ParaGraph: Accelerating Graph Indexing through GPU-CPU Parallel Processing for Efficient Cross-modal ANNS

Yuxiang Yang, Shiwen Chen, **Yangshen Deng**, Bo Tang

Southern University of Science and Technology, AlayaDB.AI

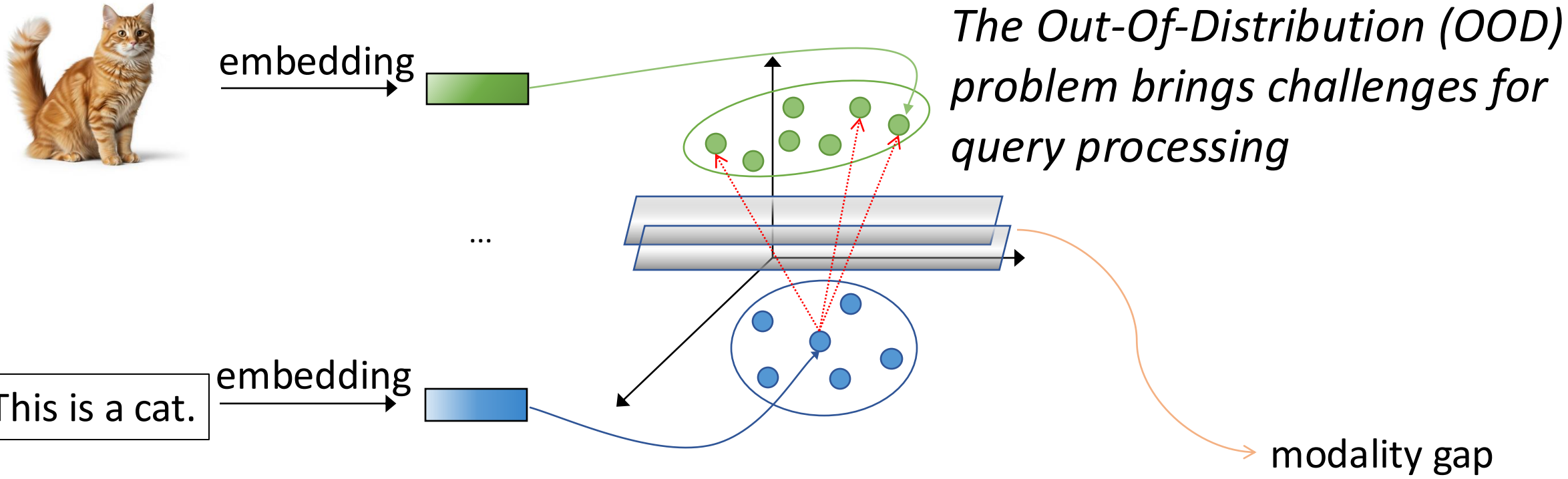
research@alayadb.ai

Approximate Nearest Neighbor Search (ANNS)

- Given a query vector, search for the closest vectors.
- Used in RAG, recommendation, and LLM inference.
- An index is built on the base data for efficient search.

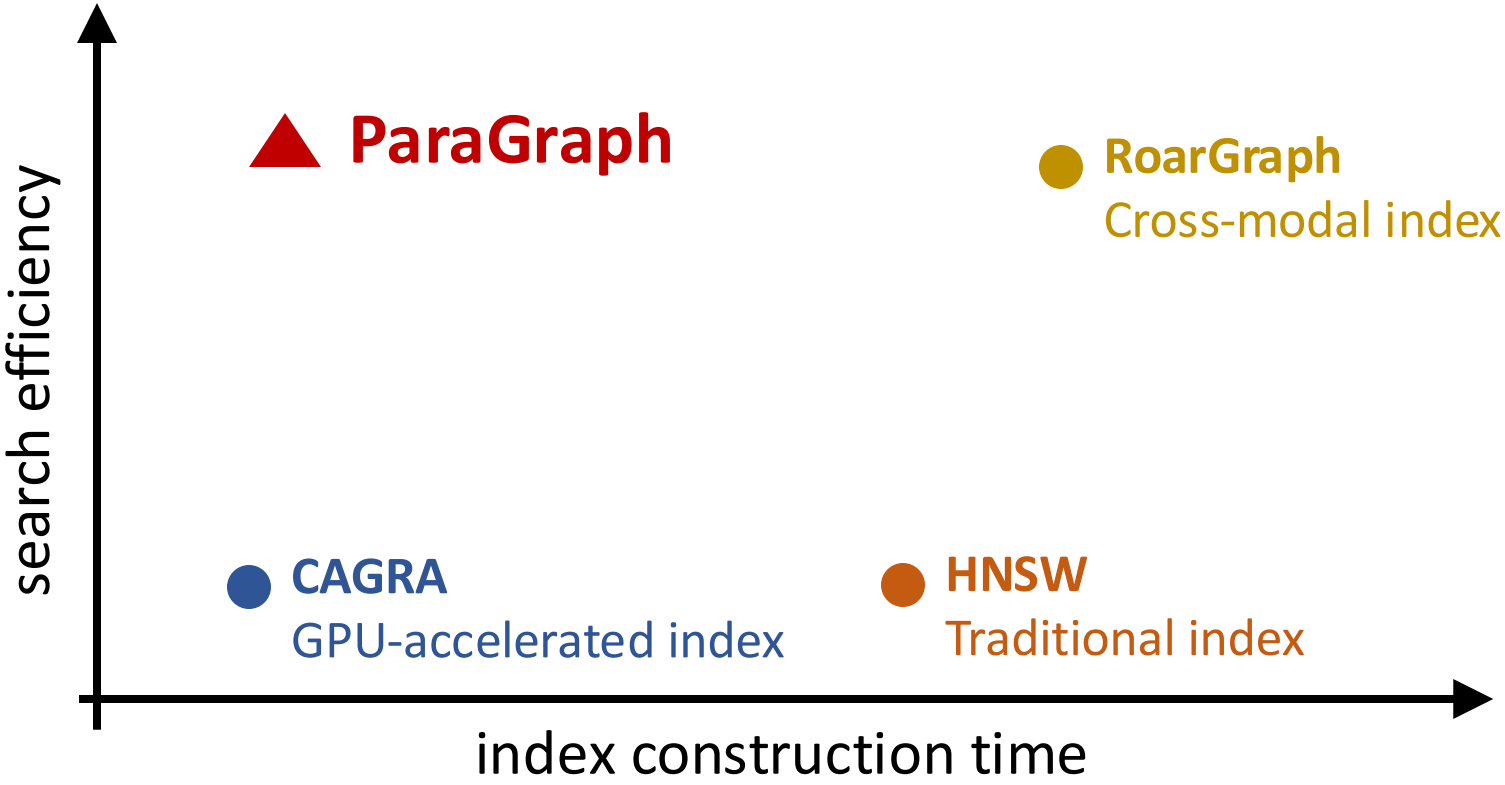
Cross-modal ANNS

An example for text-to-image ANNS



The limitation of existing methods

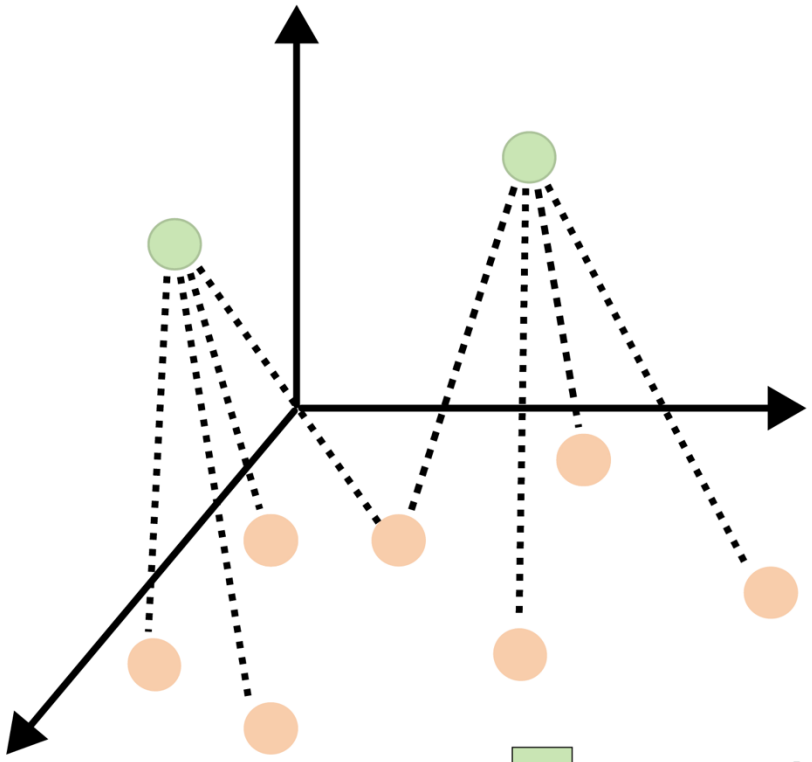
The gap between ***search efficiency*** and ***construction time***.



RoarGraph construction

1. kNN graph construction (can be offline)
2. Top-1 projection
3. Iterative search-and-refine

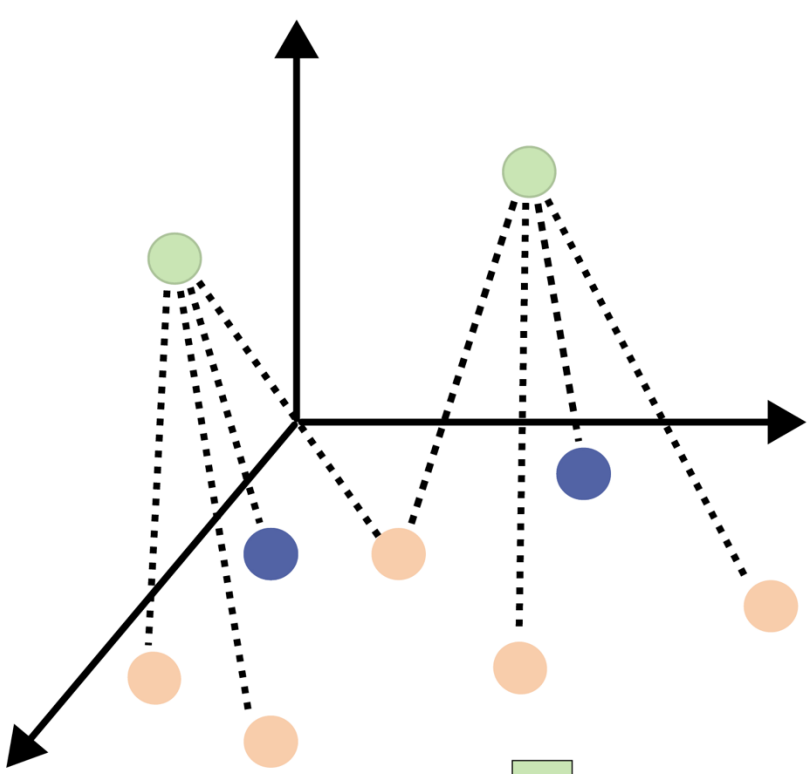
1. kNN graph construction



query id	kNN			
0	*8*	*16*	*...*	*1024*
...	...	...	...	...
i	*n-1*	...	...	...
...	...	...	...	...
q-1	*100*	*799*	*...*	*465*

query node projection node base node

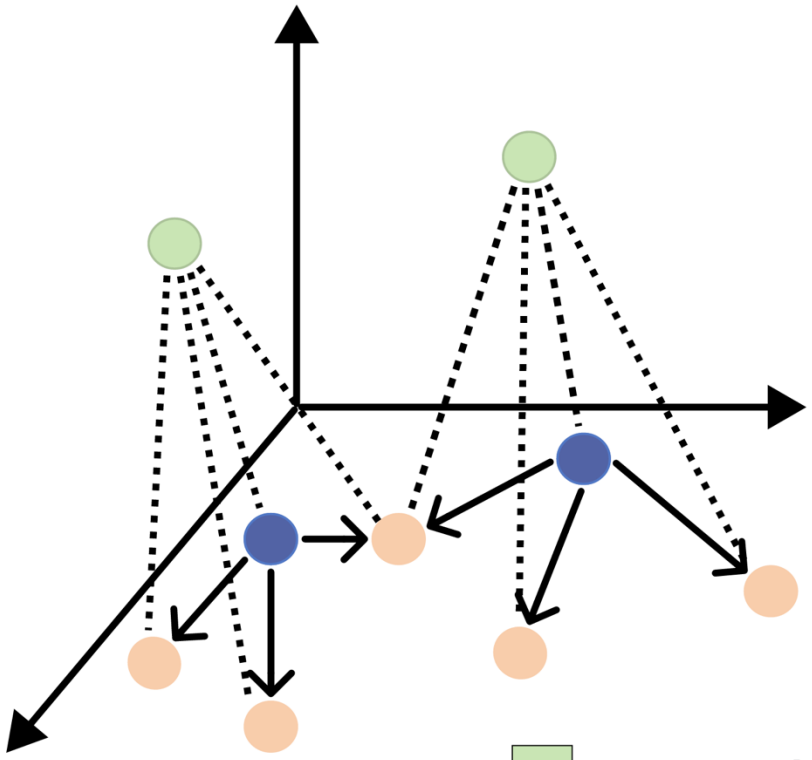
2. Top-1 projection



query id	kNN			
0	*8*	*16*	...	*1024*
...	...	...	...	...
i	*n-1*	...	...	...
...	...	...	...	...
q-1	*100*	*799*	...	*465*

query node projection node base node

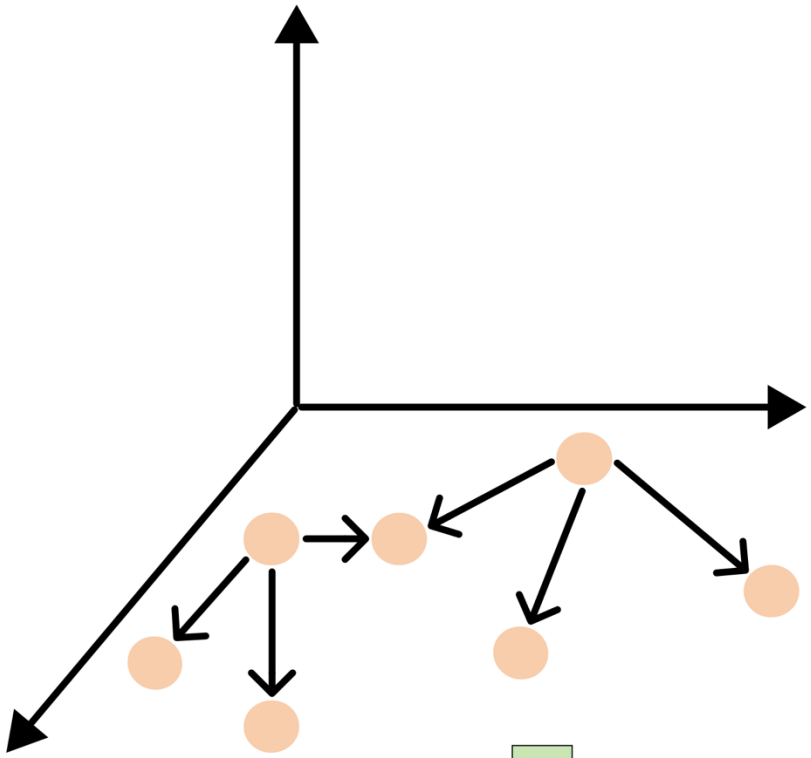
2. Top-1 projection



query id	kNN				base id	bipartite graph		
0	*8*	*16*	...	*1024*	...	...	...	...
...	...	...	...	...	8	*16*	...	*1024*
i	*n-1*	...	...	...	...	...	...	...
...	...	...	...	...	100	*799*	...	*465*
q-1	*100*	*799*	...	*465*	...	...	...	...
					n-1	...	...	...

query node projection node base node

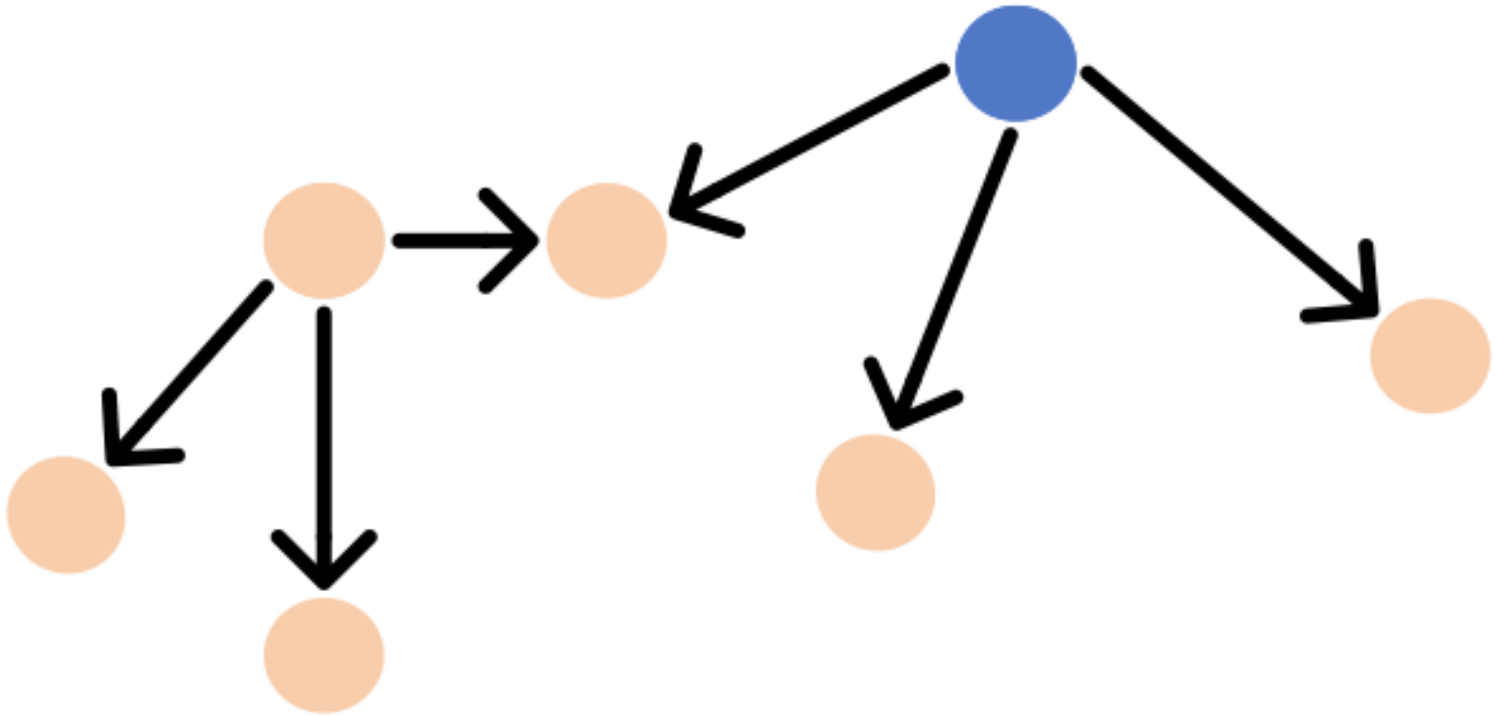
2. Top-1 projection



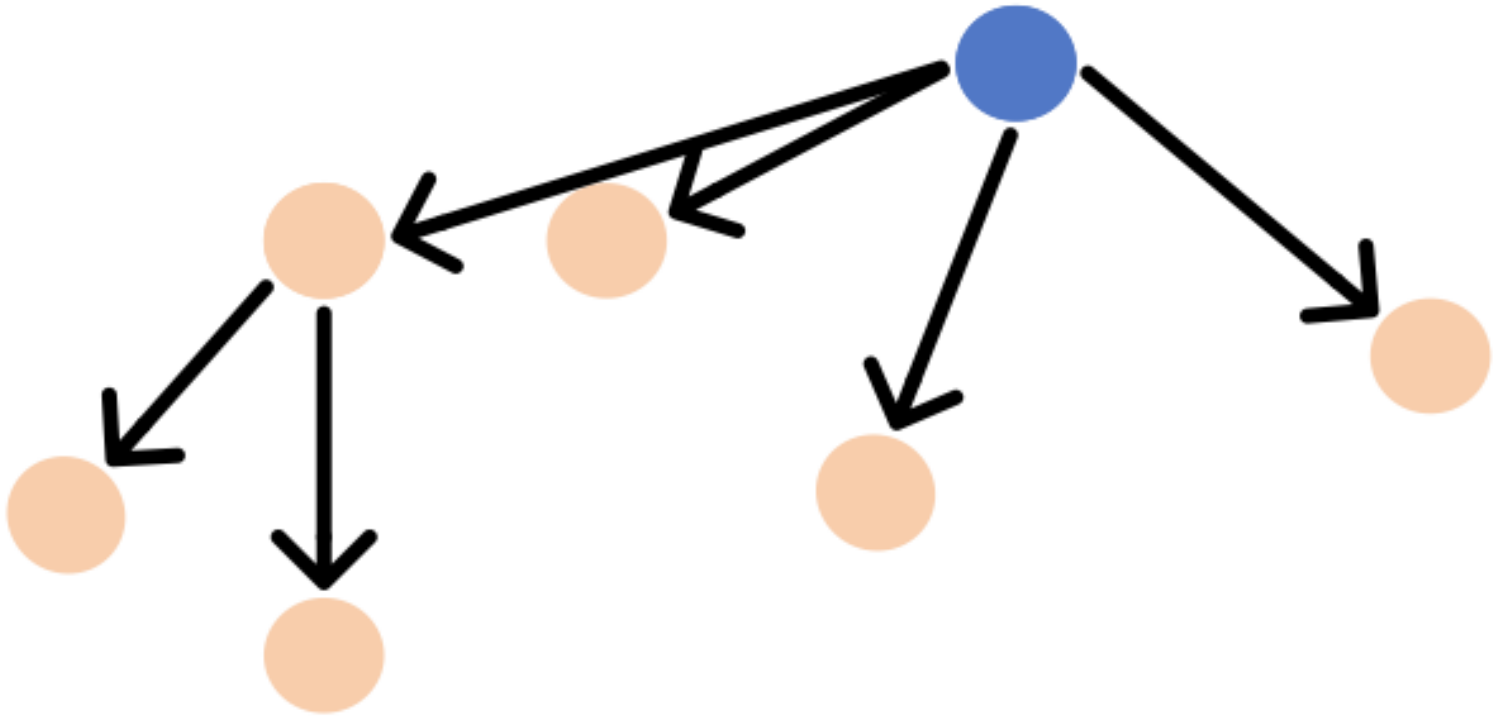
base id	neighbors		
8	16	...	1024
...	...	...	...
n-1	...	...	...
...	...	...	...
100	799	...	465

query node projection node base node

3. Iterative search-and-refine

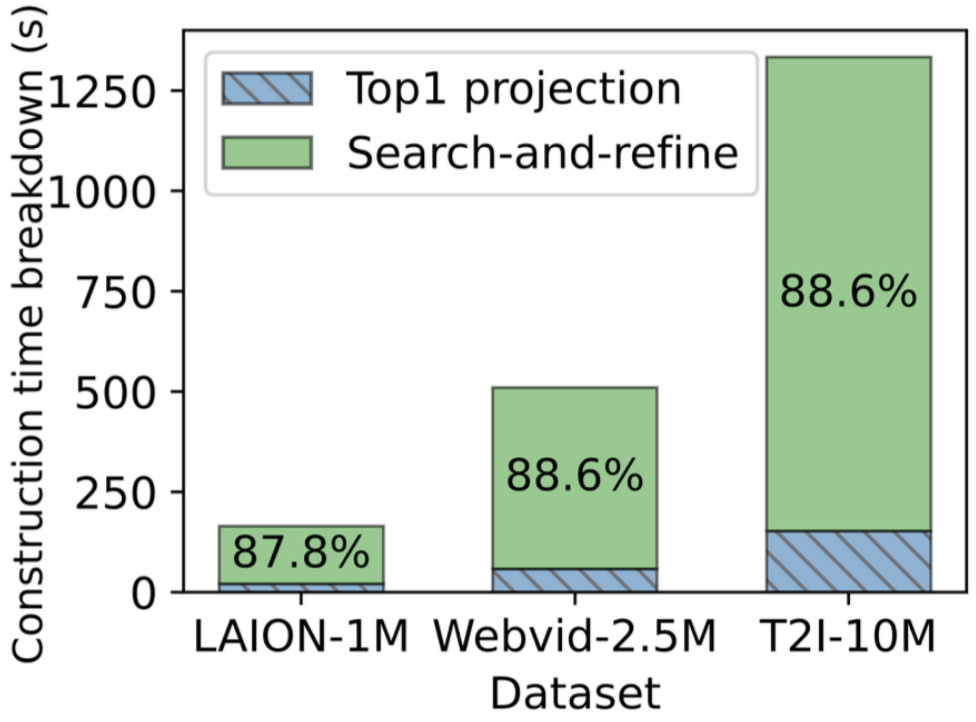


3. Iterative search-and-refine

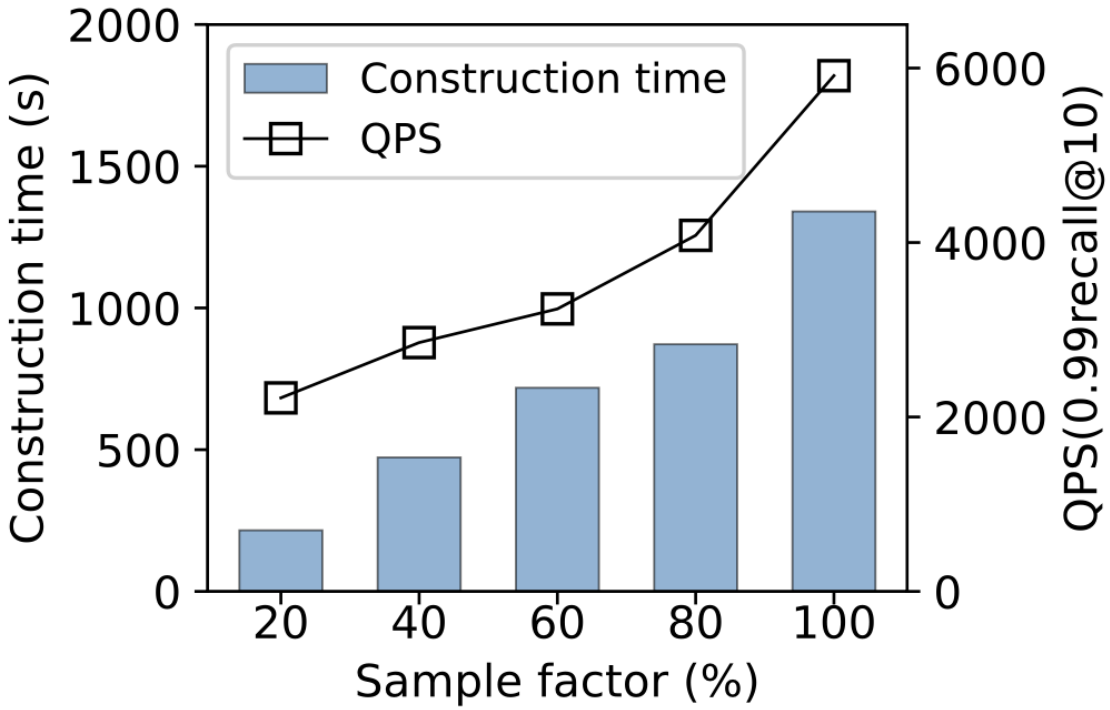


The bottleneck: search-and-refine

Simply reducing #iteration will harm the search efficiency.



Construction time breakdown



Performance of different #iteration

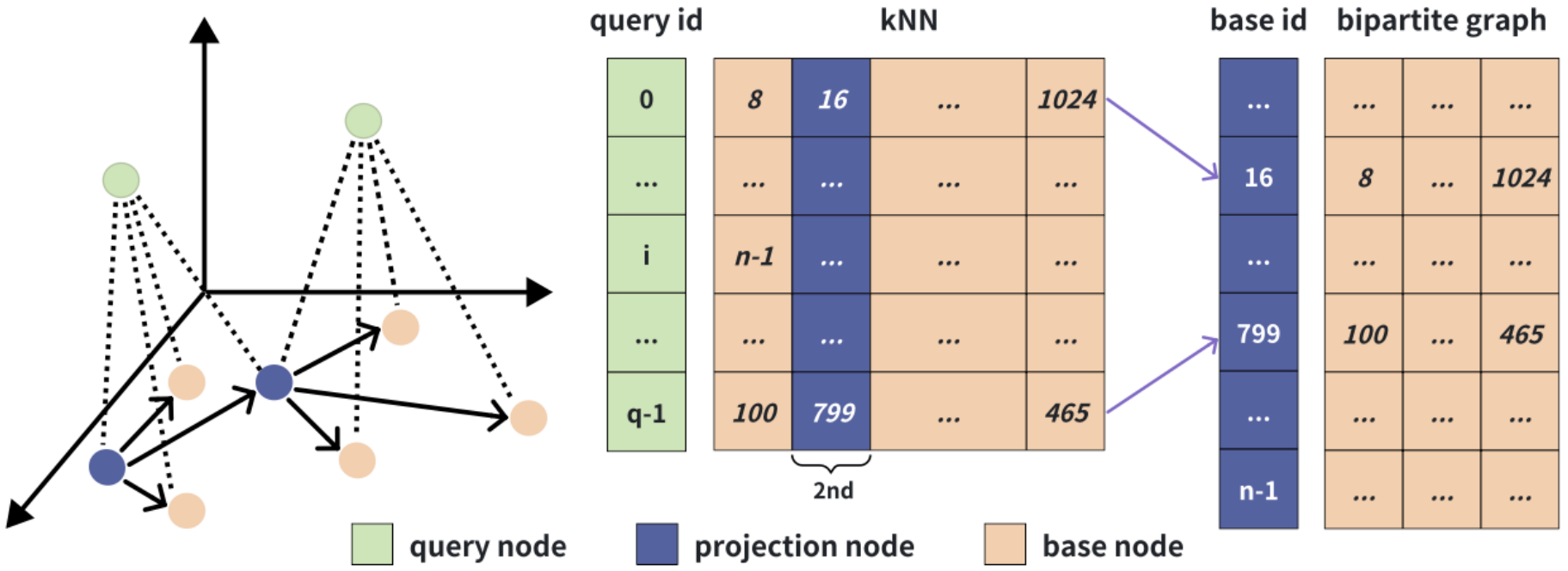
The key: projection

- Projection is very fast.
- Projection can capture the cross-modal relation.

*Can we better utilize **the power of projection** to accelerate cross-modal index construction?*

Our core idea: top-m projection

Instead of only projecting with top-1 node,
we do multiple rounds of projections using top-m nodes.

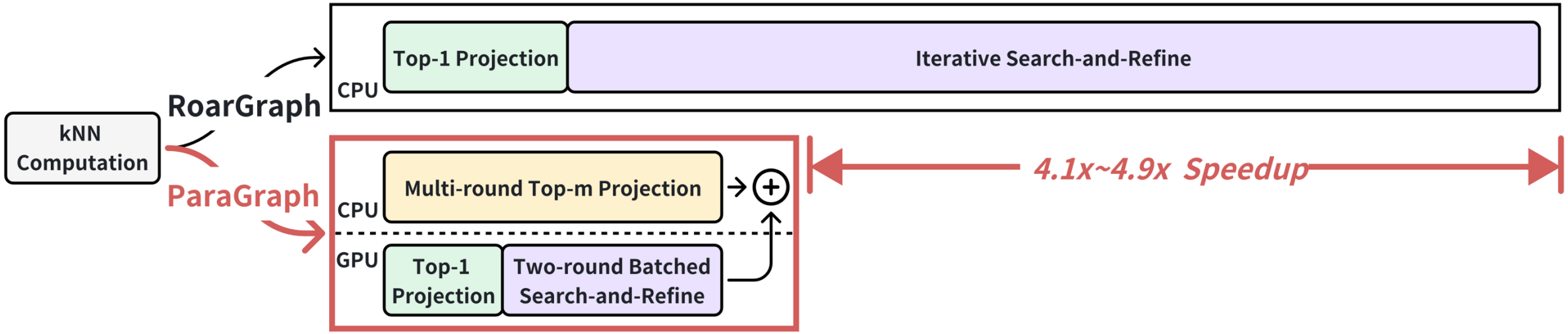


An example of top-m projection ($m = 2$)

ParaGraph overview

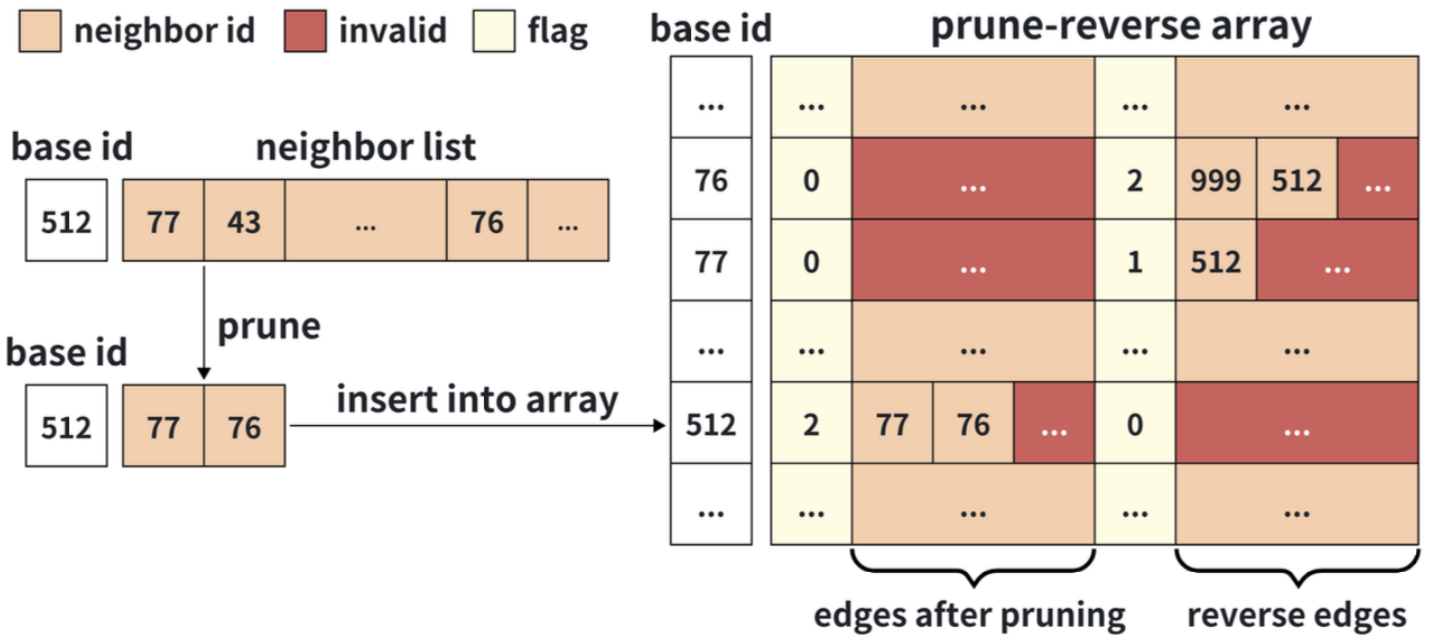
Algorithm: top-m projection + batched search-and-refine

System: CPU-GPU co-processing



New graph format for index construction

- A format that support concurrent **pruning and reversing operations** in a lock-free way.
- We can now fuse the pruning and reversing operations into one CUDA kernel.



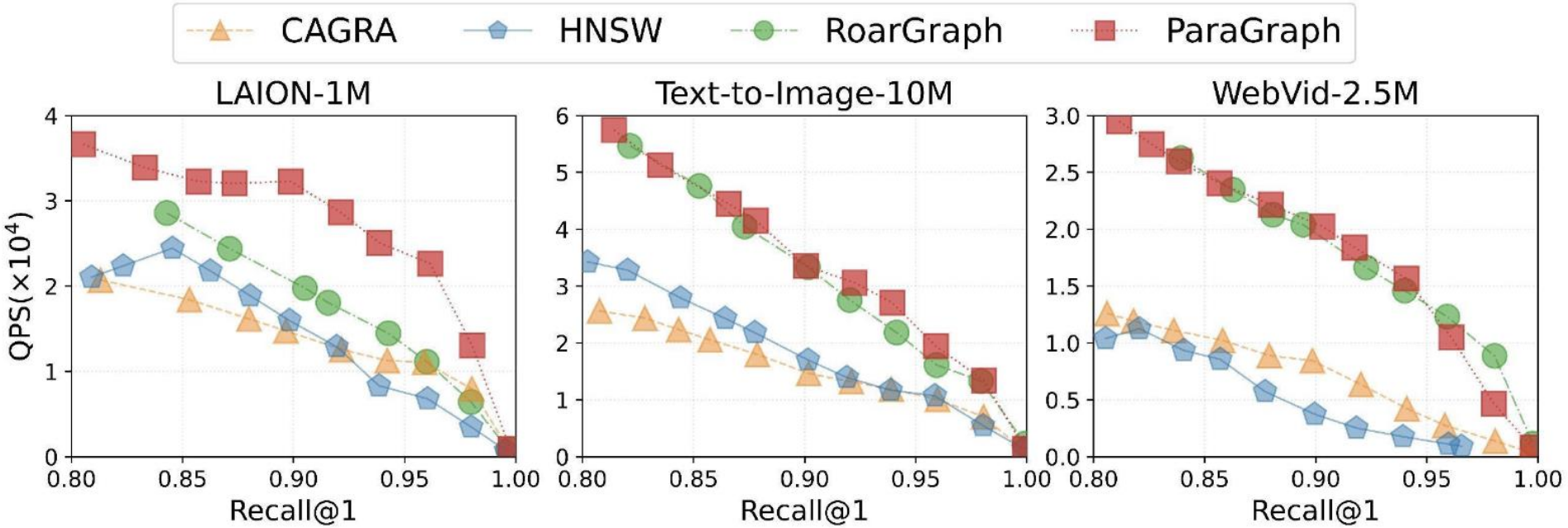
Experiment setup

- 2 Intel(R) Xeon(R) Gold 5318Y CPUs (all threads utilized)
- 1 NVIDIA A10 GPU with 24GB memory

Dataset	Size	Dimension	Modalities
Text-to-Image-10M	10000000	200	Text → Image
LAION-1M	1000000	512	Text → Image
WebVid-2.5M	2505000	512	Text → Video

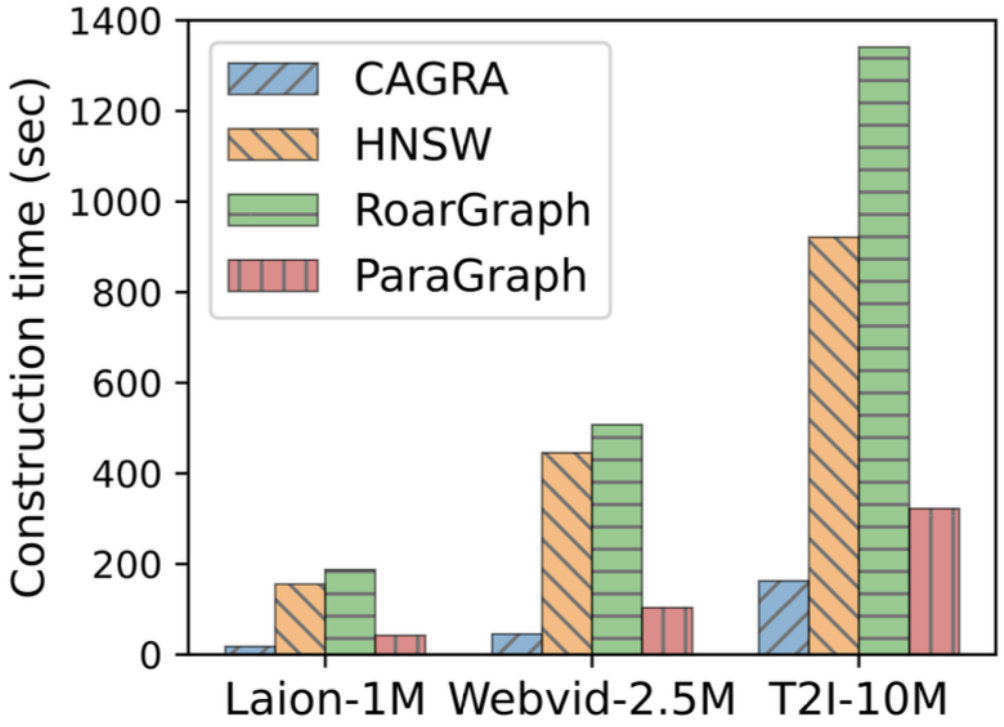
Search efficiency

Comparable or even better search efficiency than RoarGraph (SOTA)



Construction time

Up to 4.9x faster construction than RoarGraph



Takeaways

The *projection* is not only fast,
but also as *powerful* as the search-and-refine!

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

DBGroup

AlayaDB

DAMON
DATA MANAGEMENT ON NEW HARDWARE

Thank you!

research@alayadb.ai