Through the Lens of Hubness: A Revisit on Graph-Based Approximate Nearest Neighbor Search: [Experiments & Analysis]

XIAOLIANG XU, Hangzhou Dianzi University, China
HAONAN DAI, Hangzhou Dianzi University, China
CAN LI, Hangzhou Dianzi University, China
MENGZHAO WANG*, Hangzhou Dianzi University, China
QIANG YUE, Hangzhou Dianzi University, China

Graph-based Approximate Nearest Neighbor Search (ANNS) is a cornerstone of modern AI systems; however, its practical utility is undermined by a severe long-tail latency problem in which outlier queries jeopardize Service-Level Objectives (SLOs). This paper presents the first systematic study to establish the hubness phenomenon, an intrinsic property of high-dimensional data, as the fundamental cause of this performance instability. Our theoretical analysis reveals that hubness induces topologically skewed proximity graphs, characterized by overly centralized *hubs* and isolated *anti-hubs*. This topological imbalance invalidates the greedy traversal heuristic underpinning graph-based search, explaining why queries targeting anti-hubs become performance outliers. We propose a unified framework that deconstructs mainstream ANNS algorithms, reinterpreting their designs as a taxonomy of distinct hubness-mitigation strategies. Extensive experiments on datasets scaling up to 100 million vectors validate this framework, linking an algorithm's mitigation strategy to its effectiveness in balancing graph topology and optimizing outlier performance. Furthermore, our analysis uncovers critical design challenges, specifically the inherent trade-off between suppressing hubs and ensuring the reachability of anti-hubs. Building on these insights, we introduce a hubness-aware pruning optimization that efficiently refines graph topology. This approach yields performance improvements with negligible processing overhead, validating the potential of designing robust graph-based ANNS indexes by integrating hubness information.

CCS Concepts: • **Information systems** → **Nearest-neighbor search**; **Retrieval efficiency**; **Retrieval effectiveness**.

Additional Key Words and Phrases: Approximate Nearest Neighbor Search; Graph-based Vector Index; High-Dimensional Hubness

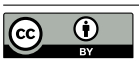
ACM Reference Format:
Xiaoliang Xu, Haonan Dai, Can Li, Mengzhao Wang, and Qiang Yue. 2026. Through the Lens of Hubness: A Revisit on Graph-Based Approximate Nearest Neighbor Search: [Experiments & Analysis]. *Proc. ACM Manag. Data* 4, 3 (SIGMOD), Article 243 (June 2026), 27 pages. https://doi.org/10.1145/3802120

1 Introduction

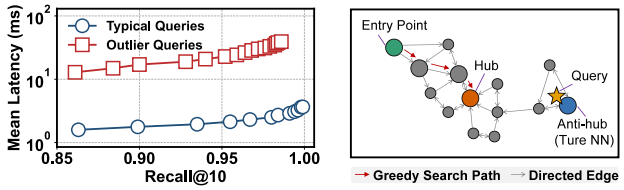
The proliferation of large-scale AI applications has established high-dimensional vector embeddings as the de facto standard for representing complex data [47, 57, 61]. Efficient similarity search over

*Mengzhao Wang is the corresponding author.

Authors' Contact Information: Xiaoliang Xu, Hangzhou Dianzi University, China, xxl@hdu.edu.cn; Haonan Dai, Hangzhou Dianzi University, China, dhn@hdu.edu.cn; Can Li, Hangzhou Dianzi University, China, lic@hdu.edu.cn; Mengzhao Wang, Hangzhou Dianzi University, China, wmzssy@hdu.edu.cn; Qiang Yue, Hangzhou Dianzi University, China, yq@hdu.edu.cn.



ACM 2836-6573/2026/6-ART243
https://doi.org/10.1145/3802120



(a) Latency–Recall **(b) Toy Example**

Fig. 1. The outlier query phenomenon in graph-based ANNS. (a) The latency-recall trade-off on the GIST dataset. (b) A conceptual illustration of hubness-induced search failure, where a high-degree hub captures the greedy search path, isolating the true nearest neighbor (anti-hub).

these vectors is integral to foundational domains such as data mining [7, 11] and recommendation systems [46, 64]. More recently, the advent of Large Language Models (LLMs) has propelled paradigms like Retrieval-Augmented Generation (RAG) to the forefront, creating a critical dependence on high-performance vector search that is increasingly supported by vector database systems [9, 51]. However, the inherent *curse of dimensionality* imposes a severe computational bottleneck on exact nearest neighbor search [33], rendering it infeasible for production systems governed by stringent real-time Service-Level Objectives (SLOs). Consequently, Approximate Nearest Neighbor Search (ANNS) has emerged as an indispensable field of study, aiming to strike an optimal balance between retrieval accuracy and query efficiency [5, 54, 58].

Among the spectrum of ANNS paradigms, graph-based algorithms have emerged as the dominant approach [17, 26, 40, 44, 53, 63], consistently delivering state-of-the-art performance. These methods operate in two distinct phases: offline index construction and online query processing. In the offline phase, the dataset is organized into a proximity graph, where vertices represent data vectors and directed edges denote proximity relationships. During the online phase, the search initiates at predefined entry points and executes an iterative greedy traversal. In each step, the algorithm transitions to the neighbor that minimizes the distance to the query, exploiting the geometric gradient to steadily converge on the global nearest neighbors. The efficacy of this paradigm is fundamentally predicated on the *transitive proximity heuristic* [13, 17]—the assumption that a neighbor's neighbor is likely also to be a neighbor.

Notwithstanding their demonstrated superiority, the practical utility of graph-based ANNS algorithms is hampered by the long-tail latency problem [4, 18, 31, 50, 56]. In real-world deployments, certain *outlier queries* exhibit substantially poorer performance than average. For these queries, attaining the target recall requires disproportionately long graph traversals, manifesting as a severe long-tail latency effect. This phenomenon is quantified in Figure 1a, which plots the latency-recall trade-off on the GIST dataset for a sample of 100 typical and 100 outlier queries. While typical queries achieve higher recall with only a moderate increase in latency, outlier queries demand an order of magnitude higher computational cost to reach the same recall targets, revealing a drastic performance disparity. This unpredictability poses a critical challenge to production systems, where ensuring low tail latency is essential to meeting stringent Service-Level Objectives (SLOs) [23, 25, 28, 34].

Prior work has sought to alleviate this performance instability along three complementary directions: (i) modeling and adapting to dataset-dependent difficulty [6, 12, 18], (ii) improving the graph construction and search procedures [17, 36, 44], and (iii) reducing the computational overhead of high-dimensional distance evaluations [1, 21, 52]. The first direction develops adaptive mechanisms tailored to specific data distributions [18, 31]. The second direction refines algorithmic components, including neighbor selection [17, 44] and routing strategies [36, 60]. The third direction leverages compression techniques (e.g., product quantization) to accelerate distance computations [22, 27].

Despite these efforts, outlier queries and the resulting long-tail latency remain persistent in practice [56]. This motivates a fundamental question: *what intrinsic factor governs the long-tail latency of graph-based ANNS?*

To answer this question, we examine the traversal behavior of mainstream graph-based ANNS algorithms on outlier queries. We find that the ground-truth nearest neighbors of these queries exhibit atypical topological characteristics, which implicates the hubness phenomenon [15, 41, 45]. Hubness is an intrinsic property of high-dimensional data and is manifested in proximity graphs as a highly skewed in-degree distribution, yielding two classes of vertices: *hubs* with disproportionately large in-degrees and *anti-hubs* with near-zero in-degrees. This structural imbalance offers an explanation for outlier queries (Figure 1b): when the true nearest neighbor is an anti-hub, greedy traversal is repeatedly attracted toward nearby hubs that dominate local choices, resulting in substantially longer search paths and the long-tail latency observed in practice (Figure 1a).

The intrinsic connection between outlier queries and hubness calls for a re-examination of graph-based ANNS from a topological perspective. In this paper, we present the first systematic study centered on this relationship. We first provide a theoretical analysis of how hubness shapes proximity-graph topology and can undermine the greedy traversal heuristic. We then revisit representative ANNS indexes through the lens of hubness and develop a unified framework that views their designs as a sequence of hubness-mitigation strategies. Building on these insights, we propose hubness-aware optimizations and distill practical implications for designing more robust indexes. To facilitate reproducibility, all artifacts are publicly available[1].

Our contributions are summarized as follows:

- We identify hubness as the root cause of outlier queries in graph-based ANNS and theoretically show how it skews graph topology and breaks the greedy traversal heuristic.
- We reinterpret mainstream ANNS designs as a taxonomy of hubness-mitigation strategies, providing a unified framework to analyze their strengths and limitations.
- We present an experimental study that empirically links hubness-mitigation efficacy to outlier-query performance, confirming that topological balance is a governing predictor of search efficiency.
- We propose a hubness-aware pruning optimization that leverages topological signals to refine graph construction, yielding more balanced indexes and improved performance with negligible indexing overhead.
- We distill key insights into the trade-off between reducing in-degree skewness and preserving local connectivity, and derive practical implications for robust graph-based ANNS design.

The rest of this paper is structured as follows. Section 2 defines the greedy search process, outlier queries, and hubness. Section 3 establishes the theoretical link between hubness, graph topology, and greedy-search failure. Section 4 revisits mainstream ANNS indexes through the lens of hubness. Section 5 provides empirical validation and evaluates our hubness-aware pruning optimization. Section 6 discusses key insights and future directions, and Section 7 concludes the paper.

2 Preliminaries

This section establishes the formal foundation for graph-based Approximate Nearest Neighbor Search (ANNS). We define the greedy search process and its properties, standardizing core terminology such as *local region* and *local proximity*. We then empirically characterize the outlier-query phenomenon that degrades performance stability and introduce the concept of hubness.

[1]https://github.com/H40NaN/GraphANNS-Hubness

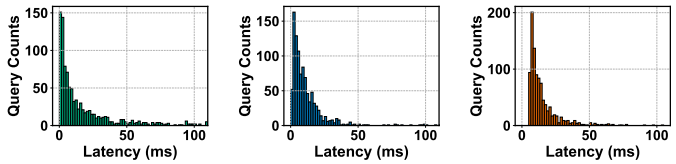


Fig. 2. The long-tail latency phenomenon in graph-based ANNS. The distribution of per-query latency to reach 99% Recall@10 on the GIST dataset is strongly right-skewed.

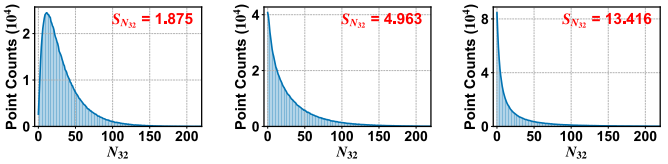


Fig. 3. In-degree distributions across representative datasets. The stark shift from SIFT's symmetric distribution to the right-skewed profiles of LAION and GIST highlights severe topological imbalance.

2.1 Graph-based Greedy Search

Graph-based methods [5, 8, 10, 59] are widely regarded as state-of-the-art solutions for high-dimensional ANNS. They typically operate in two phases: offline index construction and online search. In the construction phase, a dataset $S = \{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d$ is organized into a directed proximity graph $G = (V, E)$, where each vertex $x_v \in V$ corresponds to a data point $x_i \in S$ and each edge $(x_u, x_v) \in E$ denotes that x_v is selected as a neighbor of x_u under a similarity metric $\delta(\cdot, \cdot)$. Different graph-based indexes primarily differ in how they select and prune edges to obtain a sparse yet navigable graph.

Given a query $q \in \mathbb{R}^d$, online search traverses G to retrieve approximate nearest neighbors. For subsequent analysis, we describe this procedure via three components:

- **Entry Points.** The search starts from a seed set $S_{seed} \subset V$. While better seeds can speed up convergence, the small-world property of modern indexes often enables effective and efficient navigation even from random seeds.
- **Greedy Search Path.** The traversal induces a vertex sequence $P = (x_{p_0}, x_{p_1}, \ldots, x_{p_T})$. At step t, the algorithm explores the ***local region*** defined by the neighborhood $N(x_{p_t})$. The transition to the next vertex $x_{p_{t+1}}$ is governed by ***local proximity***—specifically, the heuristic of selecting the neighbor that minimizes the distance to q (i.e., $x_{p_{t+1}} = \arg\min_{v \in N(x_{p_t})} \delta(q, v)$). Ideally, this path is *monotonic* [17], implying $\delta(q, x_{p_{t+1}}) \leq \delta(q, x_{p_t})$.
- **Termination.** The process stops at a local optimum x_{p^*} such that $\forall x_v \in N(x_{p^*}),\ \delta(q, x_v) \geq \delta(q, x_{p^*})$.

In practice, the greedy heuristic is commonly instantiated as beam search (Algorithm 1) to reduce sensitivity to local optima. The algorithm maintains a priority queue of candidates keyed by $\delta(q, \cdot)$ and a visited set. Iteratively, it expands the closest unvisited candidate, inserts its neighbors into the queue, and prunes the queue to a fixed beam width L, finally returning the top-k results.

2.2 The Outlier Query Phenomenon

Although graph-based ANNS is efficient on average, it often exhibits long-tail latency in practice [50, 56]. In production settings, a subset of queries requires disproportionately high computation to reach the same target recall, resulting in substantial performance variance and complicating the enforcement of tail-latency Service-Level Objectives (SLOs), such as P99 latency.

Algorithm 1: Greedy Search on a Proximity Graph

Input: A graph index $G = (V, E)$; a query vector q; seed vertices S_{seed}; the number of results k; beam width L ($L \geq k$).

Output: The set $\tilde{\mathcal{R}}$ of k approximate nearest neighbors to q.

```
// Initialize candidates with seeds
```

Let C_q be a priority queue initialized with S_{seed};
Let $V_{visited} = \emptyset$ be the set of visited vertices;
Let $\tilde{\mathcal{R}}$ be a result set, initially empty;
while C_q *is not empty* **do**
    $x_{p^*} \leftarrow C_q$.pop() ; // Extract the closest candidate
    if x_{p^*} *has been visited* **then**
        Continue;
    $V_{visited}$.add(x_{p^*}); $\tilde{\mathcal{R}}$.add(x_{p^*});
    if $|\tilde{\mathcal{R}}| > k$ **then**
        $\tilde{\mathcal{R}}$.remove(farthest vertex from q);
    // A common termination condition
    if $\delta(q, x_{p^*}) > \delta(q,$ *farthest vertex in* $\tilde{\mathcal{R}})$ *and* $|\tilde{\mathcal{R}}| == k$ **then**
        Break;
    for $\forall x_v \in N_{out}(x_{p^*})$ **do** // Explore neighbors
        if $x_v \notin V_{visited}$ **then**
            C_q.add(v);
            if $|C_q| > L$ **then**
                C_q.remove(farthest vertex from q);
return The k vertices in $\tilde{\mathcal{R}}$ closest to q;

Figure 2 illustrates this phenomenon on the GIST dataset. For three representative algorithms, the per-query latency to achieve 99% Recall@10 is strongly right-skewed: most queries complete quickly, whereas a small set of *outlier queries* incur substantially higher latency than the median. The consistent presence of such outliers across algorithms suggests that long-tail latency is intrinsic to greedy graph traversal rather than an implementation artifact. Our goal is to characterize this behavior and analyze its root cause.

2.3 The Hubness Phenomenon

We posit that the *hubness phenomenon* underlies the outlier-query problem. In high-dimensional spaces, hubness is an effect in which the distribution of nearest-neighbor occurrences becomes skewed, so that a small fraction of points appear disproportionately often in other points' k-NN lists [45, 49]. To make this notion precise, we formalize hubness using the concept of k-occurrence.

Definition 1 (k-occurrence). *Given a dataset $S = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ and a similarity metric $\delta(\cdot, \cdot)$, the k-occurrence of a point $x_i \in S$, denoted $N_k(x_i)$, represents the frequency with which x_i appears in the k-nearest neighbor (k-NN) lists of other points in S. Formally:*

$$N_k(x_i) = \sum_{j=1, j\neq i}^{n} I(x_i \in k\text{-}NN(x_j)), \tag{1}$$

where $I(\cdot)$ is the indicator function and k-NN(x_j) denotes the set of k-nearest neighbors of x_j. In the context of a k-NN graph constructed on S, $N_k(x_i)$ corresponds precisely to the in-degree of vertex x_i.

This skewness in the N_k distribution induces a clear topological dichotomy: *hubs*, which have disproportionately large N_k values (high in-degree), and *anti-hubs*, which have negligible N_k values (often close to zero) and are therefore topologically isolated.

To quantify the severity of hubness, we measure the skewness of the N_k distribution via its standardized third moment [45, 48]:

$$S_{N_k} = \frac{E[(N_k - \mu_{N_k})^3]}{\sigma_{N_k}^3}. \tag{2}$$

Here, μ_{N_k} and σ_{N_k} are the mean and standard deviation of N_k, respectively. Larger positive values of S_{N_k} indicate stronger right-skewness (i.e., a heavier right tail) and thus more pronounced hubness. As shown in Figure 3, the effect becomes increasingly salient as dimensionality grows: the SIFT dataset exhibits an approximately symmetric in-degree distribution, whereas LAION and GIST display markedly right-skewed distributions with substantially higher skewness. These observations confirm that hubness is an intrinsic topological property of high-dimensional data and motivate examining its implications for greedy graph traversal.

3 Theoretical Analysis of Hubness on Graph Traversal

This section provides a theoretical analysis of how hubness distorts proximity-graph topology. We show that hubness undermines the *transitive proximity heuristic* guiding greedy traversal, providing a foundation for the outlier-query phenomenon.

3.1 Proximity Graphs in Idealized Low-Dimensional Settings

As a baseline, we consider low-dimensional data $S = \{x_1, \ldots, x_n\} \subset \mathbb{R}^d$ with small d, where points are drawn i.i.d. from a well-behaved distribution (e.g., a unimodal Gaussian [42]). In this regime, k-NN occurrences are approximately uniform across points, so the in-degree distribution $P(N_k)$ of the k-nearest neighbor graph (KNNG) concentrates around its mean and resembles that of an Erdős–Rényi (ER) random graph $G(n, p)$ [14].

Definition 2 (Erdős-Rényi Random Graph [19]). *The $G(n, p)$ model describes a probability distribution over graphs with n vertices, where each potential edge is included independently with a fixed probability p.*

In low-dimensional settings, the KNNG exhibits ER-like topology, including short average path lengths and high clustering coefficients—i.e., *small-world* structure [29, 39, 40]. This regularity supports the *transitive proximity heuristic* [13]: if x_u is close to x_v and x_v is close to x_w, then x_u is likely close to x_w.

In an ER-like topology, the heuristic is effective because a vertex's neighborhood is a statistically representative sample of its local region. Consequently, local proximity—selecting the neighbor that minimizes the distance to the query—tend to move the search toward the global target. This consistency yields an approximately monotonic trajectory and enables efficient convergence without being confounded by topological irregularities.

3.2 Topological Distortion

The topological homogeneity observed in low-dimensional settings degrades significantly under the influence of the *curse of dimensionality*. This distortion is primarily governed by the phenomenon of *distance concentration*: as $d \to \infty$, the distribution of pairwise distances becomes increasingly

concentrated, causing the ratio of standard deviation to mean, σ_Δ/μ_Δ, to converge to zero [30]. While this implies that points become metrically equidistant in a relative sense, the remaining variations in distance are not random. Instead, they are systematically dictated by the high-dimensional geometry. A seminal result by Radovanović et al. [45] demonstrates that high dimensionality disproportionately amplifies the metric advantage of points located near the geometric center of the data distribution.

Lemma 1 (Amplification of Spatial Centrality [45]). *Let x_a and x_b be two points in $\mathbb{R}^d$ drawn i.i.d. from a unimodal distribution (e.g., a Gaussian), with x_a closer to the distribution mean than x_b. Let $\bar{\delta}(x, S)$ denote the expected distance from x to all other points in S. Then the gap in their average distances, $\Delta\bar{\delta} = \bar{\delta}(x_b, S) - \bar{\delta}(x_a, S)$, is a monotonically increasing function of d for sufficiently large d.*

This lemma formalizes *spatial centrality*: points nearer the data mean are, on average, closer to other points, and this advantage strengthens with increasing d. In a KNNG, this geometric bias induces a strongly right-skewed in-degree distribution, as central points are more likely to appear in many k-NN lists. As a result, the topology departs from the homogeneous ER model and becomes polarized into high-degree hubs and near-isolated anti-hubs.

3.3 Hubness-Induced Search Failure

The topological imbalance induced by hubness invalidates the transitive proximity heuristic discussed in Section 3.1. As dimensionality increases, the graph departs from an approximately homogeneous ER-like regime and becomes increasingly centralized. In this topology, hubs act as attractors that bias greedy transitions, making local proximity a poor predictor of global progress.

3.3.1 Hub Capture. A greedy search path is susceptible to being *captured* by a hub. At each step t, the algorithm selects the next vertex $x_{p_{t+1}}$ from the current neighborhood $N(x_{p_t})$ according to local proximity. A hub vertex x_h functions as a topological attractor for this process due to the combination of high spatial centrality (a geometric property) and high in-degree (a topological property).

Theorem 1 (Hub Capture). *In a KNNG exhibiting hubness, let x_{p_t} denote the current vertex in a greedy traversal. Conditioned on a hub $x_h \in N(x_{p_t})$, the probability of selecting it next, $P(x_{p_{t+1}} = x_h \mid x_h \in N(x_{p_t}))$, approaches 1 as the spatial centrality of x_h increases relative to the other candidates in $N(x_{p_t})$.*

Proof Sketch. The event $x_{p_{t+1}} = x_h$ occurs when $\delta(q, x_h)$ is smaller than $\delta(q, x_v)$ for all other candidates $x_v \in N(x_{p_t}) \setminus \{x_h\}$. Due to its high spatial centrality, $\delta(q, x_h)$ is drawn from a distribution with a smaller mean than those of less-central neighbors. As d increases, Lemma 1 amplifies this mean separation, while distance concentration shrinks the variance, so x_h becomes the argmin with probability approaching one. Finally, the hub's high in-degree increases the likelihood that x_h appears in $N(x_{p_t})$, making capture prevalent. A formal proof is provided in Appendix A.1[1]. □

3.3.2 Anti-hub Isolation. Hub capture exacerbates the isolation of anti-hubs. Let x_a denote an anti-hub that is the true nearest neighbor of a query q. By definition, x_a has negligible in-degree $N_k(x_a)$, and thus a small gateway set $V_{\text{gate}} = \{x_v \in V \mid (x_v, x_a) \in E\}$. Therefore, a greedy search can reach x_a only if its trajectory intersects V_{gate}.

Theorem 2 (Anti-hub Isolation). *Let x_a be an anti-hub with gateway set V_{gate}. For a greedy search path $P = (x_{p_0}, x_{p_1}, \ldots, x_{p_T})$ of length T, the probability of hitting V_{gate}, $P(\exists t \le T : x_{p_t} \in V_{gate})$, approaches 0 as the skewness of the in-degree distribution S_{N_k} increases.*

Proof Sketch. We model the greedy search as a stochastic process on the graph G. The premise of high in-degree skewness (S_{N_k}) implies a high variance in spatial centrality across vertices (per Lemma 1). This variance introduces a systematic bias into the transition probabilities at each step, heavily favoring transitions toward high-centrality hubs. This functional bias partitions the graph into a strongly attractive core (hubs) and a weakly connected periphery (where V_{gate} resides). For a search of finite length, the probability of navigating into the low-density gateway set before being absorbed by a hub's basin of attraction vanishes as skewness intensifies. This dynamic renders the anti-hub topologically unreachable via greedy traversal. (See Appendix A.2[1] for the formal derivation.) □

4 Revisiting Graph-based Algorithms through the Lens of Hubness

Building on Section 3, we revisit mainstream graph-based ANNS algorithms through the lens of hubness. We argue that their effectiveness depends largely on how they (often implicitly) counteract hubness-induced distortions during index construction. As summarized in Figure 4, we organize existing methods by increasing mitigation complexity and analyze how their designs shape two core stages: *candidate acquisition* (neighbor discovery) and *neighbor selection* (edge pruning). This perspective yields a unified framework that traces the progression from topological diversification, to parameterized pruning, and finally to data-driven adaptivity.

4.1 Topological Diversification

HNSW (Hierarchical Navigable Small World graph) [40] and NSG (Navigating Spreading-out Graph) [17] constitute the foundational tier of our hubness mitigation taxonomy (Figure 4). Their primary design objective is to enforce local topological diversification. While often framed in literature as a means to improve graph navigability via *small-world* shortcuts [43, 55], we reinterpret their neighbor selection mechanism as a direct, albeit implicit, countermeasure to hub capture, functioning by breaking the edge concentration that naturally accumulates around high-centrality nodes.

4.1.1 Core Mechanism. Both HNSW and NSG employ a pruning heuristic derived from the Monotonic Relative Neighborhood Graph (MRNG) [17, 24, 54]. For a node x_p and a candidate neighbor x_i, the edge (x_p, x_i) is retained only if no existing neighbor $x_j \in N(x_p)$ provides a shorter route to x_i, i.e.,

$$\forall x_j \in N(x_p), \quad \delta(x_p, x_i) < \delta(x_j, x_i). \tag{3}$$

HNSW applies this rule online during incremental, layer-wise construction: after collecting candidates via greedy search, it prunes them to avoid redundant connections within a single dense cluster. NSG applies the same principle offline as a refinement step: starting from an approximate KNNG, it prunes each node's candidate pool to break dense edge concentrations around hubs and improve angular coverage.

4.1.2 Hubness Mitigation Analysis. The MRNG rule counteracts the amplification of spatial centrality (Lemma 1) by suppressing redundant connections to hubs. In a standard KNNG, a hub attracts edges largely because it is close to many points (i.e., it frequently minimizes $\delta(x_p, x_{hub})$). Under MRNG pruning, this distance advantage is insufficient: even when x_{hub} is close to x_p, the edge is removed if an existing neighbor $x_j \in N(x_p)$ provides a shorter route to the hub (i.e., $\delta(x_j, x_{hub}) < \delta(x_p, x_{hub})$). This *occlusion* rule filters geometrically redundant edges, limits hub in-degree growth, and promotes spatially diverse neighborhoods. As a consequence, the in-degree distribution becomes less skewed, reducing hub-induced trapping and improving reachability of peripheral regions.

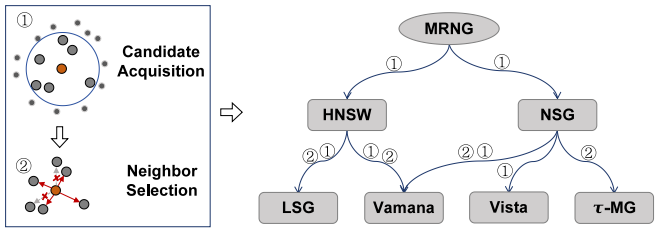


Fig. 4. A unified taxonomy of graph-based ANNS algorithms from the perspective of hubness mitigation. The framework illustrates progressive refinements in ① candidate acquisition and ② neighbor selection.

Table 1. Hubness mitigation strategies in representative graph-based ANNS algorithms.

Strategy	Algorithm	Core Mechanism	Hubness Mitigation Interpretation	Mitigation Focus
Topological Diversification	HNSW	Incremental Build with MRNG Pruning	Enforces spatial diversity of neighbors via MRNG rule, preventing over-connection to hubs	Hub Capture
	NSG	KNNG Refinement with MRNG Pruning	Applies MRNG rule to diversify a dense graph, dismantling clusters around existing hubs	Hub Capture
Parameterized Pruning	Vamana	Relaxed MRNG Rule (α-parameter)	Retains long-range edges to create navigational shortcuts to distant regions	Anti-hub Isolation
	τ-MG	Relaxed MRNG Rule (τ-monotonicity)	Ensures guaranteed traversal progress per hop, creating reliable paths to remote nodes	Anti-hub Isolation
Data-Driven Adaptivity	LSG	Local Density-based Distance Scaling	Reshapes metric space to make anti-hubs more competitive and hubs less competitive in selection	Hub Capture & Anti-hub Isolation
	Vista	Connectivity-based Resource Allocation	Boosts candidate search for anti-hubs and actively manages high-degree node connections	Hub Capture & Anti-hub Isolation

4.2 Parameterized Pruning

Vamana [26] and the τ-monotonic graph (τ-MG) [44] form the second tier of our taxonomy (Figure 4). They move beyond strict diversification by introducing parameterized pruning rules. This shift is consequential: rather than solely suppressing hubs through occlusion, these methods relax pruning to preserve long-range edges, thereby mitigating anti-hub isolation (Theorem 2) and improving reachability of sparsely connected regions.

4.2.1 Core Mechanism. Vamana implements parameterized pruning through a two-stage construction governed by a relaxation parameter $\alpha \geq 1$. In the first stage ($\alpha = 1$), it applies strict MRNG pruning to obtain a locally diversified graph. In the second stage ($\alpha > 1$), it relaxes occlusion: the edge (x_p, x_i) may be retained even when a neighbor x_j is closer to x_i, provided that

$$\delta(x_p, x_i) < \alpha \cdot \delta(x_j, x_i). \tag{4}$$

This multiplicative relaxation preserves long-range edges that would otherwise be removed by local neighbors. In contrast, τ-MG uses an additive relaxation motivated by τ-monotonicity. It retains (x_p, x_i) as long as the detour penalty is bounded by

$$\delta(x_p, x_i) < \delta(x_j, x_i) + 3\tau, \quad \tau \geq 0. \tag{5}$$

This criterion enforces a minimum rate of progress per hop, yielding traversal paths that are less susceptible to local detours.

4.2.2 Hubness Mitigation Analysis. Both methods mitigate anti-hub isolation by relaxing MRNG pruning to preserve long-range edges that connect vertices to sparsely connected regions. Despite this shared objective, their mechanisms differ. Vamana's multiplicative relaxation (α) preferentially retains long hops that act as navigational shortcuts, reducing the number of steps required to reach

isolated nodes. By contrast, τ-MG's additive relaxation (τ) is coupled with a progress guarantee, biasing the traversal toward consistent distance reduction and lowering the risk of stalling in local optima before reaching an anti-hub.

4.3 Data-Driven Adaptivity

LSG (Local Scaling Graph) [50] and Vista [18] comprise the third tier of our taxonomy (Figure 4), shifting from uniform geometric heuristics to data-driven adaptivity. Rather than applying a single pruning rule globally, these methods exploit per-node proxies inferred from the data (e.g., local density or preliminary connectivity) to modulate construction decisions. This design explicitly recognizes spatial heterogeneity in high-dimensional data and allocates connectivity accordingly.

4.3.1 Core Mechanism. LSG mitigates hubness by re-scaling the metric using a local-density proxy, since anti-hubs tend to lie in sparse regions. For each point x_i, it computes the average k-NN distance μ_i and defines a locally scaled distance

$$\delta_{ls}(x_i, x_j) = \delta(x_i, x_j) \cdot \left(\frac{\hat{\mu}^2}{\mu_i \mu_j} \right)^{\alpha}, \tag{6}$$

where $\hat{\mu}$ is the global mean of μ_i. This transformed metric then governs candidate acquisition and pruning, reducing density-induced bias in neighbor selection.

By contrast, Vista adopts connectivity-guided resource allocation. It uses the in-degree of an initial approximate KNNG as a proxy for hubness and expands the candidate budget for low-connectivity nodes (potential anti-hubs) via

$$C_i = L + (L - \min(L, N_k(x_i))), \tag{7}$$

where L is a baseline budget. It further applies dynamic edge exchange to reroute outgoing edges from high-degree nodes, limiting their centralizing effect while preserving global connectivity.

4.3.2 Hubness Mitigation Analysis. Both methods mitigate hubness by adapting construction to local data characteristics. LSG intervenes at the metric level: its local scaling reduces density-induced distance disparities (Lemma 1), which in turn suppresses the emergence of extreme in-degree skew. Vista intervenes at the topology level: using preliminary connectivity as feedback, it selectively strengthens anti-hub gateway sets (Theorem 2) while constraining the influence of hubs (Theorem 1). Together, these designs highlight a common principle: effective mitigation requires non-uniform construction rules that reflect the inherent heterogeneity of high-dimensional data.

4.4 Summary

Table 1 summarizes mainstream graph-based ANNS designs within a taxonomy of hubness mitigation. The taxonomy highlights an increasing level of control over topological skew:

- **Topological Diversification**. HNSW and NSG use rigid geometric occlusion to implicitly suppress hub formation and maintain basic navigability.
- **Parameterized Pruning**. Vamana and τ-MG relax occlusion to preserve long-range shortcuts, improving reachability of isolated anti-hubs.
- **Data-Driven Adaptivity**. LSG and Vista use distributional proxies to modulate construction decisions, enabling fine-grained and asymmetric topology control.

This perspective clarifies the design logic of existing methods and motivates the next step: explicit hubness-aware optimizations that leverage direct topological signals to refine the index.

Table 2. Overview of benchmark datasets.

Dataset	Size	Dim.	#Queries	Type
SIFT[2]	1M~100M	128	10,000	Image Features (SIFT)
LAION[3]	1M~100M	768	10,000	Image Embeddings (CLIP)
GIST[4]	1M	960	1,000	Image Features (GIST)
DBPedia[5]	0.99M	1,024	10,000	Text Embeddings (OpenAI)
Yelp[6]	1M	3,072	10,000	Text Embeddings (Gemini)

5 Experimental Evaluation

This section empirically validates the theoretical framework and mitigation strategies established in Sections 3 and 4. We structure our evaluation around four research questions:

- **RQ1:** How do intrinsic data properties (e.g., dimensionality) influence the emergence and severity of topological skew in proximity graphs? (Section 5.2)
- **RQ2:** To what extent does topological imbalance explain the performance degradation incurred by outlier queries? (Section 5.3)
- **RQ3:** Do current mitigation strategies derive their efficiency gains from the restoration of topological balance? (Section 5.4)
- **RQ4:** Can we leverage explicit hubness signals to refine graph topology and optimize search performance? (Section 5.6)

5.1 Experimental Setup

5.1.1 Evaluation Environment. To ensure fair and reproducible comparisons, we implemented all methods within a unified evaluation testbed. All experiments were run on a dedicated server equipped with an Intel Xeon Platinum 8375C CPU (2.90 GHz) and 1 TB of RAM. The codebase was compiled with GCC 13.3.0 under the `-O3` optimization flag. To eliminate I/O effects, both index construction and search were executed fully in memory, using 128 threads for indexing and 10 parallel threads for query execution.

5.1.2 Datasets. We evaluate five public benchmark datasets (Table 2), with sizes ranging from 1 M to 100 M vectors. The collection includes classic computer-vision descriptors (SIFT and GIST) as well as modern foundation-model embeddings (LAION, DBPedia, and Yelp). Together, these datasets cover a broad spectrum of intrinsic properties, with ambient dimensionalities from 128 to 3,072, enabling evaluation across heterogeneous data distributions.

A key objective of our analysis is to characterize performance heterogeneity across queries, emphasizing topologically challenging cases. To this end, we partition each workload based on inherent geometric difficulty rather than algorithm-specific artifacts. We first construct a control baseline: a standard KNNG generated via brute force, with no heuristic pruning or hubness-mitigation mechanisms, and run the full query set on this vanilla index. We then label the slowest 50% of queries (by search latency) as *outlier queries* (OQ) and the remaining 50% as *typical queries* (TQ). This procedure isolates query difficulty attributable to the underlying data geometry.

[2]http://corpus-texmex.irisa.fr/
[3]https://sisap-challenges.github.io/2024/datasets/
[4]https://www.cse.cuhk.edu.hk/systems/hash/gqr/datasets.html
[5]https://huggingface.co/datasets/filipecosta90/dbpedia-openai-1M-text-embedding-3-large-1024d
[6]https://huggingface.co/datasets/okay124/yelp-review-gemini-embedding

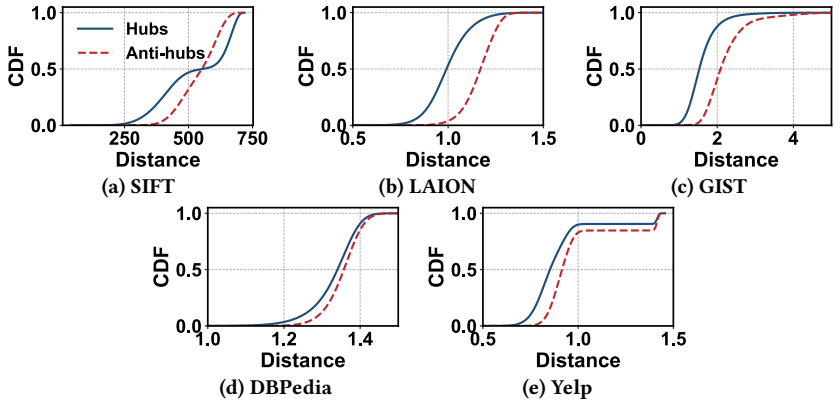


Fig. 5. Distance distributions for hubs versus anti-hubs. The CDF illustrates the L_2 distances to randomly sampled reference points. Notably, hubs consistently exhibit smaller distance compared to anti-hubs.

5.1.3 Compared Algorithms. We evaluate representative graph methods that cover the main hubness-mitigation strategies:

- **HNSW** [40][7] and **NSG** [17][8]. They use implicit geometric occlusion to suppress hub formation.
- **Vamana** [26][9] and τ-**MG** [44][10]. They utilize parameterized pruning to improve anti-hub reachability.
- **LSG** [50][11] and **Vista** [18][12]. They leverage distributional proxies to dynamically modulate the topology.

We use publicly available implementations whenever possible and integrate them into a unified testbed. Since Vista has no public codebase, we re-implemented it according to the paper on top of NSG. Unless stated otherwise, we set the maximum out-degree to $R = 32$ and the number of returned results to $k = 10$; all remaining hyperparameters follow the official recommended settings.

5.1.4 Evaluation Metrics. We evaluate search quality, efficiency, and the resulting index topology. Unless stated otherwise, all experiments use Euclidean (L_2) distance, following common practice [5, 32, 54]. Our findings are not specific to L_2, since maximum inner product search (MIPS) and cosine similarity can be reduced to nearest-neighbor search in a metric space via standard transformations [10, 20]. **(i)** *Search quality.* We report Recall@k, defined as the fraction of the true top-k neighbors retrieved. We also report the average distance ratio (ADR), computed as the mean distance of retrieved neighbors divided by that of the ground-truth neighbors; values closer to 1 indicate better approximations. **(ii)** *Search efficiency.* We report the number of distance computations (NDC) as an implementation-independent proxy for computational cost [16].We further report query latency (mean and tail, including P99.9) and throughput (queries per second, QPS). We also record hops, i.e., the number of edge traversals per query. **(iii)** *Index topology.* To quantify hubness, we compute k-occurrence (N_k), i.e., the number of times a point appears in other points' k-NN lists (equivalently, its in-degree). We additionally report the skewness S_{N_k} (the standardized third moment of the N_k distribution), where larger positive values indicate stronger hub domination.

[7]https://github.com/nmslib/hnswlib
[8]https://github.com/ZJULearning/nsg
[9]https://github.com/microsoft/DiskANN
[10]https://github.com/csypeng/taumng_sigmod
[11]https://github.com/19-hanhan/LSG
[12]https://github.com/YujianFu97/Vista-Technical-Report

Table 3. LID statistics across benchmark datasets.

	SIFT	LAION	GIST	DBPedia	Yelp
25%	15.18	17.26	34.05	21.15	26.51
50%	19.00	24.02	46.49	27.25	35.64
75%	23.22	33.22	57.24	34.13	44.54
Mean	**19.42**	**27.52**	**45.62**	**27.96**	**35.84**

5.2 Intrinsic Data Properties

We establish that hubness is not merely an algorithmic artifact but an intrinsic property of the underlying data manifold. We support this claim through two complementary analyses: (i) we use local intrinsic dimensionality (LID) to quantify manifold complexity and relate it to hubness severity; and (ii) we examine the geometric disparity between hubs and anti-hubs to empirically corroborate the amplification of spatial centrality (Lemma 1).

5.2.1 Intrinsic Dimensionality Analysis. A dataset's susceptibility to hubness is primarily determined by the intrinsic dimensionality of its underlying manifold rather than by its ambient dimension. We quantify intrinsic dimensionality using Local Intrinsic Dimensionality (LID), estimated with the Maximum-Likelihood Estimator (MLE) [2, 3]. For a point x, LID is defined as

$$\widehat{\mathrm{LID}}(x) = \left(-\frac{1}{k} \sum_{i=1}^{k} \log \frac{r_i(x)}{r_k(x)} \right)^{-1}, \tag{8}$$

where $r_i(x)$ is the distance from x to its i-th nearest neighbor. Larger LID means sparser neighborhoods [2], worsening topological skew.

Table 3 reveals a complexity ordering that is not aligned with ambient dimensionality. For example, GIST (ambient $d = 960$) has a substantially higher mean LID (45.62) than Yelp (ambient $d = 3072$, mean LID 35.84). This observation supports our claim that intrinsic dimensionality, rather than ambient dimension, better captures the geometric distortion that drives hubness. Beyond the mean, heterogeneity also matters: GIST exhibits both the highest mean LID and the largest interquartile range (IQR = 23.19), indicating pronounced non-uniformity across the manifold, whereas SIFT has a lower mean LID (19.42) and a smaller IQR (8.04). Such heterogeneity naturally separates the manifold into dense, low-LID regions (hubs) and sparse, high-LID regions (anti-hubs) [49].

5.2.2 Geometric Disparity of Hubs and Anti-hubs. To empirically validate the spatial-centrality principle, we analyze distance distributions for topologically distinct nodes. We partition each dataset by k-occurrence, selecting the top 0.01% as hubs and the bottom 0.01% as anti-hubs, and compute their distances to a random sample of 100,000 reference points.

As shown in Figure 5, the Cumulative Distribution Functions (CDFs) exhibit a clear geometric stratification. The hub distance profiles (solid blue lines) are consistently shifted left relative to those of anti-hubs (dashed red lines), indicating smaller mean distances. This confirms that hubs are geometrically more central, whereas anti-hubs lie closer to the periphery. Such a positional difference provides a geometric basis for in-degree skew: central points are more likely to "capture" greedy search trajectories. The magnitude of the separation correlates with intrinsic complexity. On the high-LID GIST dataset (Figure 5c), the gap is stark: the hub median distance is smaller than the anti-hub 25th-percentile distance. In contrast, on the low-LID SIFT dataset (Figure 5a), the CDFs are crossed with each other, indicating a weaker geometric hierarchy. Overall, these results

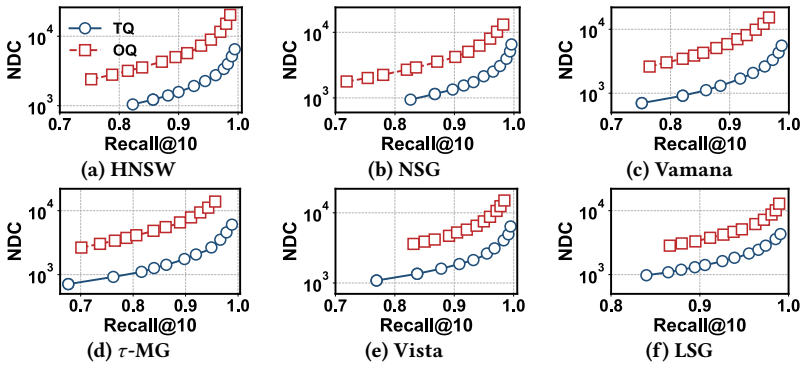


Fig. 6. Performance gap between Outlier Queries (OQ) and Typical Queries (TQ) on the GIST dataset.

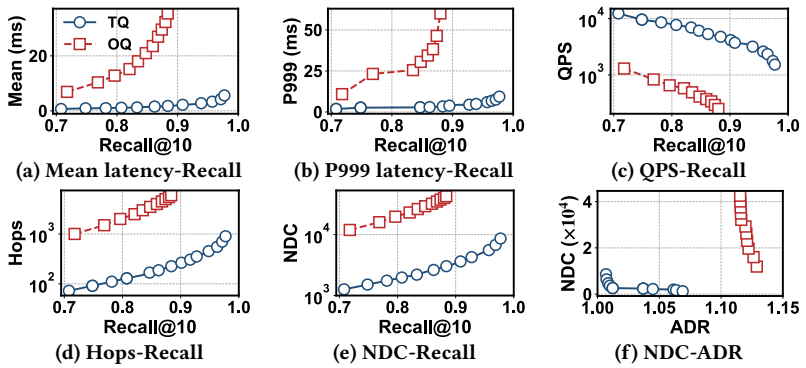


Fig. 7. Multi-dimensional performance analysis of search on a baseline KNNG for the GIST dataset.

support our hypothesis that intrinsic manifold complexity—in particular, heterogeneity—induces a center–periphery structure that impairs graph traversal.

5.3 Impact of Hubness on Search Performance

We evaluate the impact of hubness on search performance, focusing on two key dimensions: the stability of query performance and the efficacy of mitigation strategies.

5.3.1 Performance of Outlier Queries. We first quantify the performance instability induced by topological skew. Figure 6 illustrates the NDC-Recall trade-off for typical (TQ) versus outlier (OQ) queries on GIST, revealing a pervasive "stability gap" across all algorithms. Achieving high recall for OQs necessitates a disproportionate computational budget—often an order of magnitude higher than for TQs. For instance, with HNSW (Figure 6a), 90% recall costs ≈1,500 NDCs for TQs but surges to over 5,000 for OQs. This consistent disparity identifies hubness as a factor that compromises performance stability, even in state-of-the-art methods.

To dissect the sources of this instability, we analyze an unoptimized KNNG baseline. Results on GIST (Figure 7) show that OQs degrade performance across all metrics. User-facing metrics deteriorate markedly: both mean and P999 latencies increase substantially for OQs (Figures 7a and 7b), which in turn reduces throughput (QPS; Figure 7c) and threatens stringent service-level objectives (SLOs). Internal metrics expose the underlying cause. To achieve comparable recall, OQs require longer graph traversals (hops; Figure 7d) and more distance computations (NDC; Figure 7e). Moreover, the penalty extends to result quality: at the same computational cost, OQs yield a higher

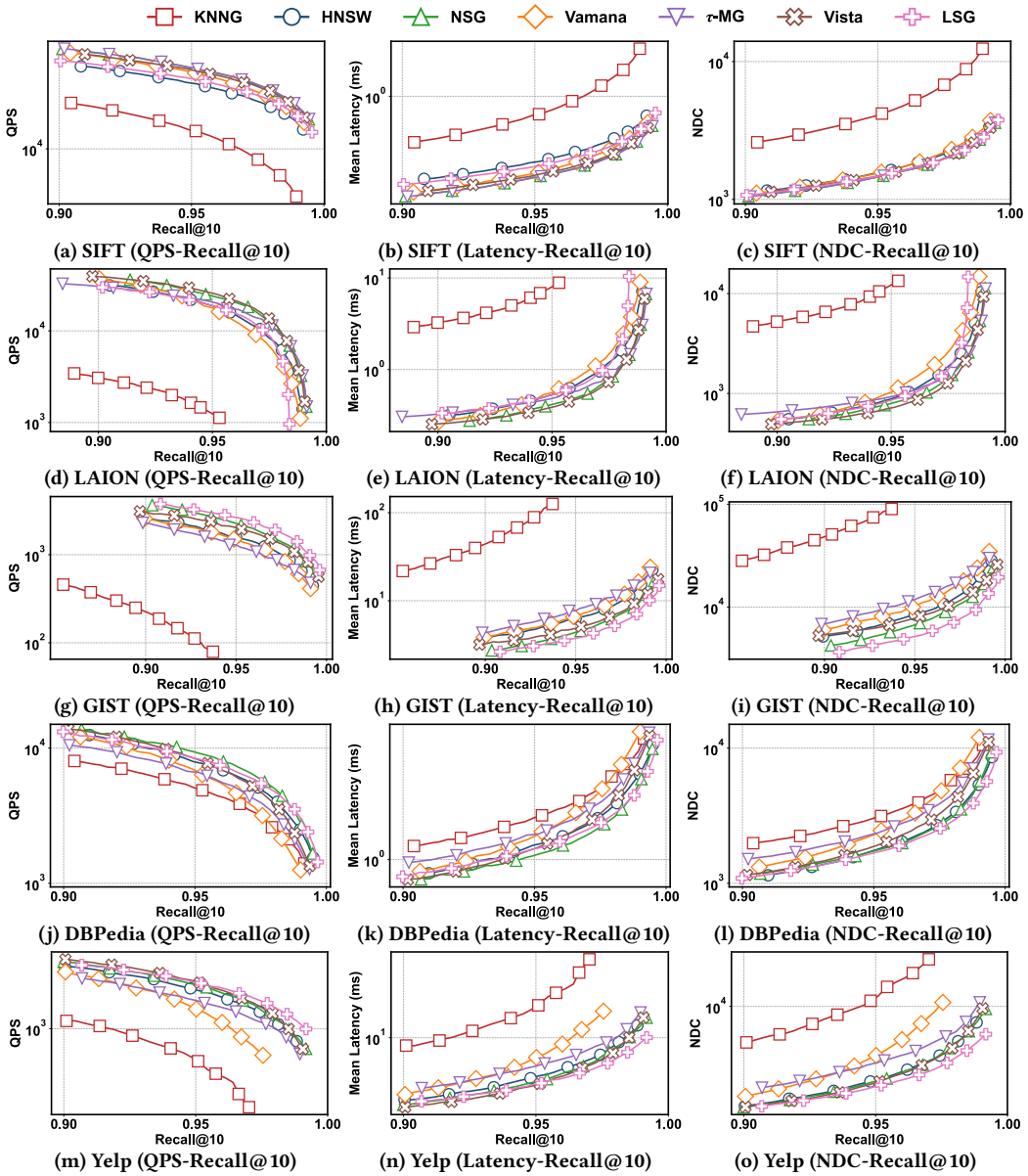


Fig. 8. Search performance comparison of graph-based algorithms across million-scale datasets.

average distance ratio (ADR), indicating a poorer approximation of the true nearest neighbors (Figure 7f). These results confirm that greedy traversal becomes both inefficient and inaccurate when navigating toward OQs.

5.3.2 Efficacy of Mitigation Strategies. We assess the effectiveness of various mitigation strategies in reducing performance instability, with a particular focus on outlier queries. Figure 8 summarizes the performance on the 1M-scale datasets, plotting *throughput (QPS)*, *mean latency*, and *NDC* against recall.

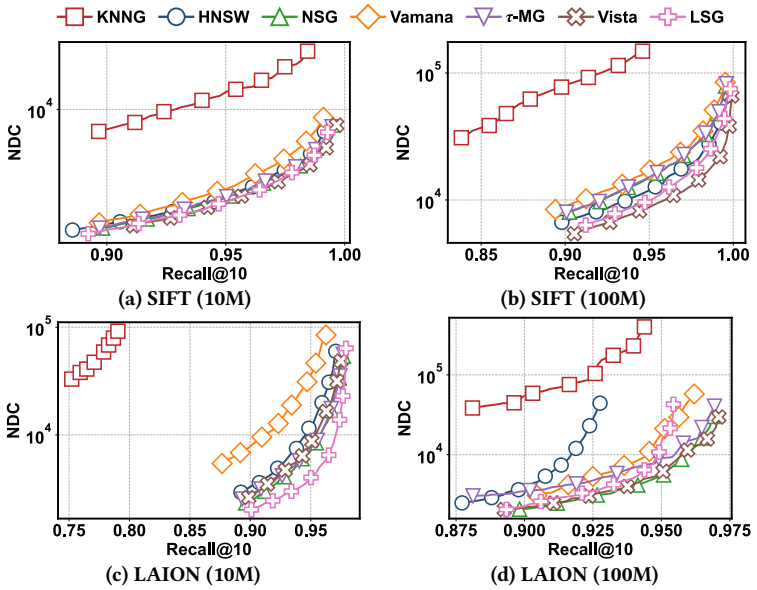


Fig. 9. NDC-Recall performance on larger datasets.

A performance hierarchy emerges, the distinctness of which is governed by both the sophistication of the mitigation strategy and the inherent complexity of the dataset. First, across all datasets and metrics, the KNNG (red squares) consistently exhibits the worst performance. This significant disparity highlights that unmitigated hubness presents a topological barrier to search efficiency; without algorithmic intervention to optimize the index, achieving high recall incurs prohibitive computational costs. Second, among the graph-based algorithms, a positive correlation is observed between the adaptability of the mitigation strategy and search efficiency, although the magnitude of this improvement is dataset-dependent.

On datasets with lower intrinsic dimensionality, such as SIFT (Figures 8a–c), the performance curves of different algorithms are tightly clustered. This indicates that when hubness is mild, the specific choice of mitigation strategy yields only marginal differences, and data-driven adaptivity provides limited benefits over standard topological diversification (e.g., HNSW and NSG). In contrast, on topologically complex datasets such as GIST (Figures 8g–i) and Yelp (Figures 8m–o), the hierarchy becomes pronounced. Here, data-driven algorithms (LSG and Vista) demonstrate a clear advantage over static methods, achieving materially higher QPS and lower latency at high recall levels (e.g., 99%). This divergence underscores that static geometric heuristics become insufficient as intrinsic complexity increases. The superior performance of LSG and Vista on complex workloads can be attributed to their capacity to adaptively modify graph topology.

Our large-scale evaluation (up to 100M vectors; Figure 9) reveals that scaling amplifies topological imbalance and exposes performance differences that are less apparent at 1M. For example, on SIFT, methods with similar performance at 1M diverge significantly at 100M (Figure 9b), indicating that even mild irregularities accumulate as graph size increases. Additionally, the KNNG remains the least efficient at all scales, reinforcing that unmitigated hubness is a scalable bottleneck that requires explicit mitigation.

5.4 Topological Analysis of Graph Indexes

We analyze the index topology to answer whether performance improvements can be attributed to hubness mitigation.

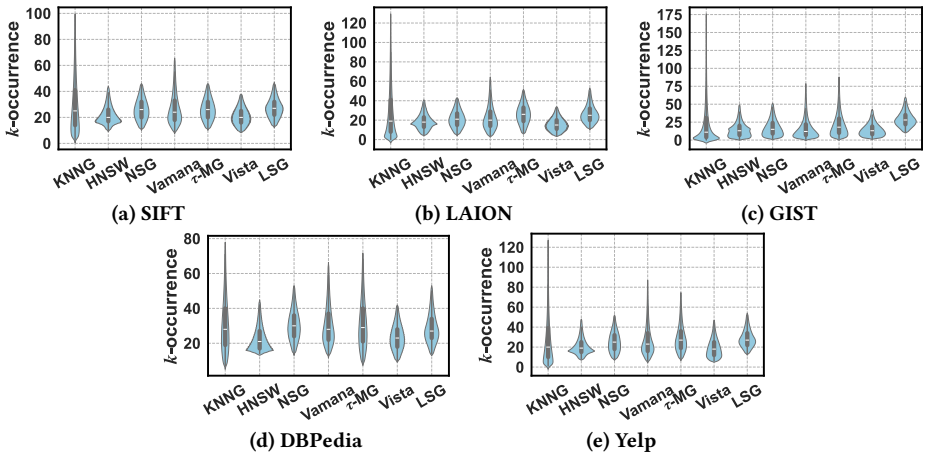


Fig. 10. In-degree (k-occurrence) distributions across five datasets, visualized as violin plots. The plots illustrate the topological balance achieved by each algorithm; narrower, less skewed distributions indicate more effective hubness mitigation.

Table 4. In-degree skewness (S_{N_k}) comparison across the evaluated graph indexes. Lower values indicate a more balanced topology.

Dataset	KNNG	HNSW	NSG	Vamana	τ-MG	Vista	LSG
SIFT (1M)	1.02	0.85	0.27	0.96	0.27	0.35	0.31
SIFT (10M)	0.94	1.00	0.31	0.79	0.31	0.29	0.36
SIFT (100M)	0.96	1.07	0.34	0.88	0.34	0.32	0.43
LAION (1M)	1.32	0.54	0.29	0.90	0.18	0.42	0.64
LAION (10M)	1.41	0.56	0.32	0.92	0.24	0.49	0.64
LAION (100M)	1.46	0.58	0.51	1.03	0.49	0.75	0.58
GIST	2.15	0.98	0.81	1.68	1.31	0.84	0.59
DBPedia	0.68	0.99	0.34	0.82	0.62	0.30	0.57
Yelp	1.44	0.99	0.37	1.29	0.90	0.68	0.57

5.4.1 In-Degree Distribution. We examine in-degree distributions, which directly quantify hubness in the graph topology. To enable stable comparisons across methods, we reduce sensitivity to extreme values by trimming the top and bottom 3% of nodes by in-degree in each graph before computing skewness.

Figure 10 and Table 4 show a clear trend toward stronger topological control. The KNNG exhibits highly skewed in-degrees (e.g., 2.15 on GIST), indicating that naive construction largely inherits the dataset's geometric imbalance. HNSW and NSG substantially reduce skewness on GIST to 0.98 and 0.81 via MRNG-style pruning. As summarized in Table 5, this improved balance is achieved with sparser graphs, suggesting that pruning primarily removes redundant edges that over-connect hubs. Vamana and τ-MG introduce a different trade-off: by relaxing the MRNG criterion to retain longer-range edges, they improve anti-hub reachability but may suppress hubs less aggressively (e.g., Vamana reaches 1.68 on GIST). In contrast, LSG and Vista employ data-driven strategies to adaptively regulate topology; for instance, LSG reduces GIST skewness to 0.59, aligning with its superior search performance. However, skewness reduction does not translate directly into performance gains. This is due to the tension between in-degree skewness and local connectivity: excessive hub suppression may remove locally valuable edges and hinder greedy traversal. Thus,

Table 5. Comparison of the mean out-degrees across the evaluated graph indexes, with the maximum degree parameter (R) set to 32 for all graphs.

Dataset	KNNG	HNSW	NSG	Vamana	τ-MG	Vista	LSG
SIFT	32	22	26	28	26	21	27
LAION	32	19	22	24	27	16	27
GIST	32	16	18	20	25	15	30
DBPedia	32	23	31	32	32	24	29
Yelp	32	19	22	24	27	16	27

Table 6. Mean in-degree (k-occurrence) of ground-truth results for the Typical (TQ) and Outlier (OQ) queries.

	KNNG		HNSW		NSG		Vamana		τ-MG		Vista		LSG	
Dataset	TQ	OQ	TQ	OQ	TQ	OQ	TQ	OQ	TQ	OQ	TQ	OQ	TQ	OQ
SIFT	3.41	1.01	4.26	4.10	6.70	7.08	1.00	1.00	6.69	7.10	5.00	5.23	6.90	8.53
LAION	0.42	0.00	2.69	1.65	3.86	1.39	1.67	0.00	5.20	1.57	2.85	1.67	4.92	4.47
GIST	5.93	1.63	5.29	3.16	7.52	4.77	4.58	2.46	8.96	5.60	6.73	4.70	9.86	9.72
DBPedia	4.35	2.79	8.44	8.95	7.93	8.65	7.11	7.35	4.72	3.90	5.12	5.51	6.55	6.43
Yelp	1.6	0.39	4.83	3.06	5.49	3.47	3.94	2.49	6.50	3.89	5.81	5.38	1.90	0.78

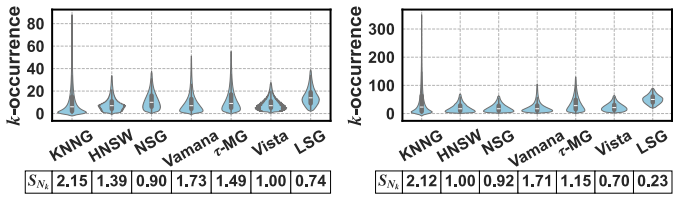
effective designs must balance global in-degree skewness with local connectivity. Beyond intrinsic complexity, data scale further exacerbates topological skew. As shown in Table 4, the KNNG on LAION becomes increasingly imbalanced as the dataset grows (1.32 at 1M to 1.46 at 100M). This trend extends even to datasets with inherently mild hubness; on SIFT, methods such as HNSW and τ-MG exhibit increased skewness at larger scales, indicating that topological irregularities naturally accumulate with graph size.

5.4.2 Ground-Truth Target Connectivity. To investigate the structural causes of outlier queries, we examine the connectivity of their ground-truth nearest neighbors. Table 6 highlights the most challenging cases by reporting the average in-degree of the 100 ground-truth nearest neighbors with the lowest connectivity.

The results generally support the connection between outlier queries and topological isolation, particularly in the unoptimized baseline. In the case of KNNG, the ground-truth neighbors for OQs exhibit negligible in-degrees (e.g., 0.00 on LAION), rendering them structurally invisible to greedy traversal. Advanced algorithms address this by strengthening these weak points; for example, LSG increases the mean OQ target in-degree on LAION from 0.00 to 4.47. Crucially, instances where OQ targets exhibit higher in-degrees (e.g., HNSW on DBpedia) highlight that query efficiency is determined by *topological reachability*, not merely raw connectivity. A target with a higher in-degree remains isolated if its incoming edges connect exclusively to other inaccessible anti-hubs. In contrast, a low-degree target becomes accessible if connected to a central hub that serves as a navigational pathway. Therefore, mitigating isolation requires not just increasing edge counts, but also establishing effective search paths from the graph's navigable core.

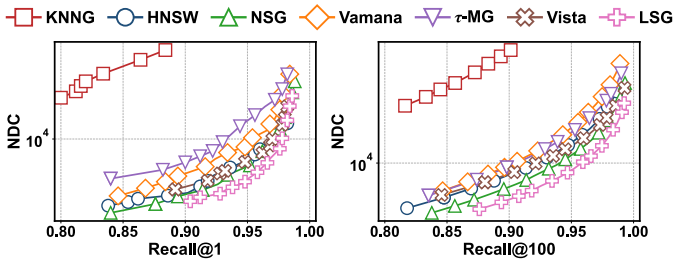
5.5 Parameter Sensitivity Analysis

We analyze the sensitivity of our results to the maximum out-degree (R) and the number of returned neighbors (k). We show that the performance hierarchy is driven by intrinsic hubness-mitigation designs rather than specific parameter choices.



(a) $R = 16$ **(b)** $R = 64$

Fig. 11. Impact of maximum out-degree (R) on the in-degree (k-occurrence) distribution for all evaluated algorithms on the GIST dataset.



(a) top-1 **(b) top-100**

Fig. 12. NDC-Recall performance on the GIST dataset for different numbers of requested neighbors (k).

5.5.1 Sensitivity to Graph Density (R). The maximum out-degree R directly controls the density of the graph index. We evaluate its impact on the topological properties of graphs constructed on the GIST dataset. As shown in Figure 11, increasing R from 16 to 64 leads to an overall reduction in in-degree skewness across all algorithms; for instance, the skewness of the HNSW index decreases from 1.39 to 1.00. This result indicates that constructing a denser graph can partially alleviate topological imbalance. However, the relative effectiveness of the mitigation approaches remains consistent. Even at higher density ($R = 64$), the data-driven algorithms (LSG, Vista) continue to produce significantly more balanced graphs than the other methods. This consistency suggests that, while density matters, the algorithmic strategy for mitigating hubness is the primary determinant of topological quality.

5.5.2 Sensitivity to Number of Neighbors (k). We verify that our conclusions hold across different search granularities by varying the number of requested neighbors, k. Our experiments confirm that the relative performance ranking among the evaluated algorithms remains stable across all tested values of k. As shown for HNSW in Figure 12, the NDC-Recall curves for $k = 1$ and $k = 100$ exhibit similar characteristics, and this pattern holds across all algorithms. On complex datasets like GIST, data-driven methods such as LSG consistently provide superior search efficiency, regardless of whether we retrieve $k = 1$ or $k = 100$ neighbors. This stability is expected. The underlying challenge—efficiently navigating a skewed graph to locate topologically isolated anti-hubs—is a structural problem whose difficulty persists irrespective of the number of neighbors being sought. Algorithms that construct more balanced graphs therefore provide a fundamental advantage for any search traversal. This consistent performance across different search granularities confirms that efficiency gains are rooted in the underlying graph topology engineered by each algorithm, rather than in the specifics of the search task.

5.6 Hubness-Aware Pruning Optimization

We propose *hubness-aware pruning* to empirically validate the potential of explicit topological correction. This lightweight post-processing strategy refines the proximity graph by leveraging

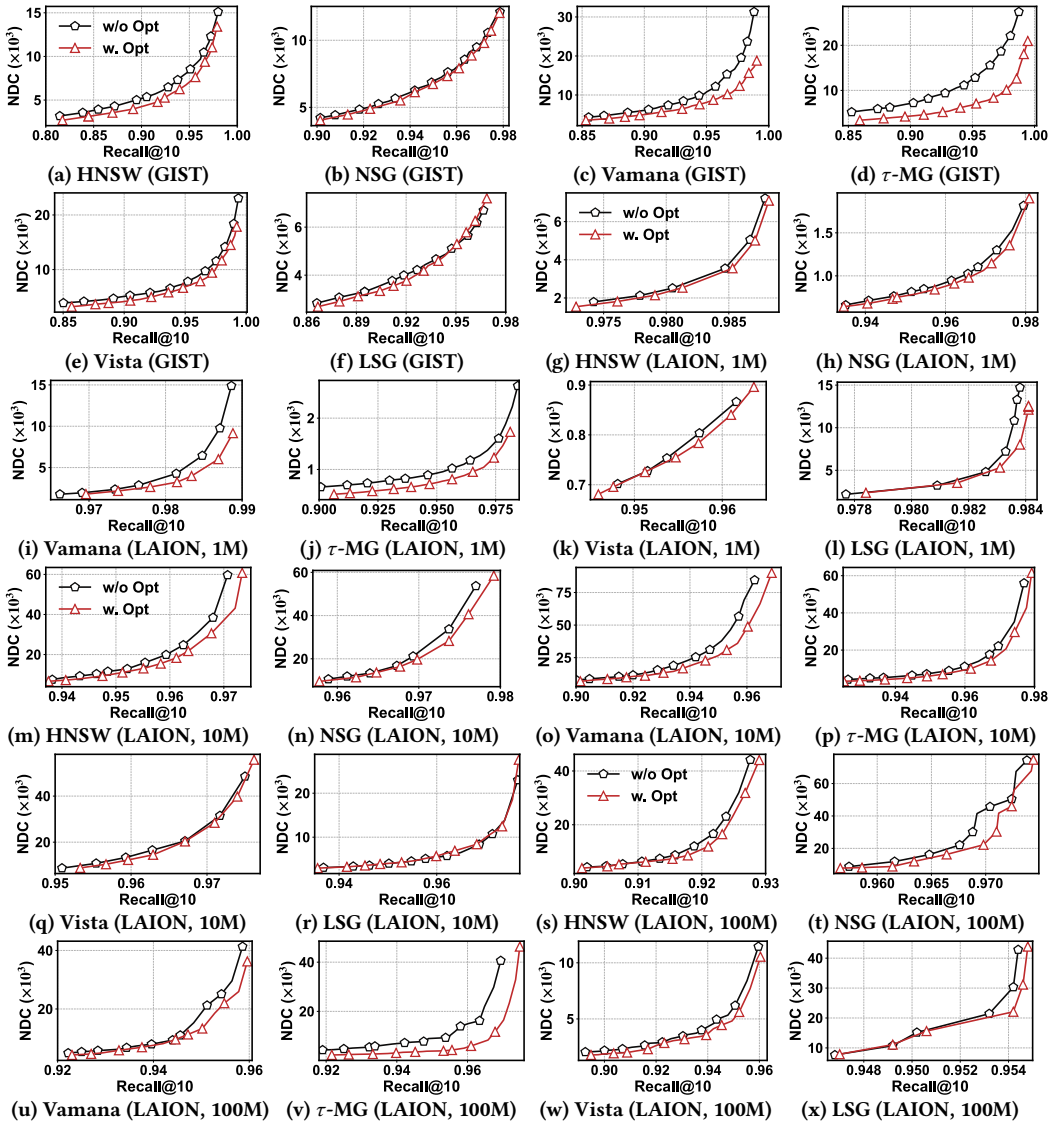
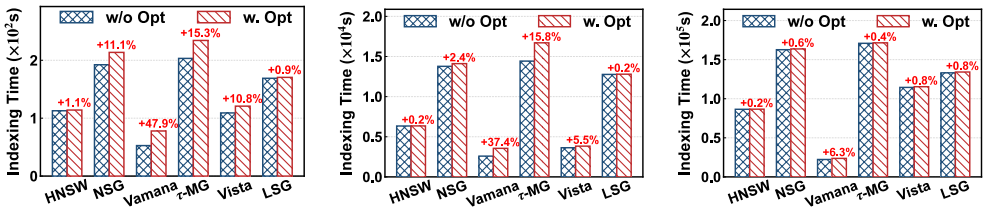


Fig. 13. Effect of hubness-aware pruning on NDC–Recall across graph-based ANNS algorithms on GIST (a–f) and LAION (g–x). Curves labeled "w. Opt" correspond to the optimized indexes and demonstrate improved search efficiency after refinement.

in-degree as a direct topological signal. Specifically, we augment the standard MRNG pruning with a dynamic factor, γ: we relax the pruning criteria ($\gamma > 1$) for low-in-degree nodes (potential anti-hubs) to foster connectivity, while tightening the criteria ($\gamma < 1$) for high-in-degree nodes (hubs) to curb over-centralization. (See our repository[1] for implementation details.)

This optimization effectively regularizes index topology. On GIST (Table 7), it consistently reduces in-degree skewness, with a pronounced effect on Vamana (1.68 → 0.92). Crucially, this efficacy extends to large-scale datasets: on LAION-100M, Vamana's skewness drops from 1.03 to 0.70. This improved balance directly translates to search efficiency. As shown in Figure 13, the NDC–Recall curves for optimized graphs shift favorably to the right, indicating reduced computational costs for



(a) LAION 1M **(b) LAION 10M** **(c) LAION 100M**

Fig. 14. Index construction time before (w/o) and after (w.) applying hubness-aware pruning.

Table 7. In-degree skewness of graphs, before (w/o) and after (w.) applying hubness-aware pruning.

		HNSW	NSG	Vamana	τ-MG	Vista	LSG
GIST	w/o	0.98	0.81	1.68	1.31	0.84	0.59
	w.	0.92	0.64	0.92	0.91	0.90	0.64
LAION (1M)	w/o	0.54	0.29	0.90	0.18	0.42	0.64
	w.	0.39	0.11	0.13	0.07	0.32	0.41
LAION (10M)	w/o	0.56	0.32	0.92	0.24	0.49	0.64
	w.	0.40	0.12	0.12	0.07	0.33	0.27
LAION (100M)	w/o	0.58	0.51	1.03	0.49	0.75	0.58
	w.	0.50	0.50	0.70	0.39	0.58	0.31

the same recall; this trend holds consistent across LAION-10M and LAION-100M. Furthermore, these gains incur negligible overhead: indexing time increases by merely +0.2% for HNSW and +0.4% for τ-MG (Figure 14c).

We observe that the magnitude of improvement is baseline-dependent: advanced methods like Vista and LSG show smaller marginal gains. This reflects the diminishing returns of post-hoc optimization, as the trade-off between suppressing hubs and preserving local greedy navigability becomes more challenging. Furthermore, as a post-processing step, this approach is strictly constrained by the connectivity of the initial graph; severely isolated anti-hubs may rarely appear in candidate pools, rendering them unrepairable. These results highlight the value of topological signals and the necessity for *hubness-native* construction strategies that structurally prevent skew accumulation during index building.

6 Discussion

This section synthesizes the primary findings of our analysis and outlines the implications for future studies.

6.1 Key Insights

6.1.1 The Connectivity-Skewness Dilemma. The design of graph-based indexes is constrained by a tension between *local connectivity*—which preserves neighborhood structure—and *global in-degree skewness*—which supports efficient global navigability. This conflict is exemplified by two boundary cases: a pure KNNG maximizes local connectivity but suffers from significant topological skew, whereas a random graph yields a more uniform in-degree distribution but lacks the nearest-neighbor structure essential for search accuracy. Consequently, modern ANNS methods occupy different points along this trade-off, each aiming to preserve local proximity while improving access to anti-hubs, thereby balancing search accuracy and efficiency.

6.1.2 Topological Isolation Drives Outlier Latency. This dilemma between local connectivity and topological uniformity manifests as the long-tail latency problem. Our findings pinpoint the cause: outlier queries are those that target topologically isolated anti-hubs. In a baseline proximity graph, these nodes have near-zero in-degrees. A greedy traversal, biased toward the gravitational pull of high-centrality hubs, cannot reliably reach them, leading to severe search inefficiency.

6.1.3 Intrinsic Complexity Governs Topological Skew. This skew is governed less by ambient dimensionality than by intrinsic manifold complexity. In particular, Local Intrinsic Dimensionality (LID) and manifold heterogeneity are key drivers of hubness. Consistent with this view, the GIST dataset—with higher mean LID and greater heterogeneity—induces substantially more pronounced in-degree imbalance than the higher-dimensional Yelp dataset. Robust index design should therefore adapt to intrinsic complexity rather than nominal dimensionality.

6.1.4 Topological Balance Predicts Performance. This insight points directly to the solution: optimizing for topological balance. Across graph-based ANNS methods, the evolution of construction heuristics reflects a steady shift toward reducing in-degree skew and improving anti-hub connectivity. Empirically, these topological indicators predict tail performance: methods that produce more balanced graphs consistently achieve lower latency on outlier queries. This reframes index design from purely geometric heuristics to explicit topological optimization.

6.1.5 The Trade-off: Hub Suppression vs. Anti-hub Reachability. Optimizing graph topology necessitates navigating a structural trade-off. As demonstrated by Vamana, strategies aimed at enhancing reachability—such as relaxing pruning criteria to connect more distant nodes—may inadvertently reinforce the prominence of central hubs. Effective mitigation of this issue demands a sophisticated edge selection policy that not only suppresses the dominance of central hubs but also ensures the establishment of reliable navigational paths to the graph's periphery, thereby maintaining global accessibility without compromising local connectivity.

6.1.6 In-Degree as a Signal for Refinement. Despite the aforementioned trade-off, direct topological control remains tractable. A node's in-degree, as an inherent property of the graph's structure, offers a valuable signal for such interventions. Our hubness-aware pruning experiment serves as a proof of concept, demonstrating the viability of this approach. The effectiveness of this post-hoc refinement provides compelling evidence that explicitly engineering for topological balance is a feasible strategy for creating more robust graph indexes.

6.1.7 Toward Implicit Topological Control. Graph-based ANNS has progressed from proximity-first designs to methods that explicitly optimize graph topology. Early approaches primarily enforced local proximity [13, 39]. Subsequent techniques—notably diversification in HNSW [40] and relaxed pruning in Vamana [26]—were introduced to improve navigability. Our analysis suggests that these mechanisms act as indirect yet effective responses to hubness: beyond adding shortcuts, they reduce hub capture and increase anti-hub reachability. This work formalizes this topological perspective and highlights directions for further algorithmic development.

6.2 Future Directions

The hubness perspective suggests several directions for future work. Below, we outline opportunities spanning index construction, query processing, storage management, and auto-tuning, with an emphasis on system-oriented implications for the database community.

6.2.1 Hubness-Native Index Construction. We advocate a shift from post-hoc hubness mitigation to *hubness-native* index construction. Existing pipelines largely induce topology as a byproduct of geometry-driven rules, treating degree skew as a downstream artifact. A more direct approach

is to incorporate topological objectives during construction. For example, an index builder can maintain in-degree statistics and enforce bounded-degree constraints: it can reject edges into saturated hubs and preferentially allocate connectivity to peripheral anti-hubs, thereby limiting skew accumulation during ingestion. This elevates topological balance from an incidental outcome of geometric heuristics to an explicit optimization target.

6.2.2 Adaptive Query Processing. The query processor can be adaptive. A promising direction is to co-design indexing and search so that traversal can exploit topological signals online. For example, a search agent can detect hub capture—e.g., consecutive hops with diminishing distance improvement—and trigger a mode switch (beam widening or randomized backtracking) to escape local optima without a full restart. Entry-point selection can likewise move beyond random seeding to topology-aware placement (e.g., gateways to distinct regions), increasing the probability of starting near sparsely connected targets.

6.2.3 Topology Management in Complex Scenarios. Topological skew is often exacerbated by practical system constraints; however, these settings also create opportunities for principled optimization. In disk-resident indexes [26, 53], topological centrality provides a useful signal for tiered storage management. High-degree hubs serve as frequently visited navigational junctions and are therefore strong candidates for placement in low-latency tiers (e.g., L3 cache, HBM, or DRAM), whereas topologically isolated anti-hubs are rarely traversed and can be assigned to capacity-oriented tiers (e.g., SSDs) to improve cost efficiency without materially degrading traversal quality. The challenge is further amplified in dynamic scenarios [35, 37], where naive insertions and deletions can introduce new degree imbalances, break navigational paths, or isolate previously reachable nodes. Developing bounded-cost update mechanisms that preserve degree regularity and global reachability—without triggering full index rebuilds—remains an important open problem for production deployments.

6.2.4 Predictive Modeling of Tail Latency. Index validation still relies heavily on workload-specific empirical benchmarks, which provide limited predictive guidance beyond the measured setting [38, 62]. A key direction is to formalize how an index's static topological structure translates into dynamic search behavior. Specifically, we seek analytical models that predict tail-latency metrics (e.g., P999 latency) from measurable topological descriptors such as in-degree skewness and anti-hub connectivity. Such models would enable topology-aware auto-tuning: by analyzing an index offline, a system could anticipate P99/SLO violations and proactively adjust search parameters (e.g., beam width) prior to deployment, reducing reliance on trial-and-error tuning.

7 Conclusion

This paper revisits graph-based ANNS through the lens of hubness, identifying this intrinsic property of high-dimensional data as a root cause of long-tail latency. Our theoretical analysis shows that hubness induces severe topological skew—manifested as over-centralized hubs and isolated anti-hubs—which undermines the greedy-traversal heuristic underlying graph search. Building on this perspective, we reinterpret mainstream algorithms as a taxonomy of hubness-mitigation strategies that counteract the resulting imbalance. Extensive experiments validate the framework and establish a quantitative link between an algorithm's mitigation strategy, the topological balance of its index, and its robustness to outlier queries. We further show that topological uniformity, quantified by in-degree skewness, predicts search performance. Finally, we present a simple topological-signal-guided optimization that explicitly rebalances the graph and yields substantial efficiency gains. Overall, our study provides a principled basis for designing robust graph indexes and suggests clear directions for future work.

Acknowledgments

This work was supported by "Pioneer" and "Leading Goose" R&D Program of Zhejiang (No. 2024C01020).

References

[1] 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.

[2] Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Ken-ichi Kawarabayashi, and Michael Nett. 2015. Estimating Local Intrinsic Dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. ACM, 29–38.

[3] Martin Aumüller and Matteo Ceccarello. 2021. The role of local dimensionality measures in benchmarking nearest neighbor search. *Inf. Syst.* 101 (2021), 101807.

[4] Martin Aumüller and Matteo Ceccarello. 2023. Recent Approaches and Trends in Approximate Nearest Neighbor Search, with Remarks on Benchmarking. *IEEE Data Eng. Bull.* 47, 3 (2023), 89–105.

[5] Ilias Azizi, Karima Echihabi, and Themis Palpanas. 2025. Graph-Based Vector Search: An Experimental Evaluation of the State-of-the-Art. *Proc. ACM Manag. Data* 3, 1 (2025), 43:1–43:31.

[6] Brankica Bratic, Michael E. Houle, Vladimir Kurbalija, Vincent Oria, and Milos Radovanovic. 2019. The Influence of Hubness on NN-Descent. *Int. J. Artif. Intell. Tools* 28, 6 (2019), 1960002:1–1960002:23.

[7] Sebastian Bruch. 2025. Advances in Vector Search. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining, WSDM 2025, Hannover, Germany, March 10-14, 2025*, Wolfgang Nejdl, Sören Auer, Meeyoung Cha, Marie-Francine Moens, and Marc Najork (Eds.). ACM, 995–997.

[8] Yuzheng Cai, Jiayang Shi, Yizhuo Chen, and Weiguo Zheng. 2024. Navigating Labels and Vectors: A Unified Approach to Filtered Approximate Nearest Neighbor Search. *Proc. ACM Manag. Data* 2, 6 (2024), 246:1–246:27.

[9] Cheng Chen, Chenzhe Jin, Yunan Zhang, Sasha Podolsky, Chun Wu, Szu-Po Wang, Eric Hanson, Zhou Sun, Robert Walzer, and Jianguo Wang. 2024. SingleStore-V: An Integrated Vector Database System in SingleStore. *Proc. VLDB Endow.* 17, 12 (2024), 3772–3785.

[10] Tingyang Chen, Cong Fu, Kun Wang, Xiangyu Ke, Yunjun Gao, Wenchao Zhou, Yabo Ni, and Anxiang Zeng. 2025. Maximum Inner Product is Query-Scaled Nearest Neighbor. *arXiv preprint arXiv:2503.06882* (2025).

[11] Shitong Dai, Jiongnan Liu, Zhicheng Dou, Haonan Wang, Lin Liu, Bo Long, and Ji-Rong Wen. 2023. Contrastive Learning for User Sequence Representation in Personalized Product Search. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*, Ambuj K. Singh, Yizhou Sun, Leman Akoglu, Dimitrios Gunopulos, Xifeng Yan, Ravi Kumar, Fatma Ozcan, and Jieping Ye (Eds.). ACM, 380–389.

[12] Liwei Deng, Penghao Chen, Ximu Zeng, Tianfu Wang, Yan Zhao, and Kai Zheng. 2024. Efficient Data-aware Distance Comparison Operations for High-Dimensional Approximate Nearest Neighbor Search. *Proc. VLDB Endow.* 18, 3 (2024), 812–821.

[13] Wei Dong, Charikar Moses, and Kai Li. 2011. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th international conference on World wide web*. 577–586.

[14] Paul Erdős, Alfréd Rényi, et al. 1960. On the evolution of random graphs. *Publications of the* (1960).

[15] Arthur Flexer and Dominik Schnitzer. 2013. Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces?. In *13th IEEE International Conference on Data Mining Workshops, ICDM Workshops, TX, USA, December 7-10, 2013*. IEEE Computer Society, 460–467.

[16] Cole Foster and Benjamin B. Kimia. 2023. Computational Enhancements of HNSW Targeted to Very Large Datasets. In *Similarity Search and Applications - 16th International Conference, SISAP 2023, A Coruña, Spain, October 9-11, 2023, Proceedings*, Oscar Pedreira and Vladimir Estivill-Castro (Eds.), Vol. 14289. 291–299.

[17] Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. 2019. Fast Approximate Nearest Neighbor Search With The Navigating Spreading-out Graph. *Proc. VLDB Endow.* 12, 5 (2019), 461–474.

[18] Yujian Fu, Cheng Chen, Yao Chen, Weng-Fai Wong, and Bingsheng He. 2025. Vista: Vector Indexing and Search for Large-Scale Imbalanced Datasets. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE, 543–556.

[19] A Ganesh and Moez Draief. 2009. A random walk model for infection on graphs. In *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools*. 1–8.

[20] Jianyang Gao, Yutong Gou, Yuexuan Xu, Yongyi Yang, Cheng Long, and Raymond Chi-Wing Wong. 2025. Practical and Asymptotically Optimal Quantization of High-Dimensional Vectors in Euclidean Space for Approximate Nearest Neighbor Search. *Proc. ACM Manag. Data* 3, 3 (2025), 202:1–202:26.

[21] Jianyang Gao and Cheng Long. 2023. High-dimensional approximate nearest neighbor search: with reliable and efficient distance comparison operations. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–27.

[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), 1–27.

[23] Zhen Gong, Xin Wu, Lei Chen, Zhenzhe Zheng, Shengjie Wang, Anran Xu, Chong Wang, and Fan Wu. 2023. Full Index Deep Retrieval: End-to-End User and Item Structures for Cold-start and Long-tail Item Recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, Jie Zhang, Li Chen, Shlomo Berkovsky, Min Zhang, Tommaso Di Noia, Justin Basilico, Luiz Pizzato, and Yang Song (Eds.). ACM, 47–57.

[24] Ben Harwood and Tom Drummond. 2016. FANNG: Fast Approximate Nearest Neighbour Graphs. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 5713–5722.

[25] Kohei Hirata, Daichi Amagata, Sumio Fujita, and Takahiro Hara. 2022. Solving Diversity-Aware Maximum Inner Product Search Efficiently and Effectively. In *RecSys '22: Sixteenth ACM Conference on Recommender Systems, Seattle, WA, USA, September 18 - 23, 2022*, Jennifer Golbeck, F. Maxwell Harper, Vanessa Murdock, Michael D. Ekstrand, Bracha Shapira, Justin Basilico, Keld T. Lundgaard, and Even Oldridge (Eds.). ACM, 198–207.

[26] Suhas Jayaram Subramanya, Fnu Devvrit, Harsha Vardhan Simhadri, Ravishankar Krishnawamy, and Rohan Kadekodi. 2019. Diskann: Fast accurate billion-point nearest neighbor search on a single node. *Advances in Neural Information Processing Systems* 32 (2019).

[27] 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), 117–128.

[28] Junkyum Kim and Divya Mahajan. 2025. An Adaptive Vector Index Partitioning Scheme for Low-Latency RAG Pipeline. *CoRR* abs/2504.08930 (2025).

[29] Jon M Kleinberg. 2000. Navigation in a small world. *Nature* 406, 6798 (2000), 845–845.

[30] Sushma Kumari and Balasubramaniam Jayaram. 2017. Measuring Concentration of Distances - An Effective and Efficient Empirical Index. *IEEE Trans. Knowl. Data Eng.* 29, 2 (2017), 373–386.

[31] 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*. 2539–2554.

[32] Mocheng Li, Xiao Yan, Baotong Lu, Yue Zhang, James Cheng, and Chenhao Ma. 2025. Attribute Filtering in Approximate Nearest Neighbor Search: An In-depth Experimental Study. *CoRR* abs/2508.16263 (2025).

[33] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. 2019. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering* 32, 8 (2019), 1475–1488.

[34] Shengwen Liang, Ying Wang, Ziming Yuan, Cheng Liu, Huawei Li, and Xiaowei Li. 2022. VStore: in-storage graph based vector search accelerator. In *DAC '22: 59th ACM/IEEE Design Automation Conference, San Francisco, California, USA, July 10 - 14, 2022*, Rob Oshana (Ed.). ACM, 997–1002.

[35] Dawei Liu, Bolong Zheng, Ziyang Yue, Fuhao Ruan, Xiaofang Zhou, and Christian S. Jensen. 2025. Wolverine: Highly Efficient Monotonic Search Path Repair for Graph-based ANN Index Updates. *Proc. VLDB Endow.* 18, 7 (2025), 2268–2280.

[36] Kejing Lu, Mineichi Kudo, Chuan Xiao, and Yoshiharu Ishikawa. 2021. HVS: Hierarchical Graph Structure Based on Voronoi Diagrams for Solving Approximate Nearest Neighbor Search. *Proc. VLDB Endow.* 15, 2 (2021), 246–258.

[37] Ruiyao Ma, Yifan Zhu, Baihua Zheng, Lu Chen, Congcong Ge, and Yunjun Gao. 2024. GTI: Graph-based Tree Index with Logarithm Updates for Nearest Neighbor Search in High-Dimensional Spaces. *Proc. VLDB Endow.* 18, 4 (2024), 986–999.

[38] Vasilis Mageirakos, Bowen Wu, and Gustavo Alonso. 2025. Cracking Vector Search Indexes. *Proc. VLDB Endow.* 18, 11 (2025), 3951–3964.

[39] Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. 2014. Approximate nearest neighbor algorithm based on navigable small world graphs. *Inf. Syst.* 45 (2014), 61–68.

[40] 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), 824–836.

[41] Priya Mani, Marilyn Vazquez, Jessica Ruth Metcalf-Burton, Carlotta Domeniconi, Hillary Fairbanks, Gülce Bal, Elizabeth Beer, and Sibel Tari. 2019. *The Hubness Phenomenon in High-Dimensional Spaces*. Springer International Publishing, Cham, 15–45. https://doi.org/10.1007/978-3-030-11566-1_2

[42] Kathleen M O'Hara. 1990. Unimodality of Gaussian coefficients: a constructive proof. *Journal of Combinatorial Theory, Series A* 53, 1 (1990), 29–52.

[43] Liana Patel, Peter Kraft, Carlos Guestrin, and Matei Zaharia. 2024. ACORN: Performant and Predicate-Agnostic Search Over Vector Embeddings and Structured Data. *Proc. ACM Manag. Data* 2, 3 (2024), 120.

[44] Yun Peng, Byron Choi, Tsz Nam Chan, Jianye Yang, and Jianliang Xu. 2023. Efficient approximate nearest neighbor search in multi-dimensional databases. *Proceedings of the ACM on Management of Data* 1, 1 (2023), 1–27.

[45] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. 2010. Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data. *J. Mach. Learn. Res.* 11 (2010), 2487–2531.

[46] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (Eds.).

[47] Saba Sturua, Isabelle Mohr, Mohammad Kalim Akram, Michael Günther, Bo Wang, Markus Krimmel, Feng Wang, Georgios Mastrapas, Andreas Koukounas, Nan Wang, and Han Xiao. 2025. Jina Embeddings V3: Multilingual Text Encoder with Low-Rank Adaptations. In *Advances in Information Retrieval - 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6-10, 2025, Proceedings, Part V (Lecture Notes in Computer Science)*, Claudia Hauff, Craig Macdonald, Dietmar Jannach, Gabriella Kazai, Franco Maria Nardini, Fabio Pinelli, Fabrizio Silvestri, and Nicola Tonellotto (Eds.), Vol. 15576. Springer, 123–129.

[48] Nenad Tomasev. 2014. The Role Of Hubness in High-dimensional Data Analysis. *Informatica (Slovenia)* 38, 4 (2014).

[49] Nenad Tomasev, Milos Radovanovic, Dunja Mladenic, and Mirjana Ivanovic. 2014. The Role of Hubness in Clustering High-Dimensional Data. *IEEE Trans. Knowl. Data Eng.* 26, 3 (2014), 739–751.

[50] Hongya Wang, Wenlong Wu, Cong Luo, Aobei Bian, Chunguang Meng, Yishuo Wu, and Ji Sun. 2025. Boosting Accuracy and Efficiency for Vector Retrieval with Local Scaling Graph. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE, 336–348.

[51] 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*. 2614–2627.

[52] Mengzhao Wang, Haotian Wu, Xiangyu Ke, Yunjun Gao, Yifan Zhu, and Wenchao Zhou. 2025. Accelerating Graph Indexing for ANNS on Modern CPUs. *Proc. ACM Manag. Data* 3, 3 (2025), 123:1–123:29.

[53] Mengzhao Wang, Weizhi Xu, Xiaomeng Yi, Songlin Wu, Zhangyang Peng, Xiangyu Ke, Yunjun Gao, Xiaoliang Xu, Rentong Guo, and Charles Xie. 2024. 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), 1–27.

[54] Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. 2021. A Comprehensive Survey and Experimental Comparison of Graph-Based Approximate Nearest Neighbor Search. *Proc. VLDB Endow.* 14, 11 (2021), 1964–1978.

[55] Zeyu Wang, Peng Wang, Themis Palpanas, and Wei Wang. 2023. Graph- and Tree-based Indexes for High-dimensional Vector Similarity Search: Analyses, Comparisons, and Future Directions. *IEEE Data Eng. Bull.* 47, 3 (2023), 3–21.

[56] Zeyu Wang, Qitong Wang, Xiaoxing Cheng, Peng Wang, Themis Palpanas, and Wei Wang. 2024. Steiner-Hardness: A Query Hardness Measure for Graph-Based ANN Indexes. *Proc. VLDB Endow.* 17, 13 (2024), 4668–4682.

[57] Jasper Xian, Tommaso Teofili, Ronak Pradeep, and Jimmy Lin. 2024. Vector Search with OpenAI Embeddings: Lucene Is All You Need. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining, WSDM 2024, Merida, Mexico, March 4-8, 2024*, Luz Angelica Caudillo-Mata, Silvio Lattanzi, Andrés Muñoz Medina, Leman Akoglu, Aristides Gionis, and Sergei Vassilvitskii (Eds.). ACM, 1090–1093.

[58] Jiadong Xie, Jeffrey Xu Yu, and Yingfan Liu. 2025. Graph Based K-Nearest Neighbor Search Revisited. *ACM Transactions on Database Systems* (2025).

[59] Shuo Yang, Jiadong Xie, Yingfan Liu, Jeffrey Xu Yu, Xiyue Gao, Qianru Wang, Yanguo Peng, and Jiangtao Cui. 2025. Revisiting the Index Construction of Proximity Graph-Based Approximate Nearest Neighbor Search. *Proc. VLDB Endow.* 18, 6 (2025), 1825–1838.

[60] Qiang Yue, Xiaoliang Xu, Yuxiang Wang, Yikun Tao, and Xuliyuan Luo. 2024. Routing-Guided Learned Product Quantization for Graph-Based Approximate Nearest Neighbor Search. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*. IEEE, 4870–4883.

[61] Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 Embedding: Advancing Text Embedding and Reranking Through Foundation Models. *CoRR* abs/2506.05176 (2025).

[62] Yunan Zhang, Shige Liu, and Jianguo Wang. 2024. Are There Fundamental Limitations in Supporting Vector Data Management in Relational Databases? A Case Study of PostgreSQL. In *40th IEEE International Conference on Data Engineering, ICDE 2024, Utrecht, The Netherlands, May 13-16, 2024*. IEEE, 3640–3653.

[63] Xi Zhao, Yao Tian, Kai Huang, Bolong Zheng, and Xiaofang Zhou. 2023. Towards efficient index construction and approximate nearest neighbor search in high-dimensional spaces. *Proceedings of the VLDB Endowment* 16, 8 (2023), 1979–1991.

[64] Ruiqi Zheng, Liang Qu, Bin Cui, Yuhui Shi, and Hongzhi Yin. 2023. AutoML for Deep Recommender Systems: A Survey. *ACM Trans. Inf. Syst.* 41, 4 (2023), 101:1–101:38.

Received October 2025; revised January 2026; accepted February 2026