ETH zürich

Chameleon: a Heterogeneous and Disaggregated Accelerator System for Retrieval-Augmented Language Models

Wenqi Jiang, Torsten Hoefler, and Gustavo Alonso

Department of Computer Science, ETH Zurich

H2RC @ SC'23, Nov. 17, 2023

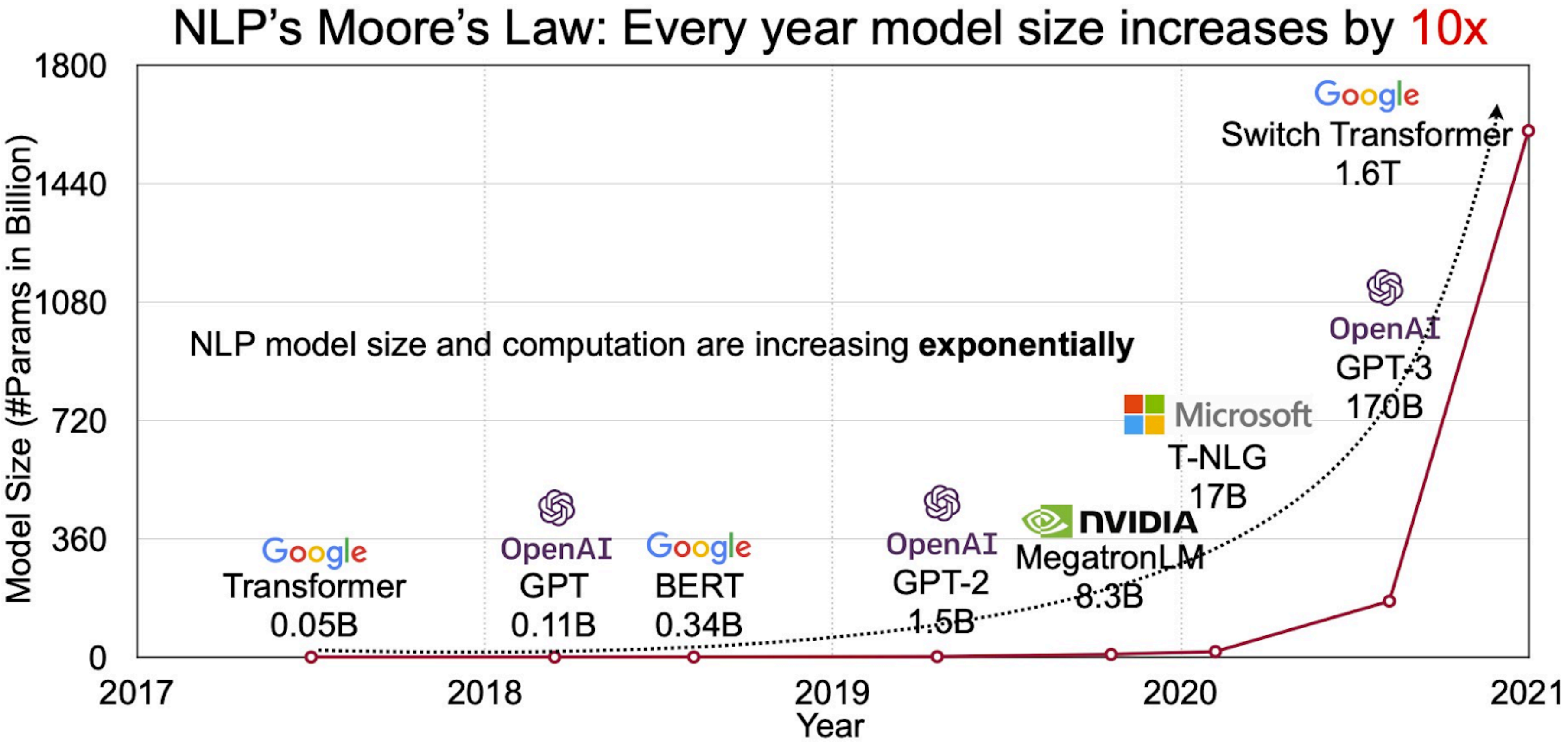
Systems@ETH zürich

Advancements of Large language models (LLMs)

ChatGPT

CONTENT CREATION

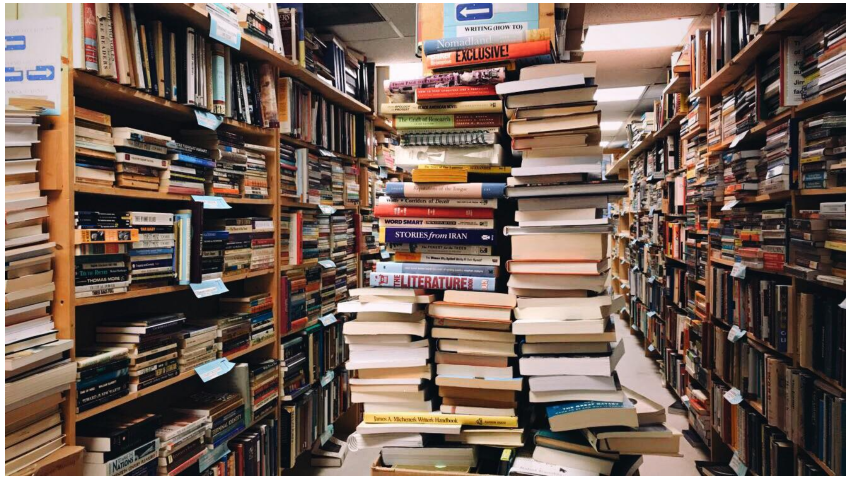
Better LLM quality relies on more parameters

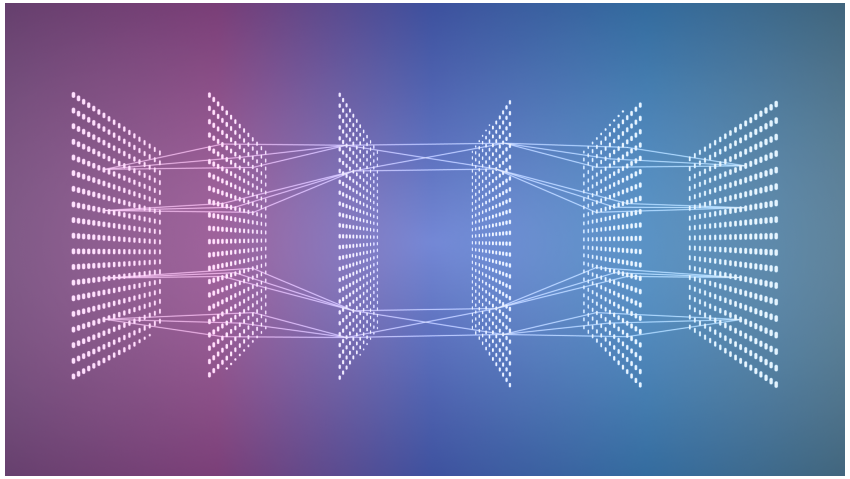


Source: https://indiaai.gov.in/article/the-future-of-large-language-models-llms-strategy-opportunities-and-challenges

Why more parameters?

LLM tries to compress textual knowledge into its parameters

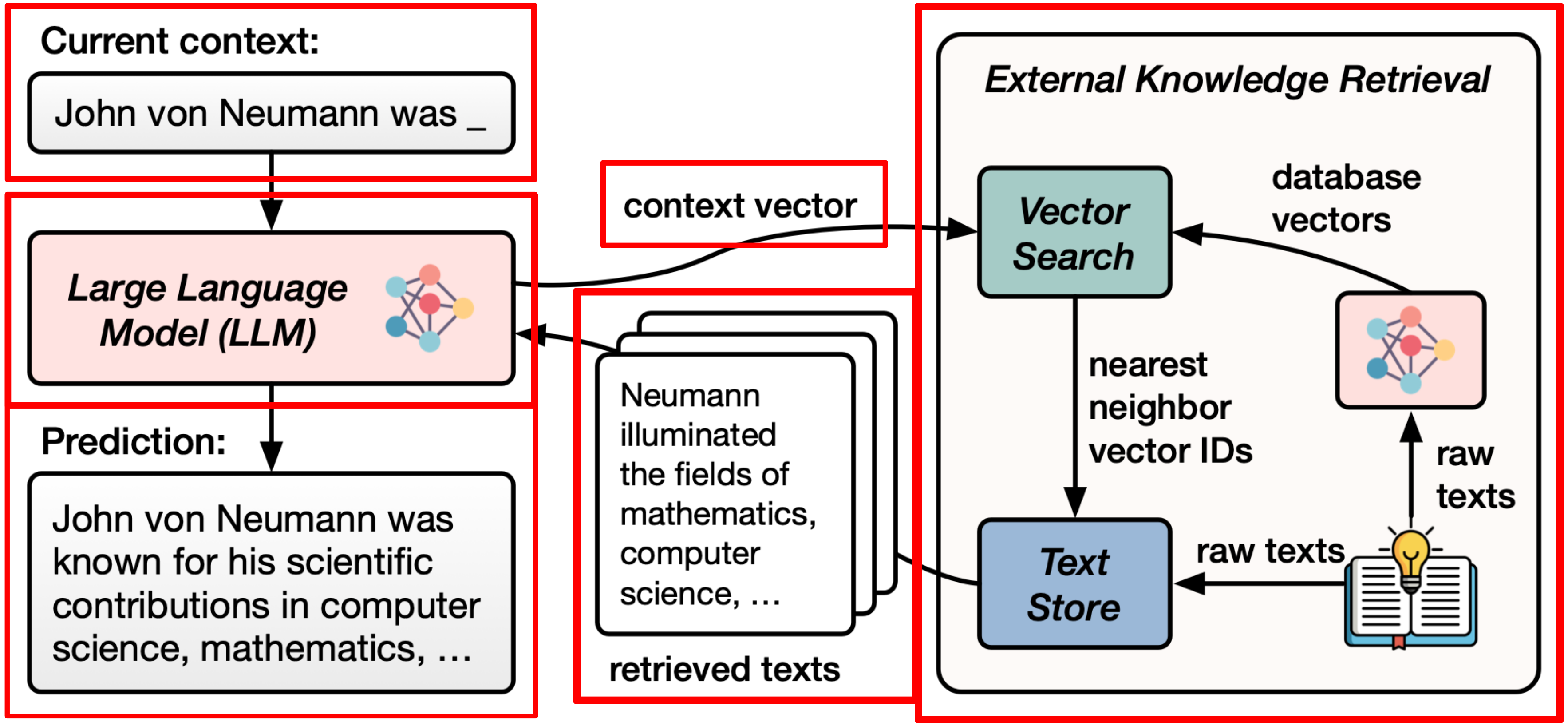




But there are serious problems by simply scaling up…

- High training and inference cost
- Cannot edit the knowledge without further training
  - Does not know the latest news
  - Hard to delete knowledge already learned from the training set
- No model personalization based on private knowledge

Retrieval-augmented generation as a rescue

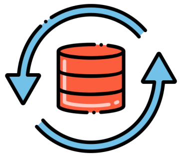


Retrieval-Augmented Language Models (RALM)

Reliability

reducing hallucinations by referencing external knowledge

Updatability

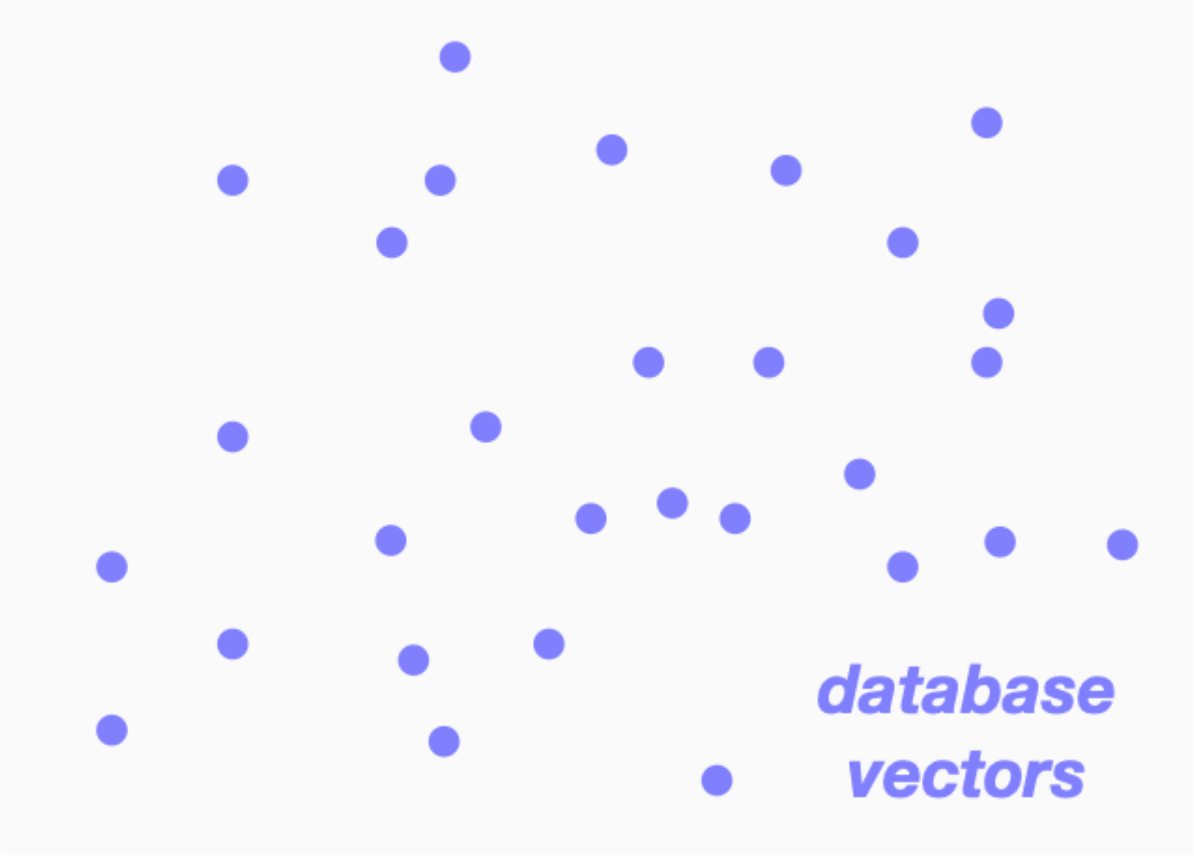


the external database can be easily updated (insertion, deletions, etc.)

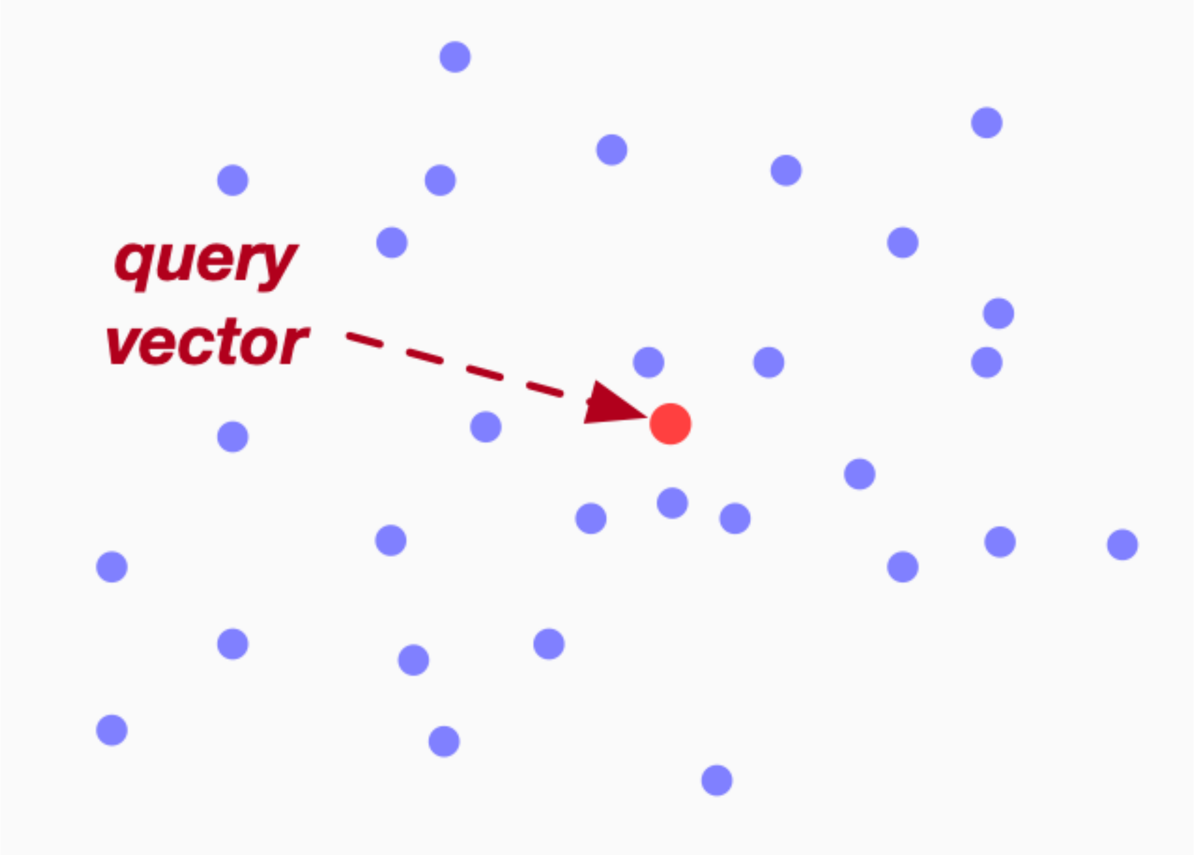
Efficiency

achieving superior generation quality with much fewer parameters

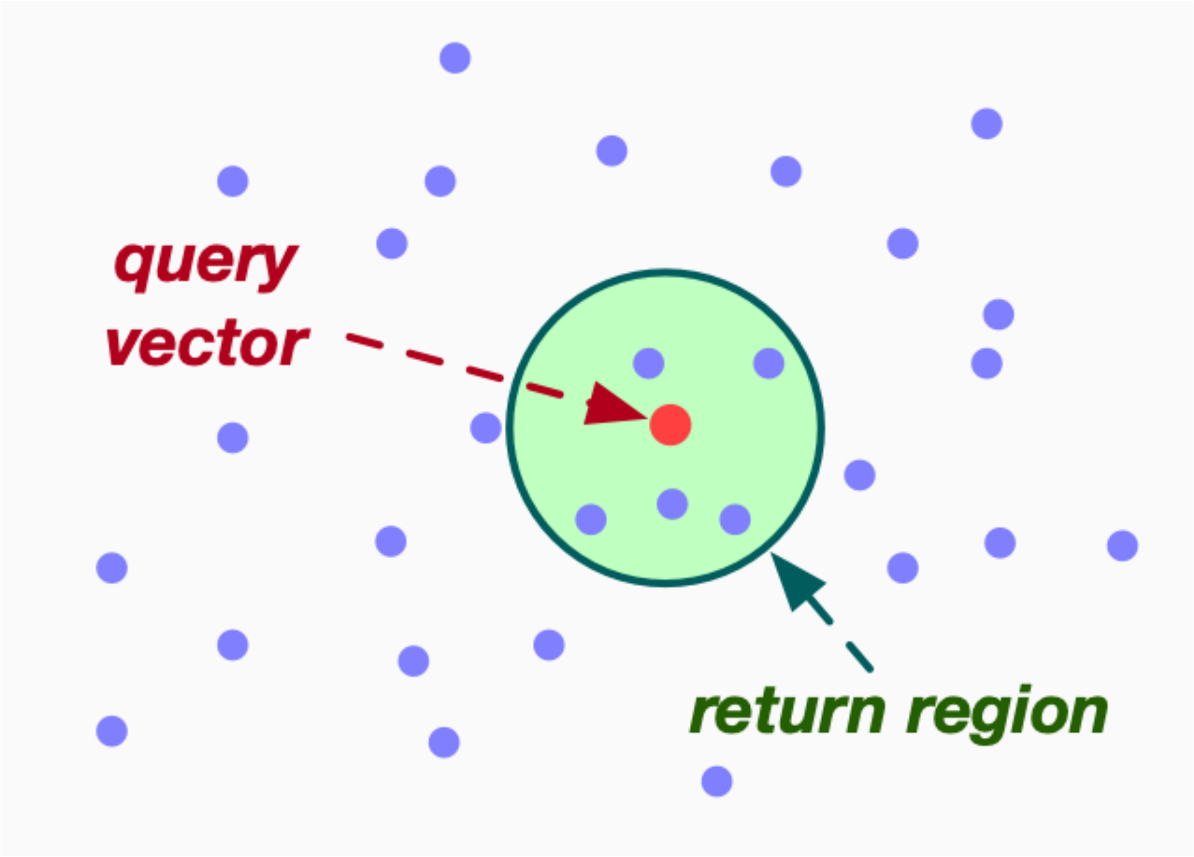
Vector search: problem definition



Vector search: problem definition



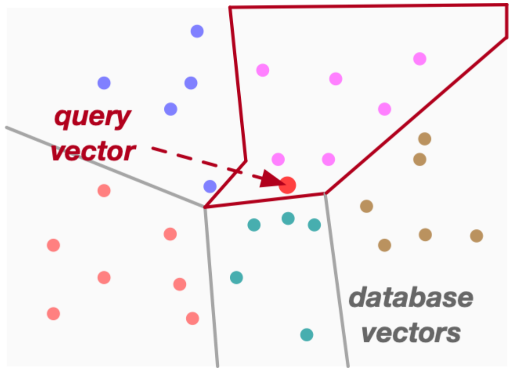
Vector search: problem definition



IVF-PQ for large-scale ANNS

Inverted-file (IVF) index

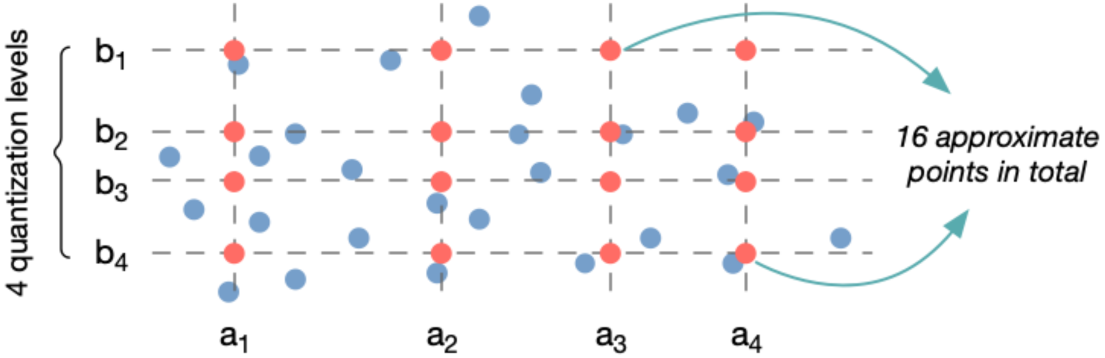
prune the search space



Product quantization (PQ)

quantize database vectors

speedup distance computation



Mapping large-scale search to CPUs

CPUs: slow at processing PQ codes

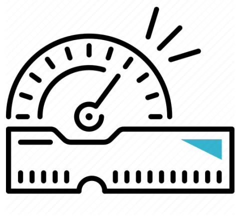
- too many cache accesses: twice per byte
- instruction dependencies: computation depends on the decoded data
- 1 GB/s per CPU core

Mapping large-scale search to GPUs

GPUs are prohibitively expensive at scale

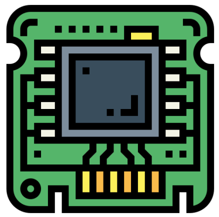
- H100 80 GB: $30K
- GPU cluster with 1 TB memory: $375K

The high bandwidth of GPUs is not fully leveraged



- multiple pass of read and write to the memory
- both PQ decoding and K-selection consume a lot of shared memory

The GPU architecture is not tailored for PQ



- waste of chip resources and energy

Proposed RALM system design principles

Both LLM inference and vector search should be fast and efficient

Principle 1: Accelerator heterogeneity

More research should be done on designing vector search accelerators

How are these accelerators connected?

Monolithic design: installing a certain number of LLM accelerators and retrieval accelerators on a same server

- not feasible for large databases
- cannot maximize accelerator utilizations due to the many RALM configurations such as retrieval intervals and model sizes

Proposed RALM system design principles

Both LLM inference and vector search should be fast and efficient

Principle 1: Accelerator heterogeneity

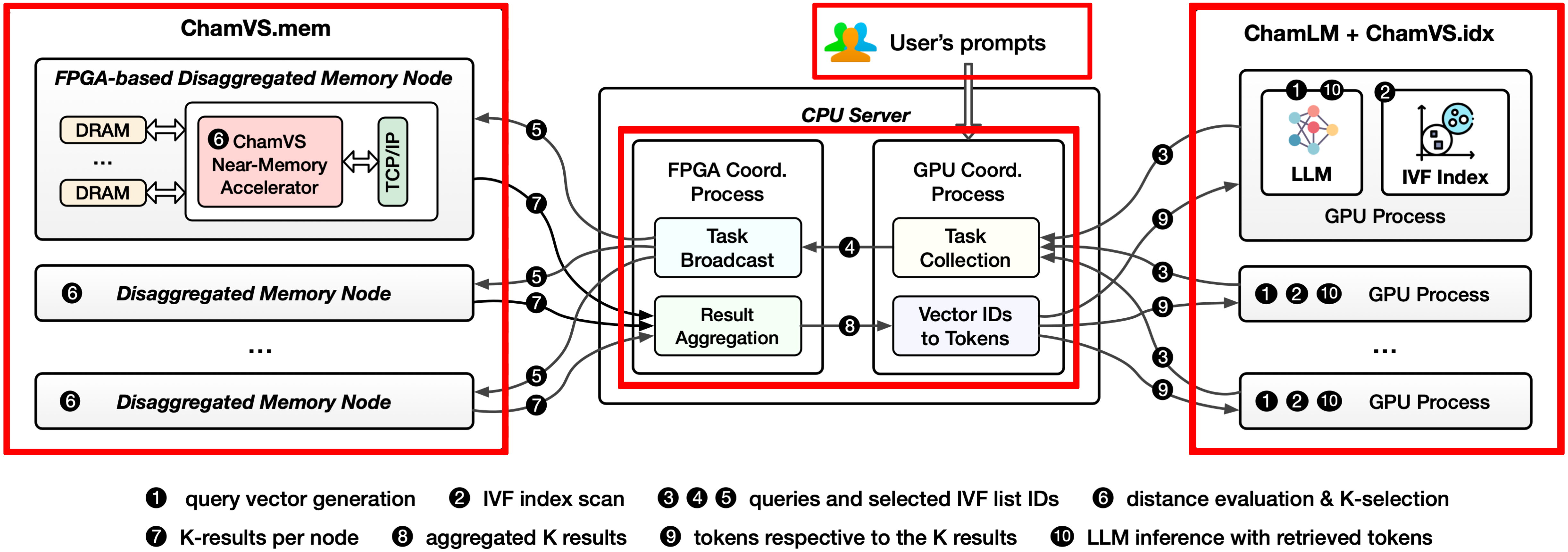
More research should be done on designing vector search accelerators

Flexibility to accommodate diverse RALM configurations

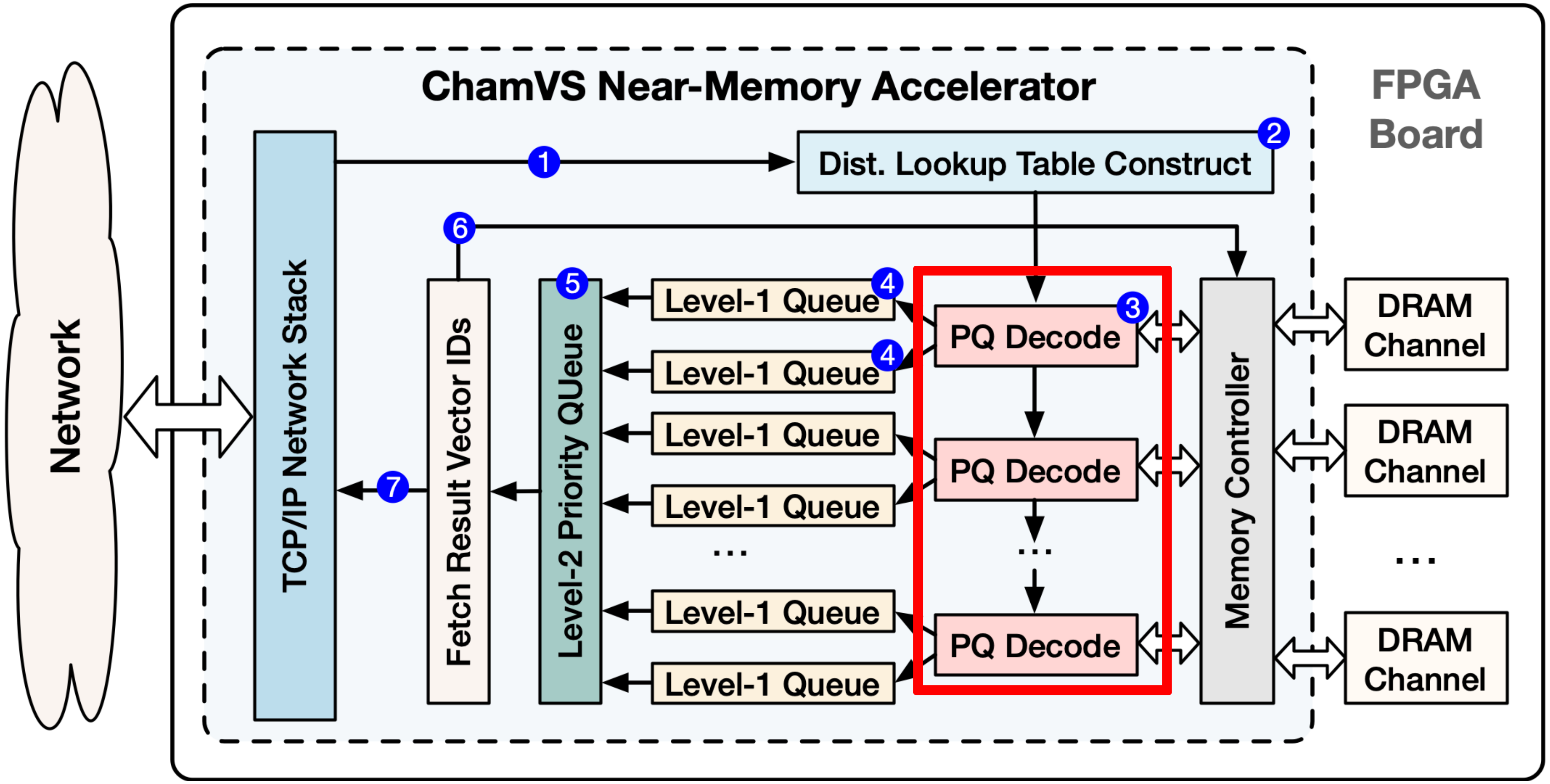
Principle 2: Accelerator disaggregation

Various performance bottlenecks and system requirements across RALMs

Chameleon overview

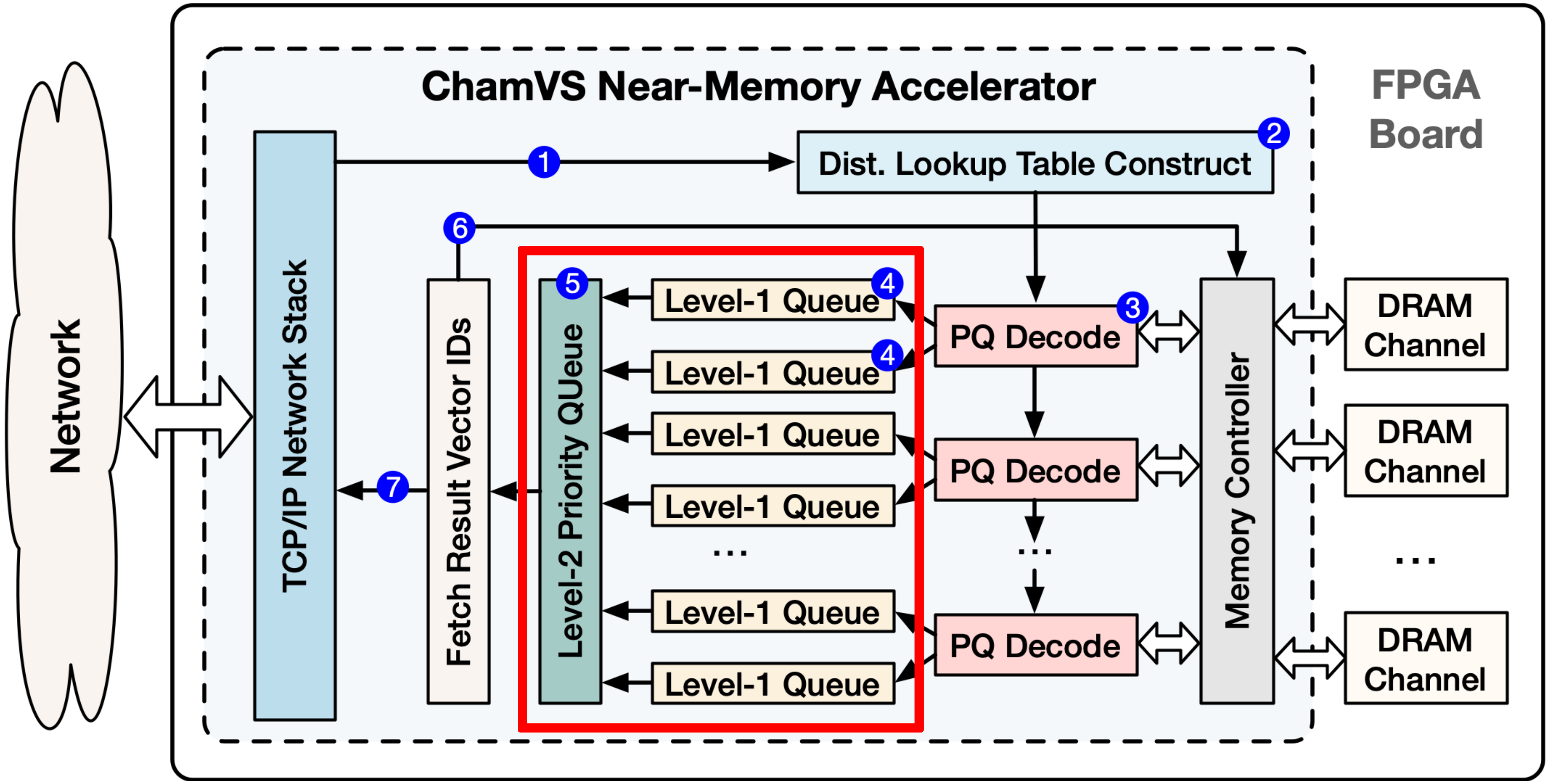


Accelerated disaggregated memory node



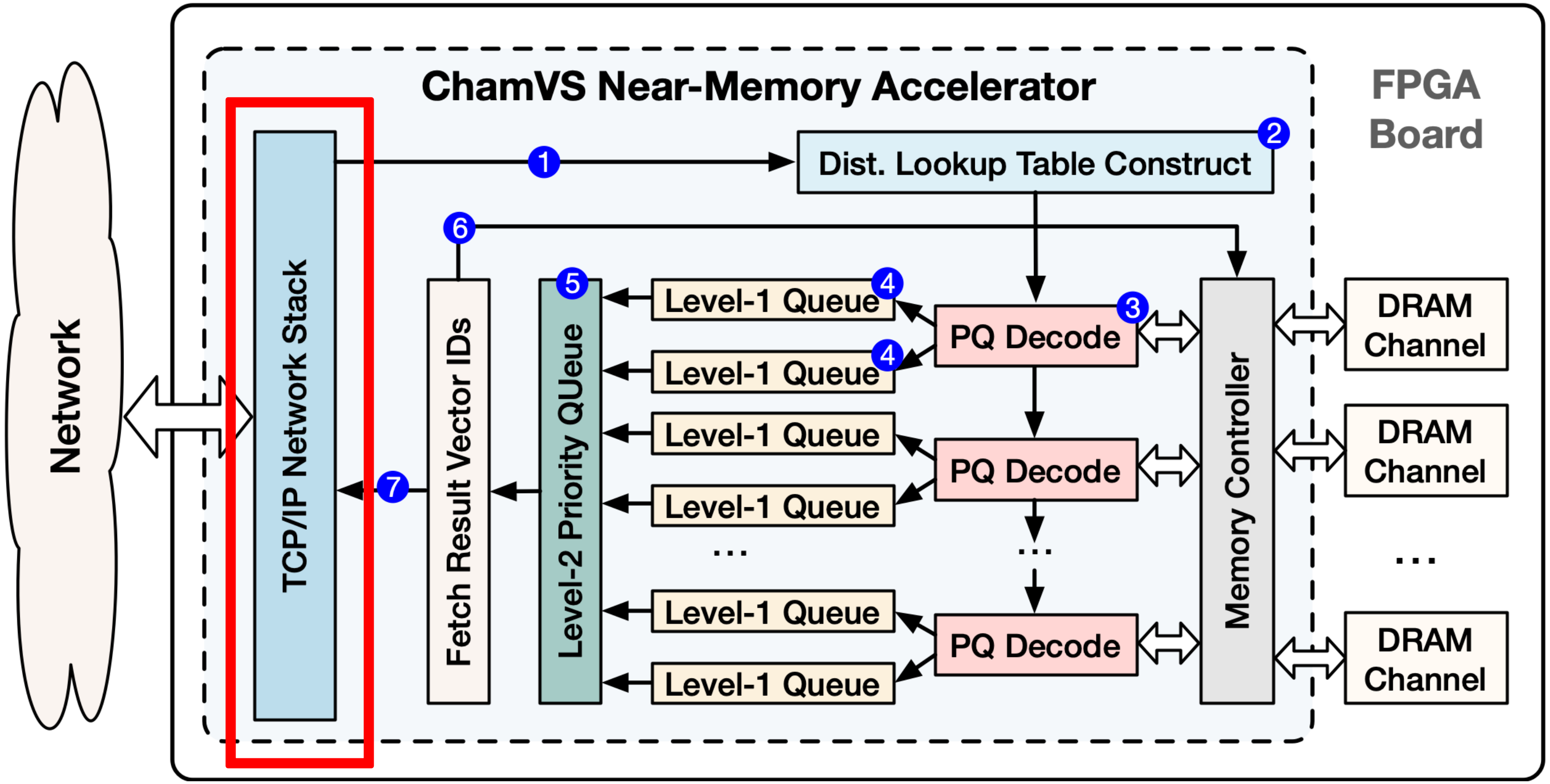
Rapidly processing quantized database vectors

Accelerated disaggregated memory node



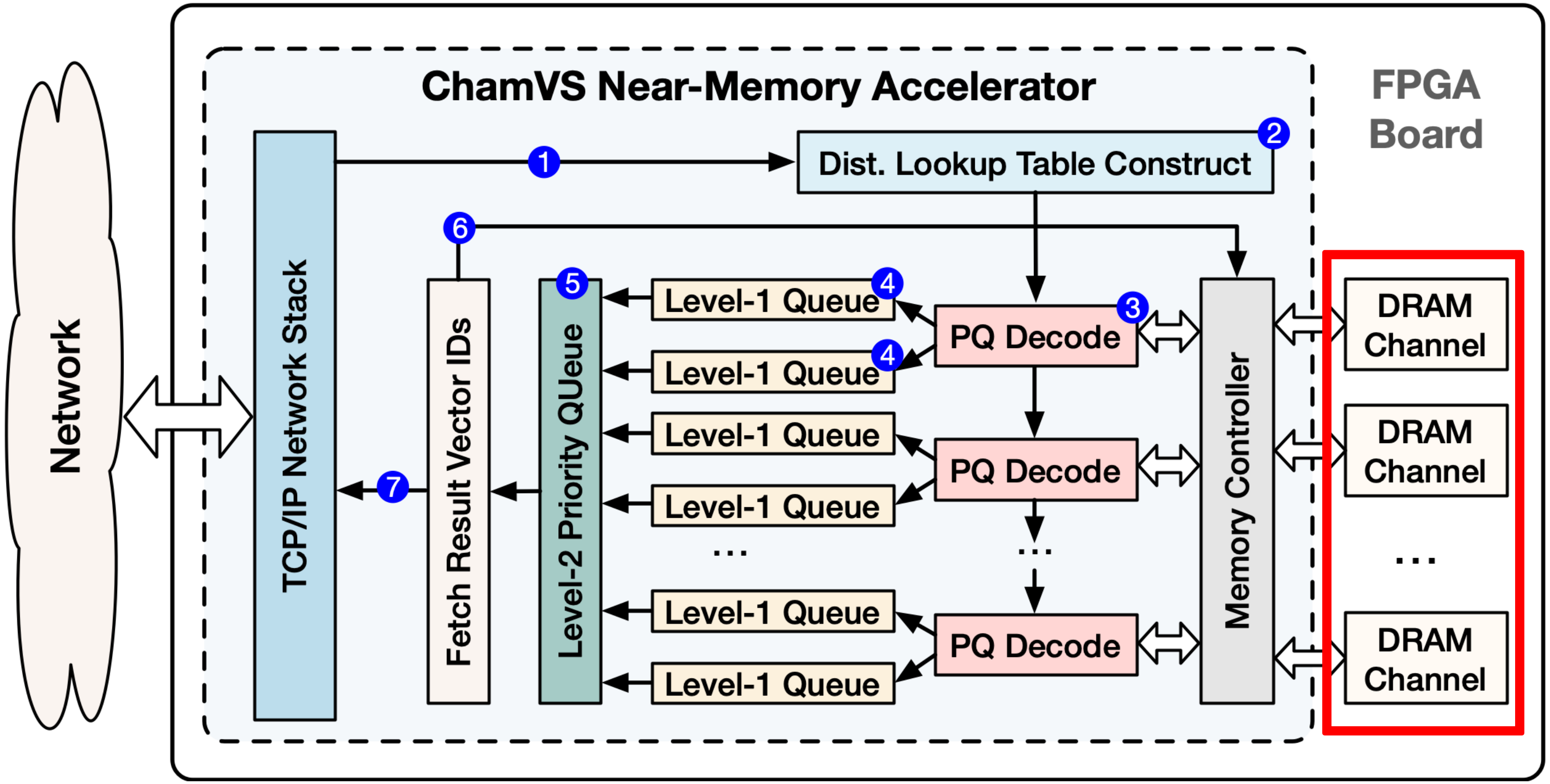
High-throughput and resource-efficient K-selection

Accelerated disaggregated memory node



Direct access to the network, bypassing PCIe

Accelerated disaggregated memory node



Operate on the physical address space; load-balance across channels

Evaluation settings

Various model architectures, sizes, and retrieval intervals

	Dim.	Layers	Heads	Param.	Interval	K
Dec-S	512	24	8	101M	1	100
Dec-L	1024	96	16	1259M	1	100
EncDec-S	512	2,24	8	158M	8/64/512	10
EncDec-L	1024	2,96	16	1738M	8/64/512	10

Evaluation settings

Vector search benchmarks of different dimensionalities

	Deep	SIFT	SYN-512	SYN-1024
#vec	1E+9	1E+9	1E+9	1E+9
D	96	128	512	1,024
m	16	16	32	64
nlist	32,768	32,768	32,768	32,768
Raw vectors (GB)	384	512	4,096	8,192
PQ and vec ID (GB)	24	24	40	72

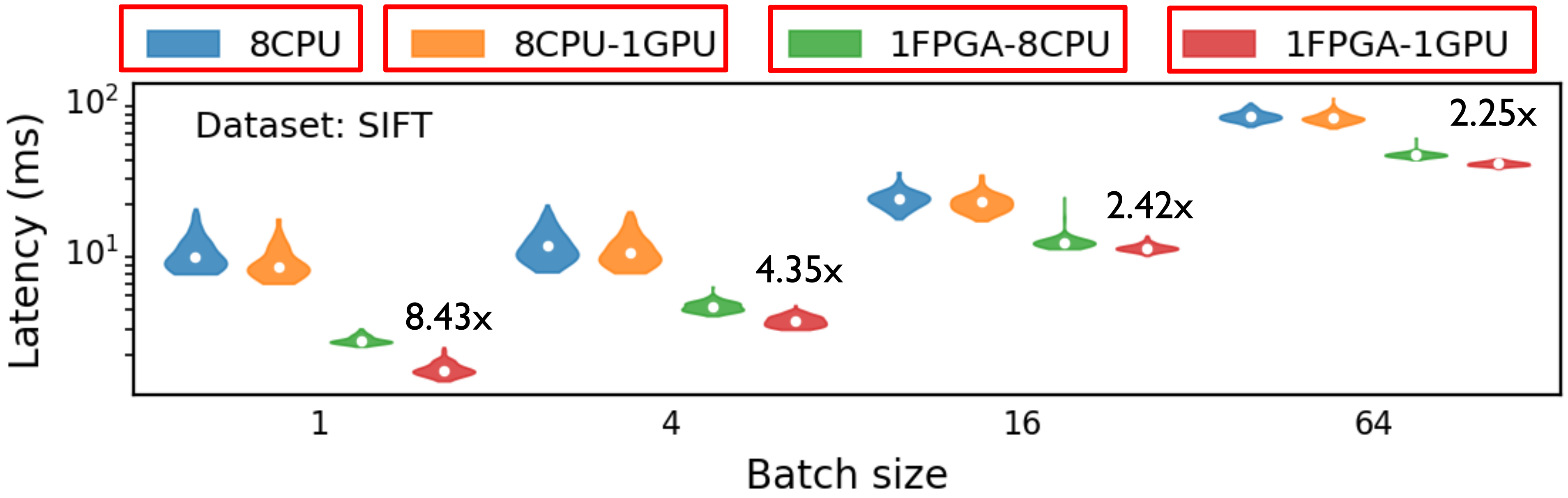
Evaluation settings

AMD Alveo U250 FPGA (16 nm) equipped with 64 GB of DDR4 memory (4 channels x 16 GB).

CPU-based vector search system with equivalent memory capacity (64 GB) and an 8-core AMD EPYC 7313 processor (7 nm) with a base frequency of 3.0 GHz and a max turbo frequency of 3.7 GHz.

NVIDIA RTX 3090 GPUs (8nm) with 24 GB GDDR6X memory.

Vector search performance (SIFT dataset)



FPGA-GPU solution achieves 1.82~16.59x speedup over CPU across datasets

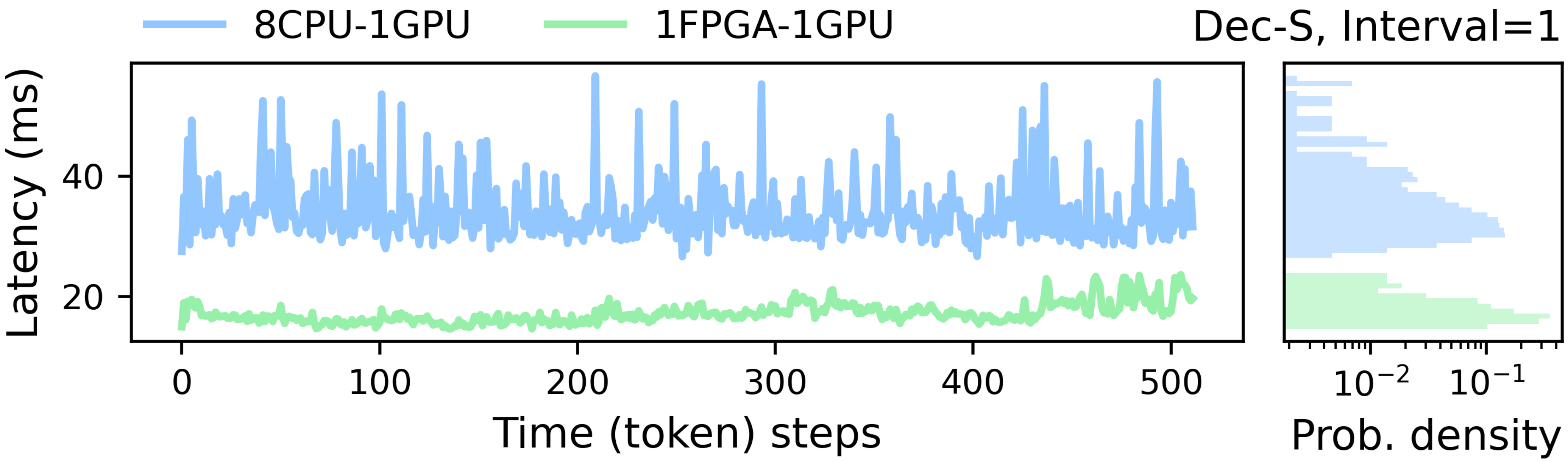
FPGA-GPU solution achieves up to 3.87x speedup over FPGA-CPU

Chameleon can take advantage of existing GPUs

End-to-end RALM latency

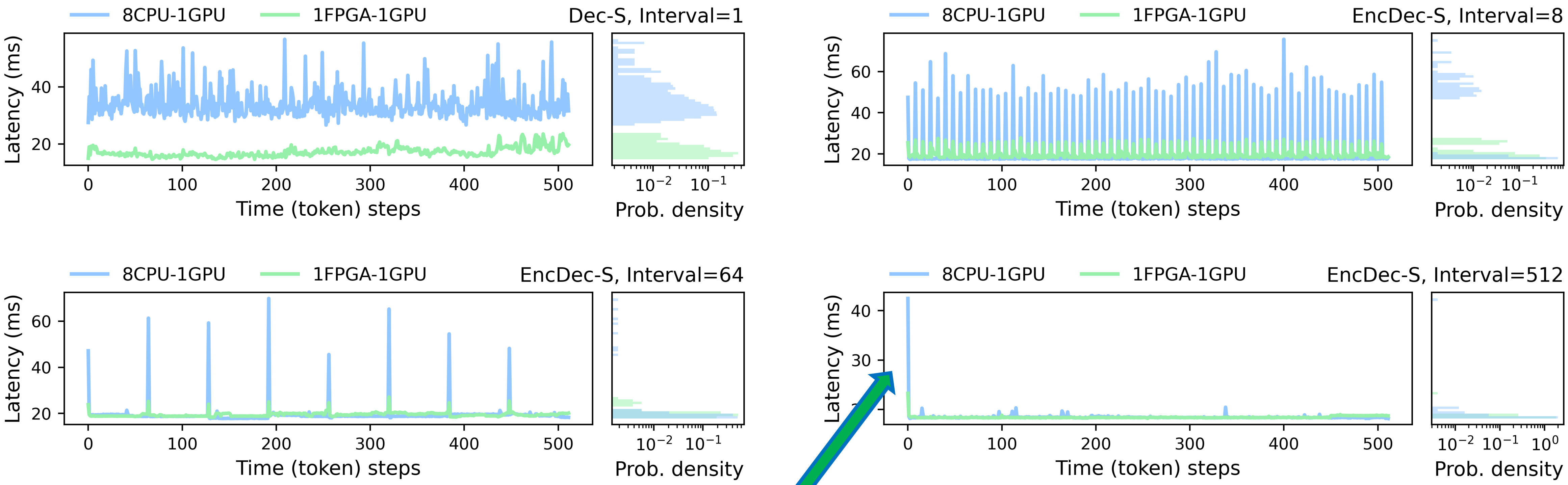
Vector search setting: CPU only versus GPU + FPGA

LLM inference setting: always use GPU



1.98x end-to-end speedup in latency

End-to-end RALM latency



Disaggregation is required to maximize utilizations and meet demands

Conclusion

Retrieval augmentation will drive the next-generation LLMs

Key design principles for RALM systems: heterogeneity and disaggregation

Chameleon: prototype those principles on CPUs, GPUs, and FPGAs

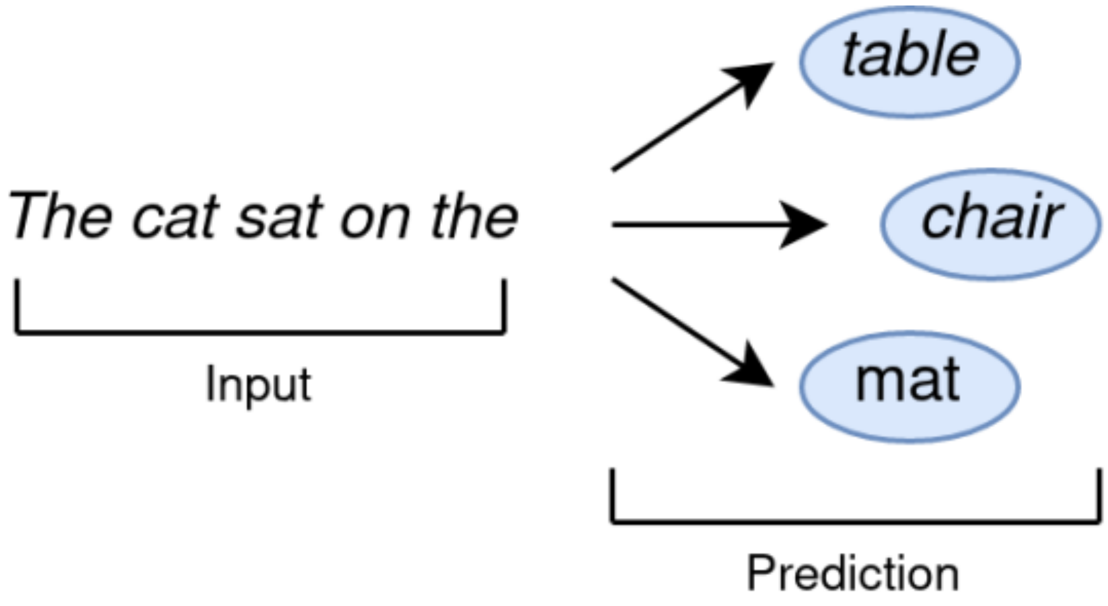
Up to 2.16x latency reduction and up to 3.18x throughput improvements

Preprint available: https://arxiv.org/pdf/2310.09949.pdf

Backup slides

Language Models

A generative large language model (LLM) is a machine learning model trained to predict the probability of a sequence of words.



What about the compression ratio?

Common Crawl: 200~300 billions of web pages

translates to 200~300 TB text data assuming 1KB per page

GPT3: 175 billion parameters

350 GB using float16

1000x compression rate!

Learns roughly rather than precisely

World Knowledge vs Linguistic Structures

ETH Zurich was ________

→ established

→ founded

→ the

Sentence found on: https://en.wikipedia.org/wiki/ETH_Zurich

World Knowledge vs Linguistic Structures

established

ETH Zurich was ________ founded

the

Sentence found on: https://en.wikipedia.org/wiki/ETH_Zurich

World Knowledge vs Linguistic Structures

ETH Zurich was founded by the ________

Swiss government

City of Zurich

European Union

Sentence found on: https://en.wikipedia.org/wiki/ETH_Zurich

World Knowledge vs Linguistic Structures

ETH Zurich was founded by the ________

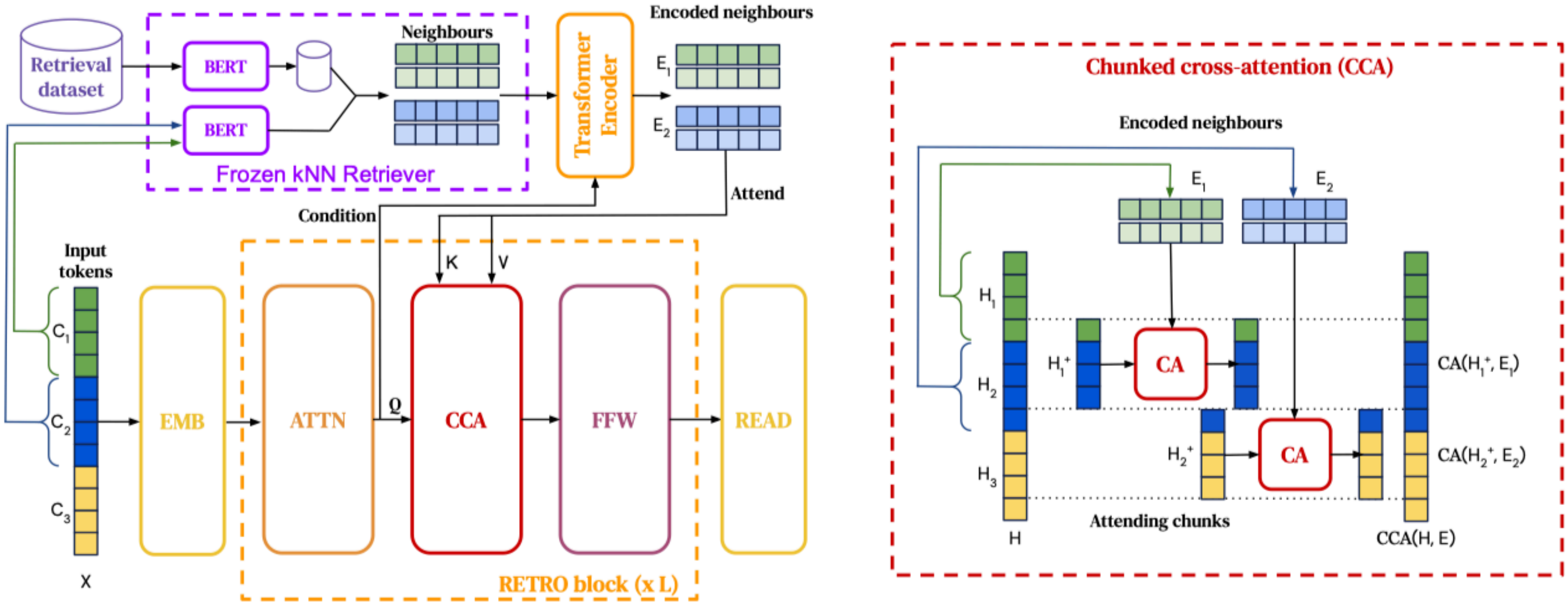
Swiss government

City of Zurich

European Union

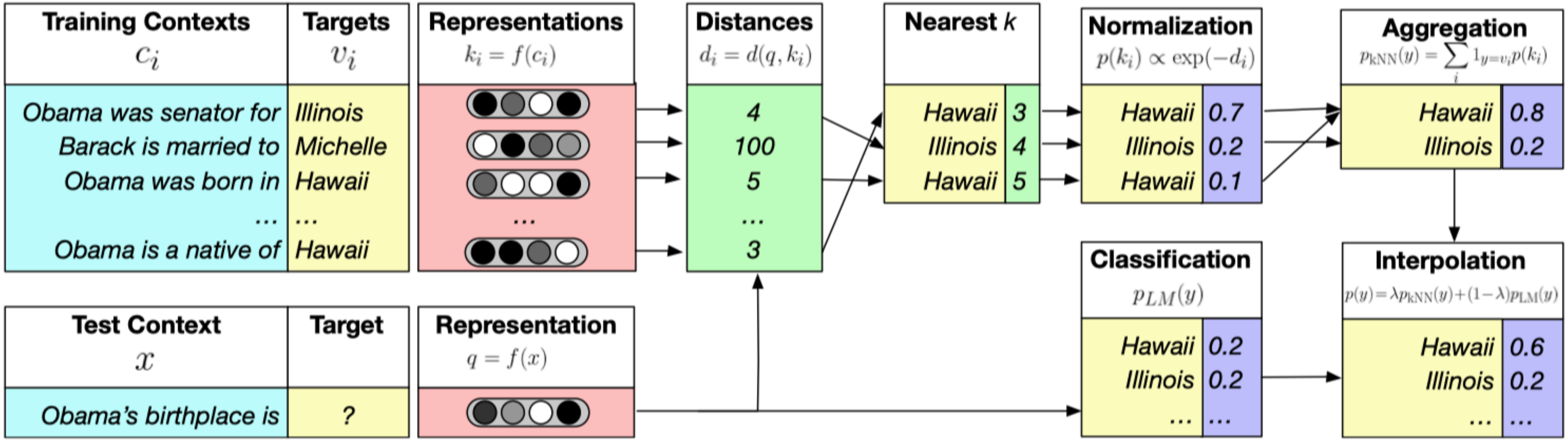
Sentence found on: https://en.wikipedia.org/wiki/ETH_Zurich

RETRO model architecture



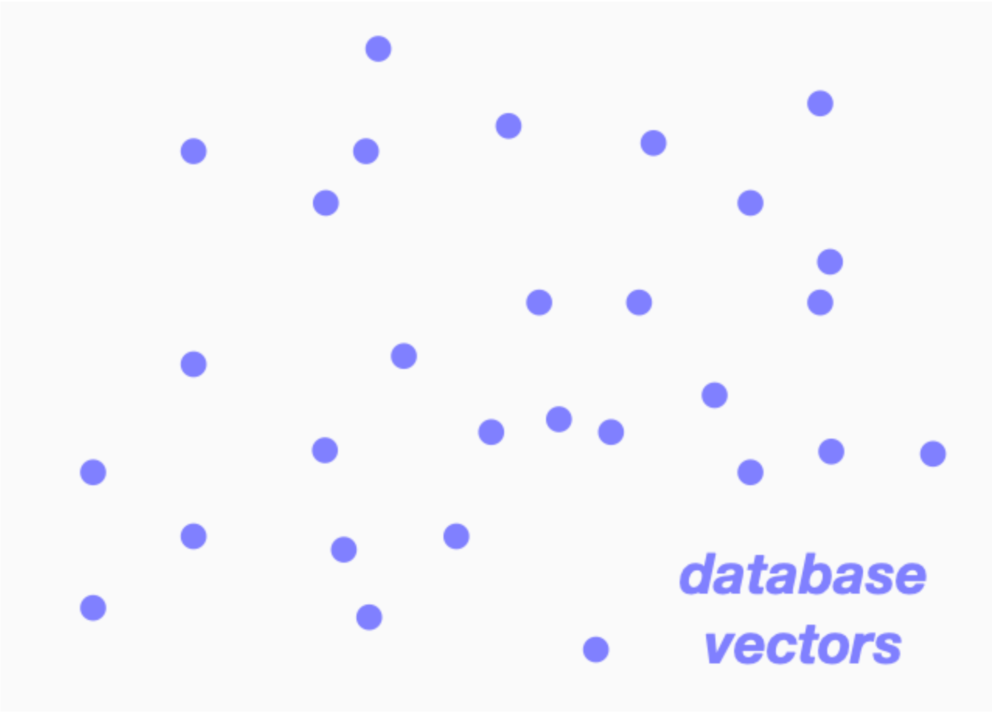
Borgeaud, Sebastian, et al. "Improving language models by retrieving from trillions of tokens." *International conference on machine learning*. PMLR, 2022.

kNN-LM model architecture



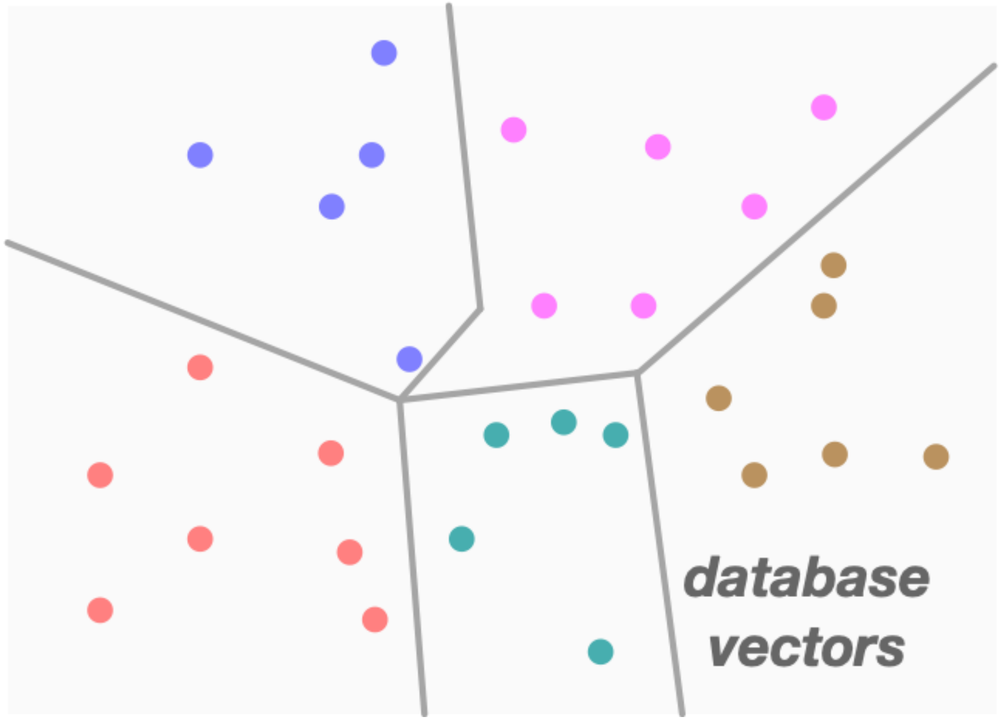
Khandelwal, Urvashi, et al. "Generalization through memorization: Nearest neighbor language models." *arXiv preprint arXiv:1911.00172* (2019).

Inverted-file (IVF) index



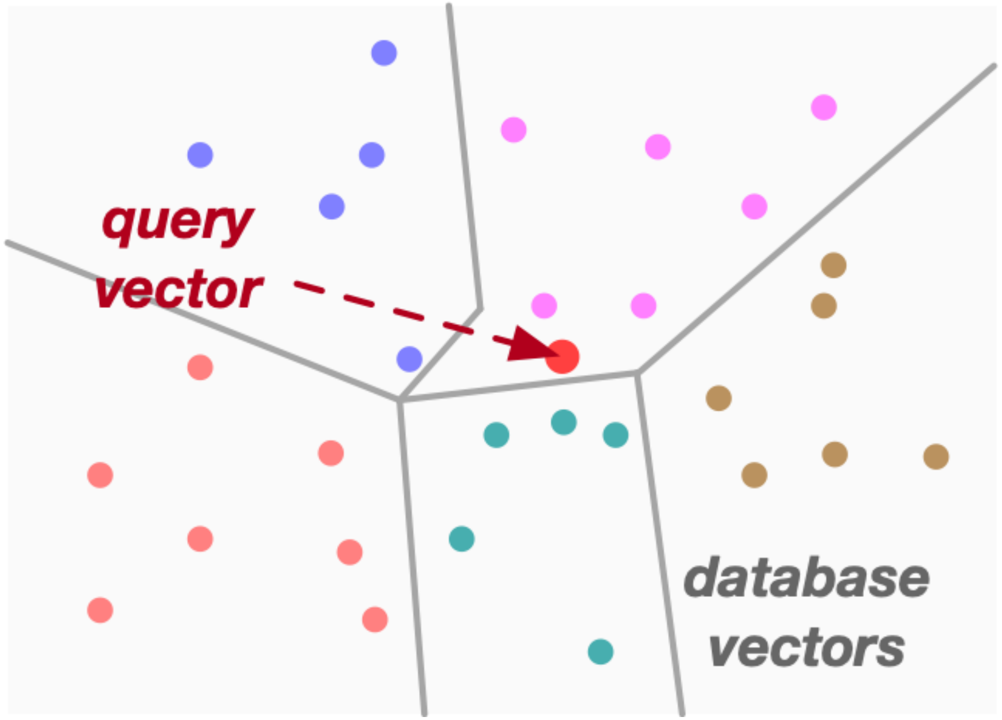
Training: cluster database vectors into IVF lists

Inverted-file (IVF) index



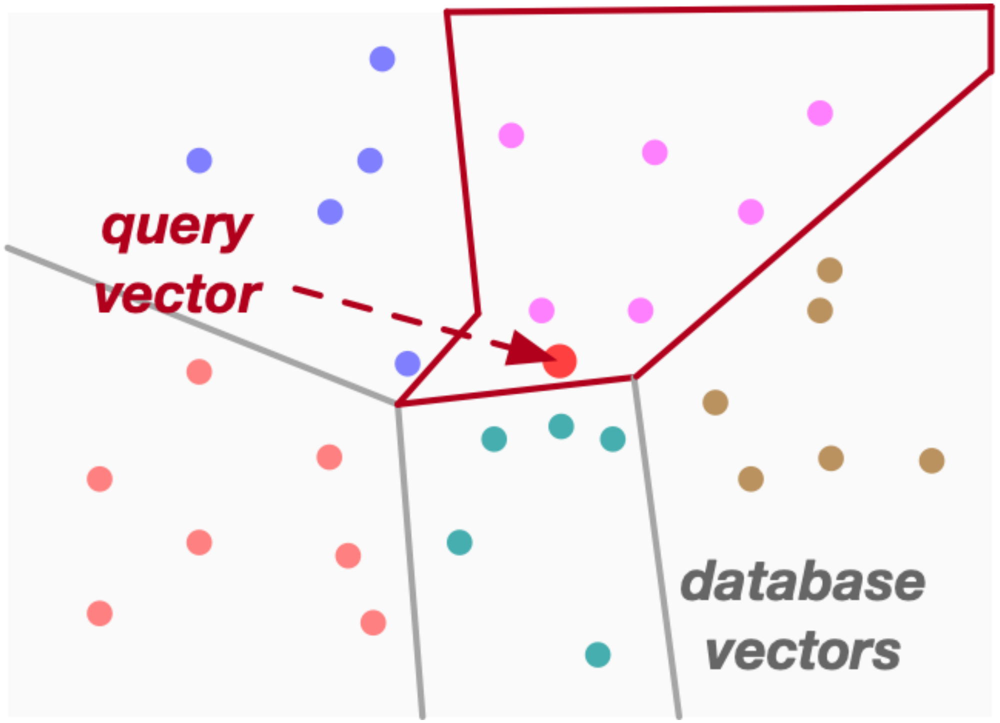
Training: cluster database vectors into IVF lists

Inverted-file (IVF) index



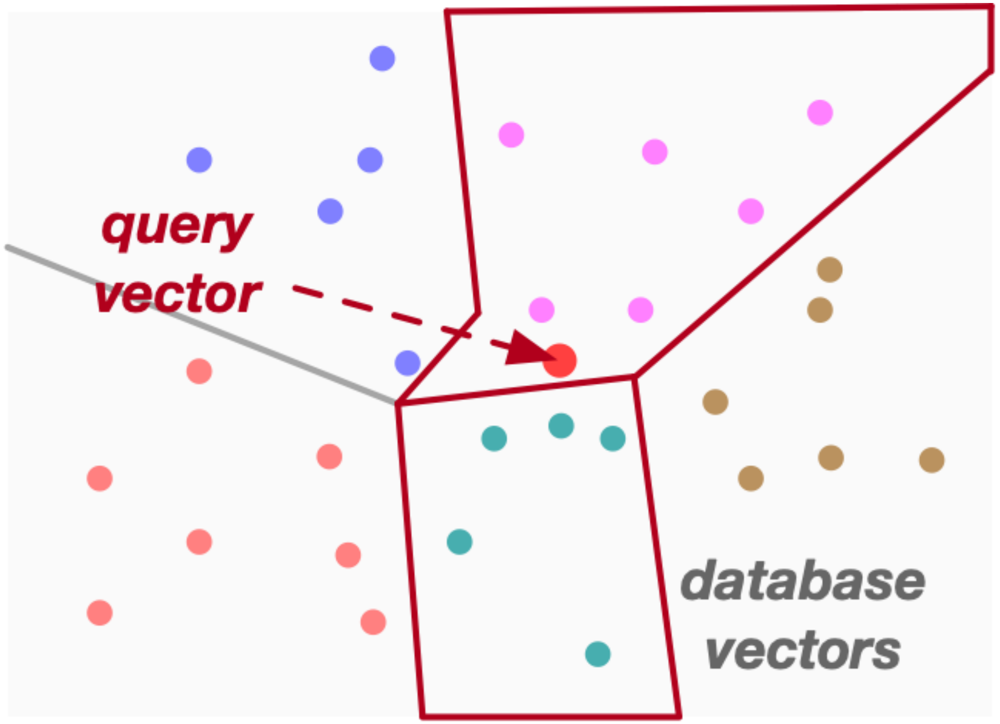
Searching: scan only a subset of IVF lists

Inverted-file (IVF) index



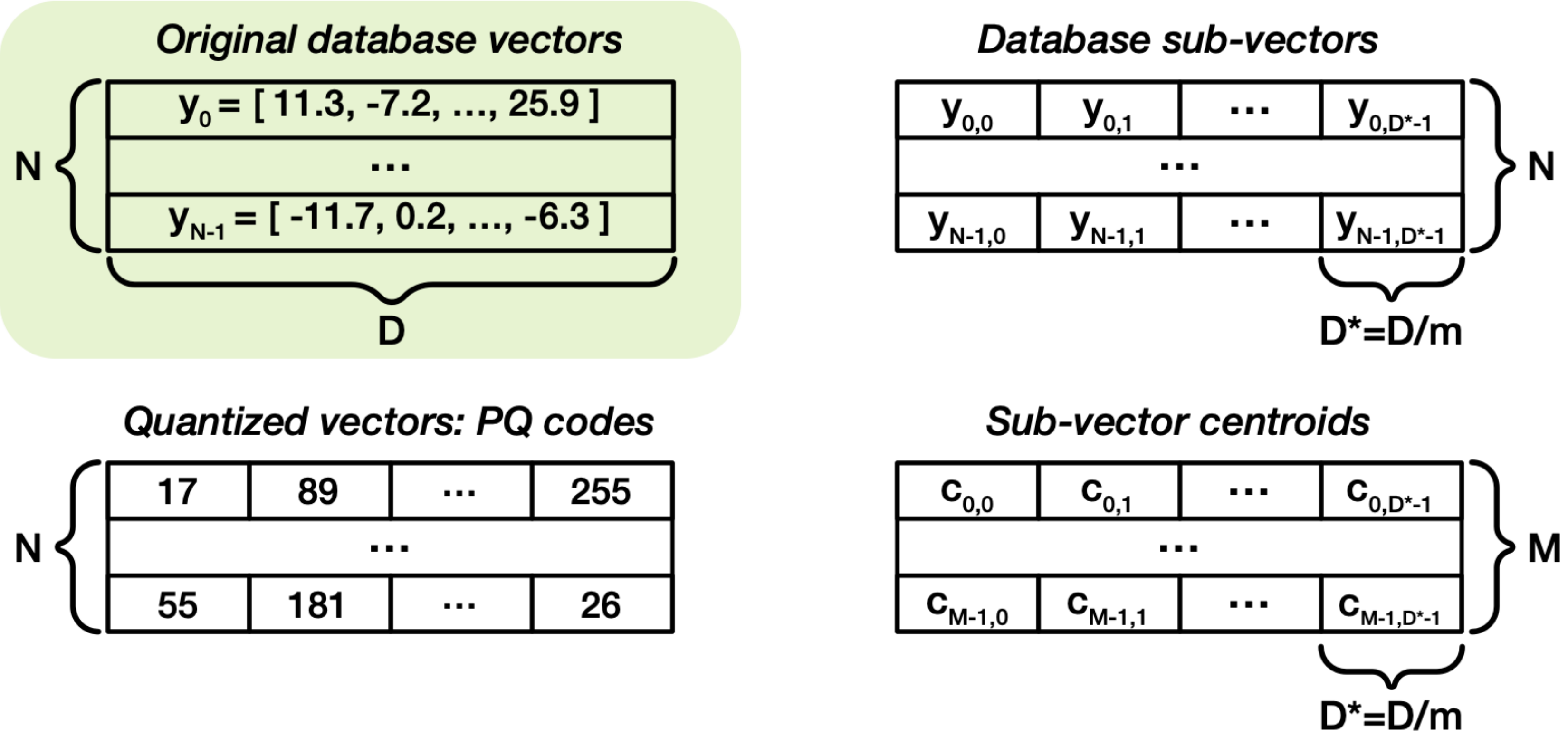
Searching: scan only a subset of IVF lists

Inverted-file (IVF) index



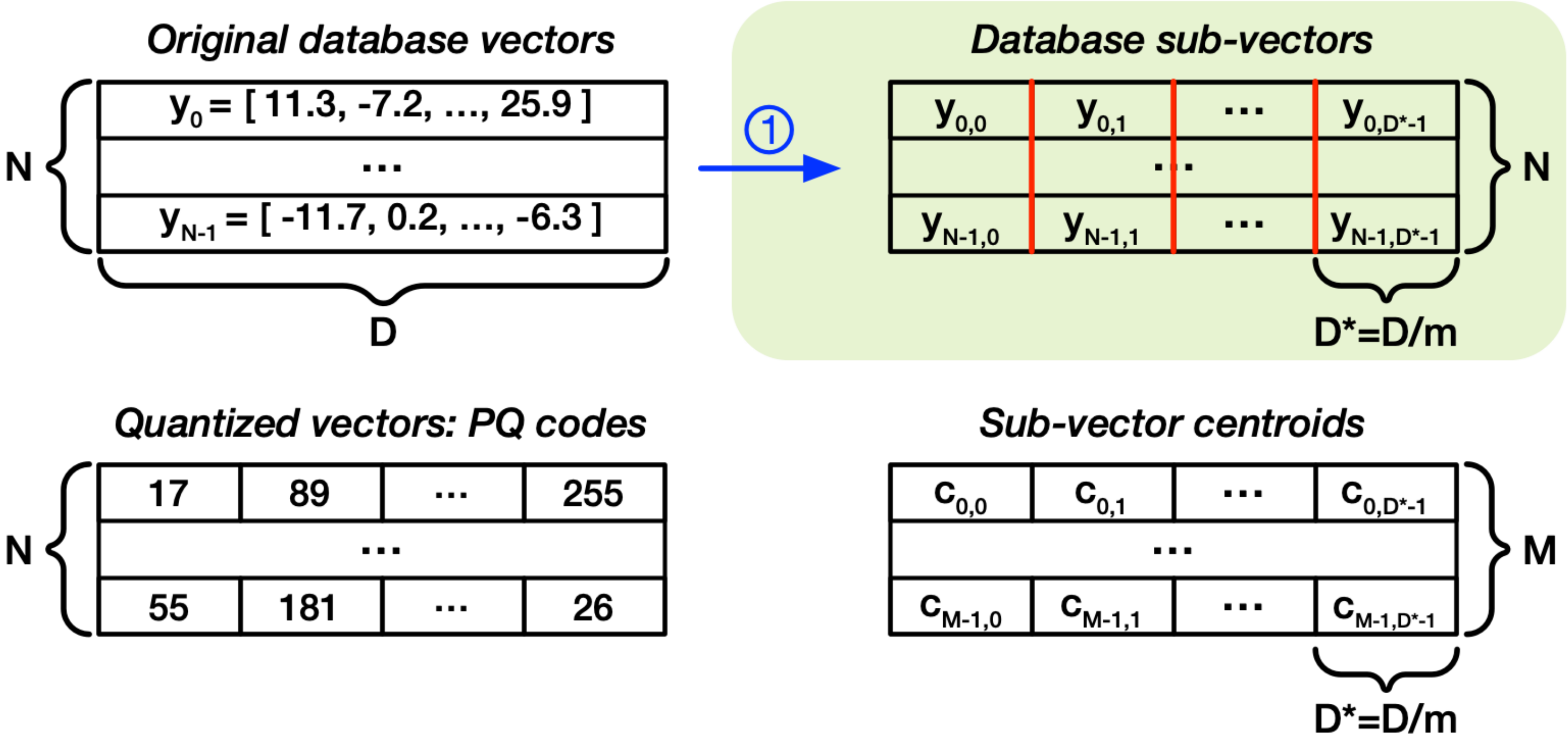
Searching: scan only a subset of IVF lists

Product quantization (PQ): training



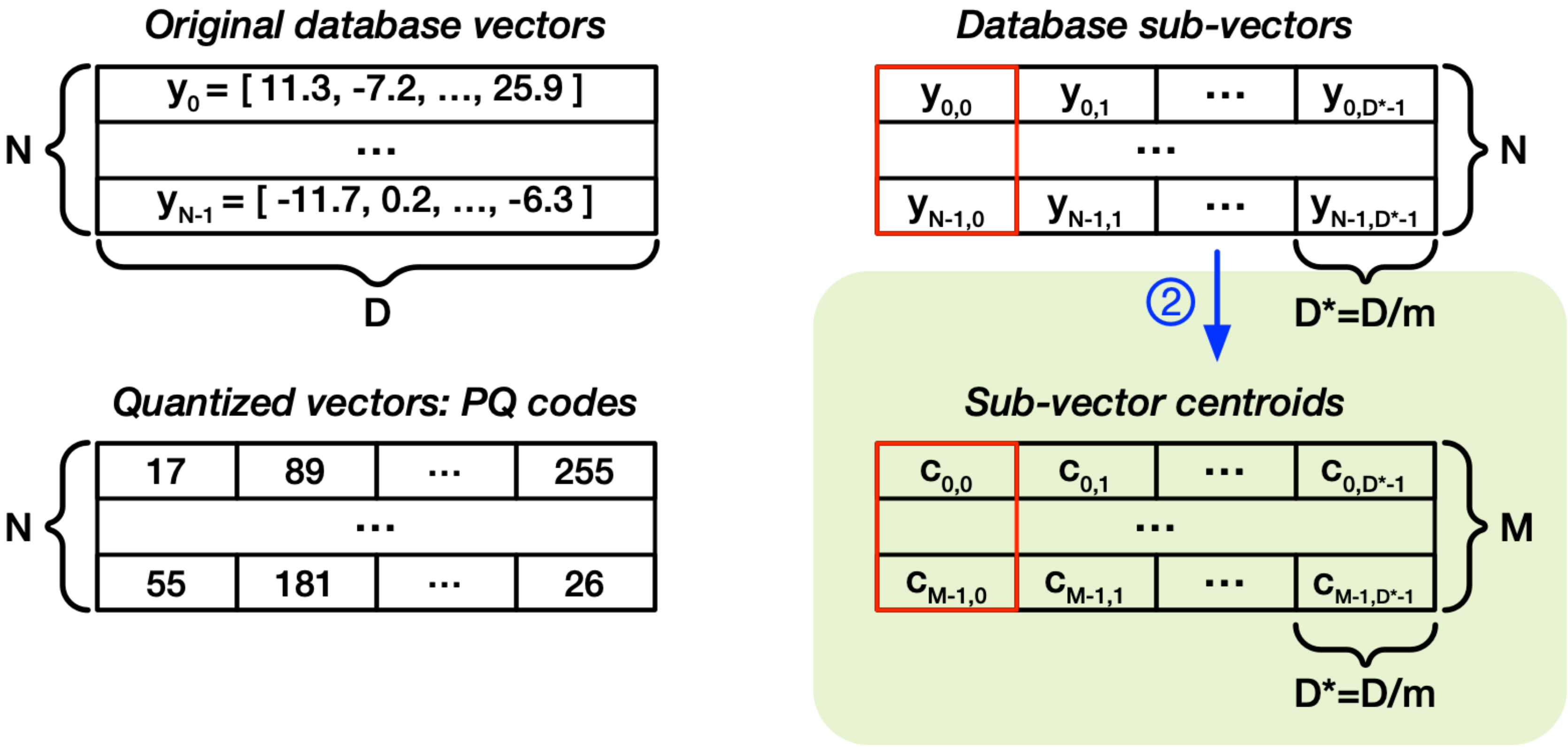
Quantize the vectors to a few bytes of PQ-codes

Product quantization (PQ): training



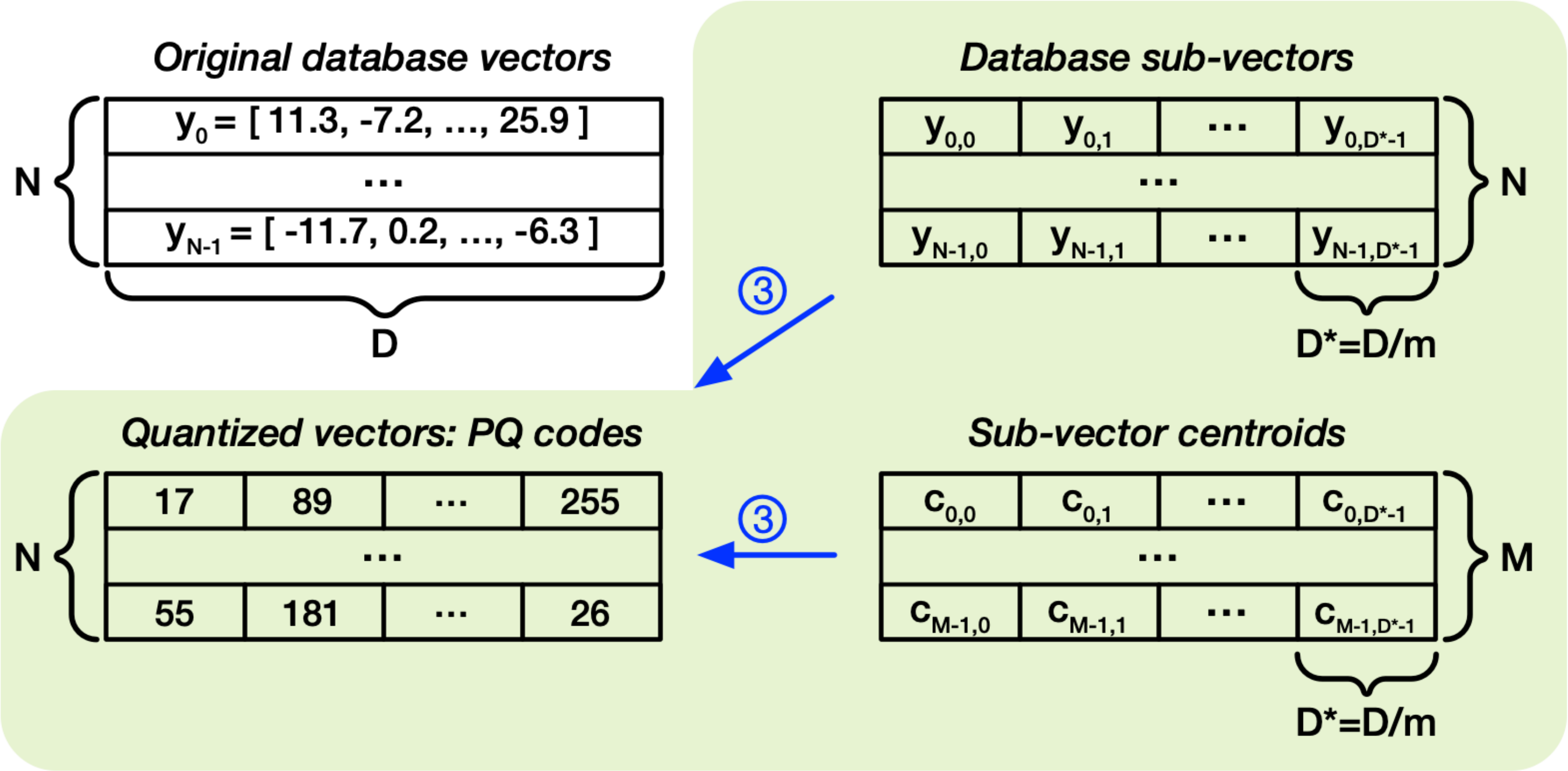
Quantize the vectors to a few bytes of PQ-codes

Product quantization (PQ): training



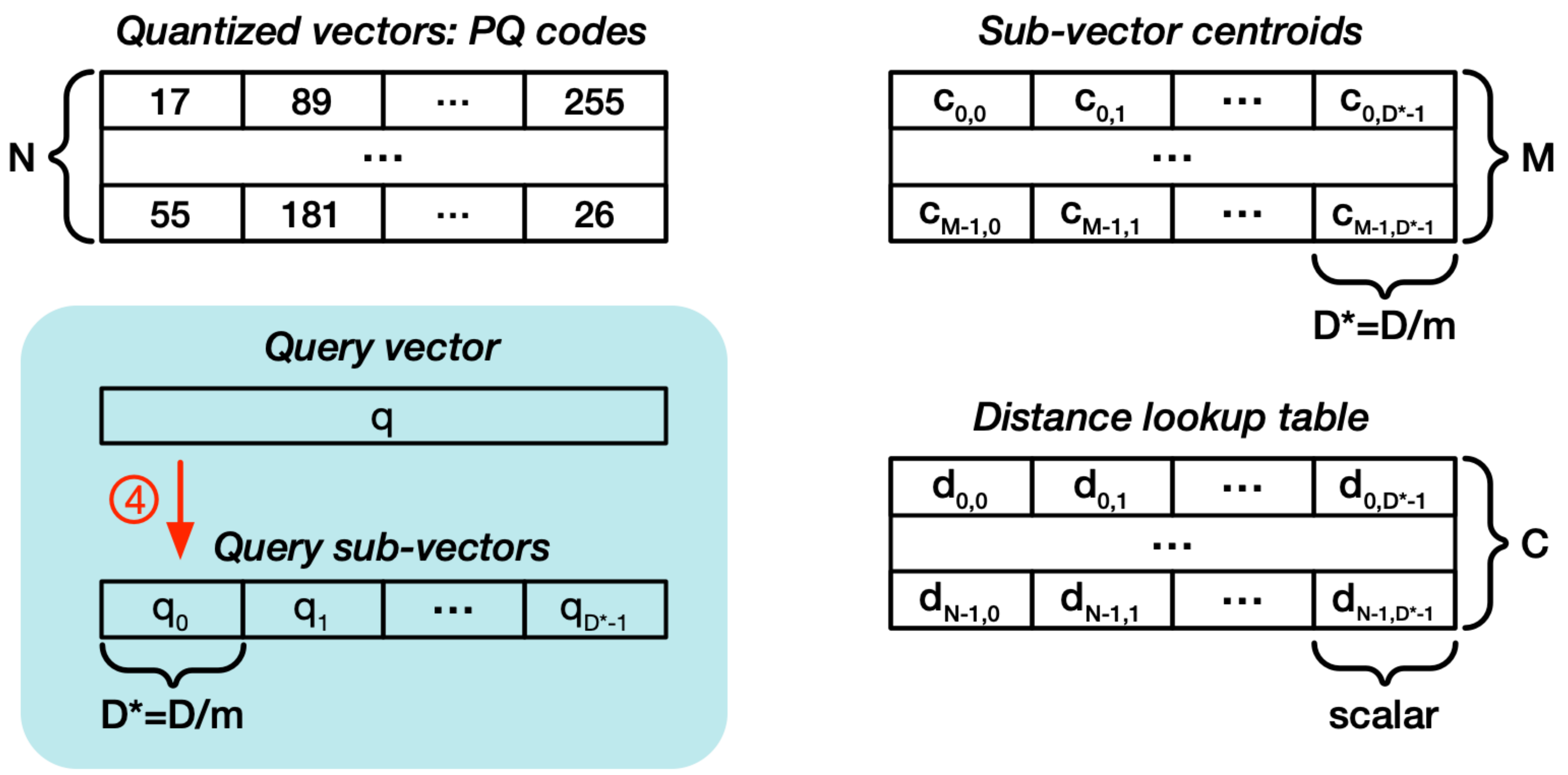
Quantize the vectors to a few bytes of PQ-codes

Product quantization (PQ): training



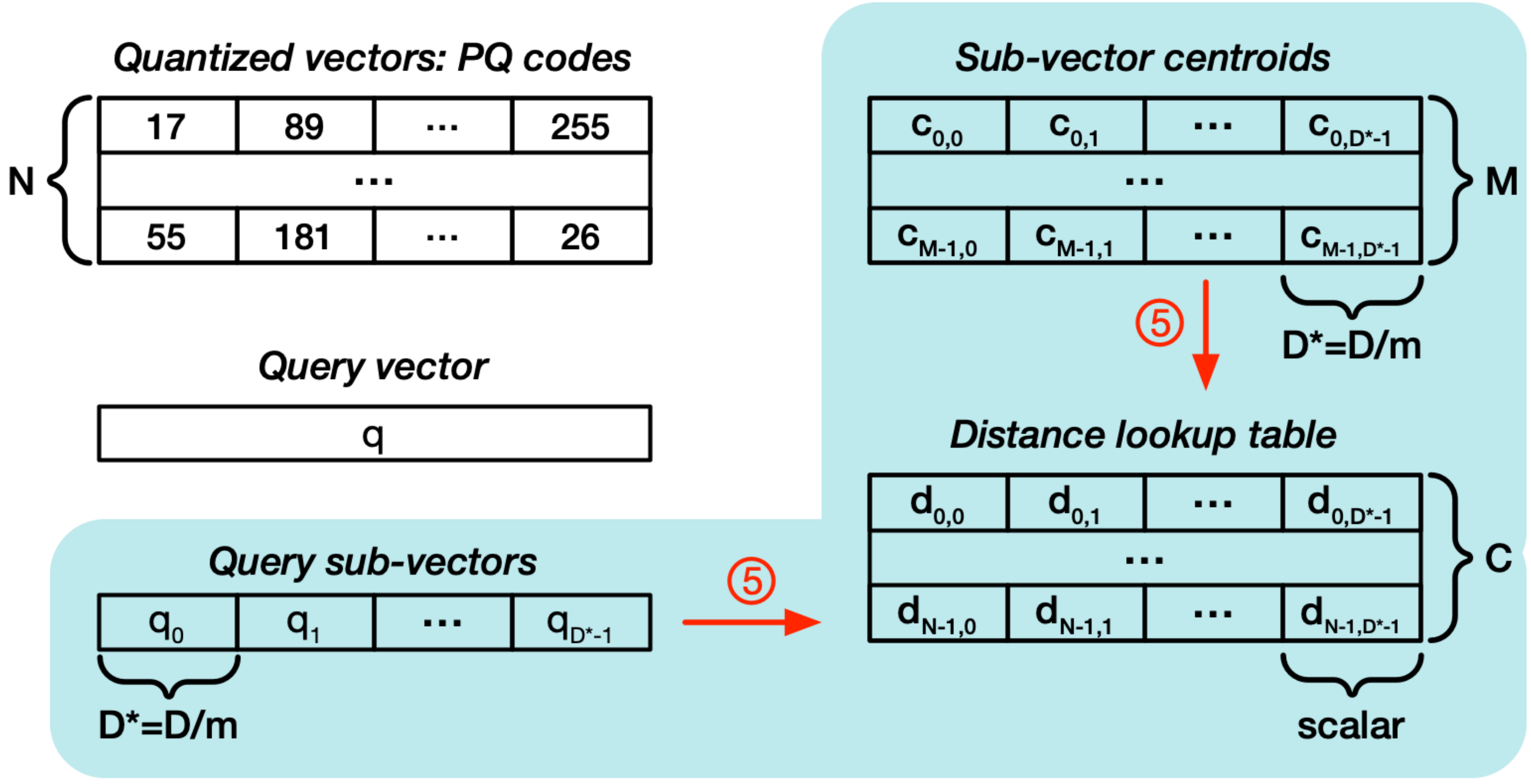
Quantize the vectors to a few bytes of PQ-codes

Product quantization (PQ): searching



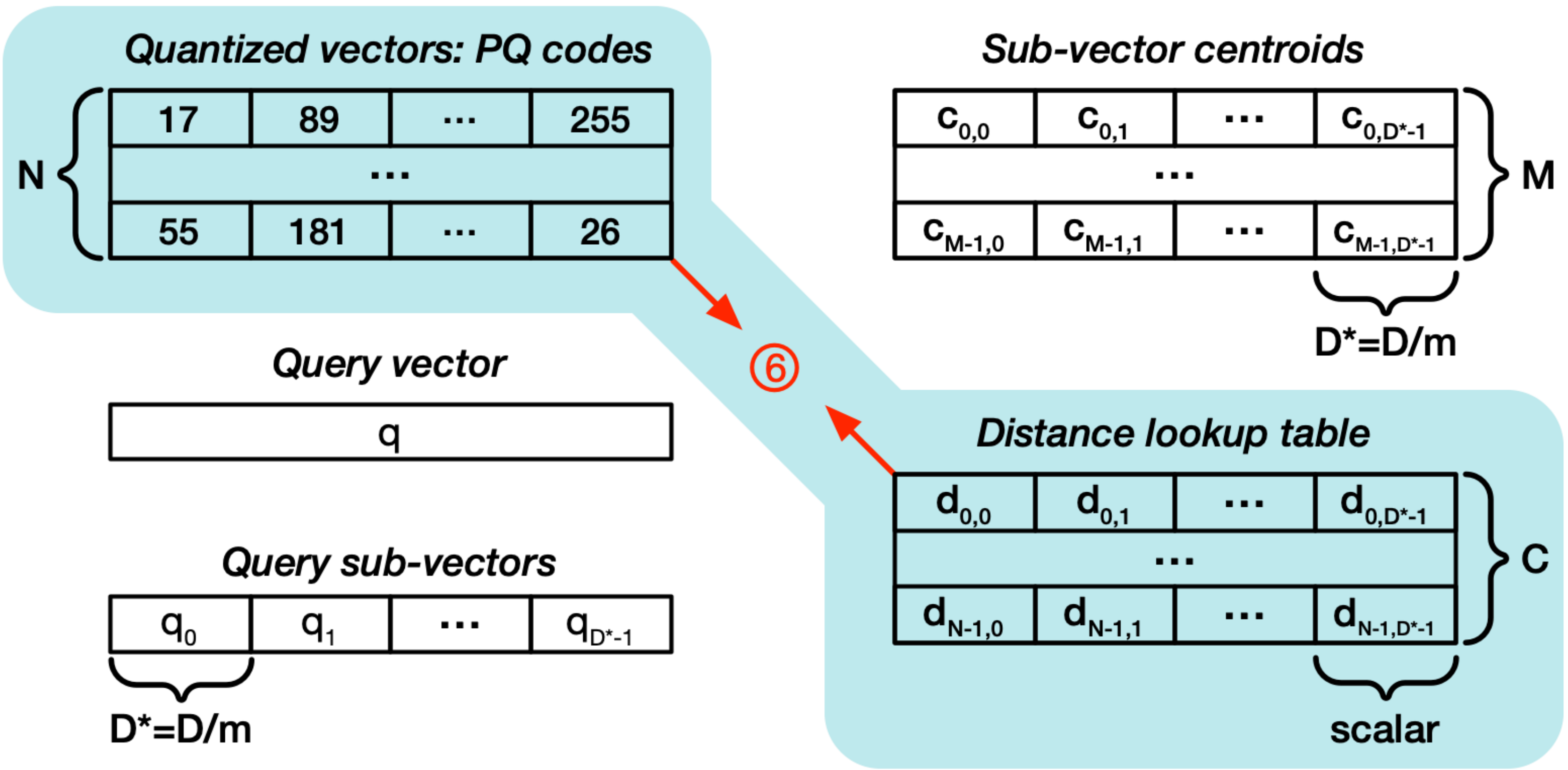
Construct distance lookup tables based on PQ-codes

Product quantization (PQ): searching



Construct distance lookup tables based on PQ-codes

Product quantization (PQ): searching



Construct distance lookup tables based on PQ-codes

System requirements for efficient RALM inference

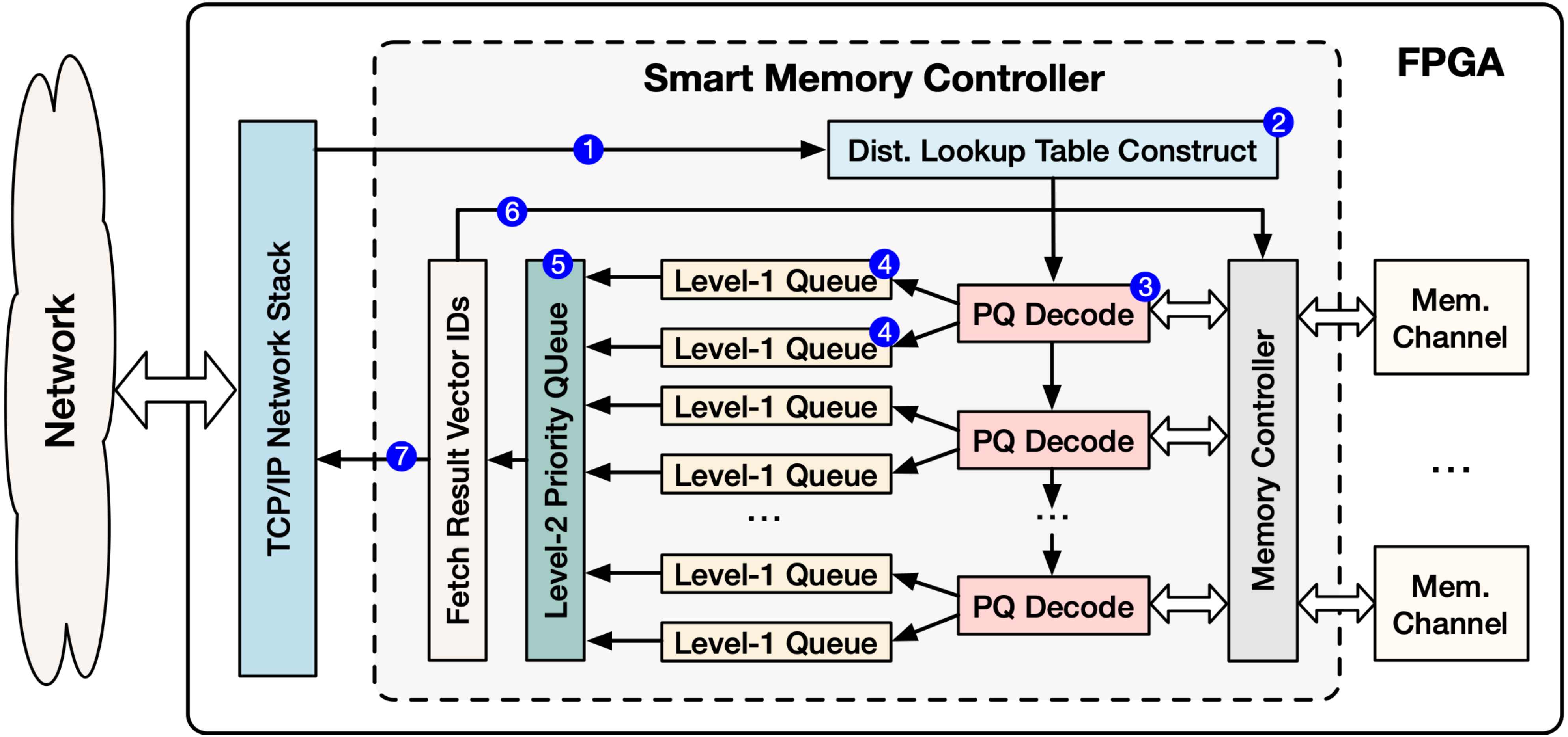
Both LLM inference and vector search should be fast and efficient

- Amdahl's law: performance gains achieved by accelerating one component are limited by the proportion of execution time of that component
- So far, many work has been focused on LLM acceleration

Flexibility to accommodate diverse RALM configurations

- Model architectures: decoder-only, encoder-decoder
- Retrieval intervals: once per token generation step ~ only once per sequence
- Various model and database sizes

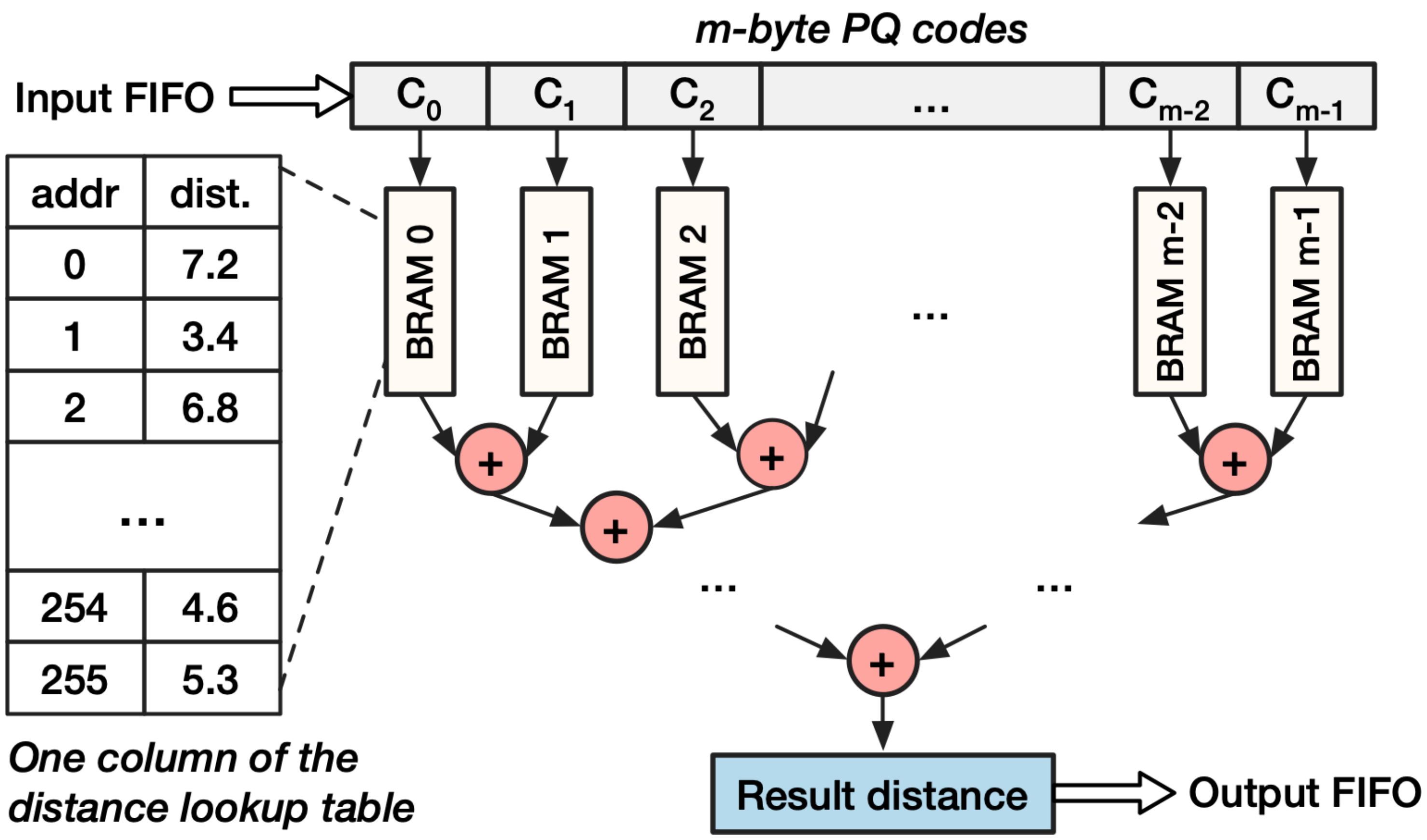
Approximate hierarchical priority queue



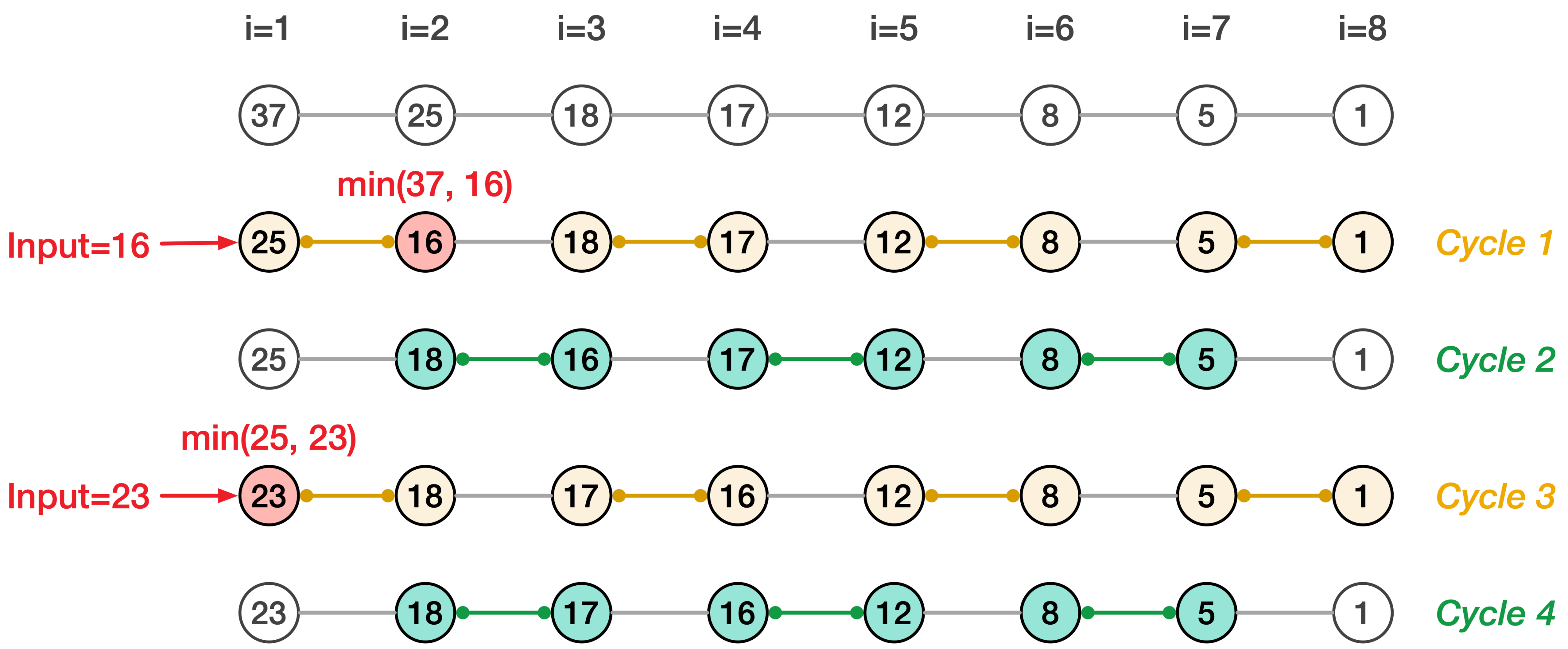
Key design insights of the near-memory accelerator

1. PQ decoding units can rapidly process quantized database vectors
2. The approximate hierarchical priority queue architecture offers high throughput while being resource-efficient
3. Operate on the physical address space to avoid virtualization overhead
4. Search workloads balanced across different channels and nodes

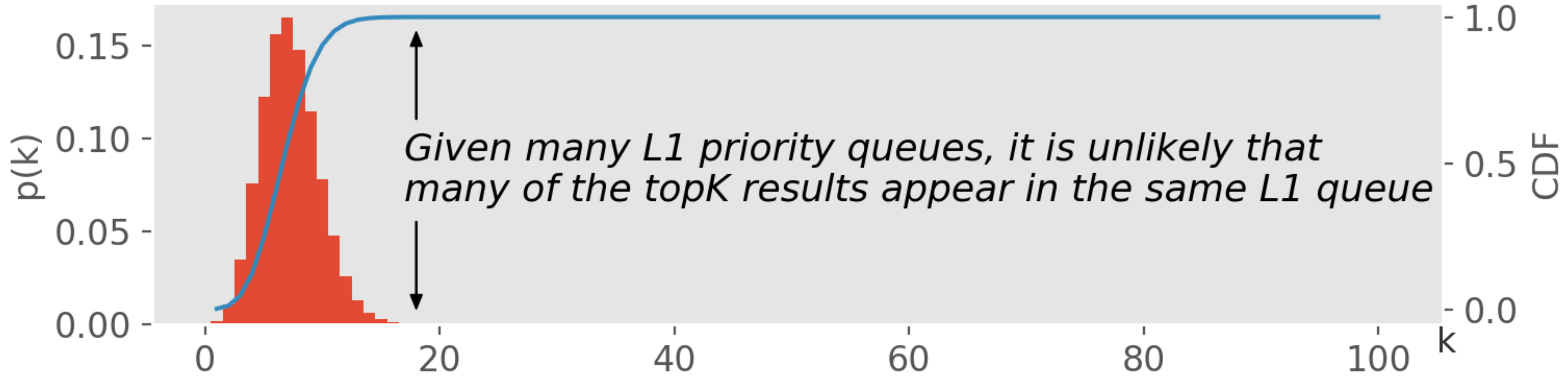
Compute distance to PQ codes

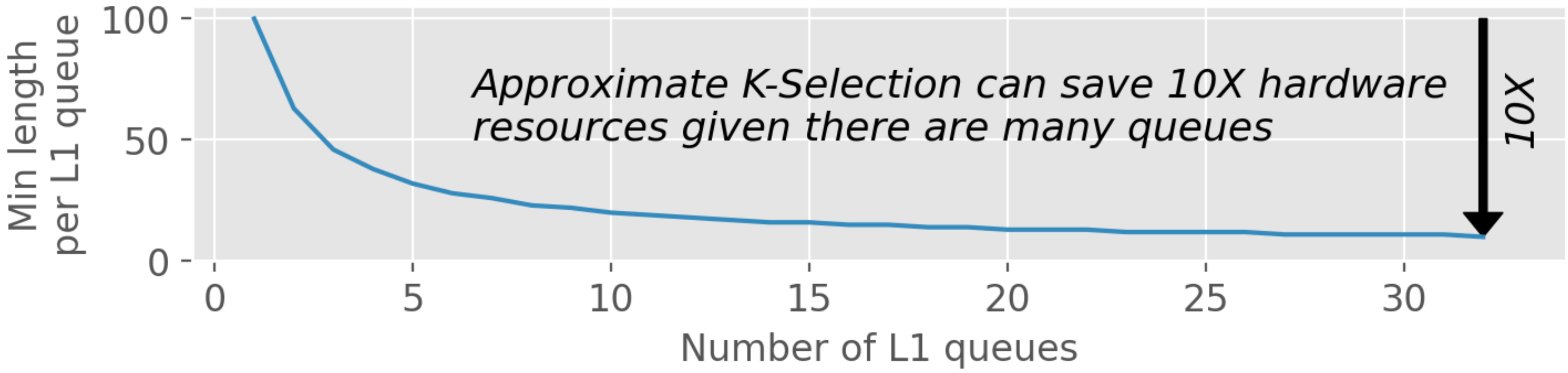


Systolic Priority Queue



Approximate hierarchical priority queue





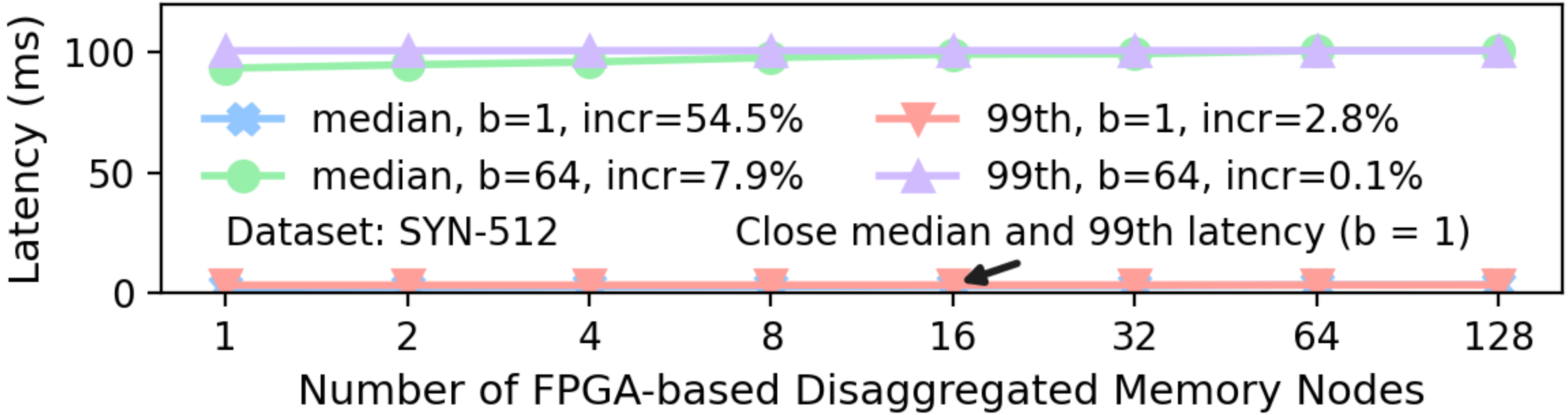
Evaluation settings

Vector search hardware combinations:

- CPU only
- GPU (IVF index) + CPU (PQ)
- CPU (IVF index) + FPGA (PQ)
- GPU (IVF index) + FPGA (PQ)

ChamVS vector search scalability

Great scalability thanks to the low latency variance per ChamVS disaggregated memory node



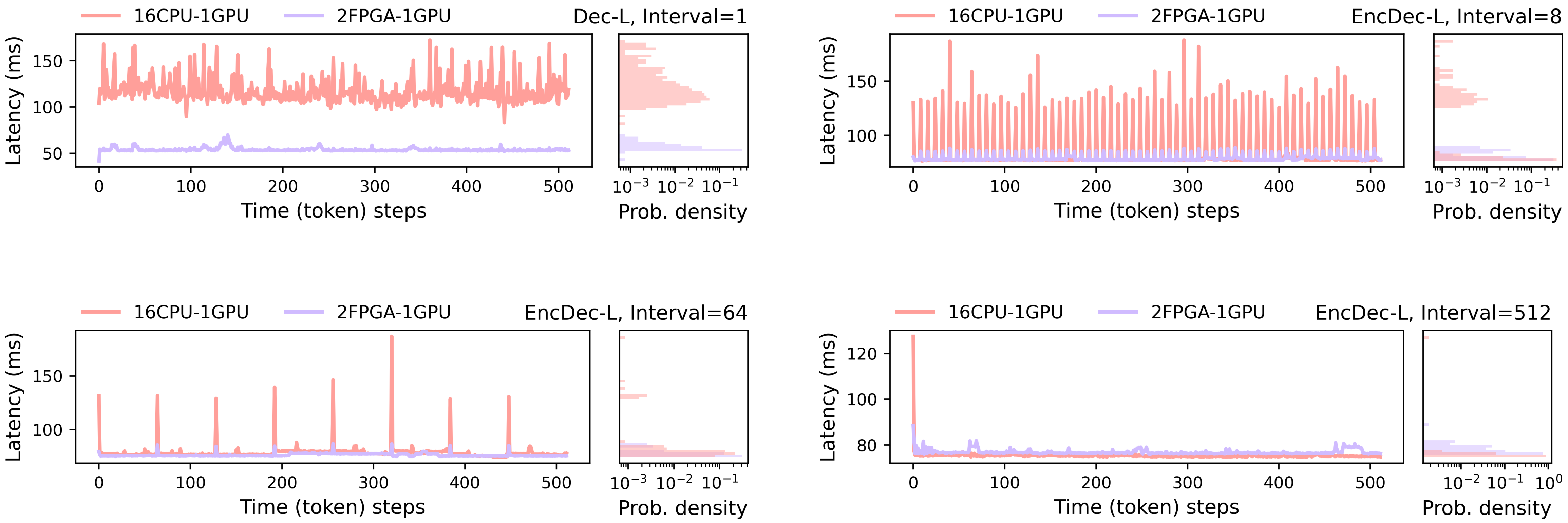
FPGA resource consumptions

The ChamVS near-memory accelerator consumes little FPGA resources on AMD Alveo U250

Can deploy it on FPGAs with more memory channels to further improve performance and cost efficiency

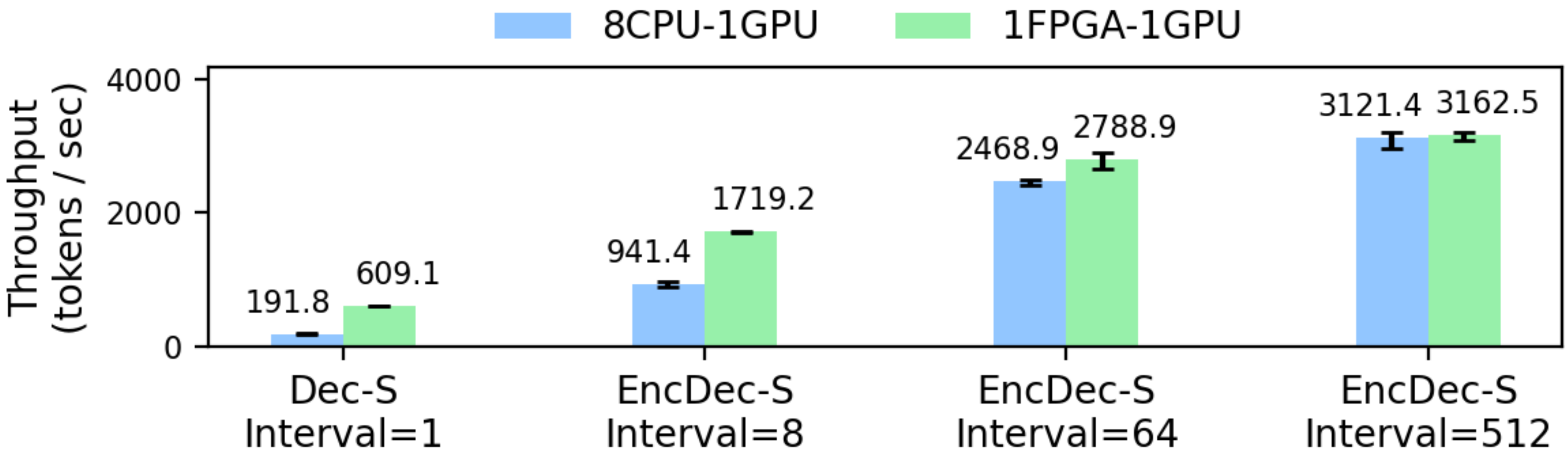
Dataset	LUT	FF	BRAM	URAM	DSP
SIFT	25.3%	16.2%	13.7%	4.4%	12.2%
Deep	23.7%	15.4%	13.0%	4.4%	10.4%
SYN-512	23.2%	15.5%	23.2%	4.4%	8.4%
SYN-1024	28.0%	19.0%	35.7%	4.4%	11.9%

RALM Latency - Large models



RALM Throughput - Large models

Up to 3.18x speedup



RALM Throughput - Large models

Up to 2.34x speedup