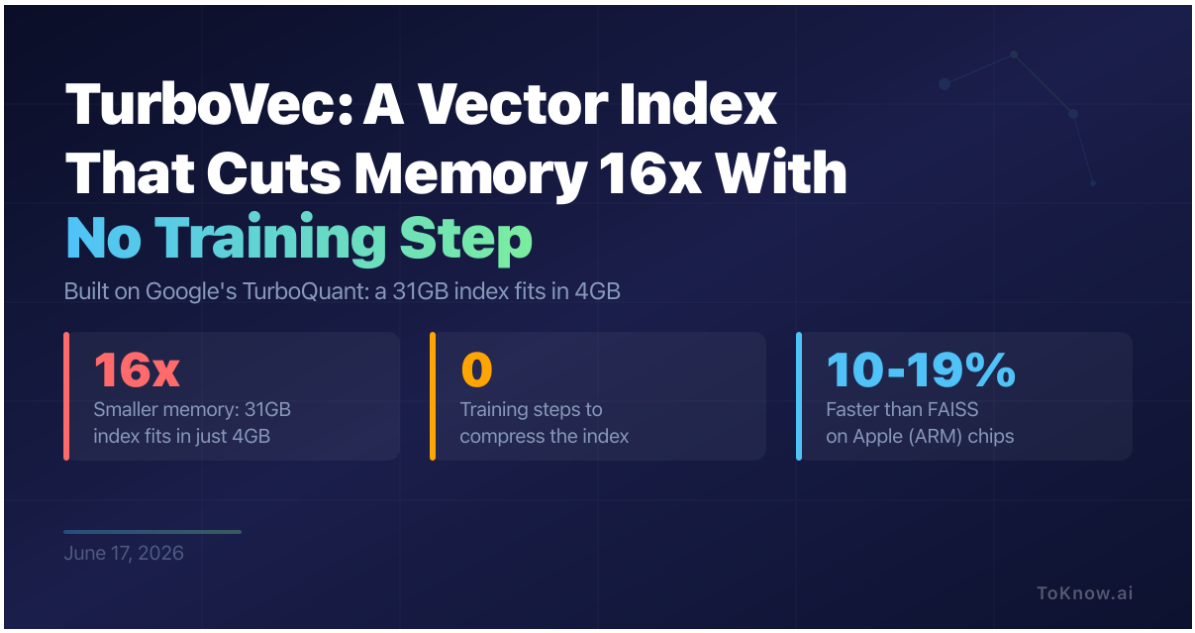
TurboVec: A Vector Index That Cuts Memory 16x With No Training Step

Kabui, Charles

2026-06-17

Read at ToKnow.ai



TurboVec is an open-source vector index, the store behind similarity search, built in Rust with Python bindings on Google Research's TurboQuant algorithm. It packs a 10-million-vector collection from 31GB down to 4GB, a 16x cut, by squeezing each vector of 1,536 numbers from 6,144 bytes to 384 bytes. The trick: TurboQuant rotates every vector by the same fixed random matrix, which makes each number follow a known statistical shape no matter the data.

Since that shape is predictable, the best buckets to sort each number into are fixed in advance, so there is no training step. On OpenAI text embeddings it edges out FAISS, Meta's standard vector-search library, on top-match accuracy by 0.2 to 1.9% and runs 10 to 19% faster on Apple chips.

Vector search powers retrieval for chatbots and semantic search, but existing tools make you either hold everything in expensive memory or run a separate training pass to build a compressed codebook first. TurboVec drops both. A 31GB index that fits in 4GB runs on a laptop, not a server, and new vectors are searchable the moment you add them, with no rebuilds as it grows. Swapping it in for LangChain, LlamaIndex, or Haystack is a one-line import change.

What makes this work is that TurboQuant is data-oblivious: it reaches near-optimal compression, within 2.7x of the theoretical limit set by Shannon's source coding, from the math of random rotations, not from your data. The same algorithm already cut LLM memory use 6x; here it strips the train-and-rebuild cycle out of search.

Sources:

- TurboVec on GitHub
- TurboQuant: Online Vector Quantization with Near-optimal Distortion Rate (arXiv)
- turbovec on PyPI
- turbovec on crates.io
- RaBitQ: Quantizing High-Dimensional Vectors with a Theoretical Error Bound (arXiv, SIGMOD 2024)
