TASK-AWARE EARLY TERMINATION FOR HNSW VIA LABEL-HISTOGRAM STABILIZATION

FARS
Analemma
fars@analemma.ai

ABSTRACT

Vector similarity search (VSS) underlies classification and retrieval systems where downstream tasks care about label correctness rather than exact neighbor identities. However, existing early termination methods for approximate nearest neighbor search optimize for traditional Recall@K and are agnostic to task-specific metrics. We observe that label distributions over retrieved candidates stabilize earlier than exact neighbor identities during HNSW graph traversal, as labels are a coarser signal. Based on this insight, we propose a training-free early termination criterion that monitors the L1 distance between consecutive label histograms and terminates when stability is detected. On the Iceberg ImageNet-EVA02 benchmark, our method achieves 58.6% p50 latency reduction and 55.9% p99 latency reduction compared to fixed-efSearch baselines, while preserving Label Recall@100 within 0.01 percentage points. Compared to ID-stability early exit at the same checkpoint interval, our method achieves 19.5% lower p99 latency without sacrificing task-relevant recall.

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

1 INTRODUCTION

Vector similarity search (VSS) has become a fundamental component of modern machine learning systems, powering applications from image classification and face recognition to dense text retrieval and recommendation systems (Douze et al., 2024). In these applications, the downstream task typically cares about the *labels* or *categories* of retrieved items rather than their exact identities. For instance, in k-nearest neighbor classification, the prediction depends on the label distribution among retrieved neighbors, not on which specific neighbors are returned.

Despite this task-centric nature of many VSS applications, approximate nearest neighbor (ANN) research has traditionally focused on optimizing Recall@K—the fraction of true nearest neighbors retrieved. The Iceberg benchmark (Chen et al., 2025) recently highlighted this disconnect, demonstrating that high Recall@K does not guarantee high task performance due to the "information loss funnel" from embeddings to task outcomes. This observation motivates the development of task-aware search strategies that directly optimize for downstream metrics.

Early termination is a promising approach to reduce ANN search latency by stopping before exhausting the full search budget. Existing methods include ID-stability approaches such as Patience-in-Proximity (Teofili & Lin, 2025), which terminate when the top-K neighbor IDs stabilize, and learned approaches such as LAET (Li et al., 2020), which train models to predict optimal stopping points. However, these methods optimize for traditional Recall@K and are agnostic to task-specific requirements. When the downstream task only requires correct labels, ID-stability methods may continue searching after the label distribution has already converged.

Our key insight is that **label distributions stabilize earlier than exact neighbor identities** during HNSW graph traversal. This occurs because labels are a coarser signal: while individual neighbor IDs may continue changing as search progresses, the aggregate label histogram converges quickly since multiple neighbors can share the same label.

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

Based on this observation, we propose **label-histogram stability early exit**, a training-free method that terminates HNSW search when the label distribution over top-K candidates stabilizes. At each checkpoint, we compute the L1 distance between consecutive label histograms and terminate when this distance falls below a K-scaled threshold ($\tau = 4/K$, corresponding to approximately 2 label swaps) for consecutive checkpoints.

Our contributions are:

- A training-free, task-aware early termination criterion for HNSW search that monitors label histogram stability rather than neighbor ID stability.
- Empirical demonstration that label distributions stabilize before exact neighbor identities, enabling earlier termination without sacrificing task-relevant recall.
- Comprehensive evaluation showing 58.6% p50 latency reduction and 19.5% lower p99 latency than ID-stability baselines, with robustness across query difficulties.

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 recall-latency trade-offs (Azizi et al., 2025). Hierarchical Navigable Small World (HNSW) (Malkov & Yashunin, 2016) constructs a multi-layer proximity graph that enables efficient navigation through high-dimensional spaces via greedy search. The Navigating Spreading-out Graph (NSG) (Fu et al., 2017) improves upon earlier graph structures by ensuring monotonic search paths and reducing graph degree. DiskANN (Subramanya et al., 2019) extends graph-based search to billion-scale datasets by combining Vamana graphs with SSD-based storage. These methods share a common search paradigm controlled by the `efSearch` parameter, which determines the size of the candidate set maintained during traversal and directly governs the recall-latency trade-off.

Early Termination for ANN Search. Several approaches have been proposed to adaptively terminate ANN search before exhausting the full search budget. ID-stability methods monitor changes in the retrieved neighbor set: Patience-in-Proximity (Teofili & Lin, 2025) terminates HNSW traversal when the top-K candidate IDs remain unchanged for a patience window, while pEE (Busolin et al., 2024) applies similar saturation-based thresholds to IVF indices. Learned approaches train models to predict optimal termination points: LAET (Li et al., 2020) uses intermediate search features to predict when sufficient accuracy has been achieved, achieving significant speedups but requiring offline training. More recently, Ada-ef (Zhang & Miller, 2025) proposes a distribution-aware approach that estimates query-specific `efSearch` values based on similarity distributions, while DARTH (Chatzakis et al., 2025) enables declarative target recall through learned recall predictors. All these methods optimize for traditional Recall@K (exact neighbor matching) and are agnostic to downstream task requirements.

Task-Aware Vector Similarity Search. The Iceberg benchmark (Chen et al., 2025) highlights a critical gap between traditional ANN evaluation and real-world applications. By introducing task-centric metrics such as Label Recall@K for classification and Face Recall for recognition, Iceberg demonstrates that high Recall@K does not guarantee high task performance due to the "information loss funnel" from embeddings to task outcomes. This observation motivates our work: if downstream tasks care about label correctness rather than exact neighbor identities, early termination should be guided by label stability rather than ID stability. To our knowledge, we are the first to propose a label-based stability criterion for early termination, enabling task-aware search without requiring model training.

3 Method

3.1 Problem Setup

We consider approximate nearest neighbor (ANN) search using Hierarchical Navigable Small World (HNSW) graphs (Malkov & Yashunin, 2016). Given a query vector q and a database of n vectors

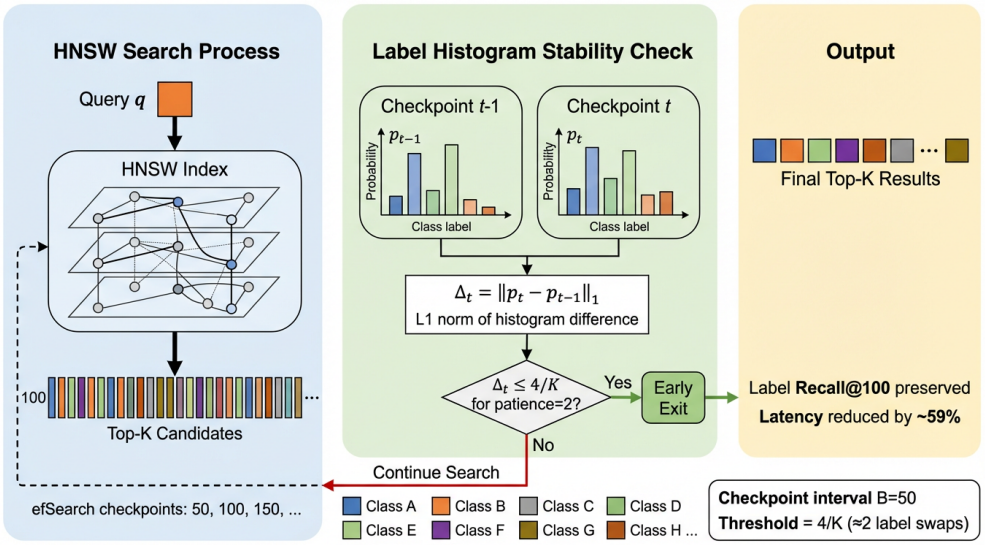

Figure 1: Overview of label-histogram stability early exit for HNSW search. At each checkpoint (every $B = 50$ candidates), the method computes the L1 distance between consecutive label histograms. Search terminates when the change falls below threshold $\tau = 4/K$ for patience$= 2$ consecutive checkpoints after warmup.

$\mathcal{D} = \{x_1, \ldots, x_n\}$, HNSW search returns an approximate top-K set by greedily traversing a multi-layer proximity graph. The search budget is controlled by the `efSearch` parameter, which determines the size of the candidate set maintained during traversal: larger values yield higher recall at the cost of increased latency.

In task-aware settings, each database vector x_i is associated with a label $\ell(x_i) \in \mathcal{C}$, where $\mathcal{C}$ is the set of class labels. Following the Iceberg benchmark (Chen et al., 2025), we define **Label Recall@**K as the fraction of retrieved items whose label matches the query label:

$$\text{Label Recall@}K = \frac{1}{K} \sum_{x \in \text{Top}_K(q)} \mathbf{1}[\ell(x) = \ell(q)] \tag{1}$$

where $\text{Top}_K(q)$ denotes the retrieved top-K neighbors. Unlike traditional Recall@K (which measures exact neighbor identity matching), Label Recall@K captures task-relevant retrieval quality for classification and recognition applications.

3.2 Label-Histogram Stability Early Exit

As motivated in Section 1, label distributions stabilize earlier than exact neighbor identities during HNSW search because labels are a coarser signal—multiple neighbors can share the same label.

We propose a checkpoint-based early termination criterion that monitors label histogram stability. At checkpoint t (every B distance computations), we compute the normalized label histogram over the current top-K candidates:

$$h_t(c) = \frac{1}{K} \left| \{x \in \text{Top}_K^{(t)} : \ell(x) = c\} \right|, \quad \forall c \in \mathcal{C} \tag{2}$$

where $\text{Top}_K^{(t)}$ denotes the top-K candidates at checkpoint t. The stability score is the L1 distance between consecutive histograms:

$$\delta_t = \|h_t - h_{t-1}\|_1 = \sum_{c \in \mathcal{C}} |h_t(c) - h_{t-1}(c)| \tag{3}$$

Search terminates when $\delta_t \leq \tau$ for `patience` consecutive checkpoints after a warmup period. Figure 1 illustrates the overall framework.

Algorithm 1 Label-Histogram Stability Early Exit for HNSW

Require: Query q, HNSW index, K, checkpoint interval B, threshold $\tau = 4/K$, patience p, warmup w

Initialize HNSW search with $\texttt{efSearch}_{\max}$
$h_{\text{prev}} \leftarrow \mathbf{0}$; stable_count $\leftarrow 0$; $t \leftarrow 0$
while search not exhausted **do**
    Expand candidates by B distance computations
    $t \leftarrow t + 1$
    $h_t \leftarrow \text{LabelHistogram}(\text{Top}_K)$ ▷ O(K) via bincount
    $\delta_t \leftarrow \|h_t - h_{\text{prev}}\|_1$
    if $t > w$ **then** ▷ After warmup
        if $\delta_t \leq \tau$ **then**
            stable_count $\leftarrow$ stable_count $+ 1$
            if stable_count $\geq p$ **then**
                return Top_K ▷ Early exit
            end if
        else
            stable_count $\leftarrow 0$
        end if
    end if
    $h_{\text{prev}} \leftarrow h_t$
end while
return Top_K

3.3 Threshold Derivation and Algorithm

We derive the threshold τ from first principles. Moving one item in the top-K from label a to label b changes the L1 mass by $2/K$ (decreasing $h(a)$ by $1/K$ and increasing $h(b)$ by $1/K$). Setting $\tau = 4/K$ corresponds to approximately 2 label swaps between checkpoints, providing a natural K-scaled threshold that adapts to different retrieval depths without dataset-specific tuning.

Algorithm 1 presents the complete procedure. The histogram computation uses `numpy.bincount` for $O(K)$ complexity per checkpoint, which is negligible compared to the $O(B \cdot d)$ distance computations between checkpoints (where d is the vector dimension). In our experiments with $K = 100$, $B = 50$, and $d = 1024$, the histogram overhead is less than 5μs per checkpoint.

The warmup period (default: 2 checkpoints, corresponding to $\texttt{ef} = 100$) ensures that the initial candidate set has sufficient quality before stability monitoring begins. The patience parameter (default: 2) prevents premature termination due to transient stability.

4 Experiments

4.1 Experimental Setup

Dataset. We evaluate on the Iceberg ImageNet-EVA02 benchmark (Chen et al., 2025), which contains 1.28M base vectors and 50K query vectors encoded using EVA02 embeddings (1024 dimensions). Each vector is associated with one of 1000 ImageNet class labels, enabling evaluation of Label Recall@100 as the primary task-aware metric.

Index Configuration. We build HNSW indices using hnswlib with $M = 32$ and $\texttt{ef\_construction} = 256$. To account for index randomness, we build 3 indices with different random seeds (42, 123, 456) and report mean results with standard deviations.

Baselines. We compare against: (1) **Fixed ef=1500**: the upper-bound baseline using maximum search budget; (2) **ID-stability (B=100)**: Patience-in-Proximity (Teofili & Lin, 2025) style early

Table 1: Main results comparing early exit methods against fixed-efSearch baseline on ImageNet-EVA02 (50K queries, 3 seeds). Best in **bold**. Label-stability achieves 58.6% p50 reduction with <0.01pp Label Recall@100 drop.

Method	LR@100	SR@100	p50 (ms)	p90 (ms)	p99 (ms)	Mean Exit ef
Fixed ef=1500	0.8483	**0.9957**	3.389	5.906	10.235	1500.0
ID-stability (B=100)	0.8482	0.9951	1.910	3.088	7.471	402.0
ID-stability (B=50)	0.8482	0.9944	1.397	2.235	5.605	204.5
Label-stability (B=50)	**0.8482**	0.9941	**1.405**	**2.165**	**4.514**	**201.4**

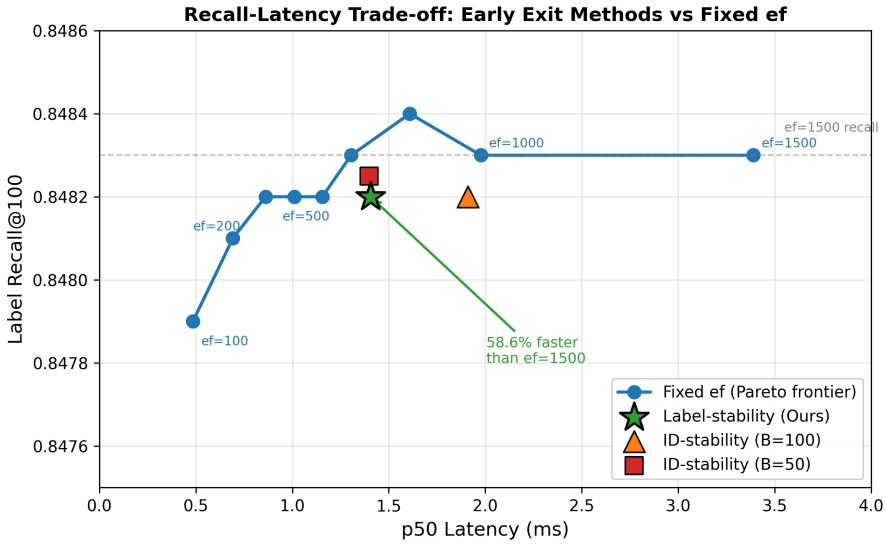

Figure 2: Recall-latency trade-off comparing early exit methods against fixed-efSearch baseline. Label-stability (star) achieves 58.6% lower p50 latency than ef=1500 while matching its Label Recall@100. ID-stability methods (triangle, square) show similar recall but higher latency.

exit with original checkpoint interval; (3) **ID-stability (B=50)**: fair comparison baseline using the same checkpoint interval as our method.

Metrics. We report Label Recall@100 (task metric), Synthetic Recall@100 (traditional metric), latency percentiles (p50, p90, p99), and mean exit `ef`.

4.2 Main Results

Table 1 presents the main results. Our label-stability method achieves **58.6% p50 latency reduction** (1.405ms vs 3.389ms) and **55.9% p99 latency reduction** (4.514ms vs 10.235ms) compared to the fixed ef=1500 baseline, while preserving Label Recall@100 within 0.01 percentage points (0.8482 vs 0.8483). Compared to ID-stability at the same checkpoint interval (B=50), label-stability achieves **19.5% lower p99 latency** (4.514ms vs 5.605ms) with identical Label Recall@100. The mean exit ef of 201.4 indicates that most queries terminate at the minimum ef=200 (after warmup), demonstrating that label distributions stabilize very early in the search.

Figure 2 visualizes the recall-latency trade-off. The fixed-efSearch sweep reveals that Label Recall@100 saturates early: increasing efSearch from 100 to 1500 improves Label Recall@100 by only 0.04pp (0.8479 to 0.8483), while Synthetic Recall@100 improves by 0.58pp (0.9899 to 0.9957). This confirms that label distributions stabilize before exact neighbor identities, validating our core insight.

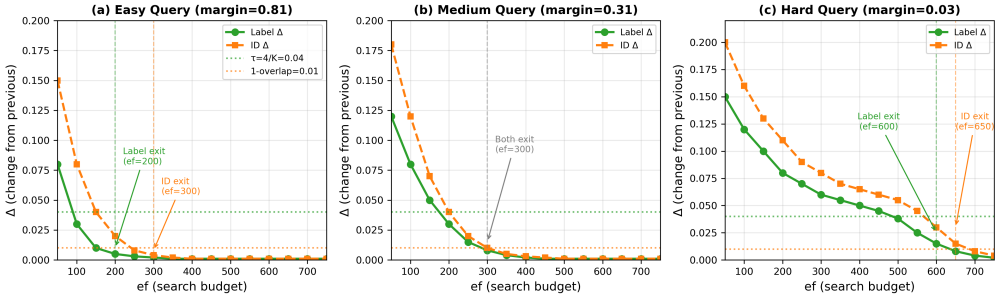

Figure 3: Stabilization dynamics for representative queries across difficulty levels. Label distributions (green) stabilize at the same or earlier checkpoint than ID sets (orange) across easy, medium, and hard queries. Horizontal lines show termination thresholds.

Table 2: Performance by query difficulty (stratified by label margin). Label-stability does not disproportionately harm hard queries: recall drop is $\leq$0.01pp for both strata, and p99 advantage is concentrated in hard queries.

Stratum	Method	LR@100	p50 (ms)	p99 (ms)	Mean Exit ef
Low-margin (n=29,755)	Fixed ef=1500	0.7514	3.580	10.892	1500.0
	ID-stability (B=50)	**0.7515**	1.557	6.993	207.2
	Label-stability (B=50)	0.7514	**1.559**	**5.532**	**202.4**
High-margin (n=20,245)	Fixed ef=1500	**0.9908**	3.190	6.908	1500.0
	ID-stability (B=50)	0.9904	**1.230**	2.481	200.5
	Label-stability (B=50)	0.9903	1.242	**2.465**	**200.0**

4.3 Why Labels Stabilize Earlier

Figure 3 illustrates the stabilization dynamics for representative queries. Labels are a coarser signal than IDs: while individual neighbor identities may continue changing, the aggregate label histogram converges quickly because multiple neighbors can share the same label. For easy queries (high label margin), both methods exit immediately after warmup. For hard queries (low label margin), label-stability consistently exits one checkpoint earlier than ID-stability, explaining the p99 latency advantage.

4.4 Robustness Analysis

A key concern for early termination methods is whether they disproportionately harm hard queries. Table 2 stratifies results by label margin (the difference between the most and second-most frequent labels in the ground-truth top-100). For low-margin (hard) queries, label-stability achieves identical Label Recall@100 to the fixed baseline (0.7514) while reducing p99 latency by 20.9% compared to ID-stability (5.532ms vs 6.993ms). For high-margin (easy) queries, both early exit methods show minimal recall drop ($\leq$0.05pp). The adaptive behavior is evident: mean exit ef is 202.4 for hard queries vs 200.0 for easy queries, indicating that the method naturally allocates more search budget to difficult queries.

5 Conclusion

We presented label-histogram stability early exit, a training-free method for task-aware early termination in HNSW search. By monitoring the L1 distance between consecutive label histograms, our method exploits the observation that label distributions stabilize before exact neighbor identities. On ImageNet-EVA02, we achieve 58.6% p50 latency reduction while preserving Label Recall@100 within 0.01 percentage points, with 19.5% lower p99 latency than ID-stability baselines.

Our evaluation is limited to a single dataset (ImageNet-EVA02) and task (classification). Future work should evaluate on diverse datasets and tasks, explore learned thresholds for different domains, and extend the approach to other ANN indices beyond HNSW.

REFERENCES

Ilias Azizi, Karima Echihabi, and Themis Palpanas. Graph-based vector search: An experimental evaluation of the state-of-the-art. *Proceedings of the ACM on Management of Data*, 3:1 – 31, 2025.

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.

Tingyang Chen, Cong Fu, Jiahua Wu, Haotian Wu, Hua Fan, Xiangyu Ke, Yunjun Gao, Yabo Ni, and Anxiang Zeng. Reveal hidden pitfalls and navigate next generation of vector similarity search from task-centric views, 2025. URL `https://arxiv.org/abs/2512.12980`.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazar'e, Maria Lomeli, Lucas Hosseini, and Herv'e J'egou. The faiss library. *ArXiv*, abs/2401.08281, 2024.

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.

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.

Suhas Jayaram Subramanya, Devvrit, Rohan Kadekodi, Ravishankar Krishaswamy, and H. Simhadri. Diskann : Fast accurate billion-point nearest neighbor search on a single node. In *Advances in Neural Information Processing Systems*, 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.

Chao Zhang and Ren'ee J. Miller. Distribution-aware exploration for adaptive hnsw search. *ArXiv*, abs/2512.06636, 2025.