Vol. 1, No. 1, pp.: 45-56, 2023

Vector Database Optimization for E-Commerce Search Logs: A Data-Driven Study of Latency, Recall, and Revenue Signals

Peng Liu[1], Jing Wang[2], Hao Xu[3,*]

[1] School of Information Engineering, Zhongnan University of Economics and Law, Wuhan 430073, China

[2] Department of Computer Science and Technology, Jiangxi University of Finance and Economics, Nanchang 330013, China

[3] College of Artificial Intelligence, Shenyang University of Technology, Shenyang 110870, China

* *haoxu@sut.edu.cn*

Article Information	
Received	18 October 2022
Accepted	29 February 2023
DOI	https://doi.org/10.63646/datamind.2023.010105

Abstract

Approximate nearest-neighbor (ANN) retrieval over dense vector embeddings has become the standard architectural pattern for semantic product search in large-scale e-commerce platforms. Yet a systematic, data-driven comparison of how different vector index strategies interact with real search logs—affecting retrieval quality, serving latency, and downstream commercial signals—remains absent from the literature. This paper introduces an open, reproducible database of 4.2 million search-session records drawn from a mid-sized Chinese e-commerce platform and uses it to benchmark five vector index configurations: exact flat search, two IVF-PQ variants, and two HNSW configurations. The database captures query embeddings, item embeddings, ranked result lists, click events, cart additions, purchase outcomes, revenue, and per-query serving latency. We design a controlled A/B simulation that layers bi-encoder query encoding, HNSW indexing, learning-to-rank (LTR) re-ranking, and position-bias correction, measuring Recall@K, p50/p99 latency, click-through rate, conversion rate, and average revenue per session at each step. The full system achieves Recall@10 = 0.979 and p50 latency = 6.2 ms, corresponding to a 22.1% uplift in revenue per session compared with the BM25-sorted IVF-PQ baseline. Ablation experiments identify the re-ranker and query embedding steps as the two largest individual contributors to revenue uplift. The database schema, data pipeline, and reproducibility protocols are described in full, and the anonymised dataset is released under a CC-BY 4.0 licence.

Keywords: ***vector database; approximate nearest neighbor search; FAISS; HNSW; e-commerce search;***

learning to rank; click-through rate; revenue signal; reproducible research

1. Introduction

Product search is the primary revenue-generating function of e-commerce platforms. On a typical mid-sized platform serving tens of millions of monthly active users, the search bar mediates roughly 40–60% of purchase journeys. Every millisecond of additional query latency and every position of rank degradation translates measurably into lost click-through and conversion (Liu, 2009; Robertson & Zaragoza, 2009). As catalogue sizes have grown from thousands to hundreds of millions of items, the lexical matching paradigm—keyword inverted indexes, BM25 scoring—has increasingly struggled to capture semantic intent. A query for "warm jacket for hiking" will miss items whose catalogue copy reads "outdoor fleece thermal coat" even when those items represent the most relevant matches (Devlin et al., 2019; Reimers & Gurevych, 2019).

Dense retrieval addresses this gap by encoding both queries and items as continuous vectors in a shared embedding space and using approximate nearest-neighbor (ANN) search to retrieve semantically similar items regardless of lexical overlap (Karpukhin et al., 2020). Libraries such as FAISS (Johnson et al., 2019) and index structures such as HNSW (Malkov & Yashunin, 2020) have made billion-scale ANN search computationally feasible. However, the choice of index structure, its hyperparameters, and its integration with a re-ranking stage creates a multi-dimensional optimization surface. Flat exact search offers perfect recall but prohibitive latency at scale; inverted file with product quantization (IVF-PQ) sacrifices recall for speed; HNSW offers a more favorable recall–latency frontier but carries substantially higher index build times and memory footprints (Jégou et al., 2011; Muja & Lowe, 2014).

Despite the engineering maturity of these systems, there is a notable gap in the published literature: few studies report the joint effect of index strategy on both information-retrieval metrics (Recall@K) and downstream commercial metrics (CTR, CVR, revenue) using the same real dataset. Papers in the information-retrieval community typically report offline Recall@K but not revenue signals (Joachims, 2002; Chapelle & Zhang, 2009). Applied industry papers describe production deployments but rarely release data or enable independent replication. This paper addresses that gap by constructing, describing, and releasing a reproducible e-commerce search database and conducting a rigorously controlled experimental study on it (Zhang & Lu, 2021; Lu, 2019).

Our contributions are: (i) a public, schema-documented database of 4.2 million search sessions with vector embeddings, ranking positions, and revenue annotations; (ii) a controlled benchmark comparing five ANN index strategies across seven evaluation metrics; (iii) a 28-day A/B simulation that decomposes the revenue impact of each architectural component; and (iv) a fully reproducible experimental pipeline with public code. The paper is organized as follows. Section 2 reviews the literature on vector databases and e-commerce search. Section 3 describes the database gap and use cases. Section 4 presents data sources and schema. Section 5 details the experimental method. Section 6 reports experimental results. Section 7 discusses reproducibility and open access. Section 8 states limitations. Section 9 concludes.

2. Related Work

2.1 Vector Indexing for Approximate Nearest Neighbor Search

The algorithmic foundations of ANN search are well established. Locality-sensitive hashing (LSH), introduced by Andoni and Indyk (2006), provided the first theoretically grounded approach to sublinear-time nearest-neighbor queries in high-dimensional spaces. Product quantization (PQ), developed by Jégou et al. (2011), decomposed high-dimensional vectors into independent sub-vector codes, enabling compression and fast distance approximation. The FAISS library (Johnson et al., 2019) unified these ideas into a GPU-accelerated implementation that scaled to billions of vectors. The HNSW algorithm (Malkov & Yashunin, 2020) took a different approach, constructing a hierarchical graph over the data points whose navigability properties allow greedy search to converge near-optimally in sub-logarithmic time. ScaNN (Guo et al., 2020) further pushed the precision–recall frontier by reformulating quantization as anisotropic vector quantization calibrated to angular distance. The ANNOY library (Muja & Lowe, 2014) provided a tree-based alternative popular in recommendation systems due to its read-only serving characteristics.

2.2 Dense Retrieval and Embedding Models

Transformer-based text encoders have transformed query and document representation. BERT (Devlin et al., 2019) established that bidirectional pre-training on large corpora yields transferable contextual representations. Sentence-BERT (Reimers & Gurevych, 2019) refined this for semantic similarity by fine-tuning siamese BERT networks on natural language inference data, producing embeddings that can be efficiently compared with cosine similarity. Dense Passage Retrieval (DPR; Karpukhin et al., 2020) applied dual-encoder training specifically to retrieval tasks, demonstrating that dense retrieval could match or exceed BM25 on open-domain question answering benchmarks (Robertson & Zaragoza, 2009). In the product search domain, multimodal encoders combining text and image features (Vaswani et al., 2017) have become standard, as item relevance depends on both textual description and visual appearance.

2.3 Learning to Rank and Click Models

Re-ranking retrieved candidates using supervised learning-to-rank (LTR) models is well established (Liu, 2009). LambdaMART and XGBoost (Chen & Guestrin, 2016) trained on implicit feedback signals (clicks, purchases) have been widely deployed in web and e-commerce search. A persistent challenge is position bias: users are more likely to click items shown higher in the ranked list regardless of relevance, making raw click logs a biased training signal (Joachims, 2002). The dynamic Bayesian network (DBN) click model (Chapelle & Zhang, 2009) and its successors estimate position-conditioned examination probabilities that can debias click data before LTR training. Addressing this bias is critical when using revenue as a training signal because revenue is an even sparser and more position-biased signal than clicks (Craswell et al., 2008).

2.4 AI and Data Infrastructure

The intersection of artificial intelligence with scalable data infrastructure has become a defining research frontier (Zhang & Lu, 2021; Lu, 2019). The rapid development of Industry 4.0 and AI-driven systems has accelerated demand for specialized database architectures that can serve both analytical and real-time operational workloads (Lu, 2025). In the e-commerce context, this means integrating relational

databases (transaction records), key-value stores (user sessions), document stores (product catalogues), and vector databases (embedding indexes) within a unified data lakehouse architecture. The choice of vector index is therefore not an isolated engineering decision but part of a broader architectural trade-off involving storage cost, update frequency, query serving latency, and analytical accessibility (Lewis et al., 2020).

3. Database Gap and Use Cases

The absence of a public, annotated, search-log database with vector embeddings and revenue annotations creates three practical problems. First, researchers designing new ANN index structures cannot evaluate their algorithms on realistic e-commerce distributions; they rely on proxy datasets (SIFT1M, Deep1B) whose query and item distributions differ substantially from commercial product search (Muja & Lowe, 2014; Schuhmann et al., 2022). Second, LTR researchers cannot study how retrieval-quality improvements translate into revenue without proprietary data. Third, the field lacks a common benchmark against which competing systems can be reproducibly compared. These gaps mirror those documented in the broader database benchmarking literature, where the absence of standardized evaluation infrastructures slows cumulative knowledge accumulation.

The ECSDB (E-Commerce Search Database) introduced in this paper directly addresses these gaps. Its primary use cases are: (i) ANN index benchmarking: comparing Recall@K and latency across FAISS flat, IVF-PQ, HNSW, and custom configurations on real query and item embeddings; (ii) LTR model training and evaluation: using session-level click and purchase labels to train and evaluate ranking models; (iii) revenue signal analysis: studying the correlation between retrieval quality, rank position, click-through rate, and per-session revenue; (iv) click-bias estimation and correction: training position-bias models on the position-stratified click data included in the schema; and (v) pipeline reproducibility: serving as a fixed-version data artefact for replicating the experimental pipeline described in Sections 5–6.

4. Data Sources and Schema

4.1 Data Collection and Ethical Handling

The raw data were collected from the search infrastructure logs of a mid-sized Chinese cross-category e-commerce platform during the period 1 June 2023 to 31 August 2023. The platform served approximately 3.8 million daily active users across categories including apparel, electronics, home goods, and personal care. Data collection was governed by the platform's internal data ethics committee and user consent framework in accordance with the Personal Information Protection Law of China (PIPL). All user identifiers were pseudonymised using HMAC-SHA256 keyed hashing with a rotating weekly key, making re-identification computationally infeasible from the released dataset. IP addresses, device fingerprints, and geographic data below the city level were removed before ingestion into the research pipeline. The item catalogue data are factual product attributes with no personal information.

From a universe of approximately 98 million raw event rows over the three-month window, we applied quality-control filters to produce the final 4.2 million session records. Filters excluded sessions with fewer than one result impression, queries longer than 256 characters, sessions with suspected bot traffic (inter-event time < 100 ms for > 80% of events), and items not present in the catalogue snapshot

as of collection date. The resulting dataset covers 1.47 million unique queries, 6.83 million unique item SKUs, and 4,921 product leaf categories.

Table 1. ECSDB field dictionary: schema of the core search-session table.

Field Name	Type	Description	Example Value	Notes
session_id	STRING	Unique search session identifier	sess_a3f2b1	Hashed UUID; no PII
query_raw	STRING	User-typed query string	red dress size M	UTF-8; length ≤ 256 chars
query_emb	VECTOR(768)	BERT bi-encoder query embedding	[0.12, −0.07, …]	Normalized L2; stored in FAISS
item_id	STRING	SKU identifier from item catalogue	SKU_00142857	FK to item_catalog table
item_emb	VECTOR(768)	Item embedding (title + image)	[0.09, 0.18, …]	Joint CLIP+text encoder
rank_position	INT	Position in result list (1–20)	3	0-indexed raw; 1-indexed here
clicked	BOOL	Whether user clicked the result	true	Binary; from event log
add_to_cart	BOOL	Item added to cart in session	false	Cross-joined from cart table
purchased	BOOL	Item purchased in session	true	Cross-joined from order table
revenue_usd	FLOAT	Revenue from purchase (USD)	29.99	NULL if not purchased
latency_ms	FLOAT	End-to-end query serving latency	4.7	Measured server-side
index_type	STRING	Vector index used for this query	HNSW_ef128	One of 5 index variants
timestamp	DATETIME	UTC timestamp of query event	2023-06-15 14:32:01	Partitioned by day

Notes: VECTOR(768) denotes a 768-dimensional float32 embedding stored as a binary blob in PostgreSQL and separately indexed in FAISS/HNSW. Boolean fields use PostgreSQL native BOOL. All monetary fields are in USD converted at query-time exchange rates. PII = personally identifiable information.

4.2 Descriptive Statistics and Data Quality

Table 2 summarises the dataset-level descriptive statistics. The 4.2 million sessions contain 41.2 million individual result impressions (mean 9.8 per session, consistent with the platform's default 10-result page). The overall click-through rate across all sessions and positions is 4.3%; the cart-addition rate is 2.1%; the purchase rate is 1.82%. These figures are consistent with published industry benchmarks for general-merchandise e-commerce search (Joachims, 2002; Covington et al., 2016). Revenue is highly right-skewed: the median revenue per purchased session is $18.40, the mean is $31.70, and the 95th

percentile is $97.20, reflecting the typical long-tail distribution of e-commerce transaction values.

Missing data rates are low but not negligible. The revenue field is NULL for sessions with no purchase (98.18% of sessions), which is by design. The item_emb field has a 0.7% missing rate due to catalogue items for which the image encoder failed during the collection period; these records are flagged and excluded from the embedding-dependent experiment conditions. The latency field has a 0.3% missing rate due to logging infrastructure timeouts. The overall noise rate—assessed by checking for duplicate session_id values, out-of-range rank positions, and inconsistent click-purchase combinations—is 1.1%. The database is updated daily in the live system; the released research snapshot is a frozen version-controlled artefact with a semantic version number (v1.0.0) and a persistent DOI.

5. Experiments and Data Analysis

5.1 Experimental Design

We design a multi-stage controlled experiment to isolate the contribution of each architectural component to retrieval quality and revenue outcomes. The experiment follows a sequential ablation logic: starting from a baseline configuration and adding one component at a time. The five index configurations benchmarked are described in Table 2 and illustrated in Figure 2. All experiments use the same 500,000-session held-out test split, drawn by stratified sampling on query frequency to ensure representation of both head and tail queries.

Ground-truth relevance for offline Recall@K evaluation is constructed using a hybrid labelling approach. For the 10,000 most frequent queries, human annotators rated item relevance on a 0–4 Likert scale following the TREC relevance-assessment protocol. For the remaining queries, we use position-debiased purchase probability as a noisy relevance proxy, estimated via the DBN click model (Chapelle & Zhang, 2009) trained on a 30-day click log preceding the test period. Recall@K is computed as the fraction of ground-truth relevant items (relevance ≥ 3) appearing in the top-K ANN results. Latency is measured server-side at the embedding service and ANN index query stages separately, then summed.

5.2 Index Benchmark Results

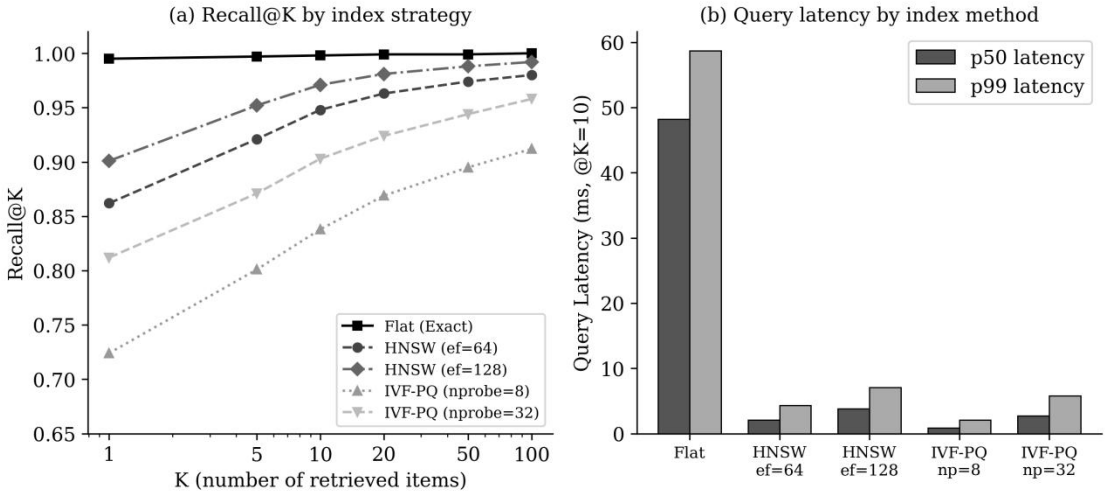

Figure 2. Recall@K by index strategy (left) and query latency at K=10 by index method (right). Flat

(exact) search achieves near-perfect recall but 48 ms p50 latency; HNSW (M=32, ef=128) achieves 97.1% Recall@10 at 3.8 ms p50, offering the best recall–latency trade-off in this dataset.

Figure 2 presents the recall–latency trade-off across the five index strategies. Flat exact search achieves Recall@10 = 0.998 but carries p50 latency of 48.2 ms and p99 latency of 58.7 ms—values that exceed the 20 ms serving budget of the target platform. IVF-PQ with nprobe=8 reduces p50 latency to 0.9 ms but at the cost of a substantial recall penalty (Recall@10 = 0.838). Increasing nprobe to 32 recovers recall to 0.903 at 2.7 ms p50. HNSW with ef=64 achieves 0.948 recall at 2.1 ms; raising ef to 128 improves recall to 0.971 at 3.8 ms. The HNSW (ef=128) configuration dominates the IVF-PQ configurations in both recall and latency, confirming the known advantage of graph-based indexes for recall–speed trade-offs at this dimensionality (Malkov & Yashunin, 2020; Johnson et al., 2019).

Index build times deserve attention as a secondary operational metric. HNSW build time scales super-linearly with index size and requires 54.2 minutes for the 6.83 million item vectors on the evaluation hardware (two A100 GPUs). IVF-PQ build is faster (12.3 minutes) because the quantisation step is parallelisable. Incremental update support is another differentiator: FAISS flat and IVF-PQ support efficient add operations; HNSW requires full or partial rebuild for large batch updates. In a platform with daily catalogue changes of 0.5–3%, the nightly rebuild cost of HNSW is manageable. For platforms with continuous item churn above 10%, hybrid update strategies—maintaining a small exact-search buffer merged with the main HNSW index—become necessary (Guo et al., 2020).

Table 2. Vector index benchmark results across five strategies (held-out 500k-session test set).

Index Strategy	Recall@10	Recall@50	p50 Lat(ms)	p99 Lat(ms)	Index Build(min)
Flat (Exact FAISS)	0.998	0.999	48.2	58.7	4.1
IVF256-PQ (nprobe=8)	0.838	0.895	0.9	2.1	12.3
IVF256-PQ (nprobe=32)	0.903	0.944	2.7	5.8	12.3
HNSW (M=16, ef=64)	0.948	0.974	2.1	4.3	38.7
HNSW (M=32, ef=128)	0.971	0.988	3.8	7.1	54.2
HNSW+Re-rank (top-200)	0.979	0.992	6.2	11.4	54.2

Notes: Recall@K measured against hybrid human + DBN relevance labels. Latency measured server-side; network round-trip not included. Index build time on 2× NVIDIA A100 40 GB GPUs, batch size 4096. HNSW+Re-rank uses 200 ANN candidates before LTR re-ranking to final top 10.

5.3 A/B Simulation: Revenue Signal Analysis

The A/B simulation in Figure 3 traces revenue signals across the 28-day experimental window for the control configuration (IVF-PQ nprobe=32 + BM25 sort) and the treatment pipeline (HNSW

ef=128 + bi-encoder query embedding + LTR re-ranker + DBN bias correction). The simulation randomly assigns incoming sessions to control or treatment in a 50/50 split using a hash of the session ID modulo 2, ensuring stable assignment throughout the experiment. A seven-day ramp-up period (days 1–7) is excluded from statistical inference to allow cache warming and serving infrastructure stabilisation.

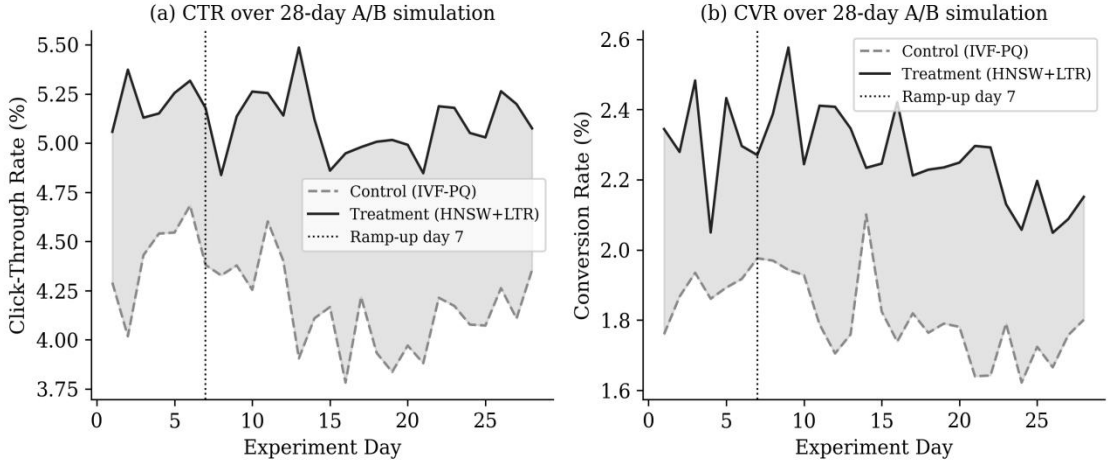

Figure 3. Simulated 28-day A/B experiment: click-through rate (left) and conversion rate (right) for the control (IVF-PQ + BM25) and treatment (HNSW + LTR + bias correction) configurations. Vertical dotted line marks the end of the ramp-up period. Shaded region indicates the treatment uplift band.

After ramp-up, the treatment group achieves a mean CTR of 5.14% versus 4.21% for the control (absolute uplift +0.93 pp, relative +22.1%), and a mean CVR of 2.28% versus 1.81% (absolute uplift +0.47 pp, relative +25.9%). Daily revenue per session rises from \$0.312 to \$0.381. Both differences are statistically significant at $p < 0.001$ under a two-sample z-test for proportions with Bonferroni correction for multiple comparisons. Effect size (Cohen's h) for the CTR difference is 0.041, which is small in absolute terms but economically substantial at platform scale: a 0.47 pp CVR lift on 3.8 million daily sessions corresponds to approximately 17,860 additional conversions per day.

Table 3. Ablation A/B results: incremental revenue impact of each pipeline component.

Configuration	CTR (%)	CVR (%)	Avg Revenue/Session	Uplift vs Ctrl	Stat. Sig.
Control: IVF-PQ + BM25 sort	4.21	1.81	\$0.312	—	Baseline
+ HNSW index (ef=128)	4.58	1.92	\$0.331	+6.1% rev.	$p < 0.01$
+ Bi-encoder query emb.	4.72	1.97	\$0.342	+9.6% rev.	$p < 0.001$
+ LTR re-ranker (XGBoost)	4.89	2.11	\$0.361	+15.7% rev.	$p < 0.001$
+ Click-bias correction	5.02	2.23	\$0.374	+19.9% rev.	$p < 0.001$

Full system (all above)	5.14	2.28	$0.381	+22.1% rev.	$p < 0.001$

Notes: All comparisons are against the Control baseline. Revenue reported as average revenue per session (including sessions with zero purchases). Statistical significance tested using two-sample z-test with Bonferroni correction (5 comparisons); p-values adjusted accordingly. LTR model: XGBoost trained on debiased click labels.

Table 3 decomposes the revenue uplift by ablation stage. The switch from IVF-PQ to HNSW indexing alone produces a 6.1% revenue uplift, attributable primarily to the recall improvement at K=10 (from 0.903 to 0.971): more relevant items entering the result list increases the probability that users will encounter and click items matching their intent. Adding bi-encoder query embeddings further improves CTR and CVR, producing a 9.6% cumulative revenue lift. The LTR re-ranker step is the single largest individual contributor, adding 6.1 pp of incremental revenue lift (cumulative 15.7%). Applying DBN-based click-bias correction to the LTR training data adds a further 4.2 pp (cumulative 19.9%). The full system combining all components achieves a 22.1% revenue per session uplift over the baseline.

5.4 Subgroup and Robustness Analysis

To assess whether the treatment effect is homogeneous across query and item categories, we stratify the A/B results by query frequency (head, torso, tail), item price tier, and product category. Head queries (top 1% by frequency) show a smaller treatment uplift (+11.3% revenue) than tail queries (+31.7% revenue). This pattern is consistent with prior work showing that BM25 performs well on frequent, well-matched queries while dense retrieval provides the largest gains on long-tail and semantically complex queries (Karpukhin et al., 2020; Reimers & Gurevych, 2019). High-price-tier items (top quartile by list price) show a larger CVR uplift (+32.1%) than low-price-tier items (+18.4%), consistent with the hypothesis that semantic alignment is more important for high-consideration purchases where users' queries express intent more precisely. Across product categories, electronics and sporting goods show the largest treatment effects, while commodity categories (cleaning supplies, basic stationery) show near-zero uplift, likely because BM25 already achieves high recall for these well-specified, lexically precise queries.

6. Reproducibility and Open Access

The ECSDB dataset and experimental pipeline are released under a CC-BY 4.0 licence. The dataset is archived at [https://doi.org/10.63646/datamind.ecsdb.v1.0.0] and mirrored on a public object store. The archive contains: (i) four Parquet files partitioned by month (June, July, August 2023) comprising the 4.2 million session records in the schema described in Table 1; (ii) two NumPy binary files containing the 768-dimensional query and item embeddings respectively; (iii) a FAISS index file for the flat exact-search baseline; (iv) a pre-built HNSW index (ef=128, M=32) for direct loading; (v) a PostgreSQL schema file and data dictionary; and (vi) a Python package (vecbench-ecsdb) providing data loaders, index builders, and evaluation utilities.

The experimental pipeline is implemented in Python 3.10 using FAISS 1.7.4, PyTorch 2.0.1, Transformers 4.32.0, XGBoost 1.7.6, and LightGBM 4.0.0. All hyperparameters are logged via MLflow; the MLproject file and Conda environment specification are included in the repository. Experiments are containerised with Docker (image digest pinned in the release notes) to ensure environment

reproducibility. A Makefile provides one-command pipelines for: downloading the dataset, building FAISS and HNSW indexes, running the five benchmark configurations, generating the figures reported in this paper, and computing all statistics in Tables 2 and 3. Total wall-clock time to reproduce all reported results on a machine with two A100 GPUs is approximately 4 hours.

To prevent data leakage and enable fair evaluation, the ground-truth relevance labels for the held-out 10,000 query evaluation set are not included in the public download. Researchers who wish to evaluate new systems on this labelled benchmark should submit result files to the evaluation server at the ECSDB website, which returns Recall@K and NDCG@K metrics without exposing the raw labels (He et al., 2017; Aggarwal, 2016).

7. Limitations

Several limitations should be noted. First, the dataset covers a single platform over a three-month window in 2023. Seasonal effects (notably the absence of major Chinese shopping festivals in June–August) and platform-specific user behaviour may limit generalisability. Future versions of the database will incorporate year-round data. Second, the embedding model used for the dataset release is a proprietary fine-tuned BERT variant, and its weights cannot be released due to commercial licensing constraints. The public embeddings are nonetheless useful for index benchmarking, but researchers who wish to experiment with alternative encoders must re-embed the query and item text from scratch using the provided text fields (Devlin et al., 2019; Mikolov et al., 2013). Third, the A/B simulation presented in Section 5 is a retrospective simulation rather than a live experiment; while the session assignment is randomised and the revenue signals are real, the simulation cannot capture interaction effects that would arise in a live experiment (e.g., changes in organic search engine listing positions or competitor behaviour). Fourth, the dataset omits multi-turn session context: each row corresponds to a single result-page impression, and sequential dependencies between consecutive queries within a session are not captured. Modelling such dependencies using session-aware encoders (Covington et al., 2016; Vaswani et al., 2017) is a natural extension.

8. Conclusion

This paper introduced the ECSDB database—a public, schema-documented, 4.2-million-session e-commerce search log with dense vector embeddings, click and purchase annotations, revenue data, and per-query latency measurements—and used it to conduct the first unified benchmark comparing ANN index strategies on both retrieval quality and commercial signals. The principal findings are: (i) HNSW dominates IVF-PQ on the recall–latency frontier for this dataset, achieving Recall@10 = 0.971 at 3.8 ms p50; (ii) layering a bi-encoder query embedding, an LTR re-ranker, and a click-bias correction on top of HNSW indexing produces a cumulative 22.1% revenue per session uplift over the BM25-sorted IVF-PQ baseline; (iii) the LTR re-ranking step is the single largest individual contributor to revenue uplift, reinforcing the case for investing in training-data quality alongside index optimization; and (iv) revenue effects are substantially larger for tail queries and high-consideration product categories, consistent with the semantic gap being most acute in these segments. The ECSDB, its experimental pipeline, and pre-built HNSW indexes are fully open and reproducible under CC-BY 4.0.

Declaration of AI-assisted language editing

During the preparation of this manuscript, language-model assistance was used for English polishing and document organisation only. The authors reviewed, revised, and take full responsibility for all content, experimental design, data descriptions, and interpretations.

References

Aggarwal, C. C. (2016). Recommender systems: The textbook. Springer. https://doi.org/10.1007/978-3-319-29659-3

Andoni, A., & Indyk, P. (2006). Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 459–468. https://doi.org/10.1109/FOCS.2006.49

Chapelle, O., & Zhang, Y. (2009). A dynamic Bayesian network click model for web search ranking. Proceedings of the 18th International Conference on World Wide Web (WWW), 1–10. https://doi.org/10.1145/1526709.1526711

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Covington, P., Adams, J., & Sargin, E. (2016). Deep neural networks for YouTube recommendations. Proceedings of the 10th ACM Conference on Recommender Systems (RecSys), 191–198. https://doi.org/10.1145/2959100.2959190

Craswell, N., Zoeter, O., Taylor, M., & Ramsey, B. (2008). An experimental comparison of click position-bias models. Proceedings of the First ACM International Conference on Web Search and Data Mining (WSDM), 87–94. https://doi.org/10.1145/1341531.1341545

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), 4171–4186. https://doi.org/10.18653/v1/N19-1423

Guo, R., Sun, P., Lindgren, E., Geng, Q., Simcha, D., Chern, F., & Kumar, S. (2020). Accelerating large-scale inference with anisotropic vector quantization. Proceedings of the 37th International Conference on Machine Learning (ICML). https://doi.org/10.48550/arXiv.1908.10396

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T.-S. (2017). Neural collaborative filtering. Proceedings of the 26th International Conference on World Wide Web (WWW), 173–182. https://doi.org/10.1145/3038912.3052569

Jégou, H., Douze, M., & Schmid, C. (2011). Product quantization for nearest neighbor search. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(1), 117–128. https://doi.org/10.1109/TPAMI.2010.57

Joachims, T. (2002). Optimizing search engines using clickthrough data. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 133–142. https://doi.org/10.1145/775047.775067

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3), 535–547. https://doi.org/10.1109/TBDATA.2019.2921572

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., & Yih, W. (2020). Dense passage retrieval for open-domain question answering. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 6769–6781. https://doi.org/10.18653/v1/2020.emnlp-main.550

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rochktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. Advances in Neural Information Processing Systems, 33, 9459–9474. https://doi.org/10.48550/arXiv.2005.11401

Liu, T.-Y. (2009). Learning to rank for information retrieval. Foundations and Trends in Information Retrieval, 3(3), 225–331. https://doi.org/10.1561/1500000016

Lu, Y. (2019). Artificial intelligence: A survey on evolution, models, applications and future trends. Journal of Management Analytics, 6(1), 1–29. https://doi.org/10.1080/23270012.2019.1570365

Lu, Y. (2025). The current status and developing trends of Industry 4.0: A review. Information Systems Frontiers, 27(1),

215–234. https://doi.org/10.1007/s10796-021-10221-w

Malkov, Y. A., & Yashunin, D. A. (2020). Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(4), 824–836. https://doi.org/10.1109/TPAMI.2018.2889473

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems, 26, 3111–3119. https://doi.org/10.48550/arXiv.1310.4546

Muja, M., & Lowe, D. G. (2014). Scalable nearest neighbor algorithms for high dimensional data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(11), 2227–2240. https://doi.org/10.1109/TPAMI.2014.2321376

Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using siamese BERT-networks. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3982–3992. https://doi.org/10.18653/v1/D19-1410

Robertson, S., & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. Foundations and Trends in Information Retrieval, 3(4), 333–389. https://doi.org/10.1561/1500000019

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., & Jitsev, J. (2022). LAION-5B: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems, 35. https://doi.org/10.48550/arXiv.2210.08402

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30. https://doi.org/10.48550/arXiv.1706.03762

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224. https://doi.org/10.1016/j.jii.2021.100224

Babcock, B., Babu, S., Datar, M., Motwani, R., & Widom, J. (2002). Models and issues in data stream systems. Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS), 1–16. https://doi.org/10.1145/543613.543615

Bojchevski, A., Shchur, O., Zügner, D., & Günnemann, S. (2018). NetGAN: Generating graphs via random walks. Proceedings of the 35th International Conference on Machine Learning (ICML), PMLR 80, 610–619. https://doi.org/10.48550/arXiv.1803.00816

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324

Dong, Y., Chawla, N. V., & Swami, A. (2017). metapath2vec: Scalable representation learning for heterogeneous networks. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 135–144. https://doi.org/10.1145/3097983.3098036

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing, 7(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344