising extensions include tail guarantees, filtered or multi-vector recall, and guarantees under workload shift.

Tags: Algorithmic, Theory, Systems, Empirical.

2 Vector Database Query Processing and Data Management

2.1 Vector Similarity Joins and Relational–Vector Operators

Literature assessment: Strong recent momentum. SIGMOD 2025 made approximate vector joins a first-class database problem; disk and offline/online designs now expose several realistic extension points.

- **Approximate range, self, and cross-collection joins.** SimJoin exploits relationships among join windows, while DiskJoin develops a disk-based join algorithm [29, 30]. Useful extensions include asymmetric collections, skew-aware scheduling, and recall-aware duplicate elimination.

- **Top-k and k-similarity joins.** SimJoin includes k-similarity support, and recent offline–online co-design targets fast approximate vector joins [29, 31]. The latter is a preprint[frontier]; a semester project should formalize the query semantics and compare shared-work execution with repeated point ANN.

- **External-memory join processing.** DiskJoin and offline–online co-design offer complementary disk and preprocessing tradeoffs [30, 31]. A promising systems project could study SSD parallelism, partition reuse, or bounded-memory execution.

- **Joins with filters, updates, or distributed inputs.** These combinations remain underexplored, but SimJoin's shared-work view and DiskJoin's partitioned execution are strong

foundations [29, 30]. The research contribution should be a new operator or cost model, not simply invoking ANN once per left-hand row.

Tags: Algorithmic, Systems, Theory.

2.2 Vector Query Optimization and Adaptive Execution

Literature assessment: Active and directly aligned with a database course. Recent papers now address cost estimation, plan integration, automatic tuning, and quality-aware stopping separately, leaving room for a unified optimizer.

- **Cost, selectivity, and candidate-count estimation.** ANNiE estimates graph-search effort for a target recall, while E2E models filtered-search effort and uses it for adaptive termination [3, 32]. Students can test transfer across datasets, indexes, and predicate correlations.

- **Plan, index, and operator selection.** VBase integrates vector search with relational query processing, and a 2026 PostgreSQL study characterizes filter-agnostic execution [33, 34]. Extensions include calibrated cost models, vector-first versus relation-first planning, and mid-query replanning.

- **Per-query adaptive stopping and effort allocation.** ANNiE predicts required work before/during graph search, whereas DARTH learns when enough candidates have been explored [3, 27]. A strong project would compare prediction error, guarantee violations, and overhead on easy versus hard queries.

- **Automatic multi-objective configuration.** VDTuner tunes vector systems across competing objectives, while DARTH exposes accuracy as a declarative execution target [27, 35]. Promising extensions include update/build cost, monetary cost, or tuning that remains stable under workload drift.

Tags: Systems, Algorithmic, ML/IR, Empirical.

2.3 Dynamic and Streaming Vector Search

Literature assessment: Very active. Recent SOSP, SIGMOD, and PVLDB work covers partition-local updates, graph repair, connectivity, and filtered dynamics, but long-running mixed workloads are still poorly understood.

- **Graph insertion, deletion, and repair.** Wolverine explicitly repairs monotonic paths after updates, while CONDA preserves graph connectivity with pruning and lazy deletion [36, 37]. Extensions can target adversarial deletion sequences, bounded repair budgets, or crash consistency.

- **Partition-based incremental maintenance.** SPFresh uses local split/reassign maintenance, while CONDA provides a modern graph-based contrast [37, 38]. Comparing the two under drift can lead to hybrid partition/graph maintenance policies.

- **Streaming, sliding-window, and range-filtered updates.** DIGRA supports dynamic range-filtered ANN, and SPFresh studies continuous billion-scale ingestion and distribution shift [38, 39]. Projects can add time expiration, bursty events, or learned triggers for local rebuilding.

- **Mixed read/write concurrency and freshness guarantees.** SPFresh separates background maintenance from foreground queries; Wolverine quantifies recall degradation caused by updates [36, 38]. A database-systems extension could define snapshot semantics, bound staleness, or control tail latency during repair.

Tags: Systems, Algorithmic, Empirical.

2.4 Embedding Lifecycle and Heterogeneous Embedding Spaces

Literature assessment: Emerging but now credible as a standalone direction. It has a SIGMOD 2026 award paper, an ICML 2026 paper, established backward-compatibility work, and multiple OOD-search systems.

- **Cross-model integration and vector linking.** Integrating Vector Databases across Embedding Models studies database integration, while Vector Linking recovers correspondences using local geometric consistency [40, 41]. Students can target noisy anchors, partial overlap, or index-aware matching.

- **Backward-compatible embedding upgrades.** Learning Backward Compatible Embeddings and its earlier CVPR formulation provide strong learning baselines [42, 43]. A vector-database project could measure how compatibility objectives affect ANN topology, filtered retrieval, and compression.

- **Partial migration and simultaneous model versions.** Cross-model database integration and backward-compatible representation learning address complementary ends of migration [40, 42]. Promising extensions include online migration schedules, mixed-version query planning, and avoiding a full rebuild.

- **Distribution mismatch between queries and stored vectors.** OOD-DiskANN is an established WWW baseline; a cross-distribution monotonic graph has been accepted at SIGMOD 2026[frontier] [4, 44]. Students can study continuously changing query distributions or multiple concurrent query encoders.

Tags: Algorithmic, ML/IR, Systems, Theory.

3 Scalable Vector Database Systems

3.1 Storage-Aware and Out-of-Core Vector Search

Literature assessment: Very active and central to the 2026 course/tutorial material. Modern SSD parallelism, layout, asynchronous I/O, and storage/compute overlap now determine the algorithm.

- **SSD-resident graph search.** DiskANN is the essential systems baseline, while Starling substantially revisits disk-resident graph organization and search [45, 46]. Extensions should focus on new storage behavior, updates, or query semantics rather than reimplementing DiskANN.

- **I/O-efficient traversal and locality-aware layout.** Starling reorganizes graph access for disk, and modern-SSD work inside pgvector combines parallel I/O, insertion reordering, and colocation [46, 47]. Students can target workload-aware relayout, write amplification, or tail I/O.

- **Memory/SSD tiering and cache admission.** SPANN separates an in-memory routing layer from SSD posting lists, while the modern-SSD study exposes cache and locality bottlenecks [47, 48]. A promising project is adaptive placement under changing query heat and memory pressure.

- **Asynchronous I/O and storage–compute overlap.** FlashANNS, accepted at SIGMOD 2026[frontier], pipelines GPU-driven I/O; the PVLDB SSD study exploits io_uring parallelism [47, 49]. Extensions include admission control, cancellation after early stopping, and shared queues for batched queries.

Tags: Systems, Algorithmic, Empirical.

3.2 GPU and Hardware-Accelerated Vector Search

Literature assessment: Very active, with ICDE, SIGMOD, PVLDB, FAST, USENIX ATC, and ASPLOS results. Hardware-aware algorithms, not direct ports of CPU HNSW, are the appropriate project level.

- **GPU-native graph traversal.** CAGRA establishes a high-performance single-GPU graph design, while Jasper supports mutable GPU graph search [50, 51]. Projects can study divergence, query batching, or tail latency for mixed query difficulty.

- **GPU index construction and updates.** CAGRA includes GPU-oriented construction, and scalable GPU graph indexing pushes construction to larger datasets [50, 52]. Extensions include incremental construction, reproducibility across GPU counts, and memory-bounded building.

- **GPU-native quantization and distance computation.** GPU IVF-RaBitQ co-designs build and search kernels; Jasper provides a mutable graph alternative [51, 53]. Students can test mixed precision, query-adaptive bit widths, or fused reranking.

- **CPU–GPU–SSD cooperative search.** FusionANNS explicitly divides filtering and reranking across tiers, while PathWeaver coordinates search across multiple GPUs [54, 55]. A strong project could optimize placement and scheduling under variable load and PCIe/NVLink contention.

- **Filtered search and hardware-aware memory layout.** VecFlow accelerates vector search with structured filtering, and PathWeaver reduces redundant multi-GPU graph work [55, 56]. Extensions can co-design predicate layouts, graph shards, and communication.

- **Specialized acceleration.** JUNO maps sparsity-aware ANNS to GPU ray-tracing cores, whereas FusionANNS shows what can be achieved with commodity heterogeneous hardware [54, 57]. An appropriate project should include an implementable simulator, FPGA prototype, or careful roofline/cost model.

Tags: HPC, Systems, Algorithmic, Empirical.

3.3 Distributed, Cloud, and Disaggregated Vector Search

Literature assessment: Active. Recent PVLDB/PACMMOD papers and SIGMOD 2026 acceptances cover distributed graph search, RDMA, stateless/serverless blocks, and scale-to-zero operation.

- **Sharding and query routing.** RED-ANNS co-designs distributed graph placement and routing; Manu provides a mature cloud-native vector-database architecture [58, 59]. Extensions can learn routing under skew or quantify recall lost by visiting only a shard subset.

- **Distributed graph search and RDMA.** RED-ANNS is a peer-reviewed baseline, while CoTra is accepted at SIGMOD 2026[frontier]and focuses on RDMA-efficient distributed search [58, 60]. Students can study consistency, communication/computation overlap, or dynamic repartitioning.

- **Locality, load balance, and communication efficiency.** RED-ANNS exposes the distributed graph tradeoff, while BatANN explores a batched distributed design at VecDB 2026[frontier][58, 61]. BatANN should be treated as a hypothesis to validate against strong routing and all-shard baselines.

- **Disaggregated, elastic, and serverless execution.** Vexless studies serverless scale-to-zero vector search, and stateless serverless blocks provide a newer PACMMOD design [62, 63]. Promising projects include cold-start mitigation, multi-tenancy, and cost/recall-aware elasticity.

Tags: Systems, Algorithmic, Empirical.

3.4 Scalable Index Construction, Ingestion, and Data Layout

Literature assessment: Very active. Construction is no longer a one-time preprocessing detail at billion-vector scale; parallel build, GPU build, ingestion, and index merging now have dedicated top-tier papers.

- **Faster proximity-graph construction.** ParlayANN develops deterministic parallel construction, while FastKCNA revisits the graph-construction bottleneck in PVLDB [64, 65]. Extensions can target filtered/OOD graphs, deterministic quality, or update-aware build objectives.

- **GPU-native construction.** CAGRA and scalable GPU graph indexing offer complementary construction pipelines [50, 52]. Students can compare build throughput, temporary memory, graph quality, and downstream query behavior rather than only kernel speed.

- **Parallel and memory-bounded billion-scale building.** ParlayANN emphasizes parallel deterministic algorithms, while scalable GPU indexing targets large-scale hardware utilization [52, 64]. Useful extensions include external-memory construction and fault-tolerant distributed builds.

- **Bulk ingestion and online construction.** SPFresh is the systems baseline for incremental in-place ingestion; Direct Path Optimization is a VecDB 2026 workshop proposal[frontier][38, 66]. A strong project would reproduce the ingestion claim and measure foreground-query interference.

- **Merging independently built indexes.** Two accepted SIGMOD 2026 papers independently study general vector-index merging and graph-index merging[frontier][67, 68]. This is unusually promising for students: compare merge quality with rebuild, then add deletion, filtering, or heterogeneous shard distributions.

Tags: Systems, Algorithmic, HPC, Empirical.

4 Retrieval Systems, Data Modalities, and Foundations

4.1 Modality- and Structure-Aware Vector Retrieval

Literature assessment: Active when the modality changes the representation, score, or index. Merely loading a new dataset into an existing vector database is not a research project.

- **Long-document retrieval and representation granularity.** LongEmbed benchmarks and extends long-context embeddings; Dense X Retrieval studies proposition-level retrieval units [69, 70]. Projects can co-design segmentation, index size, and retrieval latency for variable-length documents.

- **Multimodal documents and cross-modal robustness.** ReT targets multimodal document retrieval, while L2RM learns robust language representations for vision-language retrieval [71, 72]. Database extensions include modality-aware routing, missing modalities, and compressed joint indexes.

- **Temporal and composed video retrieval.** Composed Video Retrieval and its dense-modification successor show how text, reference images, and temporal evidence interact [73, 74]. Students can study segment-level indexes, incremental video ingestion, or efficient late interaction across frames.

- **Graph- and knowledge-structure-aware retrieval.** G-Retriever retrieves graph evidence for generation, and HippoRAG uses a knowledge graph plus personalized PageRank for long-term memory [75, 76]. Extensions should measure the systems tradeoff between graph traversal, vector search, and context quality.

Tags: Algorithmic, ML/IR, Systems, Empirical.

4.2 RAG, Agentic Retrieval, and Retrieval-System Co-Design

Literature assessment: Extremely active, but easy to scope poorly. "Build a RAG app" is not sufficient; each project should isolate a retrieval, scheduling, freshness, or end-to-end systems hypothesis.

- **Retrieval granularity and hierarchical chunking.** Dense X Retrieval advocates proposition-level units, while RAPTOR constructs a hierarchy of chunks and summaries [70, 77]. Students can jointly optimize answer quality, index footprint, update cost, and retrieval latency.

- **Retriever–reranker and sparse–dense co-design.** RankRAG trains one model for ranking and generation, while HYRR trains a reranker robust to different sparse/dense candidate generators [23, 78]. Promising projects include budget-aware candidate allocation and shared representations between stages.

- **Adaptive retrieval timing, query formation, and depth.** Adaptive-RAG routes questions by predicted complexity; DRAGIN decides when and what to retrieve during generation [79, 80]. Extensions can incorporate vector search cost, uncertainty calibration, and retrieval cancellation.

- **Iterative and graph-guided multi-step retrieval.** IRCoT interleaves retrieval with chain-of-thought, while HippoRAG retrieves over an induced knowledge graph [76, 81]. Students can

cache repeated subqueries, batch frontier searches, or compare graph and multi-vector state representations.

- **Freshness and streaming RAG.** FreshLLMs supplies a benchmark and search-augmentation approach for changing knowledge; SPFresh supplies a vector-index maintenance system [38, 82]. A strong project can connect index freshness and answer freshness, measuring staleness end to end.

- **Caching and end-to-end retrieval/inference pipelines.** RAGCache caches retrieved-knowledge states and overlaps retrieval with generation; PipeRAG pipelines the two stages [83, 84]. Extensions include cache admission by semantic reuse, speculative cancellation, and multi-tenant fairness.

Tags: Systems, ML/IR, Algorithmic, Empirical.

4.3 Theory and Formal Guarantees for Vector Search

Literature assessment: Active again, especially around navigable graphs, constrained ANN, multi-vector expressiveness, and per-query quality. Projects here need a precise model and theorem, ideally paired with experiments on practical indexes.

- **Navigability, sparsity, and greedy-search guarantees.** Navigable Graphs for High-Dimensional Nearest Neighbor Search gives modern theory, while Almost Navigable Graphs sharpens the question in a 2026 workshop paper[frontier] [85, 86]. Students can test which assumptions hold on learned embeddings.

- **Worst-case and data-dependent complexity of practical indexes.** Indyk and Xu analyze guarantees and limitations of popular implementations; the NeurIPS navigability work supplies a complementary constructive view [85, 87]. Projects can derive bounds for hybrid, dynamic, or OOD graphs.

- **Guarantees for filtered ANN.** The ICML window-filtered paper provides theoretical algorithms, while NHQ gives a high-performing constrained-search system [8, 11]. A valuable project would bridge the model gap and quantify which guarantees survive realistic correlations.

- **Multi-vector approximation and expressiveness.** MUVERA proves guarantees for fixed-dimensional encodings; recent work argues that multi-vector embeddings are strictly more expressive[frontier] [18, 88]. Extensions include lower bounds for MaxSim compression or variable-size sets.

- **Per-query recall guarantees.** DARTH introduces declarative early termination, while DARTH+ adds declarative recall and quality guarantees[frontier] [27, 28]. Projects can study distribution-free calibration, filtered queries, or guarantees under drift.

Tags: Theory, Algorithmic, Empirical.

4.4 Private, Secure, Verifiable, and Robust Vector Search

Literature assessment: Active but specialized. It is a strong option for students with cryptography or security preparation; access control and poisoning also admit database-systems projects.

- **Private or encrypted semantic search.** PANTHER provides practical private high-dimensional search, and BiSON targets billion-scale two-server private similarity search [89, 90]. BiSON is recent frontier-stage work[frontier]; projects should state the leakage and threat model explicitly.

- **Embedding inversion and information leakage.** Text Embeddings Reveal (Almost) As Much As Text and Transferable Embedding Inversion Attacks show complementary reconstruction threats [91, 92]. Extensions can test modern retrieval embeddings, quantized vectors, or utility-preserving defenses.

- **Verifiable outsourced vector search.** V3DB develops verifiable vector queries in PVLDB, while earlier work verifies graph-based ANN search [93, 94]. Promising projects include update proofs, filtered queries, and proof-size/recall tradeoffs.

- **Access control and policy-constrained ANN.** HoneyBee targets role-based access control, while CGIF provides an arbitrary-predicate substrate that can express richer policy constraints [12, 95]. Students can study policy changes, many-user sharing, and side channels caused by traversal or timing.

- **Poisoning and adversarial retrieval.** PoisonedRAG attacks the RAG knowledge base, while AgentPoison targets agent memory and retrieval [96, 97]. Good extensions include index-level detection, certified retrieval robustness, and defenses evaluated against adaptive attackers.

Tags: Security, Algorithmic, Theory, Systems.

References

[1] Ming Yang, Yuzheng Cai, and Weiguo Zheng. CSPG: Crossing sparse proximity graphs for approximate nearest neighbor search. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[2] Kejing Lu, Chuan Xiao, and Yoshiharu Ishikawa. Probabilistic routing for graph-based approximate nearest neighbor search. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 33177–33195, 2024.

[3] Zeyu Wang, Manos Chatzakis, Qitong Wang, Themis Palpanas, Peng Wang, and Wei Wang. ANNiE: A learned query cost estimator for graph-based approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 19(11):3820–3833, 2026.

[4] Shikhar Jaiswal, Ravishankar Krishnaswamy, Ankit Garg, Harsha Vardhan Simhadri, and Sheshansh Agrawal. OOD-DiskANN: Efficient and scalable graph ANNS for out-of-distribution queries. In *Proceedings of the ACM Web Conference 2023*, 2023.

[5] Meng Chen, Kai Zhang, Zhenying He, Yinan Jing, and X. Sean Wang. RoarGraph: A projected bipartite graph for efficient cross-modal approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 17(11):2735–2749, 2024.

[6] Jianyang Gao and Cheng Long. 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. SIGMOD 2024.

[7] Liwei Deng, Penghao Chen, Ximu Zeng, Tianfu Wang, Yan Zhao, and Kai Zheng. Efficient data-aware distance comparison operations for high-dimensional approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 18(3):812–821, 2024.

[8] Mengzhao Wang, Lingwei Lv, Xiaoliang Xu, Yuxiang Wang, Qiang Yue, and Jiongkang Ni. An efficient and robust framework for approximate nearest neighbor search with attribute constraint. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[9] Liana Patel, Peter Kraft, Carlos Guestrin, and Matei Zaharia. ACORN: Performant and predicate-agnostic search over vector embeddings and structured data. *Proceedings of the ACM on Management of Data*, 2(3), 2024. SIGMOD 2024.

[10] Anqi Liang, Pengcheng Zhang, Bin Yao, Zhongpu Chen, Yitong Song, and Guangxu Cheng. UNIFY: Unified index for range filtered approximate nearest neighbors search. *Proceedings of the VLDB Endowment*, 18(4):1118–1130, 2024.

[11] Joshua Engels, Ben Landrum, Shangdi Yu, Laxman Dhulipala, and Julian Shun. Approximate nearest neighbor search with window filters. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 12469–12490, 2024.

[12] Jiarui Luo, Chaoji Zuo, and Dong Deng. CGIF: Combining proximity graphs and inverted files for efficient filtered vector search over arbitrary predicates. *Proceedings of the VLDB Endowment*, 19(11):3663–3675, 2026.

[13] Siddharth Gollapudi, Neel Karia, Varun Sivashankar, Ravishankar Krishnaswamy, Nikit Begwani, Swapnil Raz, Yiyong Lin, Yin Zhang, Neelam Mahapatro, Premukumar Srinivasan, Amit Singh, and Harsha Vardhan Simhadri. Filtered-DiskANN: Graph algorithms for approximate nearest neighbor search with filters. In *Proceedings of the ACM Web Conference 2023*, 2023.

[14] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 3715–3734, 2022.

[15] Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. PLAID: An efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management*, 2022.

[16] Joshua Engels, Benjamin Coleman, Vihan Lakshman, and Anshumali Shrivastava. DESSERT: An efficient algorithm for vector set search with vector set queries. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[17] Jinhyuk Lee, Zhuyun Dai, Sai Meher Karthik Duddu, Tao Lei, Iftekhar Naim, Ming-Wei Chang, and Vincent Y. Zhao. Rethinking the role of token retrieval in multi-vector retrieval. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[18] Laxman Dhulipala, Majid Hadian, Rajesh Jayaram, Jason Lee, and Vahab Mirrokni. MUVERA: Multi-vector retrieval via fixed dimensional encodings. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[19] Yaoyang Liu, Junlin Li, Yinjun Wu, and Zhen Chen. POQD: Performance-oriented query decomposer for multi-vector retrieval. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 38789–38808, 2025.

[20] Kishen N. Gowda, Laxman Dhulipala, Lars Gottesbueren, and Rajesh Jayaram. MV-IVF: Multi-vector retrieval via multi-vector clustering. In *Fourth Workshop on Vector Data Management (VecDB at VLDB)*, 2026. Workshop paper.

[21] Magdalen Dobson Manohar, Taekseung Kim, and Guy E. Blelloch. Range retrieval with graph-based indices. In *Fourth Workshop on Vector Data Management (VecDB at VLDB)*, 2026. Workshop paper.

[22] Ziqi Yin, Gao Cong, Kai Zeng, Jinwei Zhu, and Bin Cui. BBC: Improving large-k approximate nearest neighbor search with a bucket-based result collector, 2026.

[23] Jing Lu, Keith Hall, Ji Ma, and Jianmo Ni. HYRR: Hybrid infused reranking for passage retrieval. In *Proceedings of LREC-COLING 2024*, pages 8528–8534, 2024.

[24] Shengyao Zhuang, Xueguang Ma, Bevan Koopman, Jimmy Lin, and Guido Zuccon. PromptReps: Prompting large language models to generate dense and sparse representations for zero-shot document retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4375–4391, 2024.

[25] Jiachen Zhao, Xiao Yan, and Eric Lo. Approximate diverse k-nearest neighbor search in vector database. In *42nd IEEE International Conference on Data Engineering*, pages 2971–2983, 2026.

[26] Hang Gao, Dong Deng, and Yongfeng Zhang. Vector retrieval with similarity and diversity: How hard is it?, 2026.

[27] Manos Chatzakis, Yannis Papakonstantinou, and Themis Palpanas. DARTH: Declarative recall through early termination for approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 3(4), 2025.

[28] Manos Chatzakis, Yannis Papakonstantinou, and Themis Palpanas. DARTH+: Vector search with declarative recall and guarantees, 2026. Fourth Workshop on Vector Data Management (VecDB at VLDB), workshop paper.

[29] Jiadong Xie, Jeffrey Xu Yu, and Yingfan Liu. Fast approximate similarity join in vector databases. *Proceedings of the ACM on Management of Data*, 3(3), 2025.

[30] Yanqi Chen, Xiao Yan, Alexandra Meliou, and Eric Lo. DiskJoin: Large-scale vector similarity join with SSD. *Proceedings of the ACM on Management of Data*, 3(6), 2025.

[31] Kyoungmin Kim, Lennart Roth, Liang Liang, and Anastasia Ailamaki. Fast approximate vector joins via offline-online co-design, 2026.

[32] Wenxuan Xia, Mingyu Yang, Wentao Li, and Wei Wang. E2E: Efficient filtered AKNN search via adaptive termination. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2026.

[33] Qianxi Zhang, Shuotao Xu, Qi Chen, Guoxin Sui, Jiadong Xie, Zhizhen Cai, Yaoqi Chen, Yinxuan He, Yuqing Yang, Fan Yang, Mao Yang, and Lidong Zhou. VBase: Unifying online vector similarity search and relational queries via relaxed monotonicity. In *17th USENIX Symposium on Operating Systems Design and Implementation*, pages 377–395, 2023.

[34] Duo Lu, Helena Caminal, Manos Chatzakis, Yannis Papakonstantinou, Yannis Chronis, Vaibhav Jain, and Fatma Özcan. An in-depth study of filter-agnostic vector search on a postgresql database system. *Proceedings of the ACM on Management of Data*, 4(3), 2026. SIGMOD 2026.

[35] Tiannuo Yang, Wen Hu, Wangqi Peng, Yusen Li, Jianguo Li, Gang Wang, and Xiaoguang Liu. VDTuner: Automated performance tuning for vector data management systems. In *40th IEEE International Conference on Data Engineering*, 2024.

[36] Dawei Liu, Bolong Zheng, Ziyang Yue, Fuhao Ruan, Xiaofang Zhou, and Christian S. Jensen. Wolverine: Highly efficient monotonic search path repair for graph-based ANN index updates. *Proceedings of the VLDB Endowment*, 18(7):2268–2280, 2025.

[37] Darae Lee and Min-Soo Kim. CONDA: A connectivity-aware dynamic index for approximate nearest neighbor search over evolving data. *Proceedings of the VLDB Endowment*, 19(11):3357–3370, 2026.

[38] 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. SPFresh: Incremental in-place update for billion-scale vector search. In *Proceedings of the 29th ACM Symposium on Operating Systems Principles*, 2023.

[39] Mengxu Jiang, Zhi Yang, Fangyuan Zhang, Guanhao Hou, Jieming Shi, Wenchao Zhou, Feifei Li, and Sibo Wang. DIGRA: A dynamic graph indexing for approximate nearest neighbor search with range filter. *Proceedings of the ACM on Management of Data*, 3(3), 2025.

[40] Beining Yang, Yang Cao, and Yang Ren. Integrating vector databases across embedding models. *Proceedings of the ACM on Management of Data*, 3(6), 2025. SIGMOD 2026 Best Paper Honorable Mention.

[41] Ziying Chen, Yang Cao, He Sun, Beining Yang, and Tianjian Yang. Vector linking via cross-model local isometric consistency. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. Accepted at ICML 2026.

[42] Wenbin Hu, Rahul Bansal, Kaidi Cao, Nikhil Rao, Karthik Subbian, and Jure Leskovec. Learning backward compatible embeddings. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

[43] Yantao Shen, Yuanjun Xiong, Wei Xia, and Stefano Soatto. Towards backward-compatible representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[44] Qiang Yue, Mengzhao Wang, Xiaoliang Xu, Cheng Long, Yuxiang Wang, and Jiahui Wang. Efficient and robust out-of-distribution vector similarity search with cross-distribution monotonic graph. In *Proceedings of the 2026 ACM SIGMOD International Conference on Management of Data*, 2026. Accepted paper.

[45] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnaswamy, and Rohan Kadekodi. DiskANN: Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[46] Mengzhao Wang, Weizhi Xu, Xiaomeng Yi, Songlin Wu, Zhangyang Peng, Xiangyu Ke, Yunjun Gao, Xiaoliang Xu, Rentong Guo, and Charles Xie. Starling: An I/O-efficient disk-resident graph index framework for high-dimensional vector similarity search on data segment. *Proceedings of the ACM on Management of Data*, 2(1), 2024.

[47] Joobo Shim, Jaewon Oh, Hongchan Roh, Jaeyoung Do, and Sang-Won Lee. Turbocharging vector databases using modern SSDs. *Proceedings of the VLDB Endowment*, 18(11):4710–4722, 2025.

[48] Qi Chen, Bing Zhao, Haidong Wang, Mingqin Li, Chuanjie Liu, Zengzhong Li, Mao Yang, and Jingdong Wang. SPANN: Highly-efficient billion-scale approximate nearest neighbor search. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

[49] Yang Xiao, Mo Sun, Ziyu Song, Bing Tian, Jie Sun, Jie Zhang, Zeke Wang, Zonghui Wang, Wenzhi Chen, and Fei Wu. FlashANNS: Gpu-driven asynchronous I/O pipelining for eliminating storage-compute bottlenecks in billion-scale similarity search. In *Proceedings of the 2026 ACM SIGMOD International Conference on Management of Data*, 2026. Accepted paper.

[50] Hiroyuki Ootomo, Akira Naruse, Corey Nolet, Ray Wang, Tamás Fehér, and Yong Wang. CAGRA: Highly parallel graph construction and approximate nearest neighbor search for GPUs. In *40th IEEE International Conference on Data Engineering*, pages 4236–4247, 2024.

[51] Hunter McCoy, Zikun Wang, and Prashant Pandey. GPU-accelerated ANNS: Quantized for speed, built for change. *Proceedings of the VLDB Endowment*, 19(11):3413–3426, 2026.

[52] Zhonggen Li, Xiangyu Ke, Yifan Zhu, Bocheng Yu, Baihua Zheng, and Yunjun Gao. Scalable graph indexing using GPUs for approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 3(6), 2025. SIGMOD 2026.

[53] Jifan Shi, Jianyang Gao, James Xia, Tamás Béla Fehér, and Cheng Long. GPU-native approximate nearest neighbor search with IVF-RaBitQ: Fast index build and search. *Proceedings of the VLDB Endowment*, 19(11):3454–3467, 2026.

[54] Bing Tian, Haikun Liu, Yuhang Tang, Shihai Xiao, Zhuohui Duan, Xiaofei Liao, Hai Jin, Xuecang Zhang, Junhua Zhu, and Yu Zhang. Towards high-throughput and low-latency billion-scale vector search via CPU/GPU collaborative filtering and re-ranking. In *23rd USENIX Conference on File and Storage Technologies*, 2025.

[55] Sukjin Kim, Seongyeon Park, Si Ung Noh, Junguk Hong, Taehee Kwon, Hunseong Lim, and Jinho Lee. PathWeaver: A high-throughput multi-GPU system for graph-based approximate nearest neighbor search. In *2025 USENIX Annual Technical Conference*, 2025.

[56] Jingyi Xi, Chenghao Mo, Benjamin Karsin, Artem Chirkin, Mingqin Li, and Minjia Zhang. VecFlow: A high-performance vector data management system for filtered-search on GPUs. *Proceedings of the ACM on Management of Data*, 3(4), 2025.

[57] Zihan Liu, Wentao Ni, Jingwen Leng, Yu Feng, Cong Guo, Quan Chen, Chao Li, Minyi Guo, and Yuhao Zhu. Juno: Optimizing high-dimensional approximate nearest neighbour search

with sparsity-aware algorithm and ray-tracing core mapping. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 549–565, 2024.

[58] Yue Chen, Kai Zhang, Sipeng Chen, Shihai Xiao, Xiaomin Zou, Ren Ren, Yinan Jing, X. Sean Wang, Li Cao, and Mingxiang Wan. RED-ANNS: An RDMA-enabled distributed framework for graph-based approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 19(3):399–412, 2025.

[59] Rentong Guo, Xiaofan Luan, Long Xiang, Xiao Yan, Xiaomeng Yi, Jigao Luo, Qianya Cheng, Weizhi Xu, Jiarui Luo, Frank Liu, Zhenshan Cao, Yanliang Qiao, Ting Wang, Bo Tang, and Charles Xie. Manu: A cloud native vector database management system. *Proceedings of the VLDB Endowment*, 15(12):3548–3561, 2022.

[60] Xiangyu Zhi, Meng Chen, Xiao Yan, Baotong Lu, Hui Li, Qianxi Zhang, Qi Chen, and James Cheng. CoTra: Towards efficient and scalable distributed vector search with RDMA. In *Proceedings of the 2026 ACM SIGMOD International Conference on Management of Data*, 2026. Accepted paper.

[61] Nam Anh Dang, Ben Landrum, and Ken Birman. Passing the baton: High throughput distributed disk-based vector search with BatANN. In *Fourth Workshop on Vector Data Management (VecDB at VLDB)*, 2026. Workshop paper.

[62] Yongye Su, Yinqi Sun, Minjia Zhang, and Jianguo Wang. Vexless: A serverless vector data management system using cloud functions. *Proceedings of the ACM on Management of Data*, 2(3), 2024.

[63] Daniel Barcelona-Pons, Raúl Gracia-Tinedo, Albert Cañadilla-Domingo, Xavier Roca-Canals, and Pedro García-López. Building stateless serverless vector DBs via block-based data partitioning. *Proceedings of the ACM on Management of Data*, 3(6):1–25, 2025.

[64] Magdalen Dobson Manohar, Zheqi Shen, Guy E. Blelloch, Laxman Dhulipala, Yan Gu, Harsha Vardhan Simhadri, and Yihan Sun. ParlayANN: Scalable and deterministic parallel graph-based approximate nearest neighbor search algorithms. In *Proceedings of the 29th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming*, 2024.

[65] Shuo Yang, Jiadong Xie, Yingfan Liu, Jeffrey Xu Yu, Xiyue Gao, Qianru Wang, Yanguo Peng, and Jiangtao Cui. Revisiting the index construction of proximity graph-based approximate nearest neighbor search. *Proceedings of the VLDB Endowment*, 18(6):1825–1838, 2025.

[66] Tianji Yang and Xuhao Chen. Direct path optimization for faster ingestion and sparser graph index in vector database. In *Fourth Workshop on Vector Data Management (VecDB at VLDB)*, 2026. Workshop paper.

[67] Chenzhe Jin, Yunan Zhang, Jiayi Liu, and Jianguo Wang. Efficient vector index merging in vector databases. In *Proceedings of the 2026 ACM SIGMOD International Conference on Management of Data*, 2026. Accepted paper.

[68] Zekai Wu, Jiabao Jin, Peng Cheng, Xiaoyao Zhong, Lei Chen, Yongxin Tong, Zhitao Shen, Jingkuan Song, Heng Tao Shen, and Xuemin Lin. FGIM: A fast graph-based indexes merging framework for approximate nearest neighbor search. In *Proceedings of the 2026 ACM SIGMOD International Conference on Management of Data*, 2026. Accepted paper.

[69] Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. LongEmbed: Extending embedding models for long context retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 802–816, 2024.

[70] Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. Dense X retrieval: What retrieval granularity should we use? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, 2024.

[71] Davide Caffagni, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Recurrence-enhanced vision-and-language transformers for robust multimodal document retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9286–9295, 2025.

[72] Haochen Han, Qinghua Zheng, Guang Dai, Minnan Luo, and Jingdong Wang. Learning to rematch mismatched pairs for robust cross-modal retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26679–26688, 2024.

[73] Omkar Thawakar, Muzammal Naseer, Rao Muhammad Anwer, Salman Khan, Michael Felsberg, Mubarak Shah, and Fahad Shahbaz Khan. Composed video retrieval via enriched context and discriminative embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26896–26906, 2024.

[74] Omkar Thawakar, Dmitry Demidov, Ritesh Thawkar, Rao Muhammad Anwer, Mubarak Shah, Fahad Shahbaz Khan, and Salman Khan. Beyond simple edits: Composed video retrieval with dense modifications. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20435–20444, 2025.

[75] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-augmented generation for textual graph understanding and question answering. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[76] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. HippoRAG: Neurobiologically inspired long-term memory for large language models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[77] Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *International Conference on Learning Representations*, 2024.

[78] Yue Yu, Wei Ping, Zihan Liu, Boxin Wang, Jiaxuan You, Chao Zhang, Mohammad Shoeybi, and Bryan Catanzaro. RankRAG: Unifying context ranking with retrieval-augmented generation in LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[79] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 7036–7050, 2024.

[80] Weihang Su, Yichen Tang, Qingyao Ai, Zhijing Wu, and Yiqun Liu. DRAGIN: Dynamic retrieval augmented generation based on the real-time information needs of large language

models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 12991–13013, 2024.

[81] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 10014–10037, 2023.

[82] Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc V. Le, and Thang Luong. FreshLLMs: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13697–13720, 2024.

[83] Chao Jin, Zili Zhang, Xuanlin Jiang, Fangyue Liu, Shufan Liu, Xuanzhe Liu, and Xin Jin. RAGCache: Efficient knowledge caching for retrieval-augmented generation. *ACM Transactions on Computer Systems*, 44(1), 2025.

[84] Wenqi Jiang, Shuai Zhang, Boran Han, Jie Wang, Bernie Wang, and Tim Kraska. PipeRAG: Fast retrieval-augmented generation via algorithm-system co-design. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 589–600, 2025.

[85] Haya Diwan, Jinrui Gou, Cameron Musco, Christopher Musco, and Torsten Suel. Navigable graphs for high-dimensional nearest neighbor search: Constructions and limits. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[86] Pratyush Avi and Christopher Musco. Almost navigable graphs. In *Fourth Workshop on Vector Data Management (VecDB at VLDB)*, 2026. Workshop paper.

[87] Piotr Indyk and Haike Xu. Worst-case performance of popular approximate nearest neighbor search implementations: Guarantees and limitations. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[88] Rajesh Jayaram. Multi-vector embeddings are provably more expressive than single vector embeddings. In *Fourth Workshop on Vector Data Management (VecDB at VLDB)*, 2026. Workshop paper.

[89] Jingyu Li, Zhicong Huang, Min Zhang, Cheng Hong, Jian Liu, Tao Wei, and Wenguang Chen. PANTHER: Private approximate nearest neighbor search in the single server setting. In *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, 2025.

[90] Sankha Das, Rohan Ravi, Nishanth Chandran, and Divya Gupta. BiSON: Billion-scale oblivious nearest-neighbor search in milliseconds. In *2026 IEEE European Symposium on Security and Privacy*, 2026. Accepted paper; IACR ePrint 2026/1343.

[91] John X. Morris, Volodymyr Kuleshov, Vitaly Shmatikov, and Alexander M. Rush. Text embeddings reveal (almost) as much as text. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12448–12460, 2023.

[92] Yu-Hsiang Huang, Yuche Tsai, Hsiang Hsiao, Hong-Yi Lin, and Shou-De Lin. Transferable embedding inversion attack: Uncovering privacy risks in text embeddings without model queries. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 4193–4205, 2024.

[93] Zipeng Qiu, Wenjie Qu, Jiaheng Zhang, and Binhang Yuan. V3DB: Audit-on-demand zero-knowledge proofs for verifiable vector search over committed snapshots. *Proceedings of the VLDB Endowment*, 19(10):2604–2616, 2026.

[94] Chenzhao Wang, Jilian Zhang, Xuyang Liu, Kaimin Wei, and Bingwen Feng. Verifiable graph-based approximate nearest neighbor search. In *Advanced Data Mining and Applications*, pages 3–17, 2024.

[95] Hongbin Zhong, Matthew Lentz, Nina Narodytska, Adriana Szekeres, and Kexin Rong. HONEYBEE: Efficient role-based access control for vector databases via dynamic partitioning. In *Proceedings of the 2026 ACM SIGMOD International Conference on Management of Data*, 2026. Accepted paper.

[96] Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. PoisonedRAG: Knowledge corruption attacks to retrieval-augmented generation of large language models. In *34th USENIX Security Symposium*, pages 3827–3844, 2025.

[97] Zhaorun Chen, Zhen Xiang, Chaowei Xiao, Dawn Song, and Bo Li. AgentPoison: Red-teaming LLM agents via poisoning memory or knowledge bases. In *Advances in Neural Information Processing Systems*, volume 37, 2024.