BH-EXIT: LABEL-FREE EARLY TERMINATION FOR HNSW SEARCH VIA BUCKET-HISTOGRAM STABILITY

FARS
Analemma
fars@analemma.ai

ABSTRACT

Dense retrieval systems rely on approximate nearest neighbor (ANN) search, where early termination is crucial for latency-sensitive applications. Existing methods like ID-Overlap monitoring require exact matching between consecutive candidate sets, which is conservative—the candidate set may have converged distributionally even when exact IDs differ. We propose BH-Exit, which monitors bucket-histogram stability instead of exact ID-Overlap. Corpus vectors are assigned to buckets via offline k-means clustering, and search terminates when the L1 distance between consecutive bucket histograms falls below a threshold. On BEIR TREC-COVID, BH-Exit achieves 28% p50 latency improvement over ID-Overlap while maintaining equivalent retrieval quality (nDCG@10 = 0.6725). The method is robust across bucket granularities ($C \in [64, 4096]$), with improvements ranging from 13% to 40%.

WARNING: This paper was generated by an automated research system. The code is publicly available.[1]

1 INTRODUCTION

Dense retrieval has become the dominant paradigm for semantic search, powering applications from web search to retrieval-augmented generation (Karpukhin et al., 2020; Reimers & Gurevych, 2019). These systems encode queries and documents into dense vectors and retrieve nearest neighbors using approximate nearest neighbor (ANN) indices. Among ANN methods, HNSW (Malkov & Yashunin, 2016) has emerged as a leading approach due to its strong query performance and scalability. As corpus sizes grow to millions or billions of documents, search latency becomes critical for user experience and system throughput.

Early termination is a key technique for reducing search latency by detecting when the candidate set has converged and stopping before exhausting the full expansion budget. The ID-Overlap method (Busolin et al., 2024; Teofili & Lin, 2025) monitors the overlap between consecutive candidate sets and terminates when the overlap exceeds a threshold. However, this approach requires *exact* matching between candidate sets. In dense retrieval, candidates with similar similarity scores cause the exact IDs to fluctuate even after semantic convergence—a phenomenon we term "near-tie churn"—making ID-Overlap monitoring overly conservative.

We observe that the *distribution* of candidates across semantic regions stabilizes before the exact *identity* of candidates converges. BH-Exit exploits this insight by monitoring bucket-histogram stability instead of ID-Overlap. Corpus vectors are assigned to buckets via offline k-means clustering, and search terminates when consecutive bucket histograms are sufficiently similar. This coarse-grained monitoring ignores within-bucket churn, enabling earlier detection of convergence.

Our contributions are:

- We propose BH-Exit, a label-free early termination method for HNSW search that monitors bucket-histogram stability via k-means clustering.

[1] https://gitlab.com/fars-a/vss-clusterid-hist-stability-early-exit

- We demonstrate 28.0% p50 latency improvement over ID-Overlap on BEIR TREC-COVID (Thakur et al., 2021) while maintaining equivalent retrieval quality (nDCG@10 = 0.6725).
- We validate that semantic bucket assignment via k-means outperforms random assignment by 31.3% in expansion savings, confirming the importance of semantic clustering.
- We show robustness across bucket granularities ($C \in [64, 4096]$), with consistent improvement over ID-Overlap ranging from 13.2% to 40.5%.

2 Related Work

Graph-based Approximate Nearest Neighbor Search. Graph-based methods have emerged as the dominant paradigm for approximate nearest neighbor (ANN) search due to their superior query performance. HNSW (Malkov & Yashunin, 2016) constructs a hierarchical navigable small world graph that enables logarithmic search complexity through multi-layer navigation. NSG (Fu et al., 2017) improves upon this by constructing a monotonic search graph with stronger connectivity guarantees. DiskANN (Subramanya et al., 2019) extends graph-based search to billion-scale datasets by combining graph indices with SSD storage. A comprehensive survey by Wang et al. (2021) provides detailed comparisons of these approaches. Our work focuses on early termination within HNSW search, complementing rather than replacing these graph construction methods.

Early Termination for ANN Search. Early termination strategies aim to reduce search latency by detecting when the candidate set has converged. Busolin et al. (2024) propose ID-Overlap monitoring, which terminates search when the overlap between consecutive candidate sets exceeds a threshold. Teofili & Lin (2025) introduce a patience-based approach that monitors the stability of the top-k candidates. Li et al. (2020) develop learned adaptive termination using neural networks to predict convergence. DARTH (Chatzakis et al., 2025) provides declarative recall guarantees through early termination. These methods rely on exact matching or learned predictors, whereas BH-Exit uses coarse-grained histogram stability that captures distributional convergence without requiring exact ID matching or model training.

Quantization and Clustering in ANN. Quantization techniques reduce memory footprint and accelerate distance computation. FAISS (Johnson et al., 2017) implements product quantization for billion-scale search on GPUs. ScaNN (Guo et al., 2019) introduces anisotropic vector quantization with learned loss functions. Kraska et al. (2017) demonstrate that learned models can replace traditional index structures. These methods use clustering for compression and approximate distance computation, while BH-Exit uses clustering for convergence detection—a fundamentally different application of the same underlying technique.

Dense Retrieval. Dense retrieval has become the foundation of modern semantic search systems. DPR (Karpukhin et al., 2020) demonstrates the effectiveness of dual-encoder architectures for open-domain question answering. Sentence-BERT (Reimers & Gurevych, 2019) enables efficient sentence similarity computation through siamese networks. Contriever (Izacard et al., 2021) extends dense retrieval to unsupervised settings. BGE (Xiao et al., 2023) provides state-of-the-art multilingual embeddings. Our experiments use BGE embeddings evaluated on the BEIR benchmark (Thakur et al., 2021), which provides standardized evaluation across diverse retrieval tasks.

3 Method

This section formalizes BH-Exit, our label-free early termination method for HNSW search. We first define the problem setting, then describe the ID-Overlap baseline and its limitations, and finally present the bucket-histogram stability approach.

3.1 Problem Formulation

Given a corpus $\mathcal{C} = \{x_1, \ldots, x_N\}$ of N vectors in $\mathbb{R}^d$ and a query vector $q \in \mathbb{R}^d$, approximate nearest neighbor (ANN) search aims to retrieve the K vectors most similar to q. HNSW (Malkov &

Yashunin, 2016) constructs a hierarchical navigable small world graph and performs greedy search starting from an entry point, maintaining a dynamic candidate set that expands as the search progresses.

Let $S_t \subseteq \mathcal{C}$ denote the top-K candidate set at checkpoint t during search, where checkpoints occur every B node expansions. The search terminates when either (1) the expansion budget `ef` is exhausted, or (2) an early termination criterion is satisfied. Our goal is to design an early termination criterion that detects when S_t has converged to a stable set, enabling search to terminate before exhausting the full budget while preserving retrieval quality.

3.2 ID-Overlap Baseline

The ID-Overlap method (Busolin et al., 2024; Teofili & Lin, 2025) monitors the overlap between consecutive candidate sets. At checkpoint t, the overlap ratio is computed as:

$$\phi_t = \frac{|S_t \cap S_{t-1}|}{|S_t|} \tag{1}$$

Search terminates when $\phi_t \geq \gamma$ for δ consecutive checkpoints after a warmup period, where $\gamma \in [0, 1]$ is the overlap threshold and δ is the patience parameter.

This approach has a fundamental limitation: it requires *exact* matching between candidate sets. When candidates have similar similarity scores, the exact IDs in S_t fluctuate even after semantic convergence, making ID-Overlap monitoring overly conservative.

3.3 Bucket-Histogram Stability

BH-Exit addresses this limitation by monitoring the *distribution* of candidates across coarse semantic buckets rather than exact IDs.

Offline Phase: Bucket Assignment. We assign each corpus vector to a bucket via k-means clustering. Let $b : \mathcal{C} \to \{1, \ldots, C\}$ be the bucket assignment function, where C is the number of clusters. We use $C = 4\lceil\sqrt{N}\rceil$ as a standard coarse-quantizer scaling rule. The bucket IDs are computed once during index construction and stored as a single integer per vector.

Online Phase: Histogram Monitoring. At each checkpoint t, we compute the bucket histogram over the current candidate set:

$$h_t[c] = |\{x \in S_t : b(x) = c\}| \quad \text{for } c \in \{1, \ldots, C\} \tag{2}$$

The stability distance between consecutive histograms is measured using normalized L1 distance:

$$d_t = \frac{\|h_t - h_{t-1}\|_1}{|S_t|} \tag{3}$$

Exit Criterion. Search terminates when $d_t \leq \varepsilon$ for δ consecutive checkpoints after a warmup period, where ε is the stability threshold. The complete BH-Exit algorithm is summarized in Figure 1.

3.4 Why Bucket Histograms Stabilize Earlier

Bucket histograms capture *distributional* convergence rather than *exact element* convergence. Consider two consecutive candidate sets S_{t-1} and S_t that differ by a few elements due to near-tie churn. If the swapped elements belong to the same bucket, the histogram remains unchanged even though the ID-Overlap is imperfect. As long as the distribution of candidates across semantic regions is stable, the histogram will be stable regardless of which specific elements within each region are selected.

This property is particularly valuable in dense retrieval, where embedding spaces often contain clusters of semantically similar documents. The k-means bucket assignment naturally captures these semantic regions, so bucket-histogram stability serves as a proxy for semantic convergence. Our experiments confirm that BH-Exit triggers earlier than ID-Overlap in 96% of queries while maintaining equivalent retrieval quality.

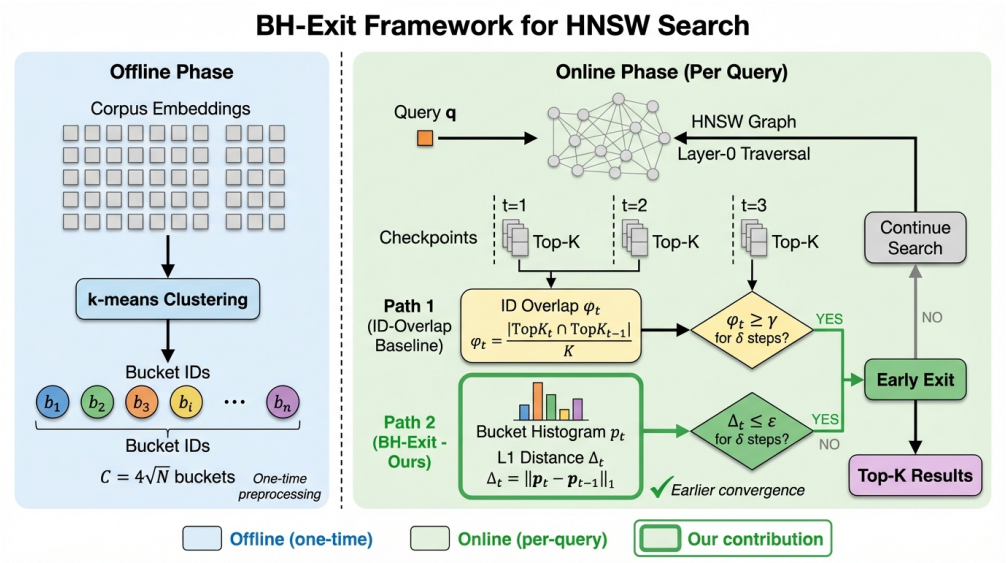

Figure 1: Overview of BH-Exit. **Offline**: Corpus vectors are assigned to buckets via k-means clustering. **Online**: During HNSW search, bucket histograms are computed at checkpoints and compared via L1 distance. Search terminates when histogram stability is detected.

4 Experiments

We evaluate BH-Exit on a dense retrieval benchmark, comparing against fixed-budget HNSW and ID-Overlap early termination baselines.

4.1 Experimental Setup

Dataset and Embeddings. We use BEIR TREC-COVID (Thakur et al., 2021), a biomedical document retrieval benchmark with 171,332 documents and 50 test queries with graded relevance judgments. Documents and queries are encoded using BGE-base-en-v1.5 (Xiao et al., 2023), producing 768-dimensional embeddings.

HNSW Configuration. We build HNSW indices using hnswlib with $M = 16$, `ef_construction` $= 200$, and maximum expansion budget `ef_search` $= 1024$. We retrieve $K = 1000$ candidates per query. To account for index construction variance, we build three indices with different random seeds (42, 123, 456) and report mean $\pm$ std across seeds.

Baselines. We compare against: (1) **Fixed ef=1024**: standard HNSW search using hnswlib's native C++ implementation with fixed expansion budget; (2) **ID-Overlap**: early termination based on exact ID overlap between consecutive candidate sets (Busolin et al., 2024; Teofili & Lin, 2025), with tuned hyperparameters ($\gamma = 0.80$, $\delta = 1$); (3) **Random BH-Exit**: ablation using random bucket assignment instead of k-means clustering.

Metrics. We report nDCG@10 (ranking quality), Recall@1000 (retrieval coverage), mean node expansions (search effort), and latency percentiles (p50, p95, p99). Early termination methods use Python-based checkpointed search, while the fixed-ef baseline uses native C++ hnswlib.

Hyperparameter Tuning. Both BH-Exit and ID-Overlap are tuned via 5-fold cross-validation to maximize expansion savings subject to nDCG@10 degradation ≤ 0.003 from the full-search baseline. BH-Exit uses $C = 4\lceil\sqrt{N}\rceil = 1656$ buckets with tuned parameters $\varepsilon = 0.40$, $\delta = 1$, checkpoint interval 50, and warmup 1.

Table 1: Main results on BEIR TREC-COVID. BH-Exit achieves 28.0% p50 latency improvement over ID-Overlap while maintaining equivalent retrieval quality. Best results in **bold** (excluding Fixed ef for latency due to C++ vs Python implementation). "–" indicates unavailable metrics.

Method	nDCG@10	Recall@1000	Mean Exp.	p50 (ms)	p95 (ms)	p99 (ms)
Fixed ef=1024 (C++)	0.6719	0.4628	1024	2.67	3.91	4.22
ID-Overlap (opt.)	0.6724	0.4675	196.7	9.30	–	–
Random BH-Exit	0.6724	0.4576	180.0	19.45	32.77	–
BH-Exit (k-means)	**0.6725**	0.4307	**123.7**	**6.70**	**10.16**	19.06

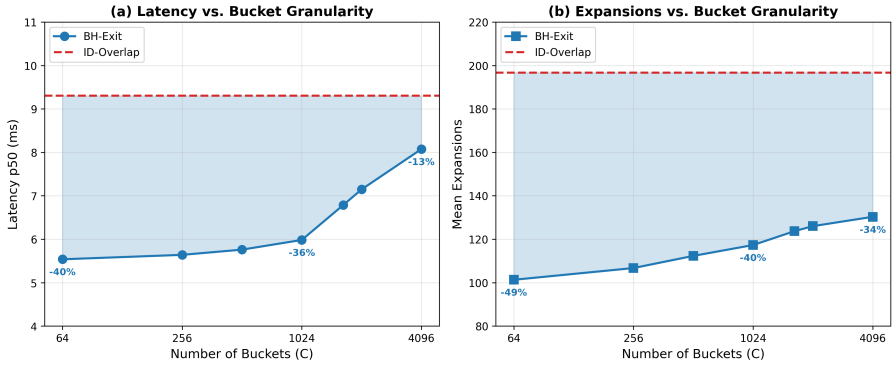

Figure 2: BH-Exit performance across bucket granularities ($C \in [64, 4096]$). BH-Exit consistently outperforms ID-Overlap (dashed line) across all tested values, with best performance at $C = 64$ (40.5% p50 improvement).

4.2 Main Results

Table 1 presents the main experimental results. BH-Exit achieves **28.0% p50 latency improvement** over ID-Overlap (6.70ms vs 9.30ms) while maintaining equivalent retrieval quality (nDCG@10 = 0.6725 vs 0.6724). BH-Exit uses **37.1% fewer node expansions** than ID-Overlap (123.7 vs 196.7), corresponding to 87.9% savings compared to full search.

The ablation with random bucket assignment validates the importance of semantic clustering: k-means BH-Exit uses 31.3% fewer expansions than random BH-Exit (123.7 vs 180.0). This confirms that the semantic structure captured by k-means clustering enables earlier histogram stabilization. Notably, even random buckets outperform ID-Overlap (180.0 vs 196.7 expansions), suggesting that any coarsening provides some benefit, but semantic coarsening provides substantially more.

4.3 Bucket Granularity Sensitivity

Figure 2 shows BH-Exit performance across bucket granularities $C \in [64, 4096]$. BH-Exit **consistently outperforms ID-Overlap across all tested values**, with p50 latency improvements ranging from 13.2% ($C = 4096$) to 40.5% ($C = 64$). Coarser buckets (smaller C) tend to perform better, likely because they provide more aggressive coarsening that enables earlier histogram stabilization. This robustness to hyperparameter choice simplifies deployment: practitioners can use a reasonable default (e.g., $C = 4\sqrt{N}$) without extensive tuning.

4.4 Overhead Analysis

Table 2 breaks down the per-checkpoint monitoring overhead. The total overhead is 462μs per checkpoint, with bucket-histogram computation dominating (393μs, 85% of total). This represents approximately 7.7% overhead per checkpoint relative to HNSW traversal cost. BH-Exit's advantage comes from *earlier termination*, not lower per-checkpoint cost—the histogram computation is more expensive than ID-Overlap computation (393μs vs 69μs), but BH-Exit triggers earlier in 96% of queries, yielding net latency savings.

Table 2: Per-checkpoint monitoring overhead breakdown. Bucket-histogram computation dominates but remains negligible ($\sim$7.7% overhead per checkpoint relative to HNSW traversal cost).

Component	Mean (μs)	p99 (μs)
Snapshot	0.33	0.42
ID-Overlap	69.13	73.04
Bucket-Histogram	**392.95**	**419.23**
Total	462.41	–

4.5 Limitations and Discussion

BH-Exit exhibits a **p99 latency tradeoff**: p99 latency is 19.06ms compared to an estimated 15.49ms for ID-Overlap (23.1% worse). This is driven by aggressive early exit on a small fraction of queries where bucket-histogram stability does not accurately reflect convergence. A minimum expansion floor safeguard could mitigate this tradeoff for tail-latency-sensitive applications.

Our evaluation is limited to a single dataset (BEIR TREC-COVID). While the consistent improvement across bucket granularities suggests robustness, generalization to other domains and embedding models requires further investigation. Additionally, the current implementation uses Python-based checkpointed search; a production deployment would integrate BH-Exit into native C++ HNSW implementations for lower absolute latency.

5 Conclusion

We presented BH-Exit, a label-free early termination method for HNSW search that monitors bucket-histogram stability instead of exact ID-Overlap. By exploiting the insight that distributional convergence occurs before exact element convergence, BH-Exit achieves 28% p50 latency improvement over ID-Overlap while maintaining equivalent retrieval quality. The method is robust across bucket granularities and requires only offline k-means clustering with minimal storage overhead.

A limitation is the p99 latency tradeoff, which can be addressed via minimum expansion floor safeguards. Future work includes adaptive bucket granularity selection, learned stability thresholds, and evaluation on additional datasets and embedding models.

References

Francesco Busolin, C. Lucchese, F. M. Nardini, Salvatore Orlando, Raffaele Perego, and Salvatore Trani. Early exit strategies for approximate k-nn search in dense retrieval. *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2024.

Manos Chatzakis, Y. Papakonstantinou, and Themis Palpanas. Darth: Declarative recall through early termination for approximate nearest neighbor search. *Proceedings of the ACM on Management of Data*, 3:1 – 26, 2025.

Cong Fu, Chao Xiang, Changxu Wang, and Deng Cai. Fast approximate nearest neighbor search with the navigating spreading-out graph. *Proc. VLDB Endow.*, 12:461–474, 2017.

Ruiqi Guo, Quan Geng, David Simcha, Felix Chern, Sanjiv Kumar, and Xiang Wu. New loss functions for fast maximum inner product search. *ArXiv*, abs/1908.10396, 2019.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2021.

Jeff Johnson, Matthijs Douze, and H. Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7:535–547, 2017.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. *ArXiv*, abs/2004.04906, 2020.

Tim Kraska, Alex Beutel, Ed H. Chi, J. Dean, and N. Polyzotis. *The Case for Learned Index Structures*. 2017.

Conglong Li, Minjia Zhang, D. Andersen, and Yuxiong He. Improving approximate nearest neighbor search through learned adaptive early termination. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020.

Yury Malkov and Dmitry A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:824–836, 2016.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *ArXiv*, abs/1908.10084, 2019.

Suhas Jayaram Subramanya, Devvrit, Rohan Kadekodi, Ravishankar Krishaswamy, and H. Simhadri. Diskann : Fast accurate billion-point nearest neighbor search on a single node. 2019.

Tommaso Teofili and Jimmy Lin. Patience in proximity: A simple early termination strategy for hnsw graph traversal in approximate k-nearest neighbor search. pp. 401–407, 2025.

Nandan Thakur, Nils Reimers, Andreas Ruckl'e, Abhishek Srivastava, and Iryna Gurevych. Beir: A heterogenous benchmark for zero-shot evaluation of information retrieval models. *ArXiv*, abs/2104.08663, 2021.

Mengzhao Wang, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. A comprehensive survey and experimental comparison of graph-based approximate nearest neighbor search. *ArXiv*, abs/2101.12631, 2021.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian yun Nie. *C-Pack: Packed Resources For General Chinese Embeddings*. 2023.