A Vertical Layout for Vector Similarity Search

Leonardo Kuffo*, **Peter Boncz**
CWI Database Architectures group

DBDBD ‘24

Vector Similarity Search











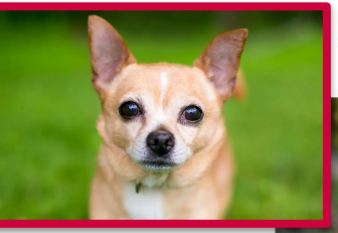






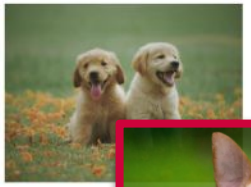
CWI

QUERY

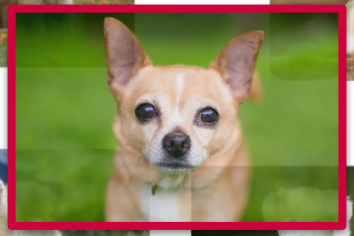


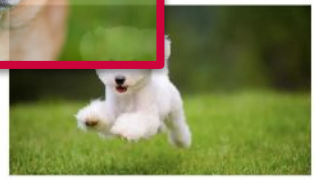
CWI

QUERY





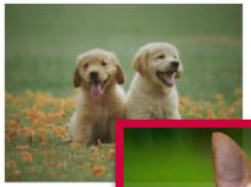


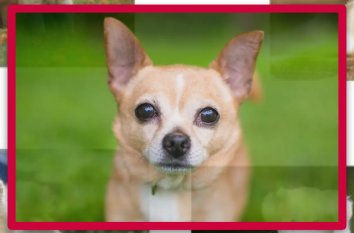


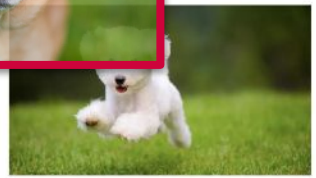


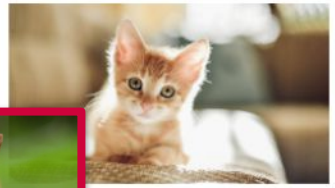

CWI

Result









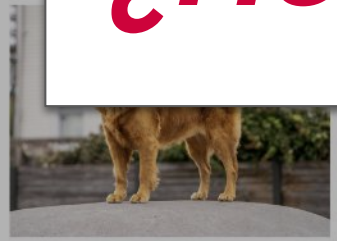
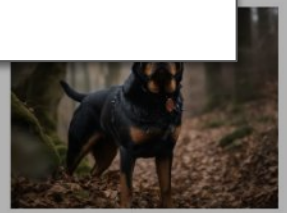
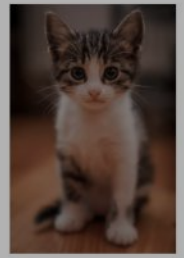




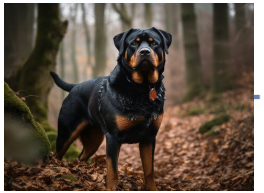


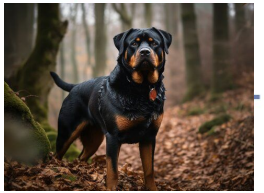




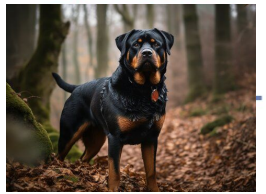






¿How?





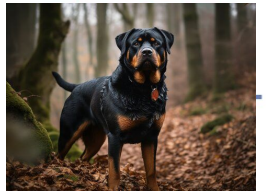
Embedding Model



Embedding Model

0.6	0.3	0.1	...
0.8	0.5	0.3	...
0.4	0.2	0.9	...

Objects as vectors

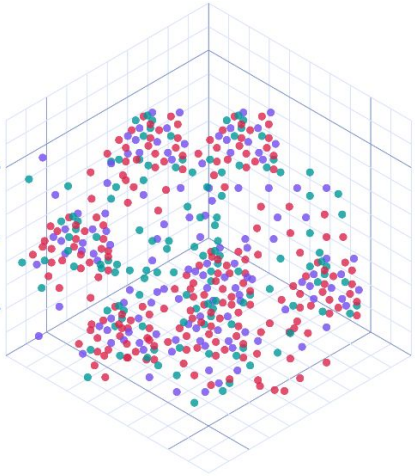


Embedding Model

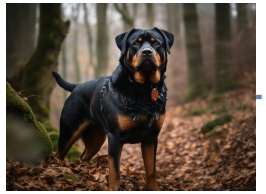
0.6	0.3	0.1	...

0.8	0.5	0.3	...

0.4	0.2	0.9	...



Objects as vectors

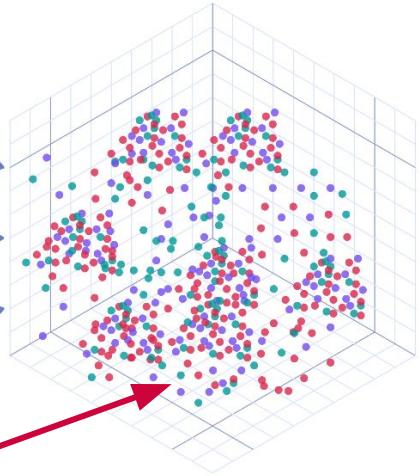


Embedding Model

0.6	0.3	0.1	...

0.8	0.5	0.3	...

0.4	0.2	0.9	...



Objects as vectors

0.1	-0.3	0.8	...

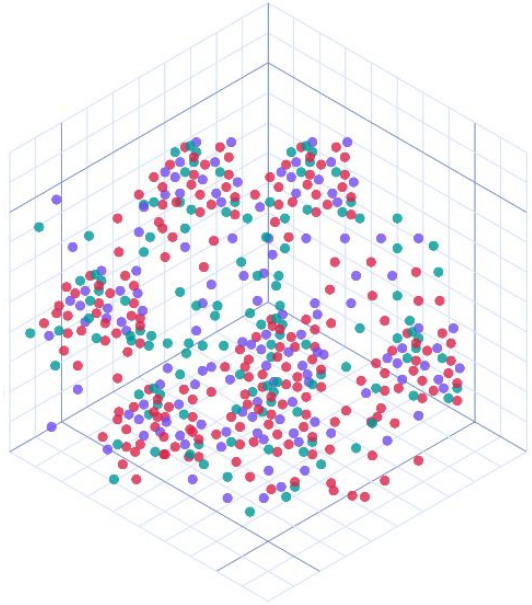
Euclidean Distance, Cosine Similarity, Dot Product, Manhattan...

Cool use cases

- Information Retrieval
- Recommendation Systems
- **In the LLMs era**: Retrieval Augmented Generation (RAG)

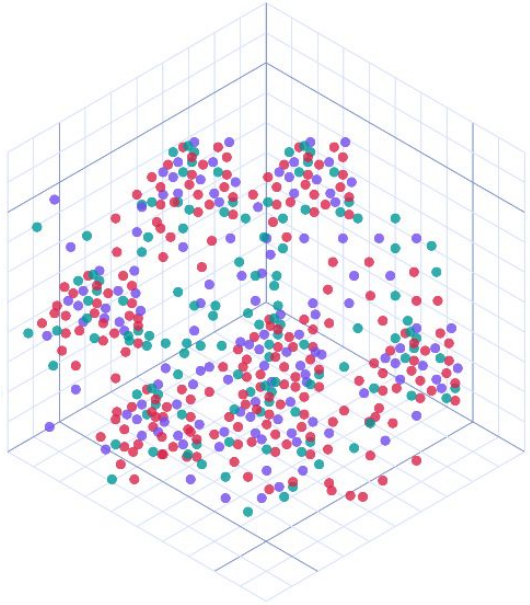
CWI

Vector similarity search is expensive



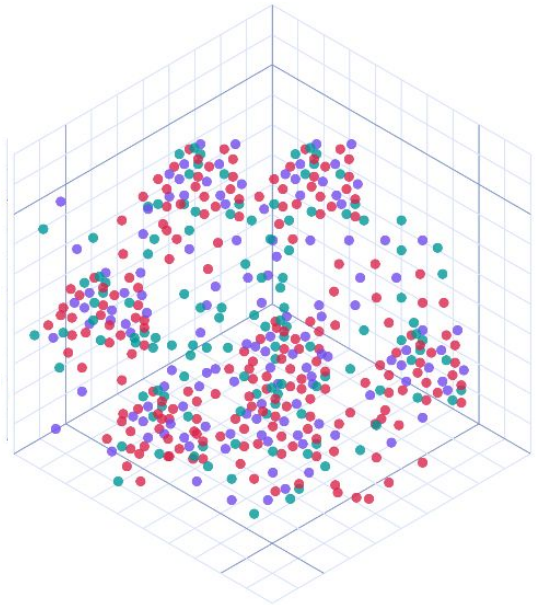
CWI

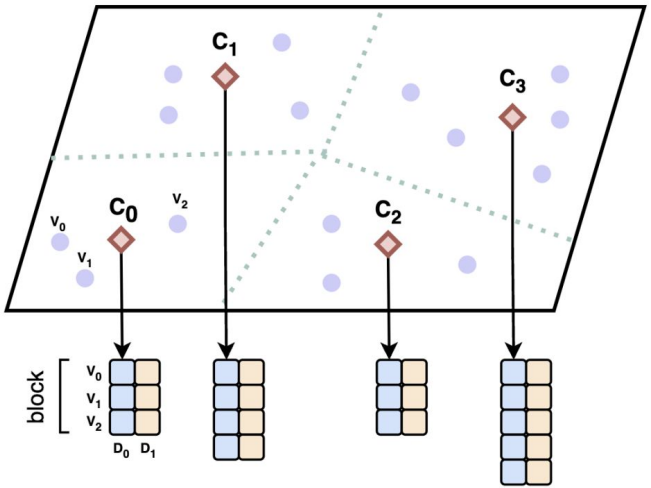
Vector similarity search is expensive



...unless we trade-off exactness

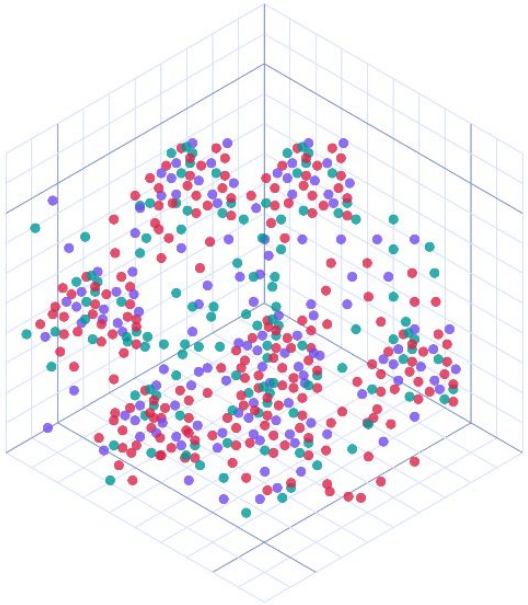
Vector indexes to speed up search

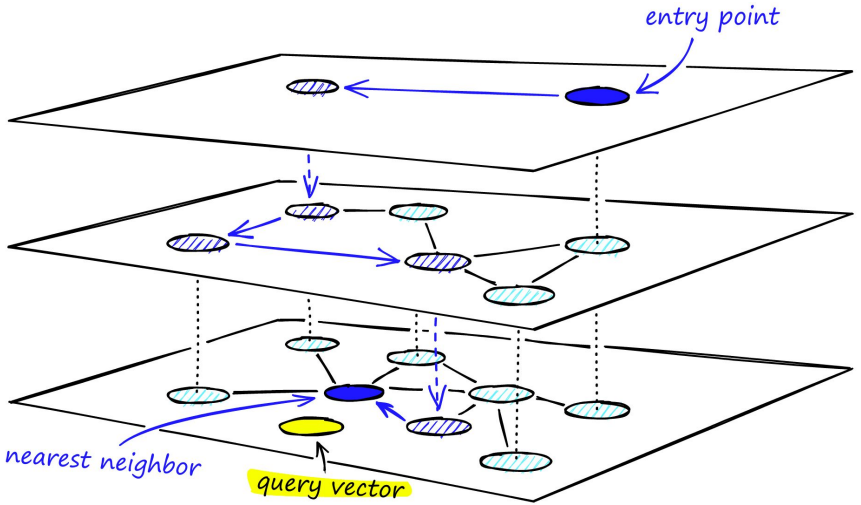




[2010] Jegou, Herve, Matthijs Douze, and Cordelia Schmid. "Product quantization for nearest neighbor search."

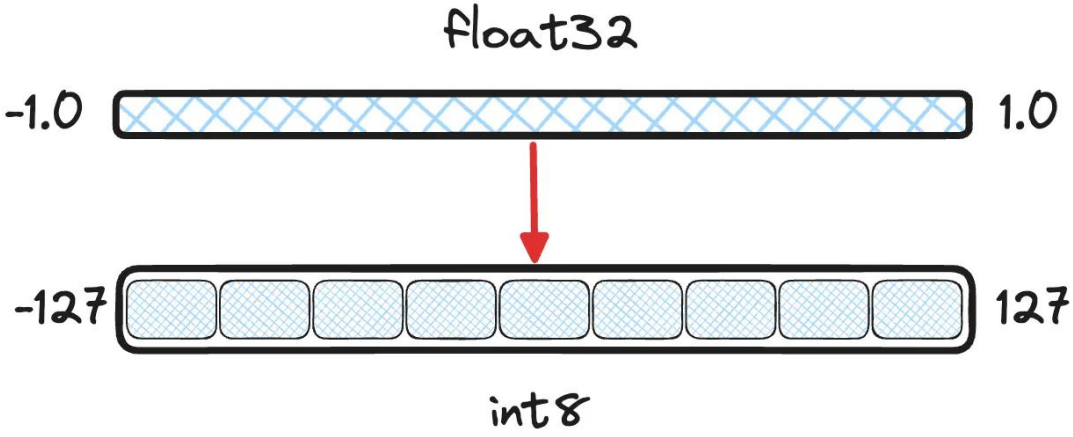
Vector indexes to speed up search





[2018] Malkov, Yu A., and Dmitry A. Yashunin. "Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs."

Quantization to speed up search



Challenges of Vector Similarity Search

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**
  - **Heavy-weight indexes** that are hard to maintain (HNSW)

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**
  - **Heavy-weight indexes** that are hard to maintain (HNSW)
  - How to efficiently **pushdown predicates** (hybrid search)

[2024] Patel, L., et al. ACORN: Performant and Predicate-Agnostic Search Over Vector Embeddings and Structured Data.

CWI

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**
  - **Heavy-weight indexes** that are hard to maintain (HNSW)
  - How to efficiently **pushdown predicates** (hybrid search)
  - Most quantization techniques lack **locality adaption**

[2023] Aguerrebere, C., et al. Similarity search in the blink of an eye with compressed indices.
[2024] Aguerrebere, C., et al. Locally-Adaptive Quantization for Streaming Vector Search. https://github.com/intel/ScalableVectorSearch

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**
  - **Heavy-weight indexes** that are hard to maintain (HNSW)
  - How to efficiently **pushdown predicates** (hybrid search)
  - Most quantization techniques lack **locality adaption**
- Search runtime dominated by **distance calculations**

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**
  - **Heavy-weight indexes** that are hard to maintain (HNSW)
  - How to efficiently **pushdown predicates** (hybrid search)
  - Most quantization techniques lack **locality adaption**
- Search runtime dominated by **distance calculations**

Recent research proposes **pruning** of dimensions on the distance calculation

However, they can still lose to full **SIMD** distance calculations

[2002] de Vries, Arjen P., et al. Efficient k-NN search on vertically decomposed data.
[2023] Gao, J., & Long, C. High-dimensional ANNS: with reliable and efficient distance comparison operations.
[2024] Yang, M., et al. Bridging Speed and Accuracy to Approximate k-Nearest Neighbor Search.

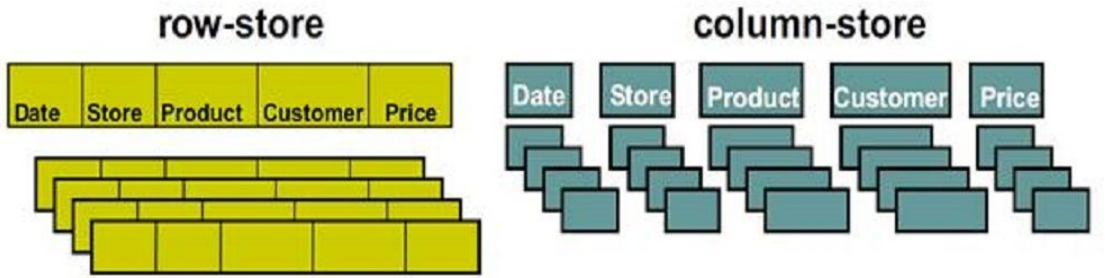
CWI

Challenges of Vector Similarity Search

- Facing similar challenges of **Analytical Databases**
  - **Heavy-weight indexes** that are hard to maintain (HNSW)
  - How to efficiently **pushdown predicates** (hybrid search)
  - Most quantization techniques lack **locality adaption**
- Search runtime dominated by **distance calculations**

Can we borrow ideas
from Analytical Databases?

Storage



- Better opportunities for **compression**
- **Efficient scans** of attributes
- **Vectorized** query execution

What if we store vectors in a Vertical Layout?

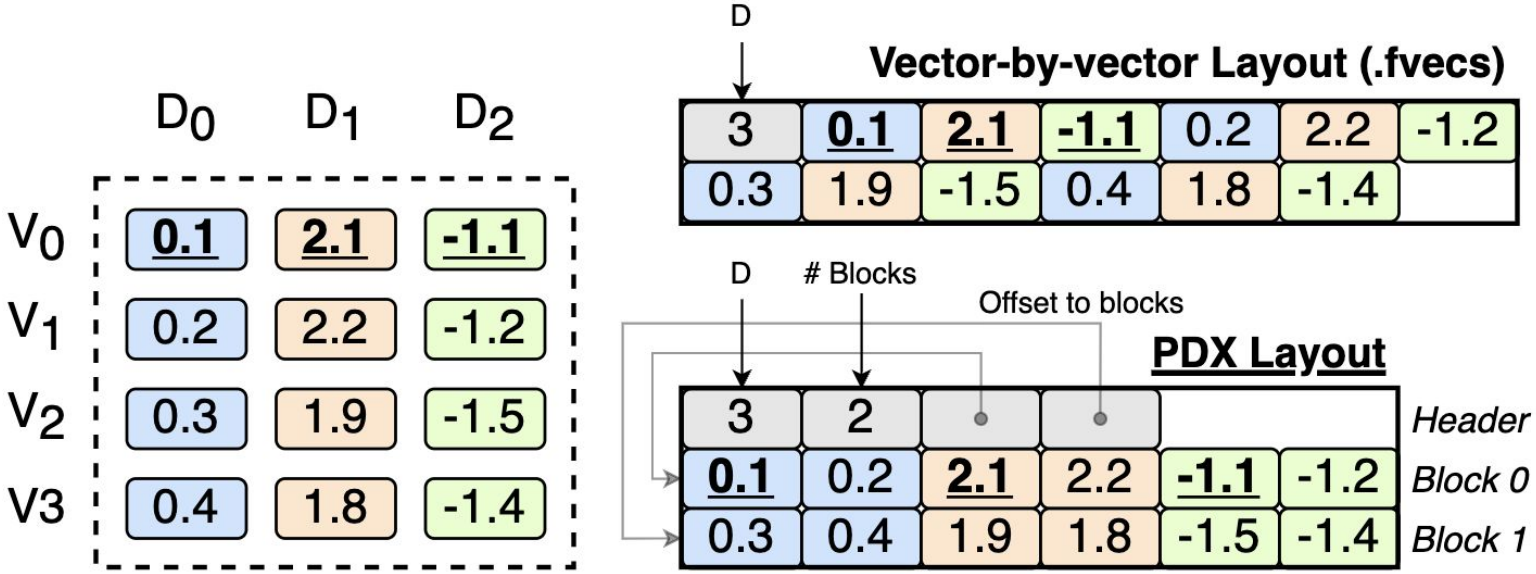
CWI

Storage

- **PDX: Vertical Layout** for Vectors

Storage

- **PDX: Vertical Layout** for Vectors



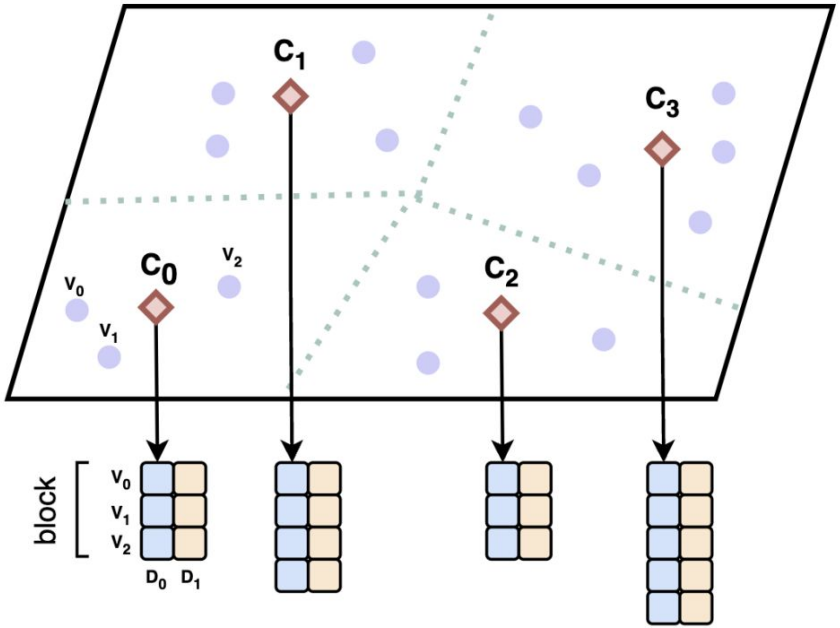
[2002] de Vries, Arjen P., et al. Efficient k-NN search on vertically decomposed data.

[2025] Kuffo, L., Krippner E., & Boncz, P. PDX: A Data Layout for Vector Similarity Search *(under-review)*

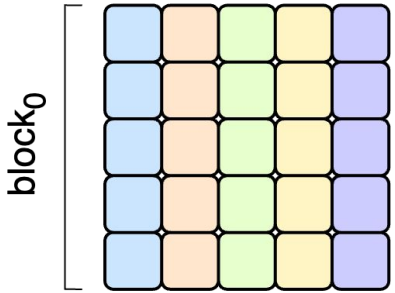
Storage: A foundational change

- **PDX: Vertical Layout** for Vectors
- Allows for efficient **dimensions-pruning** during search

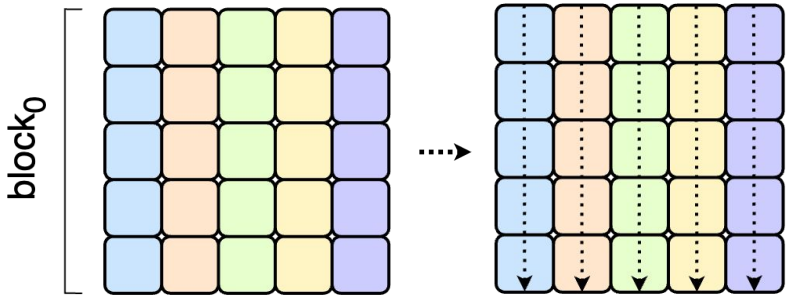
A search that (reliably) prunes dimensions



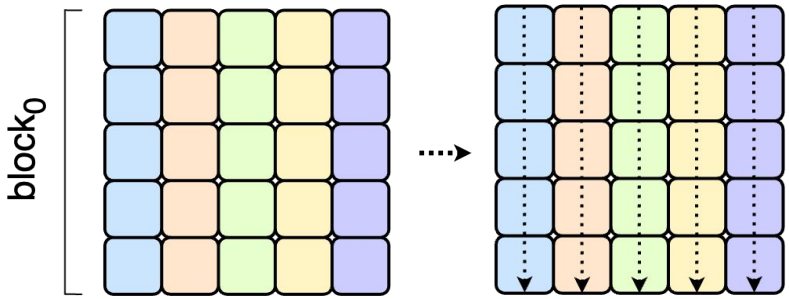
A search that (reliably) prunes dimensions

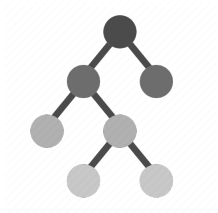


A search that (reliably) prunes dimensions

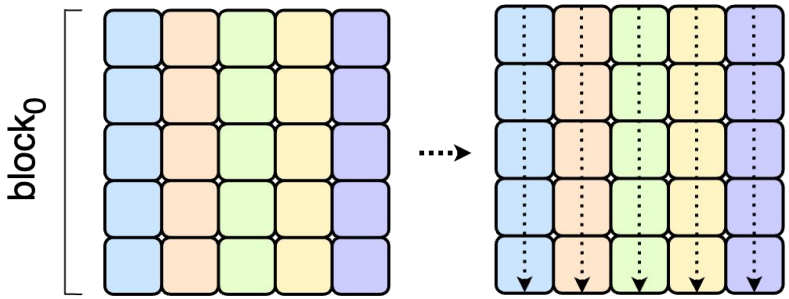


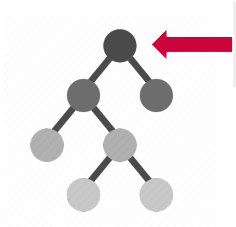
A search that (reliably) prunes dimensions





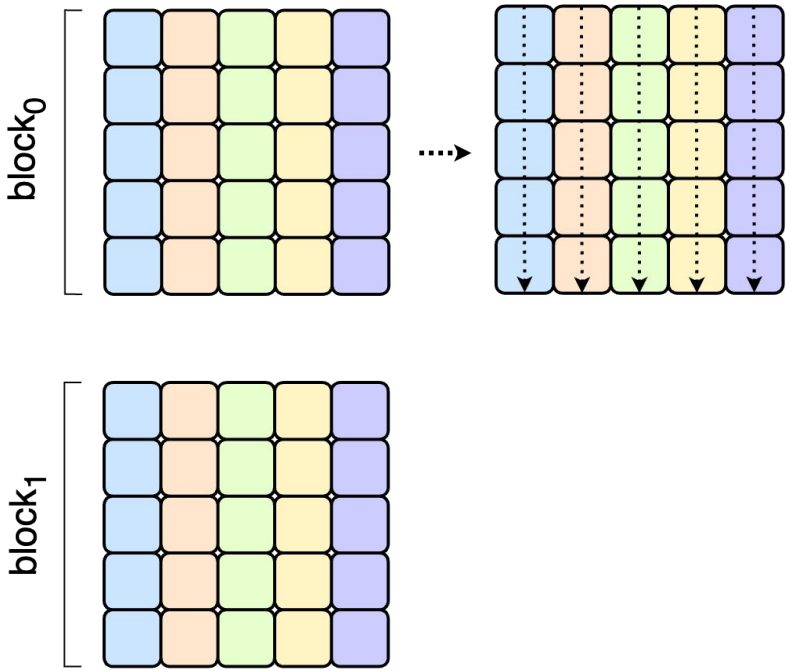
A search that (reliably) prunes dimensions

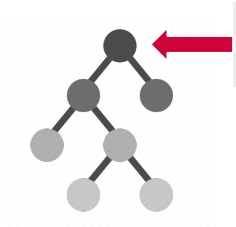




Pruning threshold

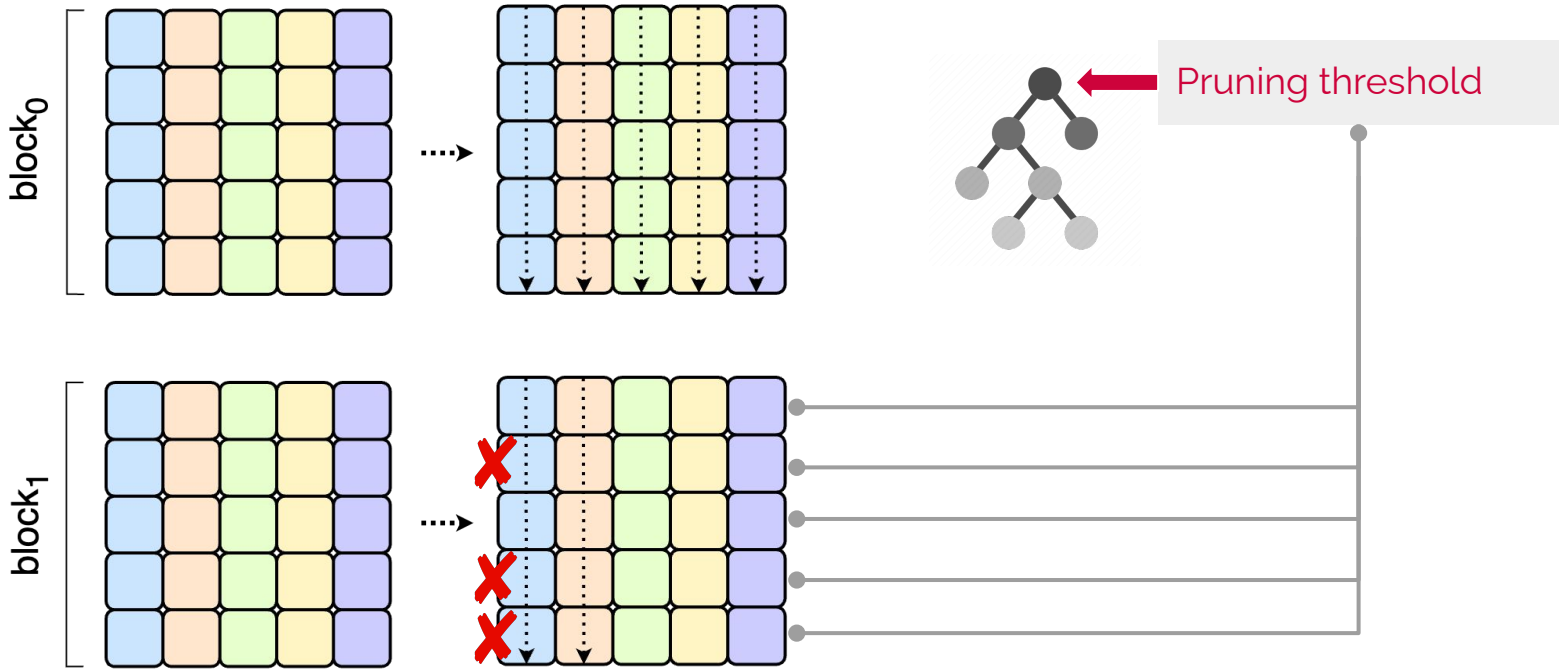
A search that (reliably) prunes dimensions



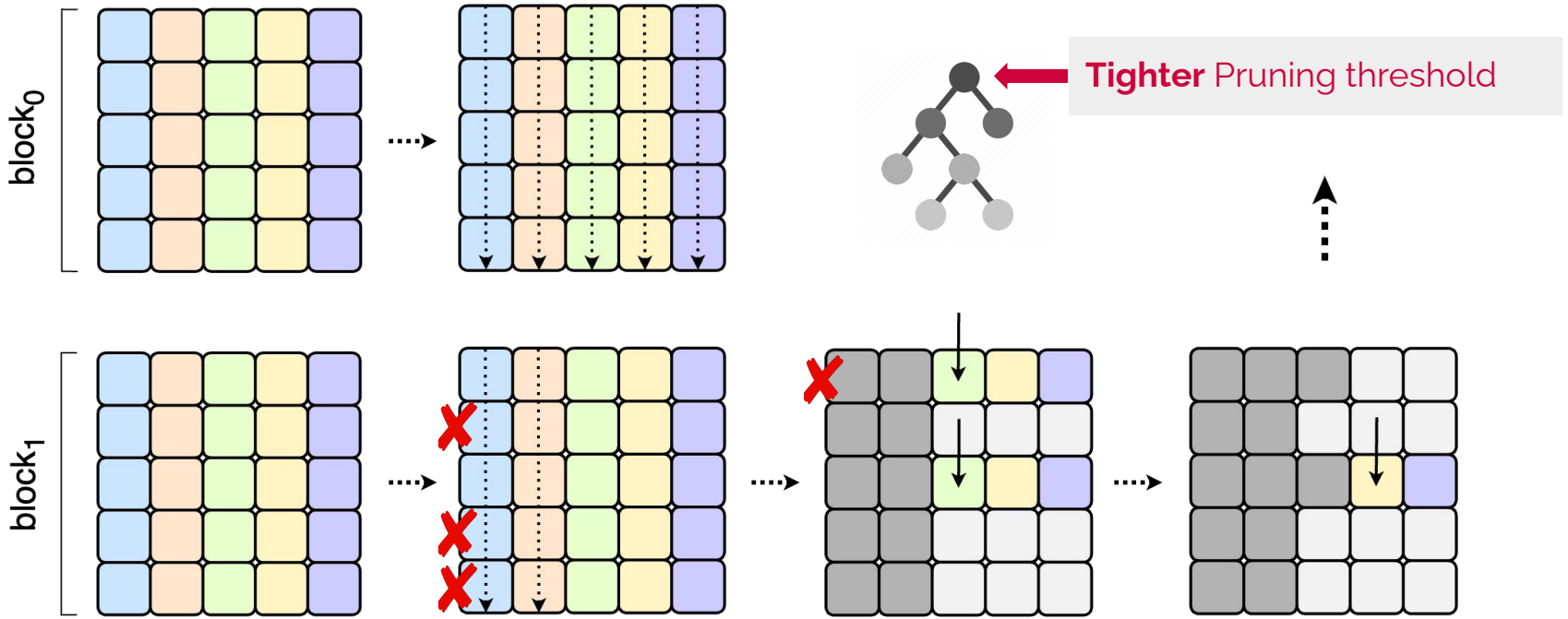


Pruning threshold

A search that (reliably) prunes dimensions



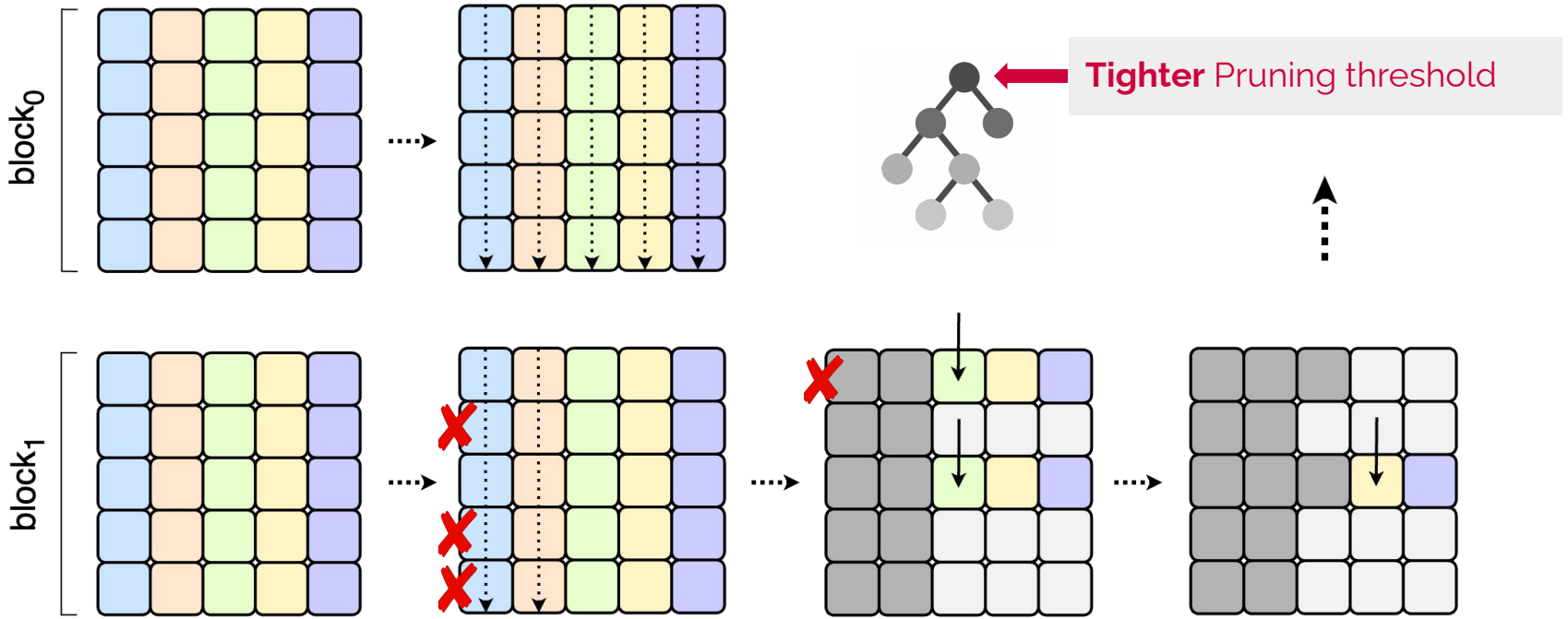
A search that (reliably) prunes vectors



[2025] Kuffo, L., Krippner E., & Boncz, P. PDX: A Data Layout for Vector Similarity Search *(under-review)*

A search that (reliably) prunes vectors

PDXearch



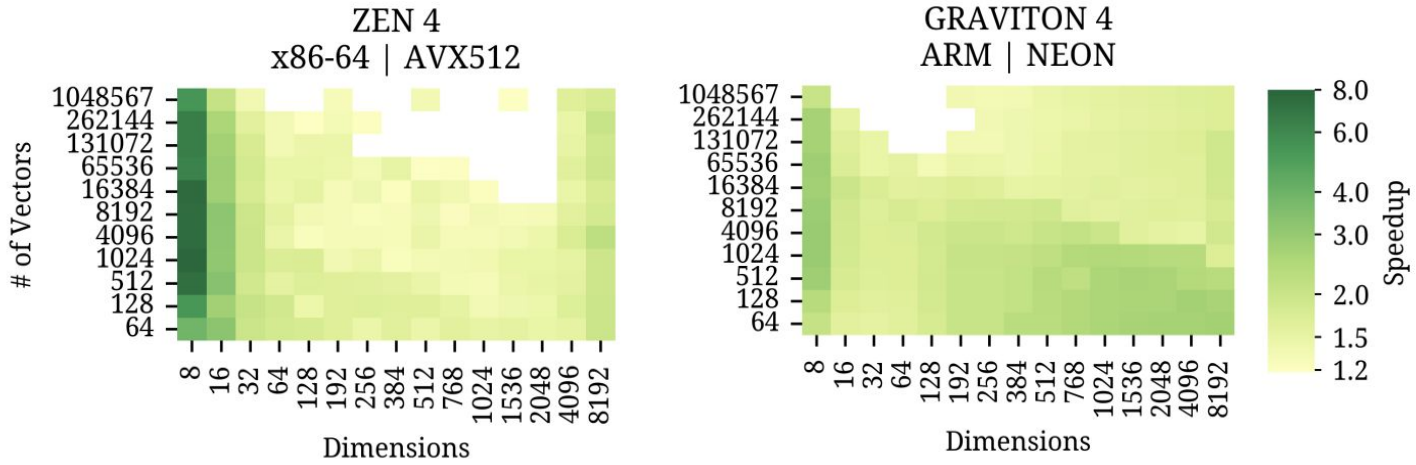
[2025] Kuffo, L., Krippner E., & Boncz, P. PDX: A Data Layout for Vector Similarity Search *(under-review)*

Storage: A foundational change

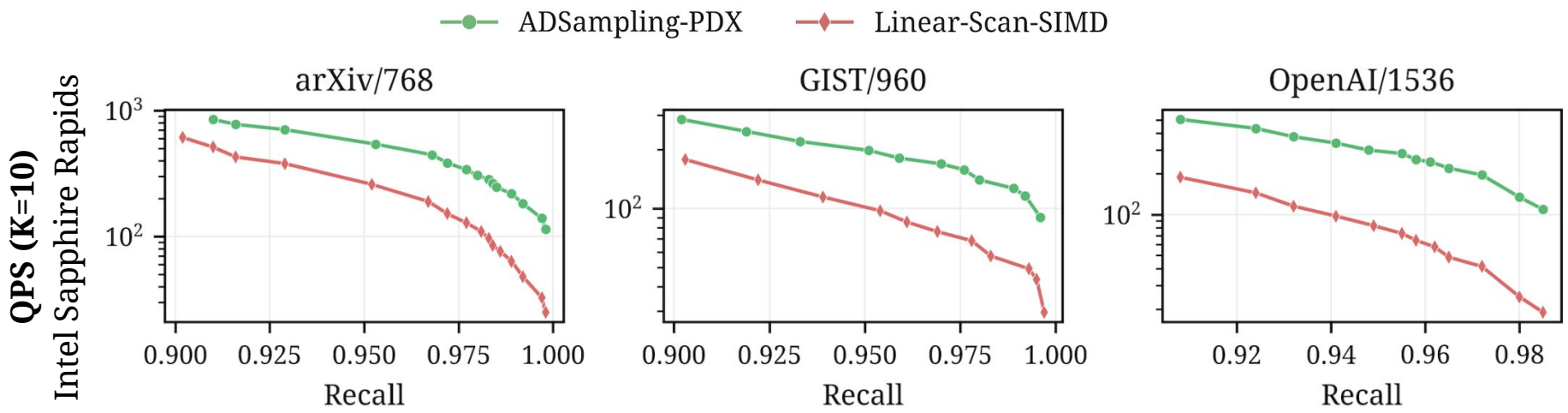
- **PDX: Vertical Layout** for Vectors
- Allows for efficient **dimensions-pruning** during search ✔

Storage: A foundational change

- **PDX: Vertical Layout** for Vectors
- Allows for efficient **dimensions-pruning** during search ✔
- Distance computation (L2) **faster** than SIMD kernels
  - Only using **Scalar code** that is auto-vectorized (in NEON, AVX2 and AVX512)

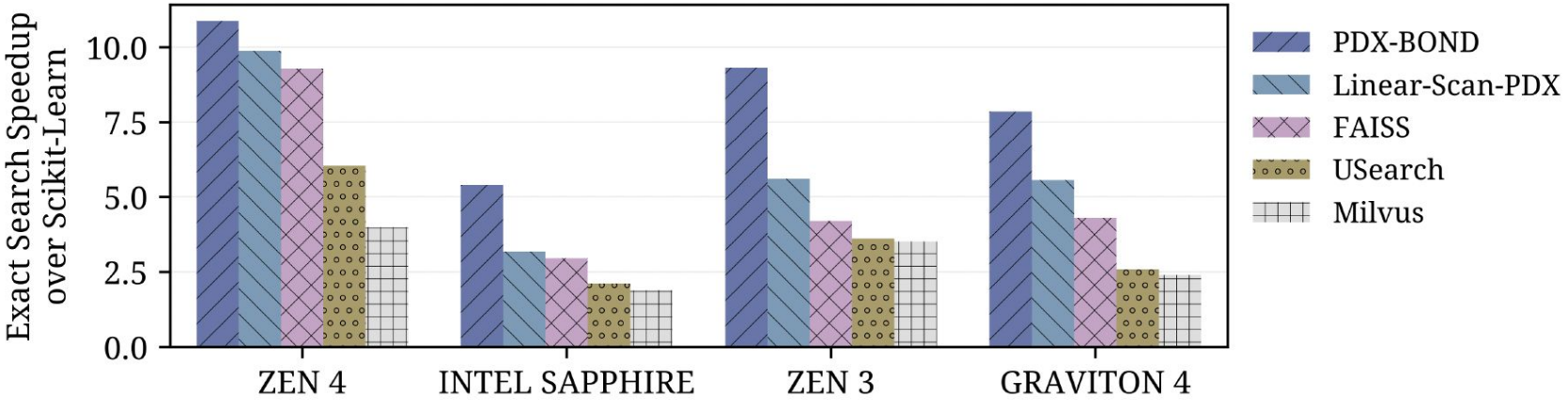


A search that (reliably) prunes vectors → on IVF



arXiv	D=768	N=2.25M
GIST	D=960	N=1M
OpenAI (dbpedia)	D=1536	N=1M

A search that (reliably) prunes vectors



Storage: A foundational change

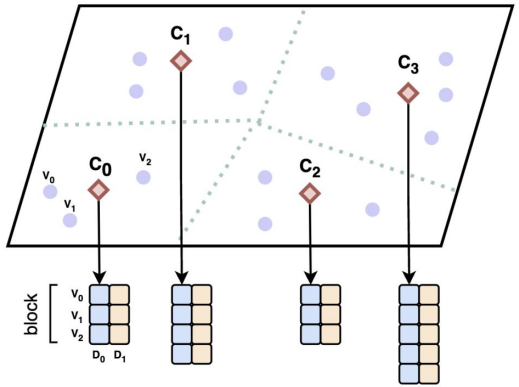
- **PDX: Vertical Layout** for Vectors
- Allows for efficient **dimensions-pruning** during search ✔
- Distance computation (L2) **faster** than SIMD kernels ✔

Storage: A foundational change

- **PDX: Vertical Layout** for Vectors
- Allows for efficient **dimensions-pruning** during search ✔
- Distance computation (L2) **faster** than SIMD kernels ✔
- Lightweight **Compression**

LEP: Lossily Encoded floating-Points (WIP)

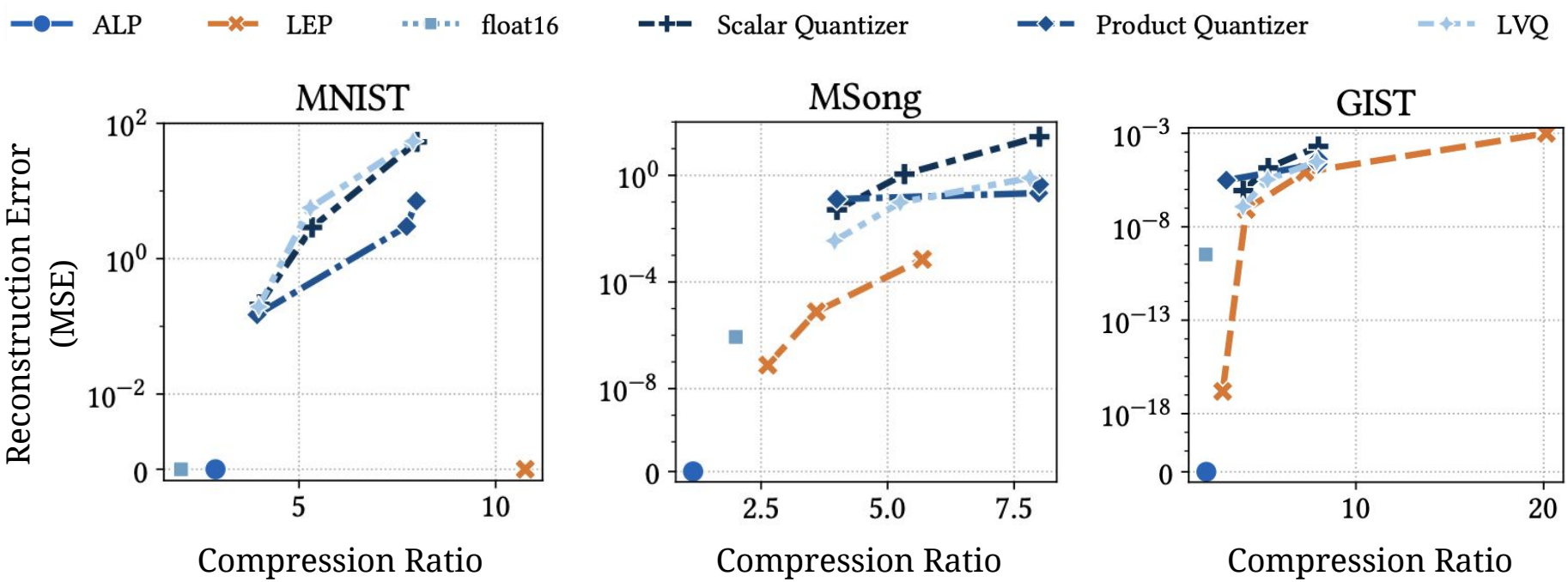
- Local adaptivity per dimension & per block
  - Specially effective on space-partitioning indexes (IVF)
  - Thanks to the **PDX** layout
- ALP without lossless verification (in a nutshell)
- Encoding of out-of-distribution values
- Encoding of repetition



[2023] Afroozeh, Azim, Leonardo X. Kuffo, and Peter Boncz. "ALP: Adaptive Lossless floating-Point Compression."

[2025] Krippner, E. Rethinking Vector Embeddings Search for Analytical Database Systems. (MSc thesis)

LEP: Lossily Encoded floating-Points (WIP)



[2025] Krippner, E. Rethinking Vector Embeddings Search for Analytical Database Systems. (MSc thesis)

Summary

PDX | **A Vertical Layout** for Vectors

- Distance calculations **faster** than SIMD kernels ✔
- Improve distance evaluation latency by **pruning dimensions** ✔
- Adaptive lightweight **compression per dimension (wip)** ✔

Future Work

- Better **pruning strategies**
- Implement PDXearch on **GPUs**
- Do IVF indexes *the right way*
    - *Lightweight* | *Updatable* | *Compressed* | *Pruning*

A Vertical Layout for Vector Similarity Search

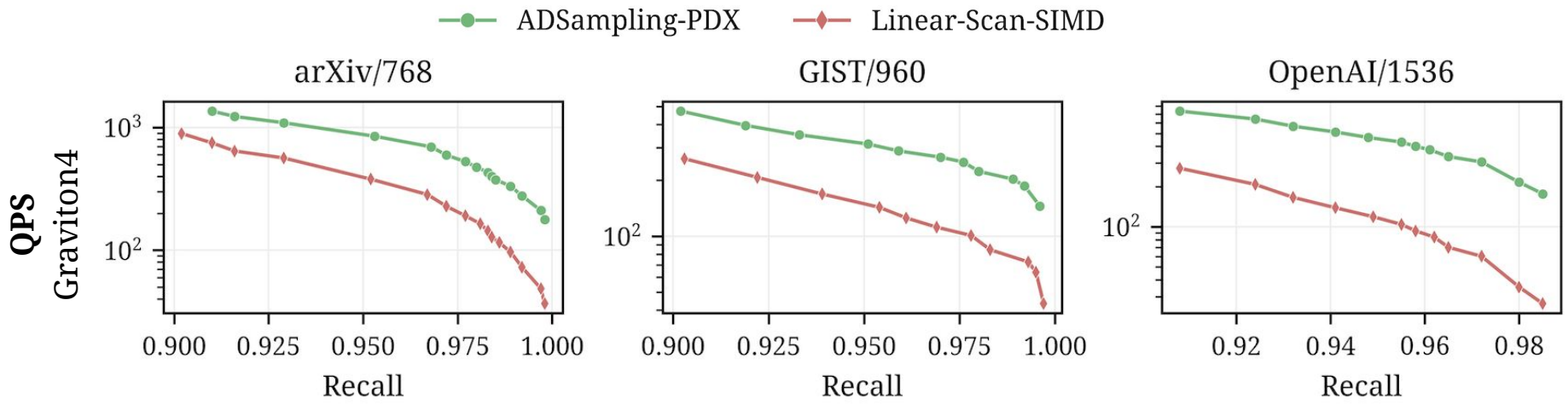
Leonardo Kuffo*, **Peter Boncz**
CWI Database Architectures group

22/11/24

A Vertical Layout for Vector Similarity Search

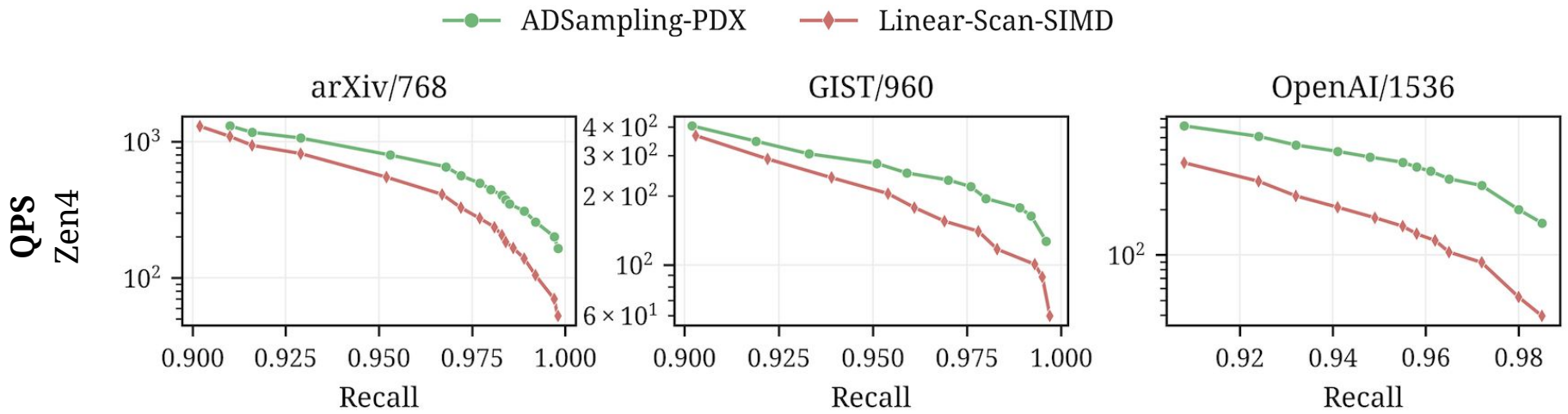
SUPPLEMENTARY SLIDES

A search that (reliably) prunes vectors → on IVF



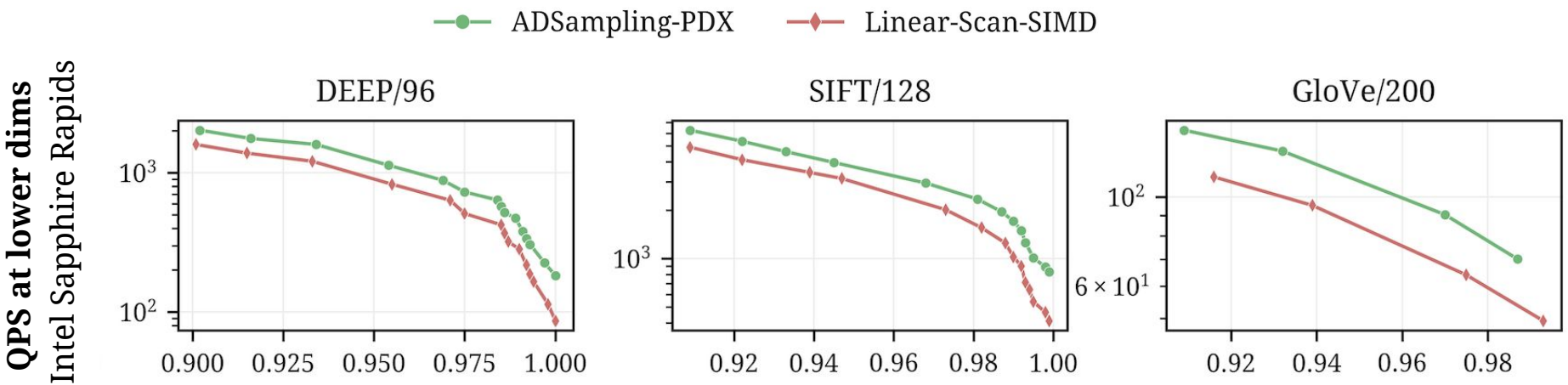
arXiv	D=768	N=2.25M
GIST	D=960	N=1M
OpenAI (dbpedia)	D=1536	N=1M

A search that (reliably) prunes vectors → on IVF

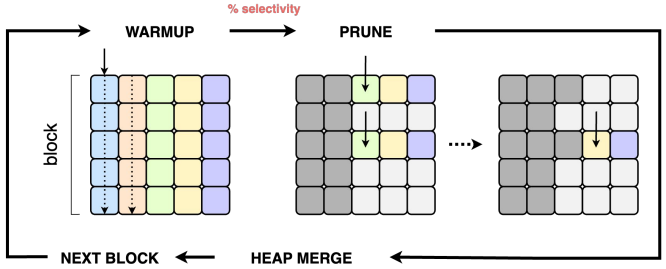


arXiv	D=768	N=2.25M
GIST	D=960	N=1M
OpenAI (dbpedia)	D=1536	N=1M

A search that (reliably) prunes vectors → on IVF



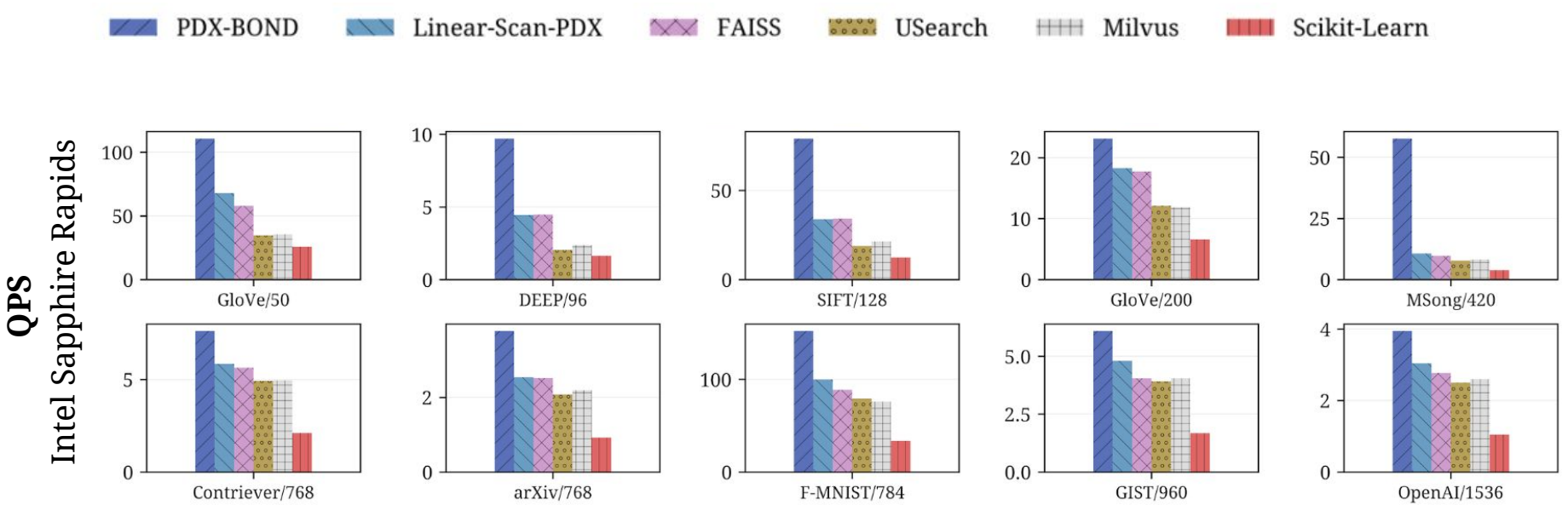
A search that (reliably) prunes vectors



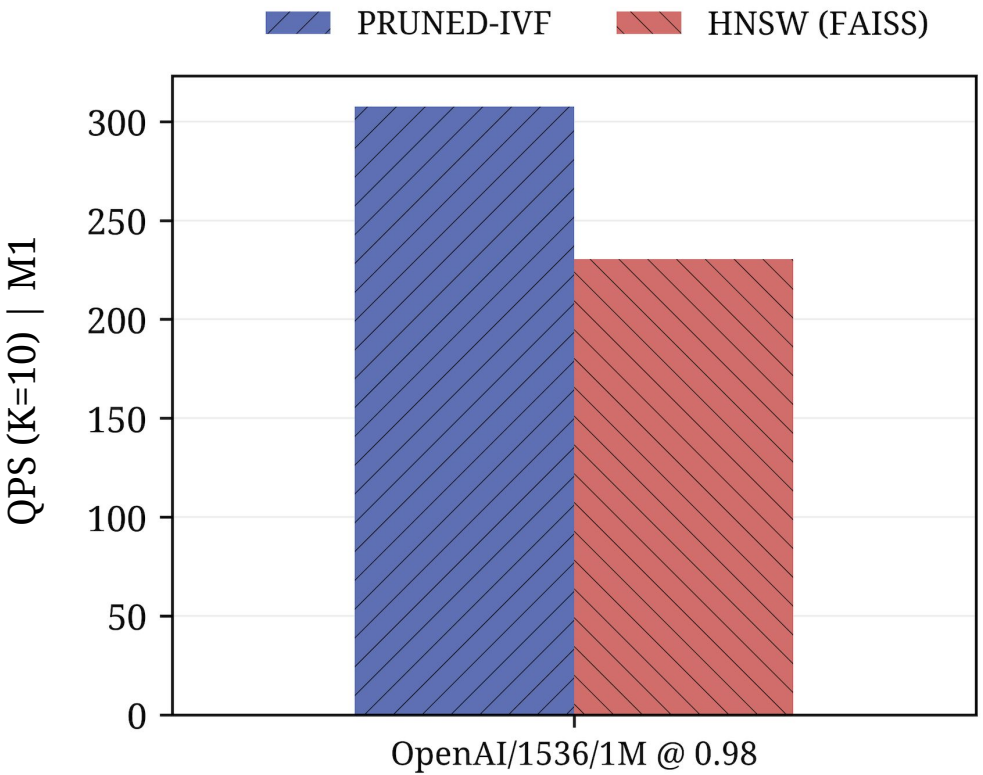
- Which **distance metrics**?
  - Monotonic ones (L2, Dot in normalized vectors [0, 1])
- How ***early*** does pruning happen?
  - As early as 2% of dimensions in some datasets
  - Depends on the query, dataset and pruning algorithm
- Find benefits only with the PDX layout
- Pruning Algorithms:
  - **ADSampling** → Approximate (<0.001 loss of recall)
  - **BSA** → Approximate
  - **PDX-BOND** → Exact | Prioritize dimensions at query-time
- ↑ gains at ↑ dimensionalities

[2023] Gao, J., & Long, C. High-dimensional ANNS: with reliable and efficient distance comparison operations.
[2024] Yang, M., et al. Bridging Speed and Accuracy to Approximate k-Nearest Neighbor Search.
[2025] Kuffo, L., Krippner E., & Boncz, P. PDX: A Data Layout for Vector Similarity Search *(under-review)*

A search that (reliably) prunes vectors → on Exact

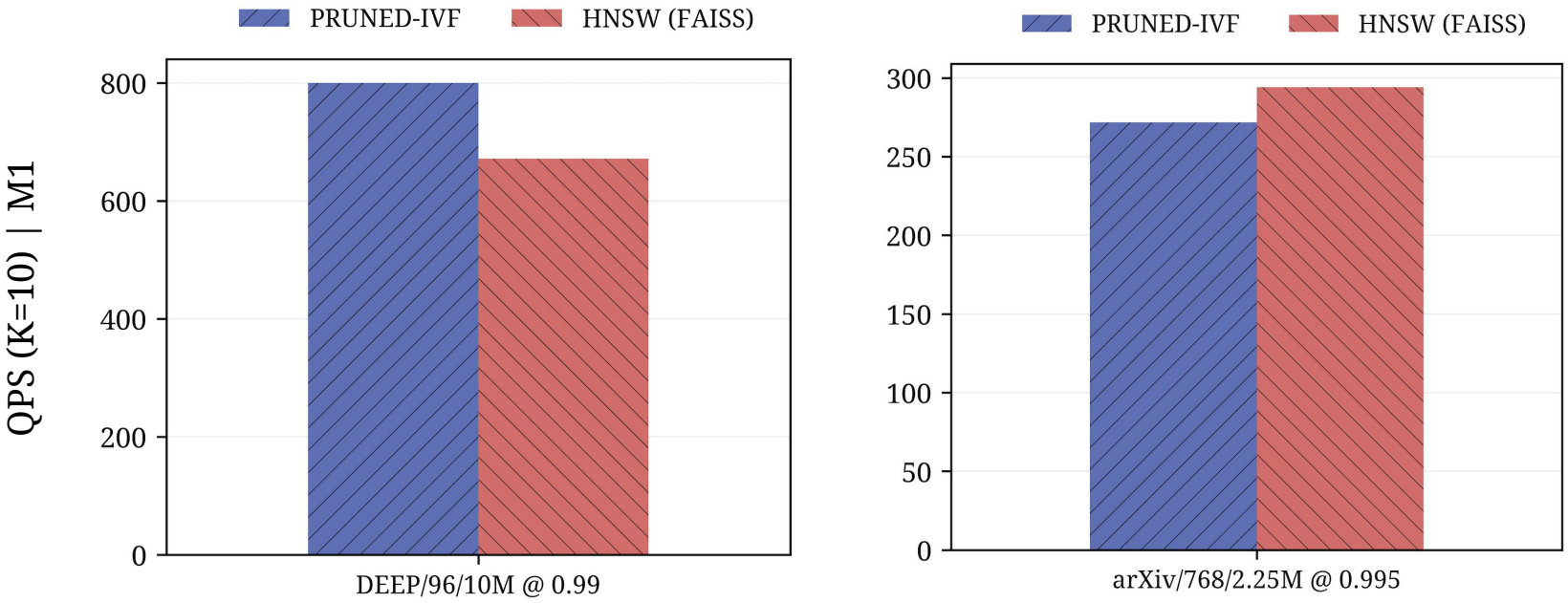


A search that (reliably) prunes vectors → vs HNSW (wip)

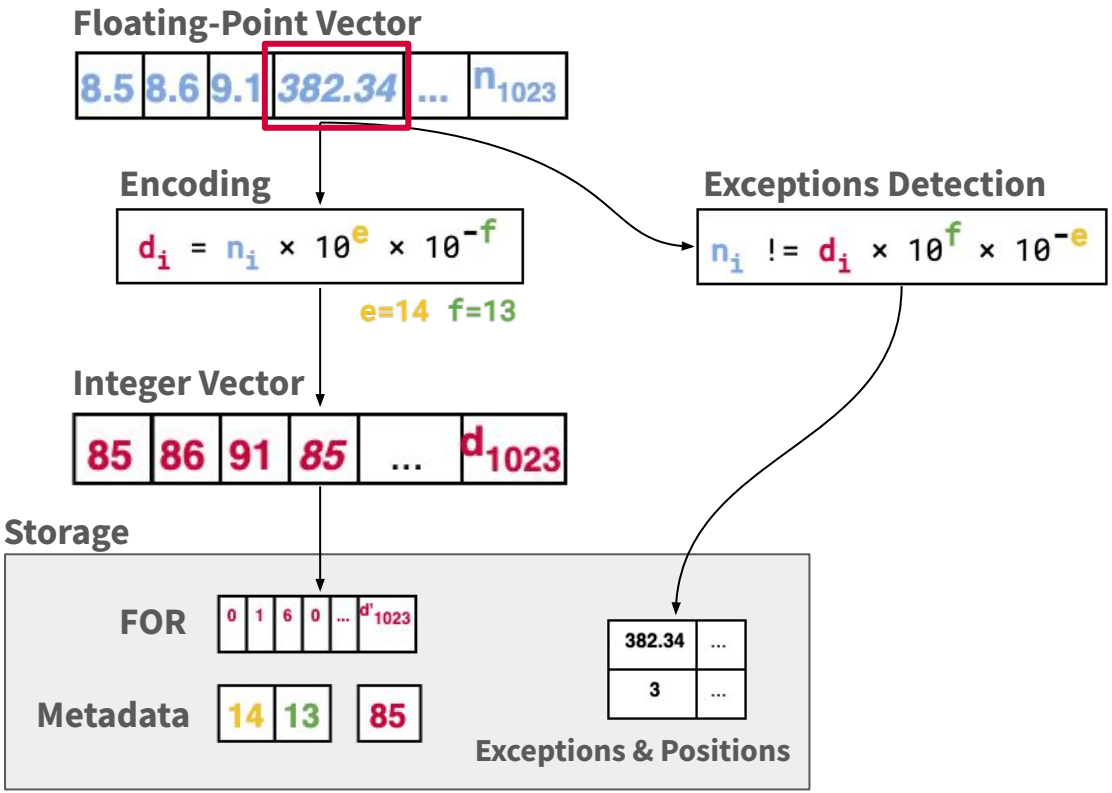


- **K = 10**
- HNSW **efConstruction = 64**
- HNSW **M = 24**
- IVF **N° Probed = 256**

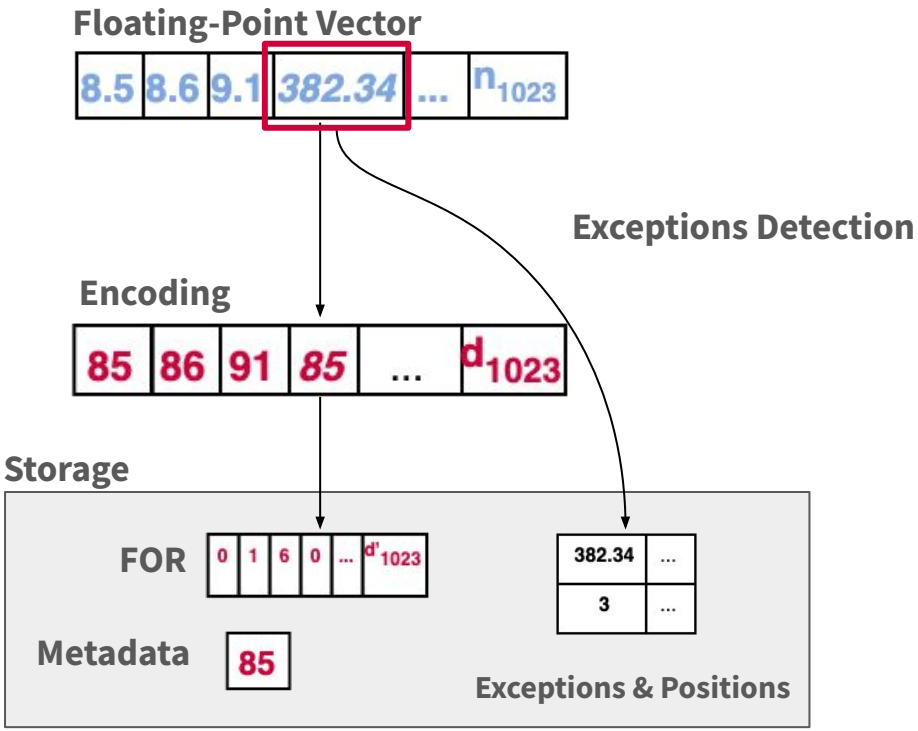
A search that (reliably) prunes vectors → vs HNSW (wip)



Lightweight Compression: ALP (lossless)



Lightweight Compression: ALP (lossless)



Across Architectures