WunDeeDB.jl: A easy to use, zero config, WAL, SQLite backend vector database

March 21 2025

Summary

WunDeeDB.jl is a Julia package that provides a disk-backed system for storing, searching, and managing embedding vectors at scale, influenced by disk-oriented graph-based ANN techniques (Pan, Sun, and Yu 2023; Jayaram Subramanya et al. 2019; Singh et al. 2021) and the broader insights from hierarchical small-world graphs (Malkov and Yashunin 2018; Wang et al. 2021). By maintaining embeddings in a SQLite database and optionally using graph-based indices, WunDeeDB.jl minimizes in-memory overhead while supporting efficient similarity searches on commodity hardware. Its design also facilitates integration with common vector-database or ML pipelines that rely on embedding retrieval.

In contrast to fully in-memory approaches, WunDeeDB.jl leverages disk-based storage and user-configurable adjacency (e.g., HNSW, LM-DiskANN, or fallback linear search), allowing large-scale data to be handled without saturating RAM. It supports incremental insertions and deletions, ensuring the index remains up-to-date as datasets evolve. By combining these disk-native strategies with tunable BFS expansions and adjacency pruning, WunDeeDB.jl enables robust nearest neighbor searches for high-dimensional embeddings.

Features include:

- **SQLite-backed embeddings** with automatic consistency checks on dimension and data type.
- **Optional ANN indexing** (HNSW, LM-DiskANN) or linear fallback, selectable at DB initialization.
- **Incremental insert/delete** operations that update disk-based adjacency structures to keep pace with dataset changes.
- **Configurable BFS expansions** (e.g., `EF_SEARCH`, `EF_CONSTRUCTION`) to balance search speed vs. recall.

With these capabilities, **WunDeeDB.jl** offers a practical and scalable solution for disk-based embedding management, building on research showing the viability of disk-native approaches for large ANN indexes (Pan, Sun, and Yu 2023;

Jayaram Subramanya et al. 2019; Singh et al. 2021).

Statement of Need

Approximate Nearest Neighbor (ANN) search is a key element in recommendation systems, large-scale retrieval, and embedding-based machine learning (Wang et al. 2021). Traditional in-memory approaches often face significant memory demands and slower scaling when dealing with more points than can fit in core memory. By persisting adjacency structures on disk rather than in RAM, WunDeeDB.jl tackles these bottlenecks—building on disk-based ANN research (Pan, Sun, and Yu 2023; Jayaram Subramanya et al. 2019; Singh et al. 2021) and provides:

1. **Reduced Memory Overhead**: Only a fraction of data resides in memory, making it feasible to handle larger datasets on commodity hardware.

2. **Dynamic Updates**: Graph-based indexing supports insertions and deletions, allowing adaptation to evolving or streaming data.

3. **High Recall**: Adjusting BFS expansions and adjacency parameters yields near state-of-the-art accuracy in neighbor retrieval.

4. **Scalable & Simple Architecture**: Built using Julia's performance ecosystem, WunDeeDB.jl integrates disk operations with numeric libraries, and is straightforward to install.

This approach benefits practitioners needing large-scale nearest neighbor indices without requiring specialized clusters or massive RAM. WunDeeDB.jl's minimal setup, clear usage patterns, and example-driven documentation lower the barrier to entry for embedding-based ANN workflows in resource-constrained environments.

Jayaram Subramanya, Suhas, 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. https://doi.org/https://doi.org/10.1145/3543507.3583552.

Malkov, Yu A, 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): 824–36. https://doi.org/10.1109/TPAMI.2018.2889473.

Pan, Yu, Jianxin Sun, and Hongfeng Yu. 2023. "Lm-Diskann: Low Memory Footprint in Disk-Native Dynamic Graph-Based Ann Indexing." In *2023 Ieee International Conference on Big Data (Bigdata)*, 5987–96. IEEE. https://doi.

org/10.1109/BigData59044.2023.10386517.

Singh, Aditi, Suhas Jayaram Subramanya, Ravishankar Krishnaswamy, and Harsha Vardhan Simhadri. 2021. "Freshdiskann: A Fast and Accurate Graph-Based Ann Index for Streaming Similarity Search." *arXiv Preprint arXiv:2105.09613*. https://doi.org/10.48550/arXiv.2105.09613.

Wang, Mengzhao, Xiaoliang Xu, Qiang Yue, and Yuxiang Wang. 2021. "A Comprehensive Survey and Experimental Comparison of Graph-Based Approximate Nearest Neighbor Search." *arXiv Preprint arXiv:2101.12631*. https://doi.org/https://doi.org/10.14778/3476249.3476255.