Accelerating String-key Learned Index Structures via Memoization-based Incremental Training

Minsu Kim
Jinwoo Hwang
Guseul Heo
Seiyeon Cho
Divya Mahajan†
Jongse Park

KAIST
Georgia Institute of Technology†

KAIST School of Computing

Georgia Tech

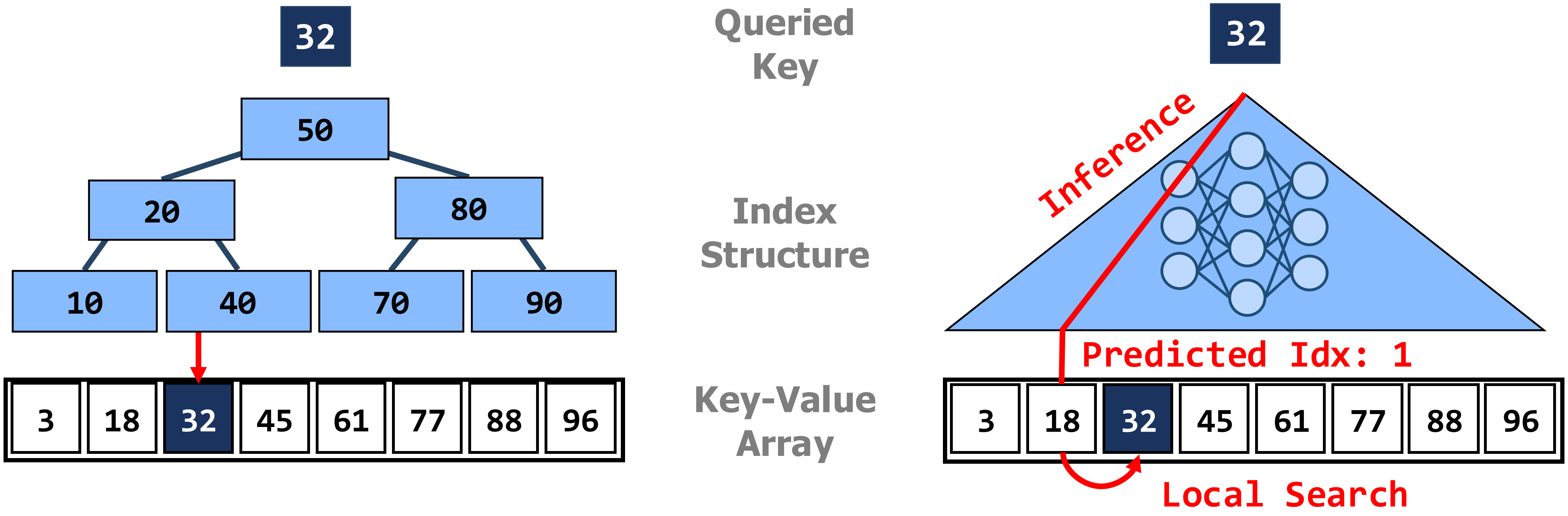
VLDB2024 GUANGZHOU

What is *learned index* anyway?

Learned Index Structure [1]

Traditional Index Structure

Learned Index Structure



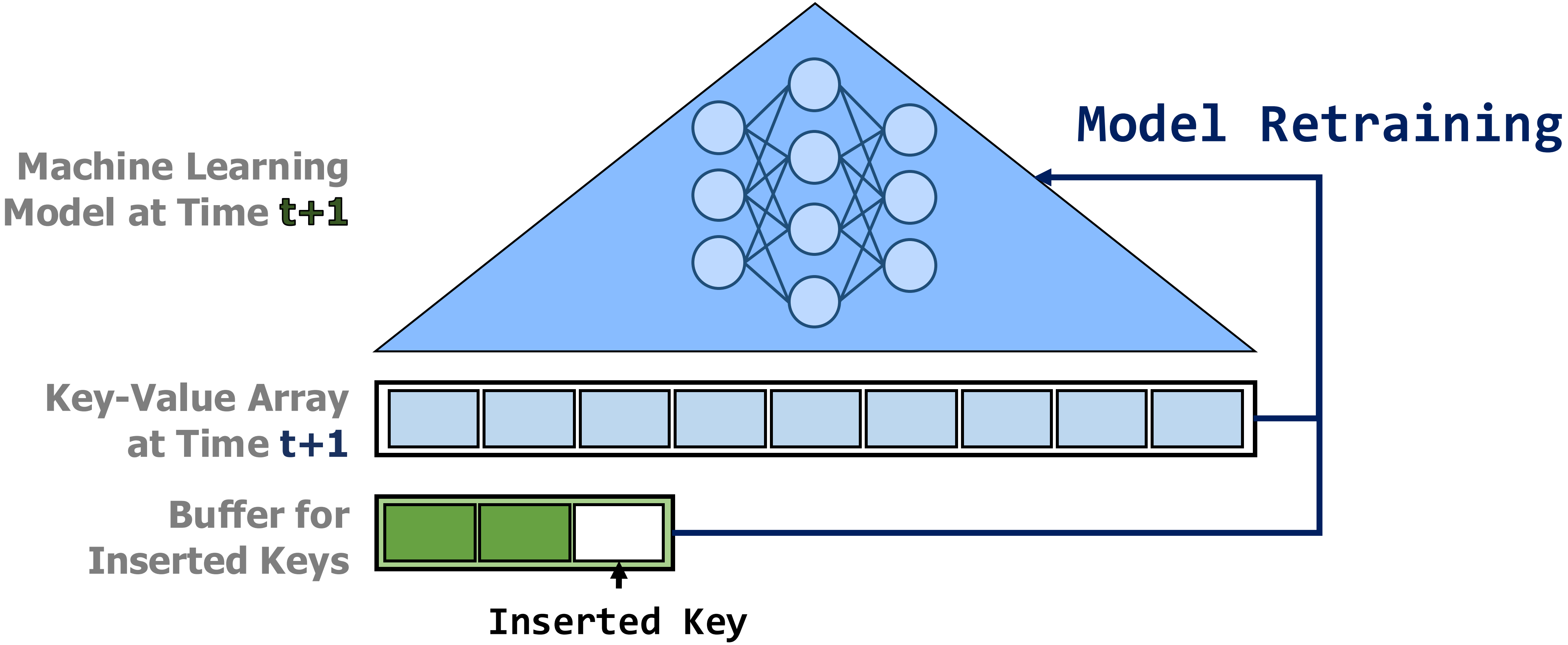
[1] Kraska et al., The Case for Learned Index Structures, SIGMOD 2018

Learned Index Structure

	Traditional Index	Learned Index
Time Complexity	▲	▼
Performance	▼	▲
Index Size	▲	▼

- **Example Applications**
  - **Database**: BOURBON (OSDI 2020)
    Learned Bigtable (ML for Systems at NeurIPS 2020)
  - **DNA Sequencing**: BWA-MEME (Bioinformatics 2022)
    BLESS (ISCA 2024)
  - **Embedded Sensor**: Ding et al. (SENSORNETS 2023)

Updatable Learned Index [2, 3, 4]



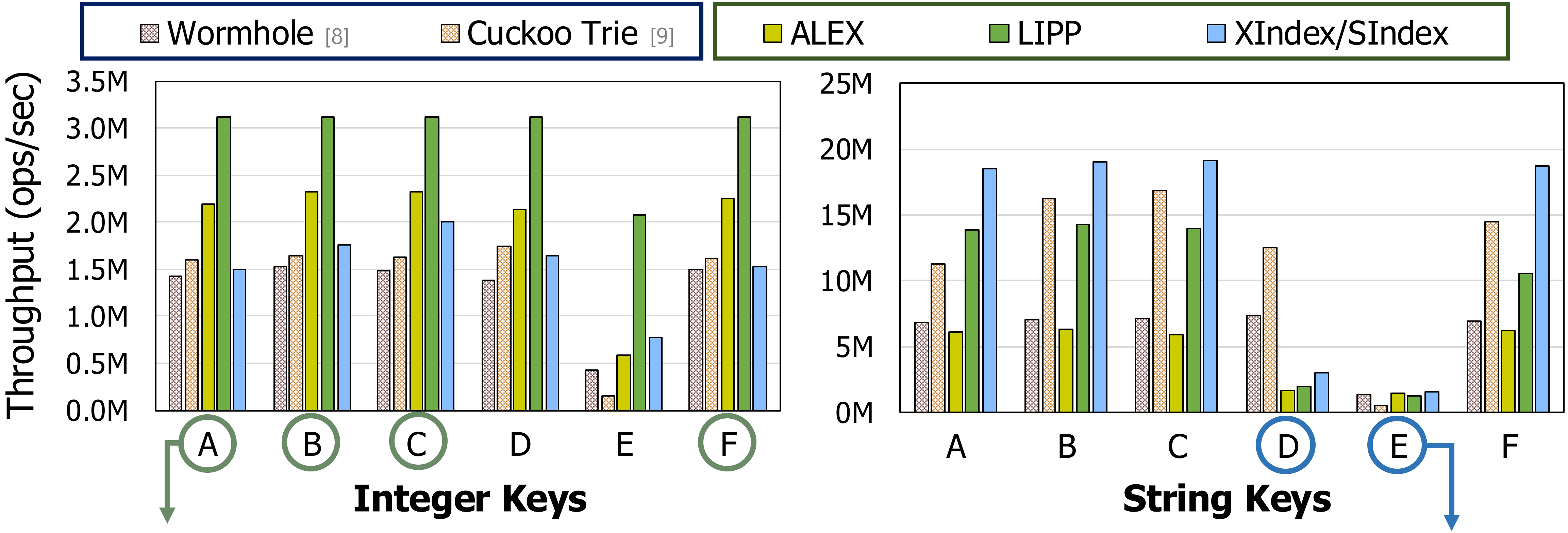
Updatable learned indexes require periodic retraining using the **entire keys**

[2] Ding et al., ALEX: An Updatable Adaptive Learned Index, SIGMOD 2020
[3] Tang et al., XIndex: A Scalable Learned Index for Multicore Data Storage, PPoPP 2020
[4] Wang et al., SIndex: A Scalable Learned Index for String Keys, APSys 2020

Performance of Updatable Indexes

** Used YCSB (Yahoo Cloud Serving Benchmark) workloads*

***Non*-Learned Indexes** | **Learned Indexes**



Read-only Workload

Read-Write Workload

String-key learned indexes show **poor performance** for **read-write workloads**

[8] Wu et al., Wormhole: A Fast Ordered Index for In-memory Data Management, Eurosys 2019
[9] Zeitak et al., Cuckoo Trie: Exploiting Memory-Level Parallelism for Efficient DRAM Indexing, SOSP 2022

Importance of String-Keys

- String-key key-value store assists GenAI of Cisco by indexing 3,600+ documents [5]

  [5] Omar Santos, Cisco Powers Secure, Responsible Artificial Intelligence Innovation at Scale with MongoDB, 2024

- Elasticsearch improves quality of GenAI model by 20% with the aid of string-key key-value store [6]

  [6] LG CNS, Elasticsearch와 AI 검색 모델 통합으로 검색 정확도 고도화하다, 2024

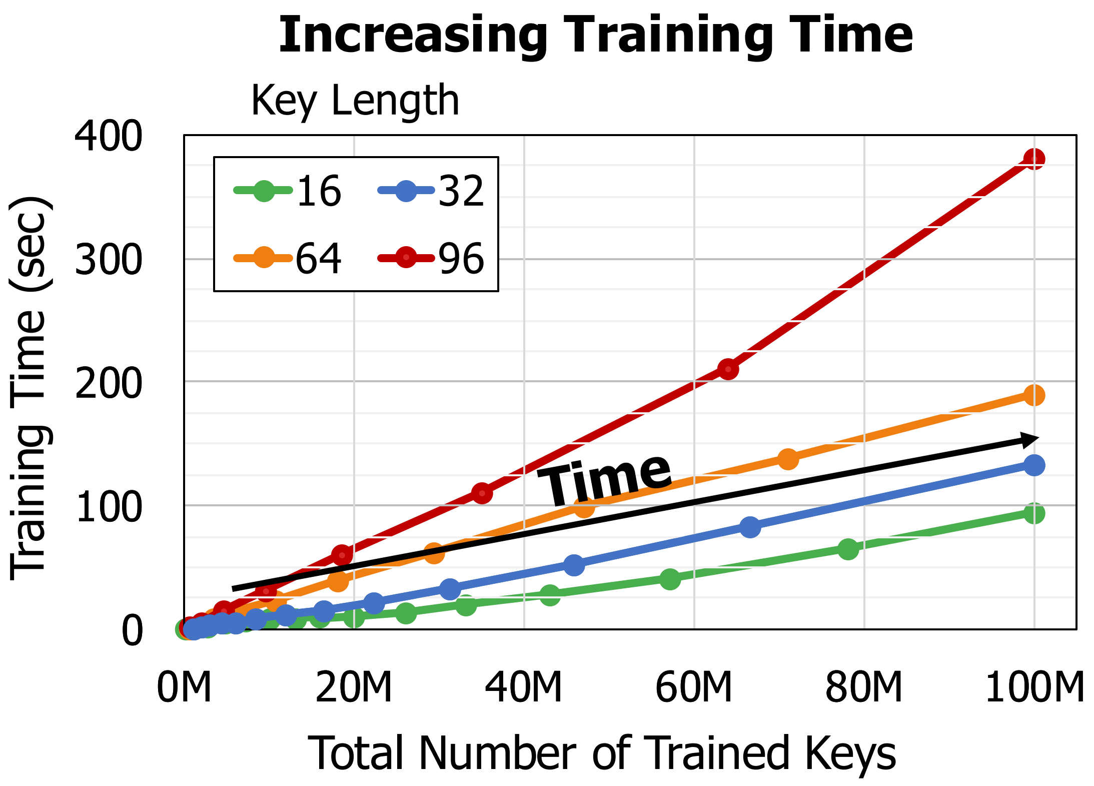
- Customer app should handle 10,000+ personalized requests per second for over 100 million items with under 40ms delay [7]

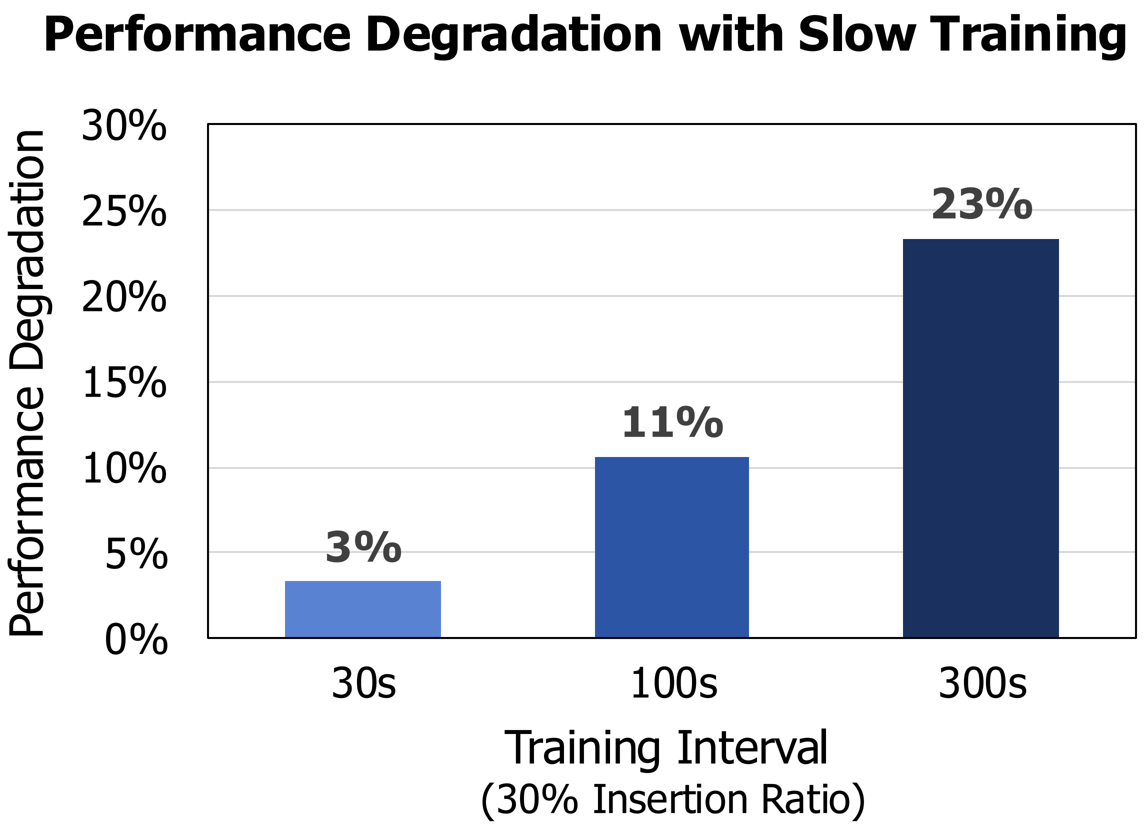
  [7] Delivery Hero Helps Customers Navigate more than 100 Million Products with MongoDB Atlas Search, 2024

Bottlenecks of Learned Index Training

1. Bad scalability & performance due to accumulated keys

Accumulated keys **degrade the performance** of learned index
by delaying updates of ML model





Bottlenecks of Learned Index Training

2. QR Decomposition Operations are Expensive

- Most learned indexes use **linear regression** for their ML model
- Solving linear regression involves **QR decomposition**

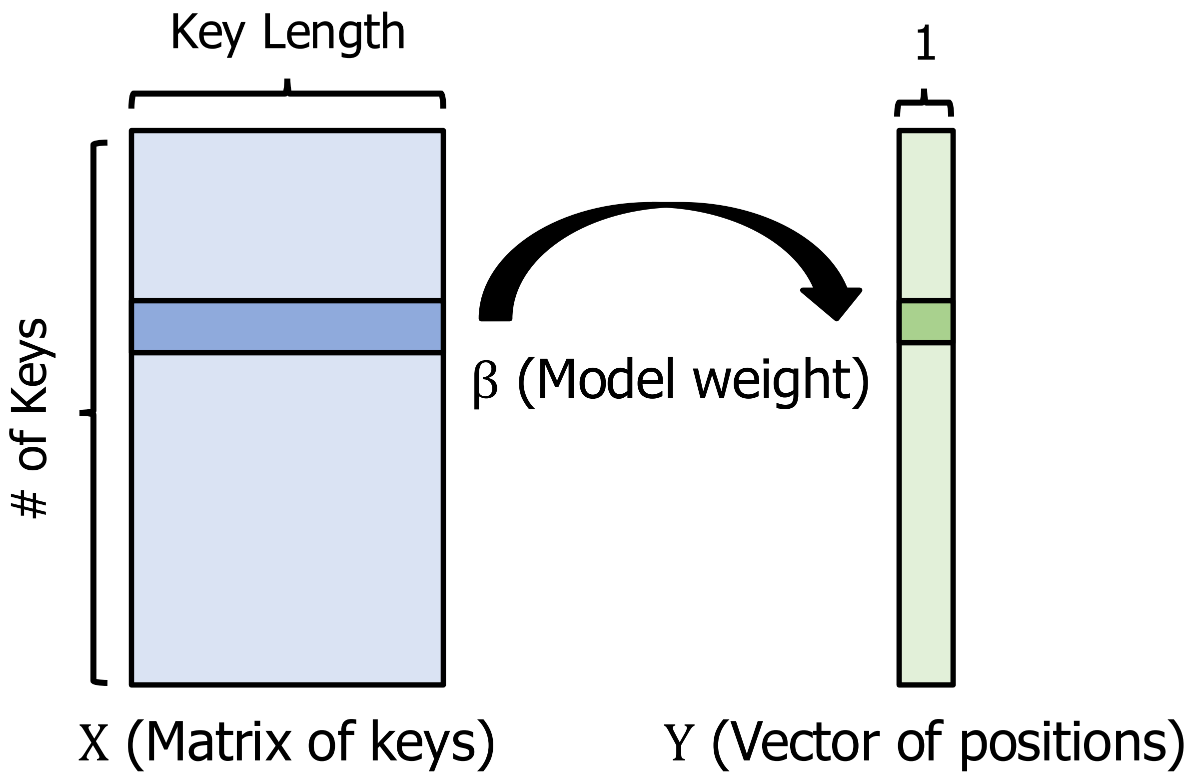
Linear Regression Model

$$X\beta = Y$$

Linear Regression Solution

$$\beta = \left(\mathbf{R}^{-1}\mathbf{R}^{-1^{T}}\right)X^{T}Y$$

, where $\mathbf{X} = \mathbf{QR}$



Bottlenecks of Learned Index Training

2. QR Decomposition Operations are Expensive

- **QR decomposition** is the major bottleneck when training
- **R Inverse and GEMM** are the second longest

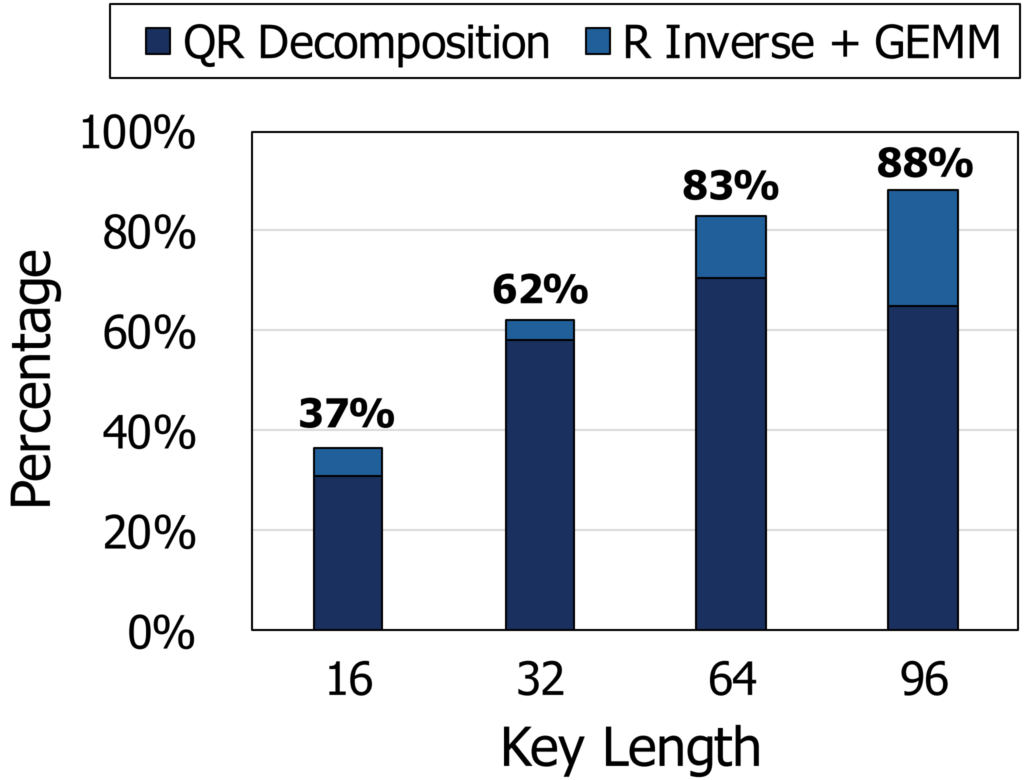
Linear Regression Model

$$X\beta = Y$$

Linear Regression Solution

$$\beta = \left(\mathbf{R}^{-1}\mathbf{R}^{-1^{T}}\right)X^{T}Y$$

, where $\mathbf{X} = \mathbf{QR}$



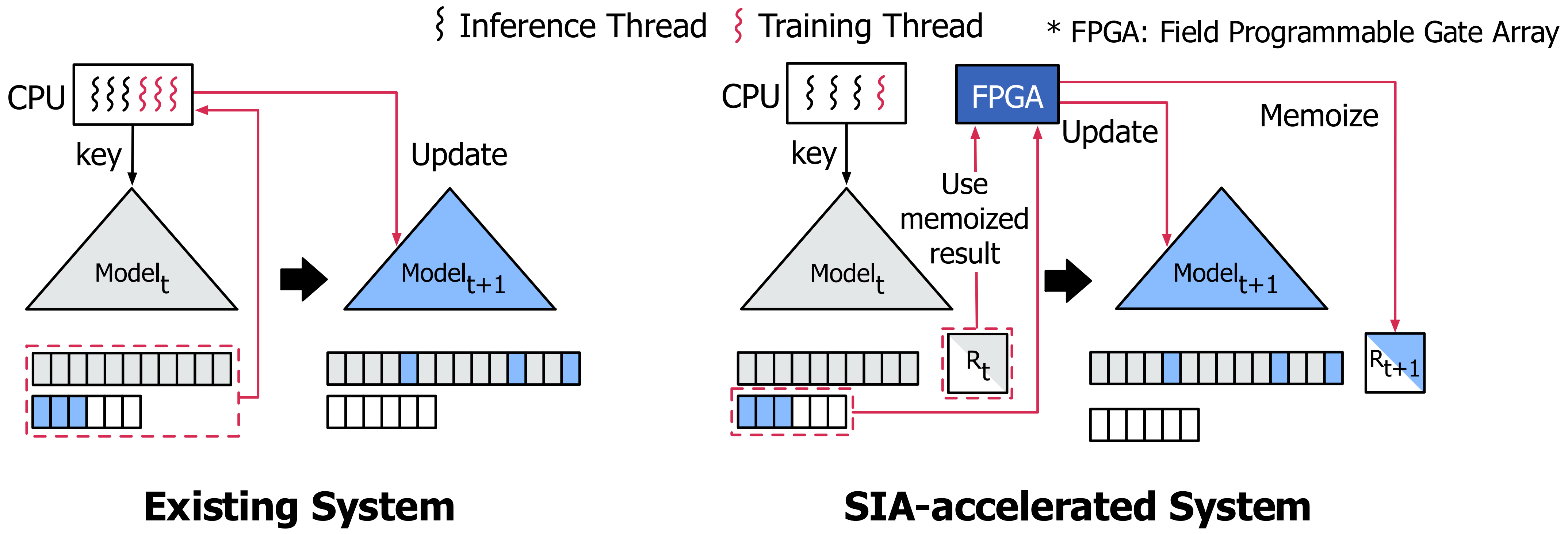
Existing String-key Learned Index Systems Offer Limited Performance

SIA: System Overview

Algorithm-Hardware Co-designed String-key Learned Index System

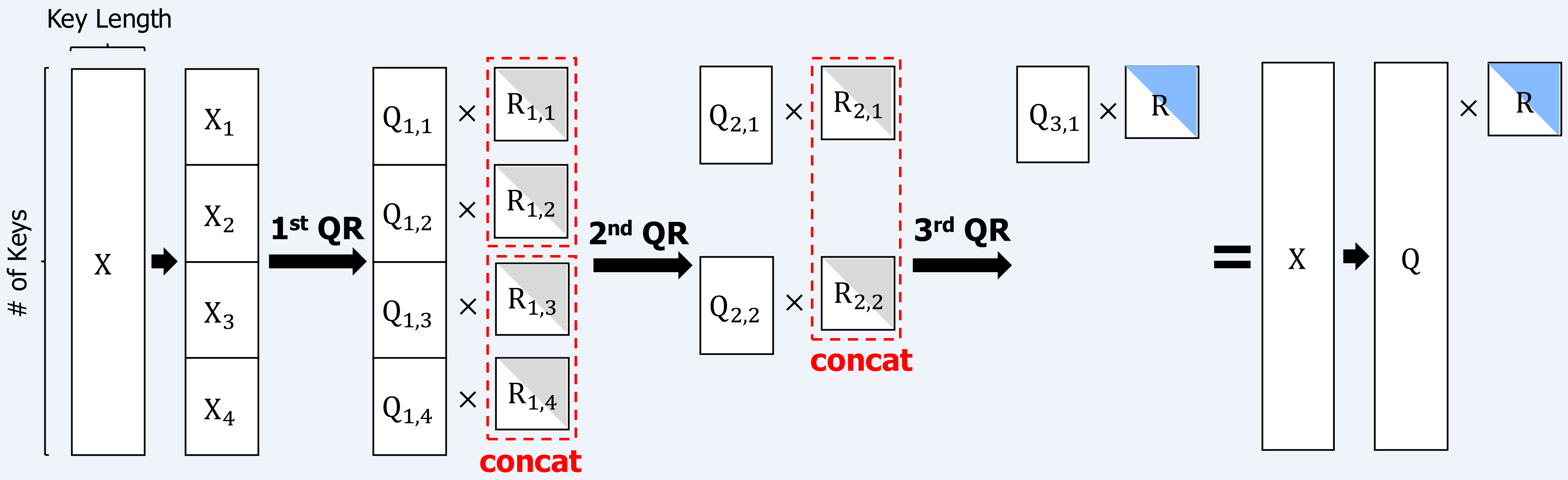
① **Algorithm** that reuses memoized intermediate results

② **Hardware** that offloads index training with FPGA accelerator



Insight from Parallel QR Decomposition

- Existing parallel QRD offers advantage to **tall-and-skinny** matrices
- Parallel QRD ensures **mathematical equivalence**



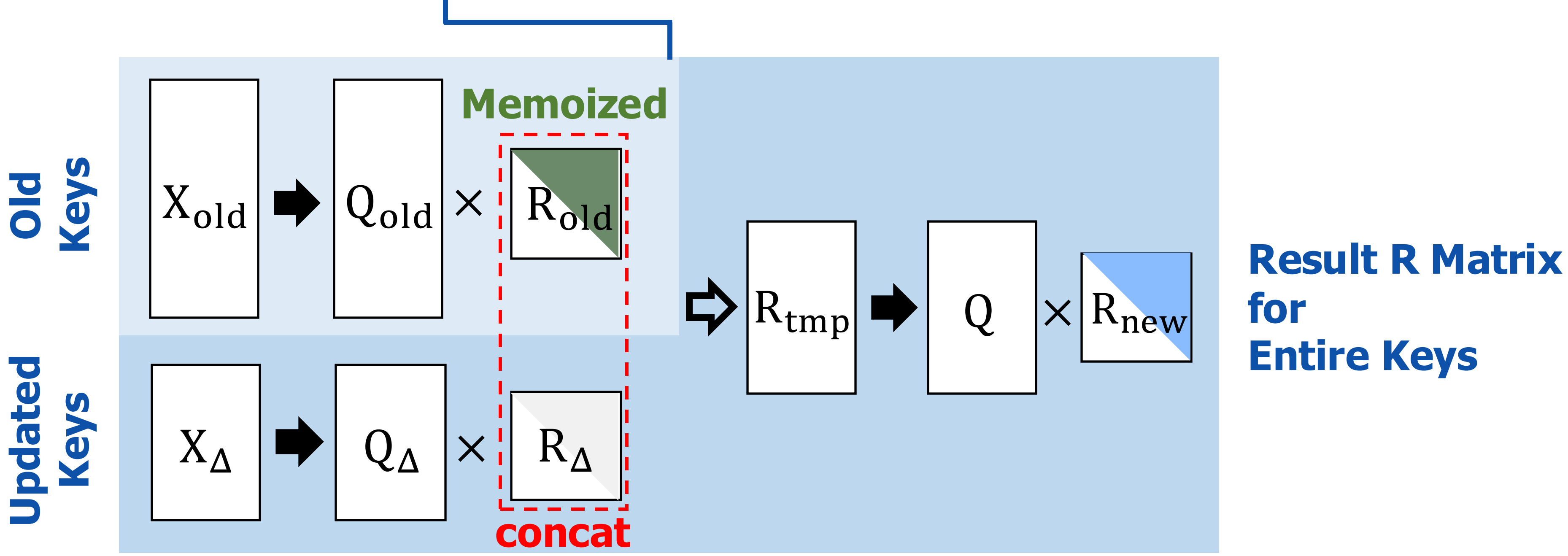
Algorithm Design

Incremental Index Learning

- Incremental index learning **reduces costly QRD** via memoization

Linear Regression Solution

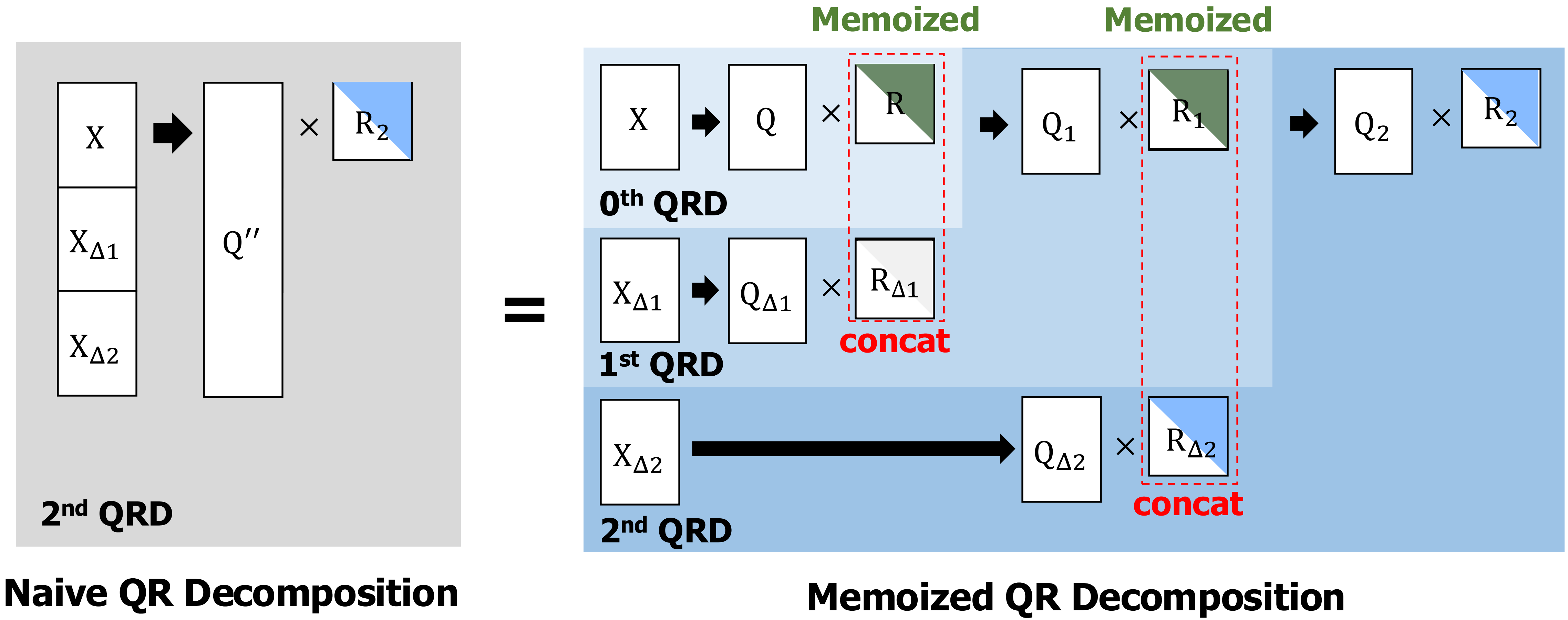
$$\beta = (\mathbf{R}^{-1}\mathbf{R}^{-1^{T}})X^{T}Y \text{ where } \mathbf{X} = \mathbf{QR}$$



Algorithm Design

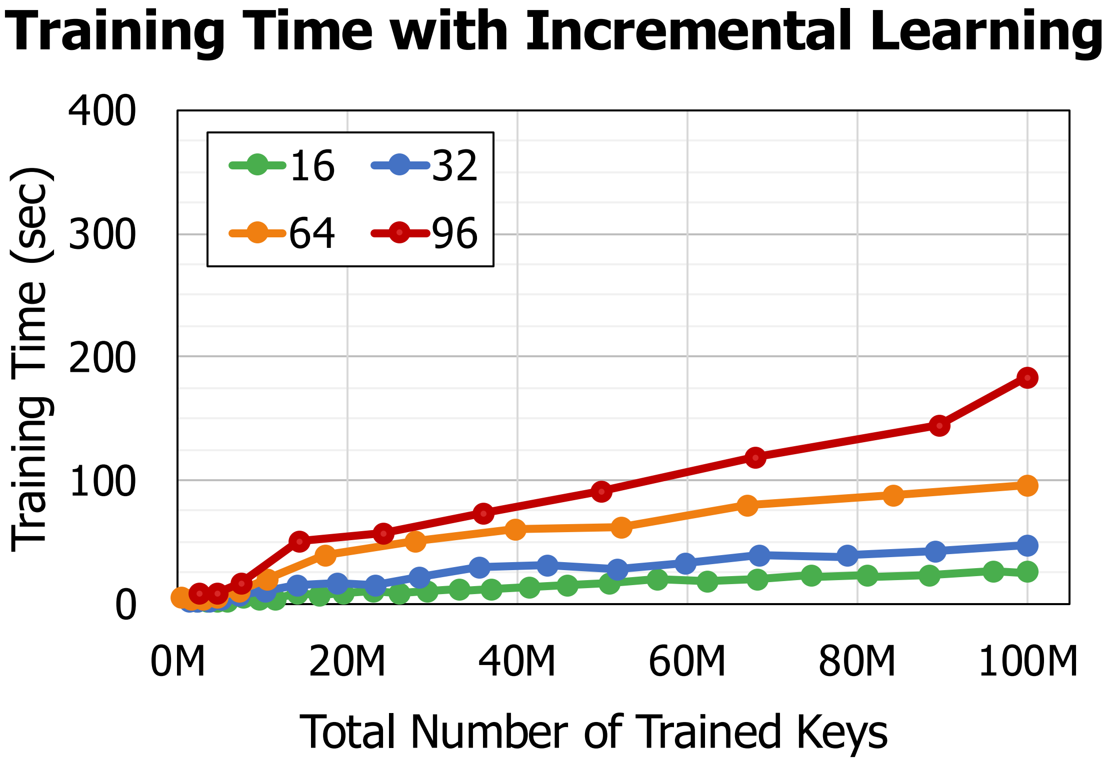
Incremental Index Learning

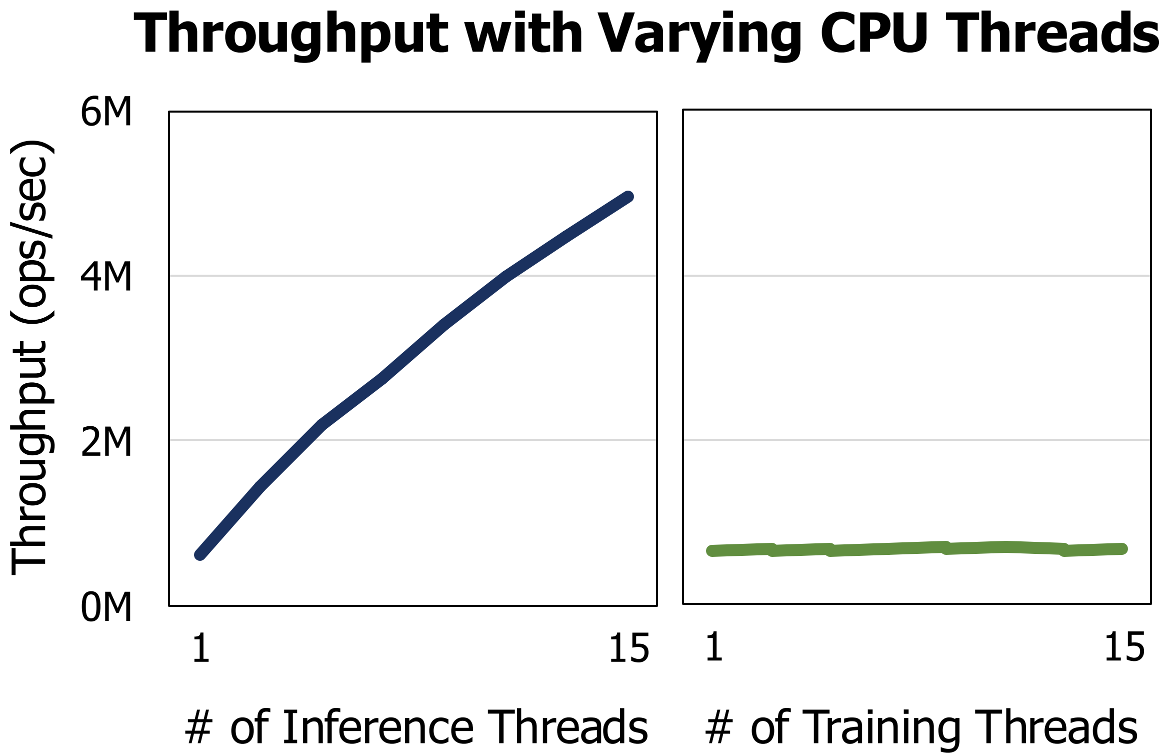
- There is no need to perform QRD for entire key matrix



Why Do We Need Hardware Acceleration?

CPU-only solution is still slow due to **low efficiency in training**





Hardware Design

Hardware Selection: FPGA

- **Reconfigurable**
  Reprogrammable without changing hardware
- **Customizable**
  Programmable with user custom hardware logic
- **Parallelizable**
  Simultaneous operation of multiple logic blocks
- **Area & Energy Efficient**
  High performance at low operating cost

Field Programmable Gate Array

Hardware Design

FPGA Accelerator Architecture

Linear Regression Solution

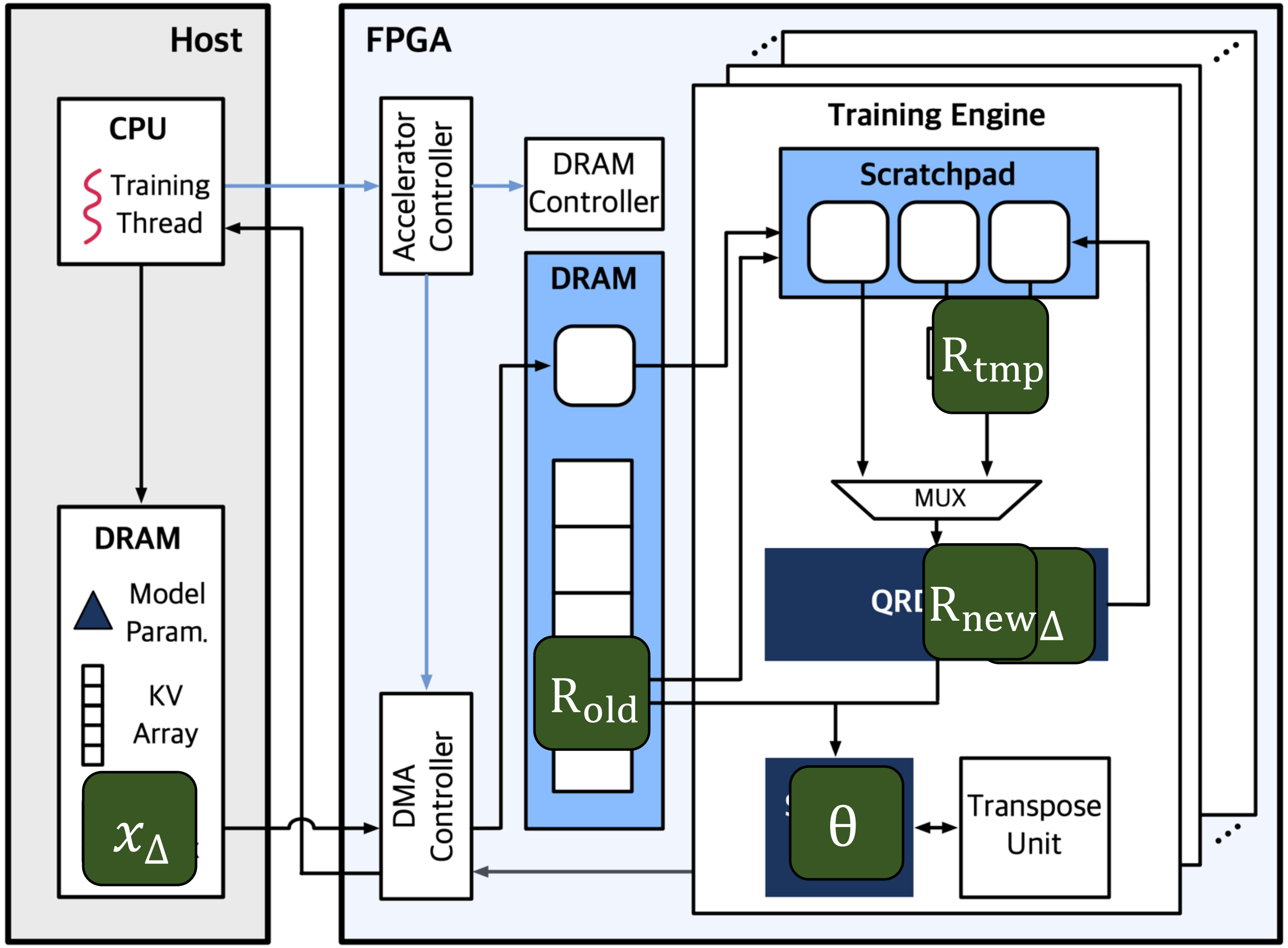
$\beta = (\mathbf{R}^{-1}\mathbf{R}^{-1^T})X^TY$ where $\mathbf{X} = \mathbf{QR}$

FPGA accelerator calculates

$$\boldsymbol{\theta} = \left(\mathbf{R}^{-1}\mathbf{R}^{-1^T}\right)$$

with incremental index learning

Calculation result is returned to host CPU



Evaluation Methodology

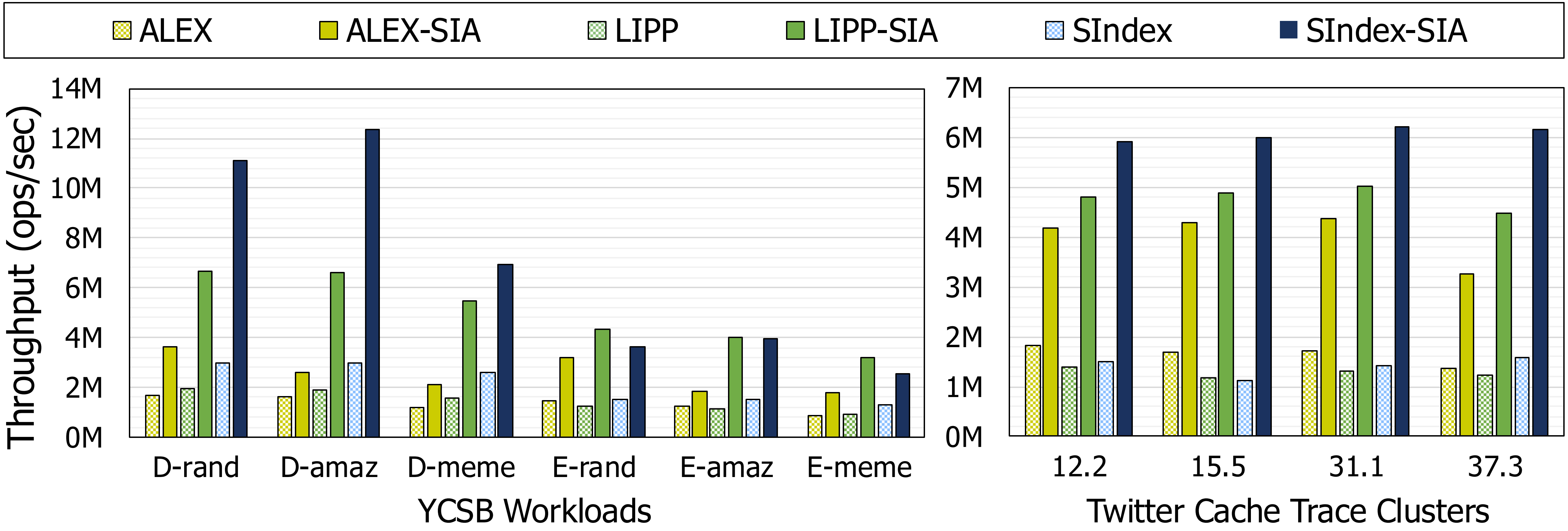
- **Baselines**
  - Wormhole[*]
  - Cuckoo Trie [*]
  - SIndex [**]
  - ALEX [**]
  - LIPP [**]

[*] Non-learned indexes
[**] Updatable learned indexes

- **FPGA**
  - Intel Arria 10 GX-1150 (Synthesized to 272MHz)

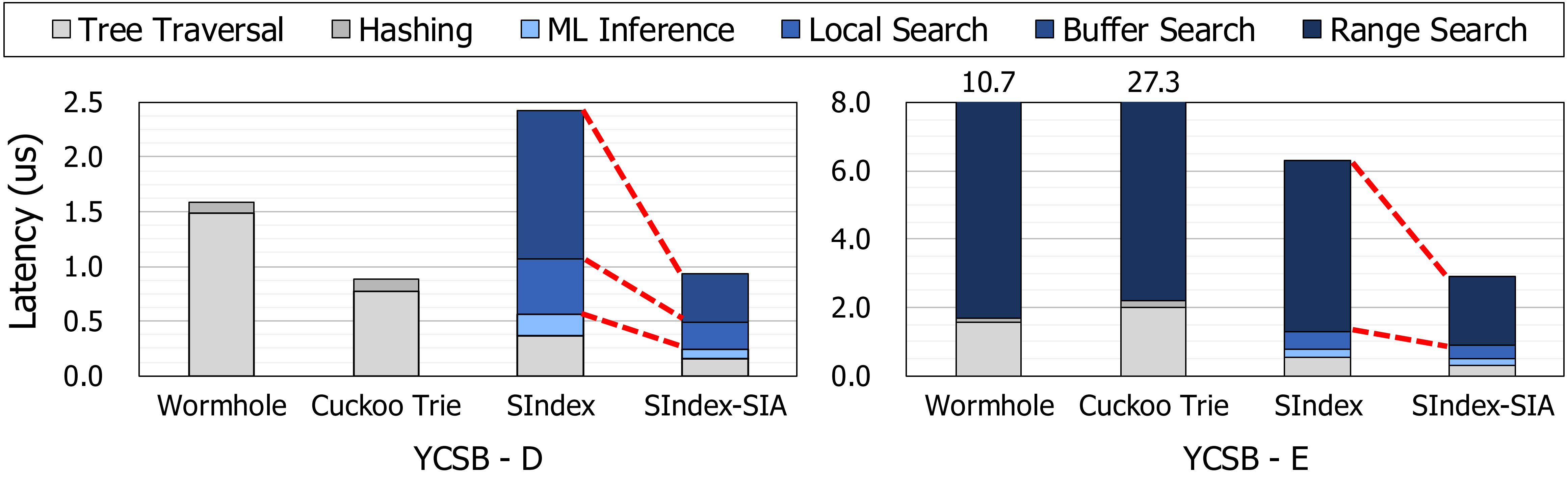
<table>
<tr><th>Dataset</th><th colspan="2">Workload</th></tr>
<tr><td>"amaz"
Amazon review dataset</td><td rowspan="3">YCSB – D
Read & Insert queries</td><td rowspan="3">YCSB – E
Range & Insert queries</td></tr>
<tr><td>"meme"
Memetracker dataset</td></tr>
<tr><td>"rand"
Randomly generated strings</td></tr>
<tr><td>Twitter Cache Trace
12.2, 15.5, 31.1, 37.3</td><td colspan="2">Twitter Cache Trace
12.2, 15.5, 31.1, 37.3
Read & Insert Queries</td></tr>
</table>

Performance Evaluation



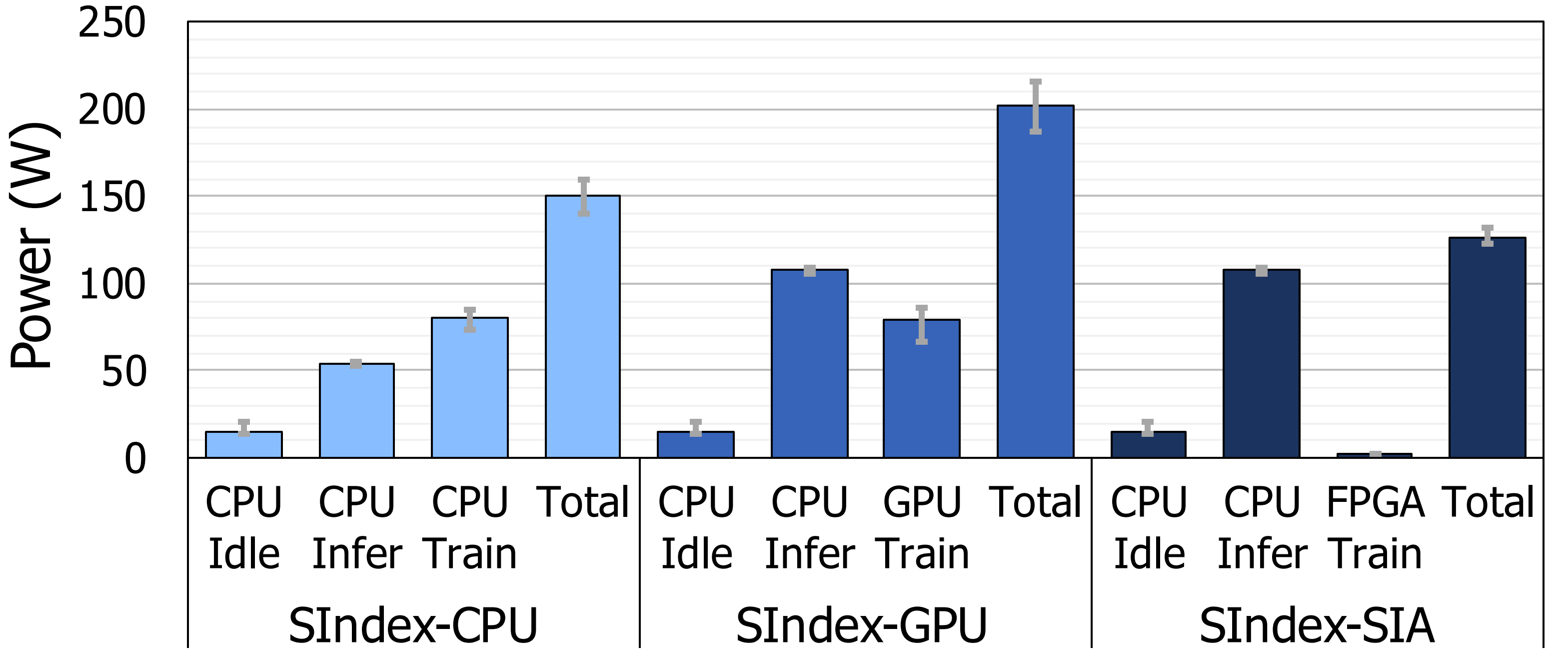
Learned indexes with SIA shows an average of **2.9x throughput improvement** compared to learned indexes without SIA

Latency Breakdown



Learned Index with SIA benefits from **reduced search time**
due to "freshness" of learning model

Energy Efficiency Evaluation



	Normalized Performance per Watt
SIndex-CPU	1.00x
SIndex-GPU	1.67x
SIndex-SIA	**2.89x**

* CPU: Intel Xeon Gold 6226R
* GPU: NVIDIA RTX 2080 TI

SIA achieves higher energy efficiency with **low energy usage of FPGA**

(**28x** less than NVIDIA RTX 2080 TI GPU)

Suitable for continuous retraining of **learned index system**

More Results in Paper

- **Hardware Resource Utilization**
- **Memory Consumption Comparison**
- **Ablation Study**
- **Throughput with Different Query Distribution**
- **Implication of Lazy Delete Query Handling**

Conclusion

- **SIA**
  - Fast and efficient
    string-key learned index
    through CPU-FPGA heterogeneous acceleration

- **Results**
  - **2.9x** performance boost on existing string-key learned indexes

- **Work-in-Progress and Future Directions**
  - Exploration of use cases for learned indexes in domain-specific applications
    - Vector DB for LLM (i.e., RAG)
    - Databases for ML training data