Declarative Recall for Approximate Vector Search

Manos Chatzakis
Supervised by Prof. Themis Palpanas
Université Paris Cité, LIPADE, F-75006 Paris, France
manos.chatzaki@gmail.com

ABSTRACT

Approximate nearest neighbor vector search (ANNS) is the backbone of a wide range of data management tasks, but its practical use remains challenging because it requires extensive manual tuning. Users must map algorithm-specific hyperparameters to the desired recall–latency trade-off, a process that is costly and often suboptimal for individual queries. This thesis addresses these challenges by introducing declarative recall for ANNS through early termination, allowing users to specify only the desired recall target without requiring hyperparameter tuning. We first present DARTH, which establishes declarative recall for ANNS using recall prediction by adaptively terminating each query's search once the target recall has been reached. We then present DARTH+, which improves DARTH's prediction features and training efficiency and introduces probabilistic quality guarantees for individual-query results through quantile regression. Finally, we demonstrate the state-of-the-art performance of our approaches and discuss future directions.

VLDB Workshop Reference Format:
Manos Chatzakis. Declarative Recall for Approximate Vector Search. VLDB 2026 Workshop: PhD Workshop.

VLDB Workshop Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://mchatzakis.github.io/declarative-recall/.

1 INTRODUCTION

Approximate nearest neighbor search (ANNS) is a core operation in many data management applications [13]. Given a query vector in a high-dimensional space, ANNS aims to identify its top-k (k-NN) nearest vectors, while relaxing the exactness constraint to enable extremely fast search. This efficiency has led to the widespread academic and industrial adoption of ANNS methods. However, ANNS algorithms typically expose several hyperparameters that users must tune to achieve the desired performance. Moreover, ANNS exhibits a fundamental trade-off between latency and recall, i.e., the fraction of correctly identified nearest neighbors among the top-k results. As a result, users targeting specific recall levels must still perform extensive manual hyperparameter tuning. They typically construct a mapping between ANNS hyperparameter values and the average recall achieved over a tuning query workload [15].

Proceedings of the VLDB Endowment. ISSN 2150-8097.

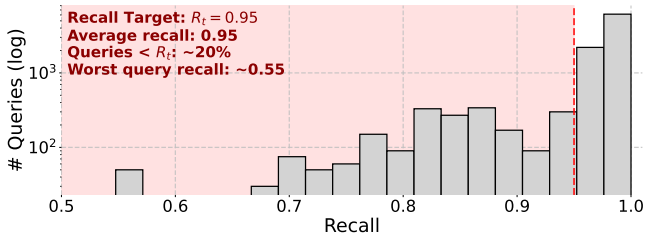


Figure 1: Query recall distribution for DEEP100M dataset (HNSW, $k = 50$, 1000 queries, $R_t = 0.95$). Average recall is equal to target, but using same hyperparameters for all queries leads to several queries with recalls lower than the target.

Unfortunately, this tuning process raises several challenges. First, constructing the mapping requires exploring a complex multidimensional hyperparameter space, which is slow and resource-intensive. Users typically select an ANNS algorithm and a tuning query workload, initialize the algorithm's hyperparameters for the desired recall target, construct and search the index, and then examine the average achieved recall. If the average recall is not close enough to the target, new values for the parameters should selected, making the creation of this mapping a task that requires several hours. Second, and more importantly, the mapping is based on average recall over the tuning workload, which can lead to suboptimal results for individual queries even after expensive tuning.

Figure 1 illustrates this phenomenon. The figure shows the query recall distribution of 1000 queries from the DEEP100M [1] dataset for $k = 50$ and $R_t = 0.95$, using HNSW [12] hyperparameters tuned to achieve precisely 0.95 average recall. Although the average recall matches the target, approximately 20% of queries fall below it, with the worst-performing query reaching only 0.55 recall. This happens because using the same hyperparameters for all queries can waste effort on easy queries while causing harder queries to miss the recall target, even when the workload-level average recall is satisfied.

Through this thesis, our goal is to develop a solution that effectively addresses the challenges described above by introducing the concept of declarative recall for ANNS. With declarative recall, ANNS is made available to users through an interface that simply requires them to declare a desired recall target, without worrying about tuning the hyperparameters of the underlying ANNS method. Our contributions begin with DARTH [4], a novel method for ANNS that provides declarative recall through recall prediction and early termination [2]. DARTH leverages a novel set of input features together with an ML-based recall predictor to adaptively predict the recall of individual queries during search and terminate the search once the declared recall target has been reached. DARTH is designed to operate seamlessly with a variety of vector search indexes, including k-NN graphs, e.g., HNSW [12], and IVF indexes, e.g., FAISS-IVF [7].

Furthermore, we introduce DARTH+ [5], which improves upon DARTH in several ways. It introduces novel recall prediction features and significantly reduces training time: training of the predictor completes within only a few minutes, even for datasets containing hundreds of millions of vectors. At the same time, DARTH+ provides accurate probabilistic guarantees on individual predicted query recalls through quantile regression.

Our work constitutes the first general solution to the problem: works for k-NN graph and IVF indexes, supports all common distance measures, allows users to choose a different target recall for each query at query time, and users do not need to set any parameters. The experimental evaluation demonstrates that our methods achieve the state-of-the-art performance, significantly outperforming all baselines. We conclude our paper with a discussion of our plans for future work.

2 BACKGROUND AND RELATED WORK

Nearest Neighbor Vector Search. Given a collection of vectors V, a query q, a distance (or similarity) measure D, and a number k, k-Nearest Neighbor vector Search (NNS) refers to the task of finding the k most similar vectors (nearest neighbors) to q in V, according to D. The nearest neighbors can be exact [3] or approximate [8] (in the case of Approximate Nearest Neighbor Search, ANNS). When dealing with approximate search, which is the focus of this thesis, search quality is evaluated using two key measures: (i) *result quality*, usually quantified using recall (the fraction of actual nearest neighbors that are correctly identified) (ii) *latency*, i.e., the time required to perform the query search.

ANNS Indexes. ANNS tasks are efficiently addressed using specialized ANNS indexes [13]. These approaches construct an index structure over the vector collection V, enabling rapid query answering times. The most popular categories for ANNS indexes are the k-NN graph (e.g., HNSW [12]) and IVF (e.g., FAISS-IVF [7]). The k-NN graph indexes create a graph over V by representing vectors as nodes, with edges between them reflecting some measure of proximity between the nodes, and they perform ANNS by performing greedy search with effort ef over the constructed graph structure. The IVF indexes cluster the vectors of V into *nlist* partitions (e.g., using k-means), and then search the vectors of the first *nprobe* closest partitions of a query.

Declarative Recall Declarative recall supports queries of the form $ANNS(q, I, k, R_t)$, where q is the query vector, I is an ANNS index, k is the number of nearest neighbors to be retrieved, and R_t is the declarative target recall value. The objective is to approximately retrieve the k-nearest neighbors of q using I, achieving a recall of at least R_t with high probability, while optimizing the search time. We assume that the user-declared target recall R_t should be attainable by the index I. Specifically, if the recall that the index I achieves using plain search (i.e., search without early termination) for the query q is R_q^h, then $R_t \leq R_q^h$. This condition is easy to satisfy practically by utilizing available heuristics for hyperparameter settings that enable very high recalls (e.g., > 0.99).

3 OUR WORK

Completed Work. We observe that a query configured with parameters that enable very high recall will naturally achieve all lower recall levels during the search process. This behavior is illustrated in Figure 2, where each curve shows the recall progression of a query on the SIFT100M [9] dataset using an HNSW index. For example, if the algorithm were stopped early, at 2 ms, Query 1, shown by the blue curve, would achieve 0.8 recall. By contrast, the "easy" Query 2 reaches recall 1 around the 1.8 ms mark and has already reached 0.8 recall by the 1 ms mark. The time spent after reaching the target recall is therefore wasted. In contrast, Query 3 reaches only approximately 0.6 recall at the 2 ms mark. Figure 2 shows that, for each query, multiple recall targets can be achieved well before the ANNS search naturally completes. We observe that, if we can estimate the recall of a query at any point during the search, we can provide an efficient declarative recall solution for any query and any recall target, without the need for parameter tuning. However, determining the current recall is not trivial because different queries have different levels of hardness (i.e., required search efforts) and reach the target recall at different points in time.

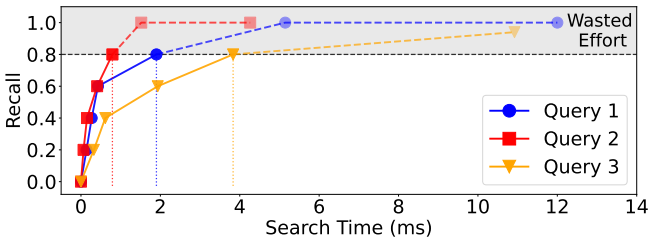


Figure 2: Early Termination margins for target recall 0.8, for 3 queries on the HNSW index. The last point of each curve represents the point where the HNSW search normally terminates: much after reaching the target recall.

The above key observation is leveraged by DARTH [4]. DARTH uses a carefully designed recall predictor model, which is dynamically invoked at selected points during the search to predict the current recall and decide whether to early terminate or continue the search, based on the specified recall target. Designing an early termination approach is complex, as it requires addressing multiple challenges to develop an efficient and accurate solution. First, we identify the key features of the ANNS search that serve as reliable predictors of a query's current recall at any point during the search. These features capture both the progression of the search, by tracking metrics such as distance calculations, and the quality of the nearest neighbors found, by examining specific neighbors and their distance distributions, and are presented in Table 1 for k-NN graphs (features are similar and generalizable for other types of ANNS indexes). Note that generating the training data with those features takes only a few minutes, and if groundtruth generation is needed for the recall calculation, it can be also extremely fast through optimized libraries for exact NNS [6].

Moreover, we need to select an appropriate recall predictor model to train on our data. We choose a Gradient Boosting Decision Tree (GBDT) [10] because of its strong performance in regression tasks and efficient training time. The GBDT recall predictor has extremely fast training times, which are negligible compared to typical ANNS index creation times.

However, an accurate recall predictor alone is not sufficient to provide an efficient solution for declarative recall: if the recall predictor is invoked too frequently, the cost of inference will cancel

Type	Features	Description
Index	*nstep*	Search step
	ndis	No. distance calculations
	ninserts	No. updates in the NN result set
NN Distance	$firstNN$	Distance of first NN found
	$closestNN$	Distance of current closest NN
	$furthestNN$	Distance of current furthest (k-th) NN
NN Stats	*avg*	Average of distances of the NN
	var	Variance of distances of NN
	med	Median of distances of NN
	perc25	25th percentile of distances of NN
	perc75	75th percentile of distances of NN

Table 1: Input features of DARTH's recall predictor.

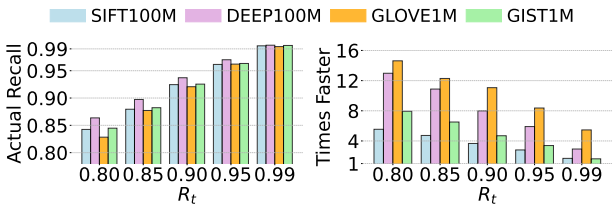
out the benefits of early termination. To address this challenge, we develop an adaptive prediction interval method, which dynamically adjusts the invocation frequency. The method invokes the recall predictor more frequently as the current recall approaches the recall target, ensuring both accuracy and efficiency.

Figure 3 provides an overview of DARTH's performance on HNSW, showing the actual average recall achieved and the corresponding speedups, compared to plain HNSW search without early termination for each corresponding index, for each recall target R_t across different datasets and for $k = 50$.

The graphs show that DARTH reaches and exceeds each R_t, while also delivering significant speedups of up to 15×, with an average of 6.8× and a median of 5.7×, compared to plain HNSW search without early termination. As expected, the speedup decreases for higher recall targets, since more search effort is required before termination as R_t increases. One of the major advantages of DARTH as a run-time adaptive approach is that it can adjust the termination point of the search for harder queries without requiring any additional configuration. In contrast, competing approaches answer queries using static parameters that are fixed for all queries in a given workload and are selected based on the validation query workload. We demonstrate this behavior in practice through a wide range of experiments that compare the performance of the different approaches on query workloads of increasing hardness.

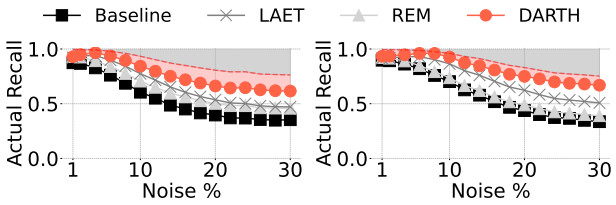
Figure 4 reports the actual recall achieved by each method for $k = 50$ and $R_t = 0.90$ across all datasets, as query hardness, represented by the added gaussian noise percentage, increases for each query workload from 1% to 30%. The graphs also show the actual recall achieved by the plain HNSW index, shown by the red line, which represents the maximum attainable recall for each noise. The results demonstrate that DARTH is the most robust approach, reaching recall very close to the declared R_t across the entire range of noise values, and especially for noise values where R_t is attainable by the plain HNSW index, i.e., up to 10–12%. The performance of the competing approaches deteriorates considerably, achieving recall values far from the target, especially as queries become harder under higher-noise configurations. DARTH achieves this level of robustness by considering a wide variety of search features to determine whether to apply early termination, rather than relying solely on the data distribution, and, at the same time, it adaptively decides the termination point for each query individually.

Ongoing Work. DARTH+ [5] introduces several optimizations, achieving similar, and sometimes better, recall predictor performance with much faster training. This is achieved through a dynamic logging frequency mechanism that collects training data more frequently during the most informative phases of the search, when recall is high, and less frequently during the early stages of



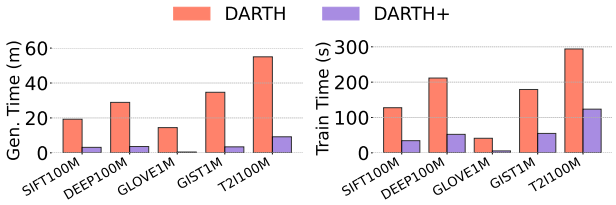
(a) Achieved Recalls **(b) Speedup**

Figure 3: DARTH summary for HNSW ($k = 50$).



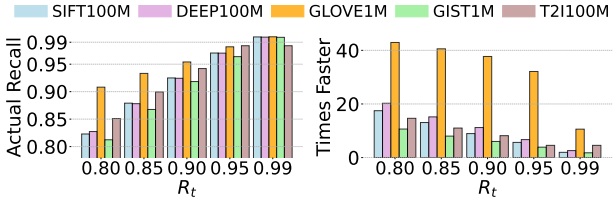
(a) SIFT100M **(b) DEEP100M**

Figure 4: Recall for varying noise ($R_t = 0.90$, $k = 50$). The red line indicates the maximum attainable recall from the plain HNSW index.



(a) Generation Time **(b) Training Time**

Figure 5: DARTH and DARTH+ training data generation and training times ($k = 50$).



(a) Achieved Recall **(b) Speedup**

Figure 6: DARTH+ summary for IVF ($k = 50$).

the search, when recall is low. In addition, DARTH+ adaptively terminates the training data collection process for queries that have already reached maximum recall, substantially reducing both the training data size and the training time. DARTH+ also introduces a novel set of query-dimension-aware recall prediction features, further improving accuracy. Finally, DARTH+ provides probabilistic quality guarantees on the achieved recall for individual queries, a feature designed for recall-sensitive applications where it is critical for individual queries to reach the target recall, such as retrieval in legal and medical domains [14]. DARTH+ employs a quantile predictor model that provides a lower bound on recall at each point in the search where a recall prediction is used. This lower bound comes with a probabilistic guarantee on its accuracy, e.g., the lower bound is correct with probability 90%. DARTH+ can then terminate

the search procedure based on this lower bound. This provides a guarantee on the quality of the results of each query: the query reaches the recall target with a probability specified by the user at execution time. We note that the quantile predictor is trained on the same data as our recall predictor, with essentially no additional overhead to the overall training procedure of DARTH+.

Our experiments demonstrate that both DARTH and DARTH+ exhibit similar recall performance, with DARTH+ performing slightly better, having on average only 10% of its queries fall below the target recall, compared to 13% for DARTH. At the same time, DARTH+ achieves significantly faster training times, thanks to the dynamic sampling interval and high-recall cutoffs it uses during training. This improvement is attributed to two design choices. First, DARTH+ uses a dynamic sampling interval that does not affect predictor performance, yet reduces the number of collected samples. Second, DARTH+ applies a high-recall cutoff that terminates training queries when the number of distance calculations performed after reaching the highest possible recall exceeds 30% of the total distance calculations performed for the other recall ranges. Together, these mechanisms significantly reduce the amount of training data needed for model training.

Figure 5(a) presents the time required to generate the training data for DARTH and DARTH+ across all datasets and values of k. We observe that DARTH+ is up to 28.5× faster than DARTH in training data generation, with an average speedup of 11.5× and a median speedup of 7.7×. DARTH+ also produces substantially smaller training datasets, containing on average 14.5× fewer training samples than DARTH. Figure 5(b) shows the time required to train the recall predictor model using the training data generated by DARTH and DARTH+. The results show that, thanks to the updated training mechanism of DARTH+ and the significantly smaller training datasets it requires, recall predictor training can be completed in only a few seconds. Across all datasets and values of k, DARTH+ achieves up to a 9.7× speedup in model training compared to DARTH, with an average speedup of 6.1× and a median speedup of 5.4×. Overall, our analysis shows that DARTH+ is substantially faster to train while matching the recall performance of DARTH. Regarding the probabilistic quality guarantee our experiments demonstrate that DARTH+ provides very accurate empirical accuracy, deviating only by 0.2% from the required guarantee on average.

Finally, DARTH+ better supports the IVF index (the implementation in DARTH was preliminary). Figure 6 presents the performance of DARTH+ on all of our datasets for IVF and $k = 50$. Figure 6(a) shows that the recall achieved by DARTH+ for IVF, using 1K queries, always meets or exceeds the target. Figure 6(b) depicts the corresponding speedups achieved by DARTH+. DARTH+ achieves speedups of up to 44.9× compared to plain IVF search, with an average speedup of 13.4× and a median speedup of 9.6×.

Future Work. Our immediate future focus is to extend declarative recall to filtered vector search. In filtered vector search, structural attributes are combined with high-dimensional vectors, and the goal is to return the top-k most similar vectors to a query that also satisfy an attribute-based filter. This setting is more challenging than standard ANNS because filters can have different selectivities, i.e., different fractions of vectors satisfy each filter, and different correlations with the vector space. Correlation captures whether the closest vectors to the query are also those that satisfy the filter, and can be positive, negative, or absent (no correlation) [11]. Different combinations of selectivity, correlation, and recall target can require very different search efforts, even for the same query vector. For example, negative correlation with high selectivity can require much more search than positive correlation with low selectivity for the same recall target. As a result, selecting predefined static parameters for filtered search becomes substantially more difficult, and potentially infeasible, because selectivity may not be known in advance and correlations are generally unknown before starting the search. Therefore, designing a parameter-free, declarative-recall approach for filtered ANNS is an important future direction.

We will tackle this problem by expanding our framework with new solutions that can effectively detect and exploit the different correlations and selectivities in order to achieve high-accuracy results in filtered ANNS, as well.

4 CONCLUSIONS

The goal of this thesis is to provide a complete and comprehensive declarative recall interface for vector ANNS. We propose DARTH, the first approach for declarative recall through recall prediction, and exhibits SotA performance. We are developing DARTH+, which improves the efficiency of DARTH and extends it with probabilistic quality guarantees on individual query recalls. In future work, we will develop declarative recall solutions for filtered ANNS.

REFERENCES

[1] Artem Babenko and Victor Lempitsky. 2016. Efficient indexing of billion-scale datasets of deep descriptors. In *CVPR*.

[2] Manos Chatzakis, Francesca Del Gaudio, Sophia Sideri, and Themis Palpanas. 2026. The Quest for Faster ANN Vector Search. In *EDBT*.

[3] Manos Chatzakis, Panagiota Fatourou, Eleftherios Kosmas, Themis Palpanas, and Botao Peng. 2023. Odyssey: A Journey in the Land of Distributed Data Series Similarity Search. *PVLDB* (2023).

[4] Manos Chatzakis, Yannis Papakonstantinou, and Themis Palpanas. 2025. DARTH: Declarative Recall Through Early Termination for Approximate Nearest Neighbor Search. *SIGMOD* 3, 4 (2025).

[5] Manos Chatzakis, Yannis Papakonstantinou, and Themis Palpanas. 2026. DARTH+: Approximate Nearest Neighbor Search with Declarative Recall and Quality Guarantees. (2026). https://hal.science/hal-05566027

[6] Francesca Del Gaudio, Manos Chatzakis, Gayathiri Ravendirane, Botao Peng, and Themis Palpanas. 2026. DaiSy: A Library for Scalable Data Series Similarity Search. *arXiv preprint arXiv:2603.27719* (2026).

[7] Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2025. The Faiss Library. *IEEE Transactions on Big Data* (2025).

[8] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2020. Return of the lernaean hydra: Experimental evaluation of data series approximate similarity search. *PVLDB* (2020).

[9] Hervé Jégou, Romain Tavenard, Matthijs Douze, and Laurent Amsaleg. 2011. Searching in one billion vectors: re-rank with source coding. In *ICASSP*.

[10] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *NeurIPS* (2017).

[11] Duo Lu, Helena Caminal, Manos Chatzakis, Yannis Papakonstantinou, Yannis Chronis, Vaibhav Jain, and Fatma Özcan. 2026. An In-Depth Study of Filter-Agnostic Vector Search on a PostgreSQL Database System. *SIGMOD* (2026).

[12] Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *PAMI* (2018).

[13] James Jie Pan, Jianguo Wang, and Guoliang Li. 2024. Survey of vector database management systems. *VLDBJ* (2024).

[14] Eugene Yang. 2024. Contextualization with SPLADE for high recall retrieval. In *SIGIR*.

[15] Tiannuo Yang, Wen Hu, Wangqi Peng, Yusen Li, Jianguo Li, Gang Wang, and Xiaoguang Liu. 2024. VDTuner: Automated Performance Tuning for Vector Data Management Systems. In *ICDE*.