LITS: An Optimized Learned Index for Strings

Yifan Yang, Shimin Chen
SKLP, ACS, Institute of Computing Technology, CAS
University of Chinese Academy of Sciences

Overview

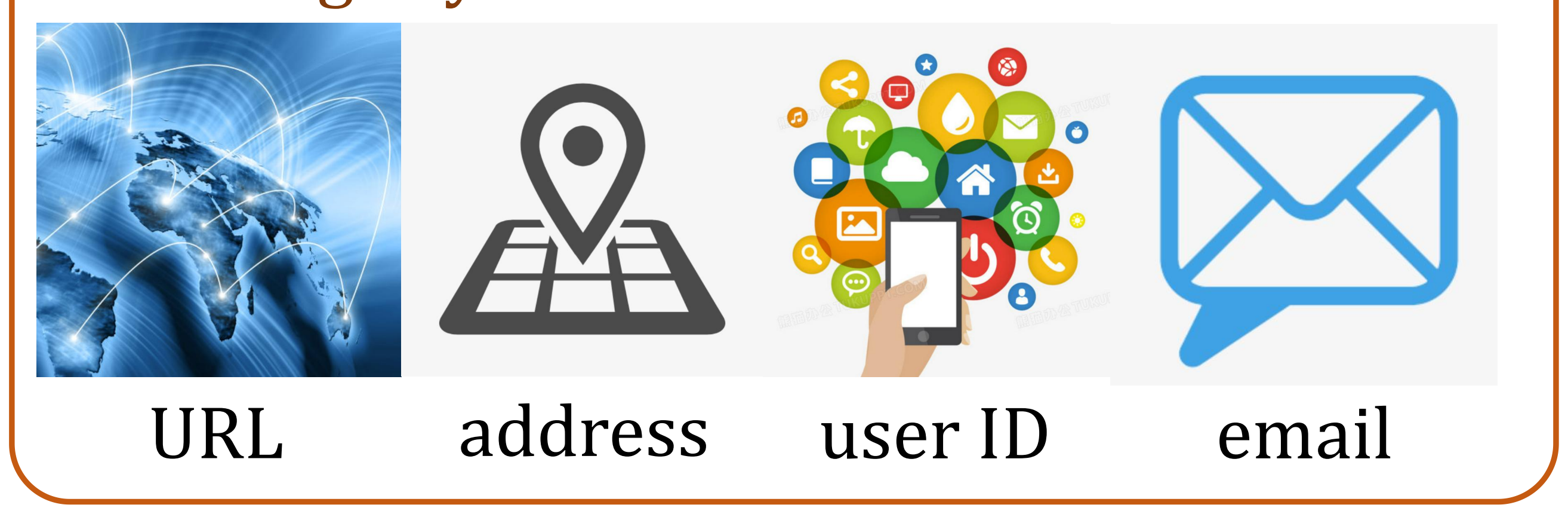
Background

- existing learned indexes fail to outperform traditional indexes when indexing **string keys**

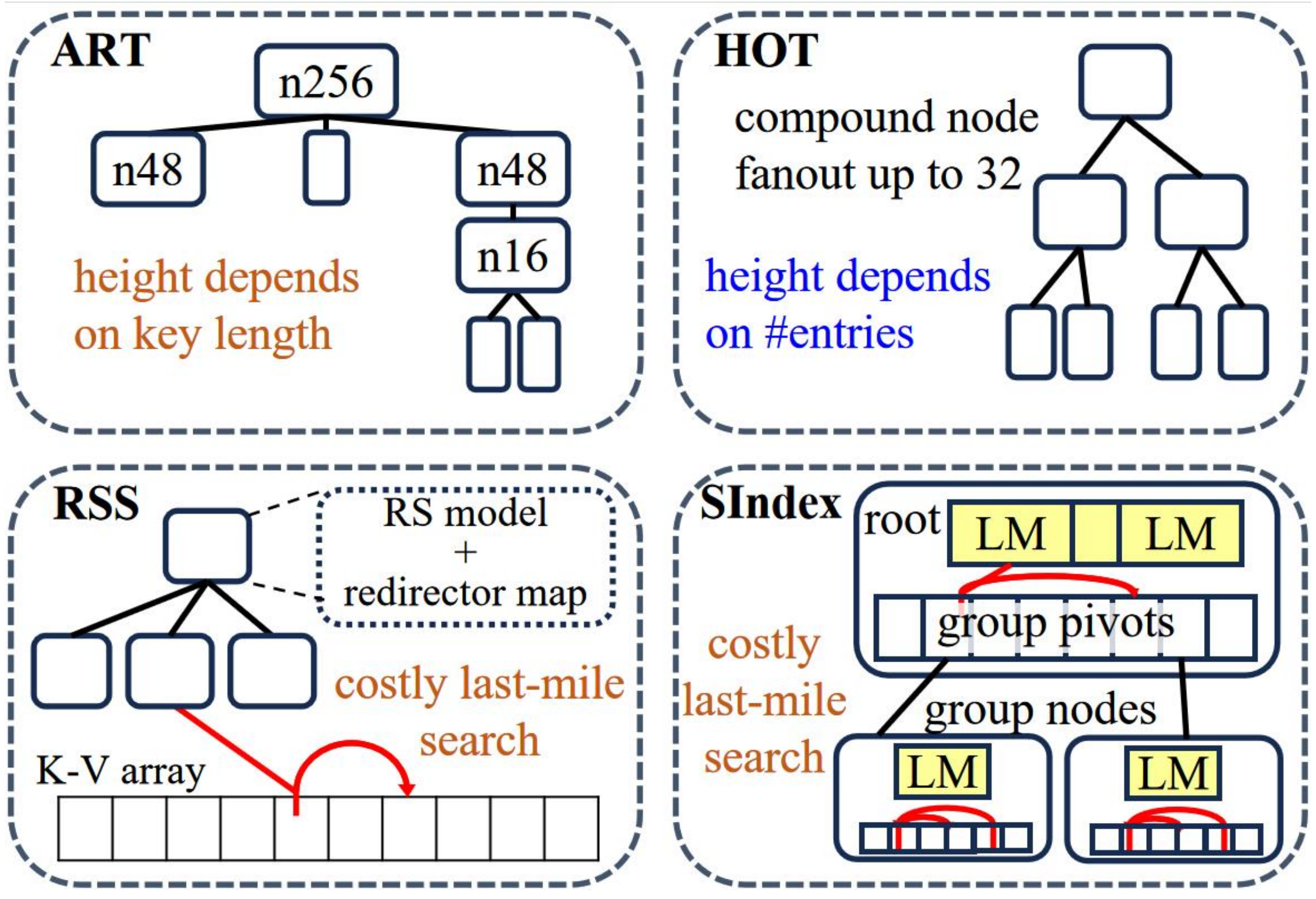
Characteristics of String keys

- long, variable-sized, with skewed prefixes

string keys are common in real-world



Existing Indexes Optimized for Strings

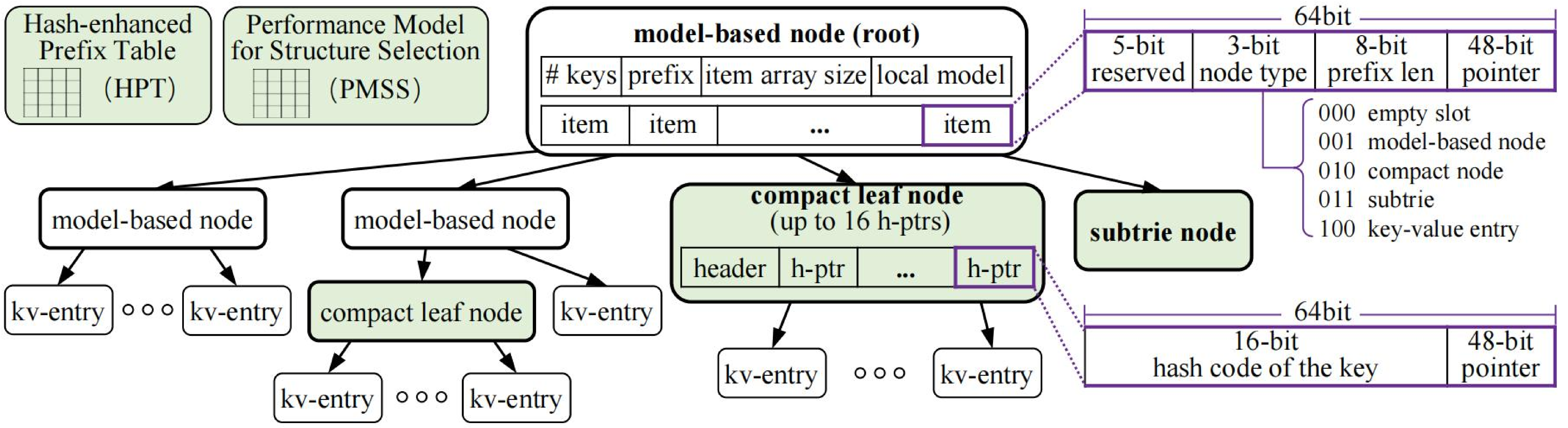


	tree height	node search	last-mile search
ART	×	√	n/a
HOT	--	√	n/a
RSS	√	√	×
SIndex	√	√	×
LITS	√	√	n/a

our solution: LITS

Learned Index with Hash-enhanced Prefix Table and Sub-tries

LITS Structure



64bit

5-bit reserved	3-bit node type	8-bit prefix len	48-bit pointer

000 empty slot
001 model-based node
010 compact node
011 subtrie
100 key-value entry

64bit

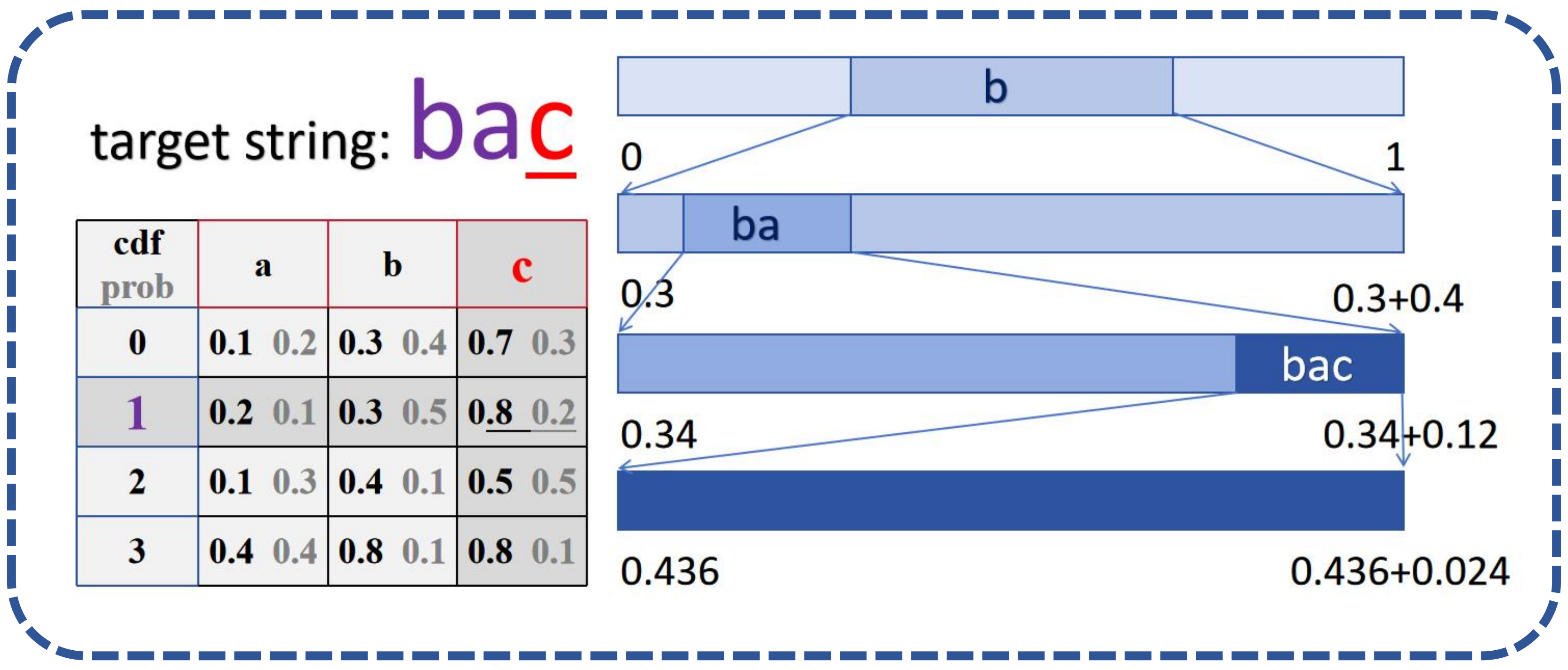
16-bit hash code of the key	48-bit pointer

Hash-enhanced Prefix Table

CDF Model in LITS: Hash-enhanced Prefix Table (HPT)

Idea of HPT: learn the pattern of a string data set by better approximating $prob(c|P)$

($prob(c|P)$: conditional probability of the next character being c given the prefix P)



cdf prob	a	b	c
0	0.1 0.2	0.3 0.4	0.7 0.3
1	0.2 0.1	0.3 0.5	0.8 0.2
2	0.1 0.3	0.4 0.1	0.5 0.5
3	0.4 0.4	0.8 0.1	0.8 0.1

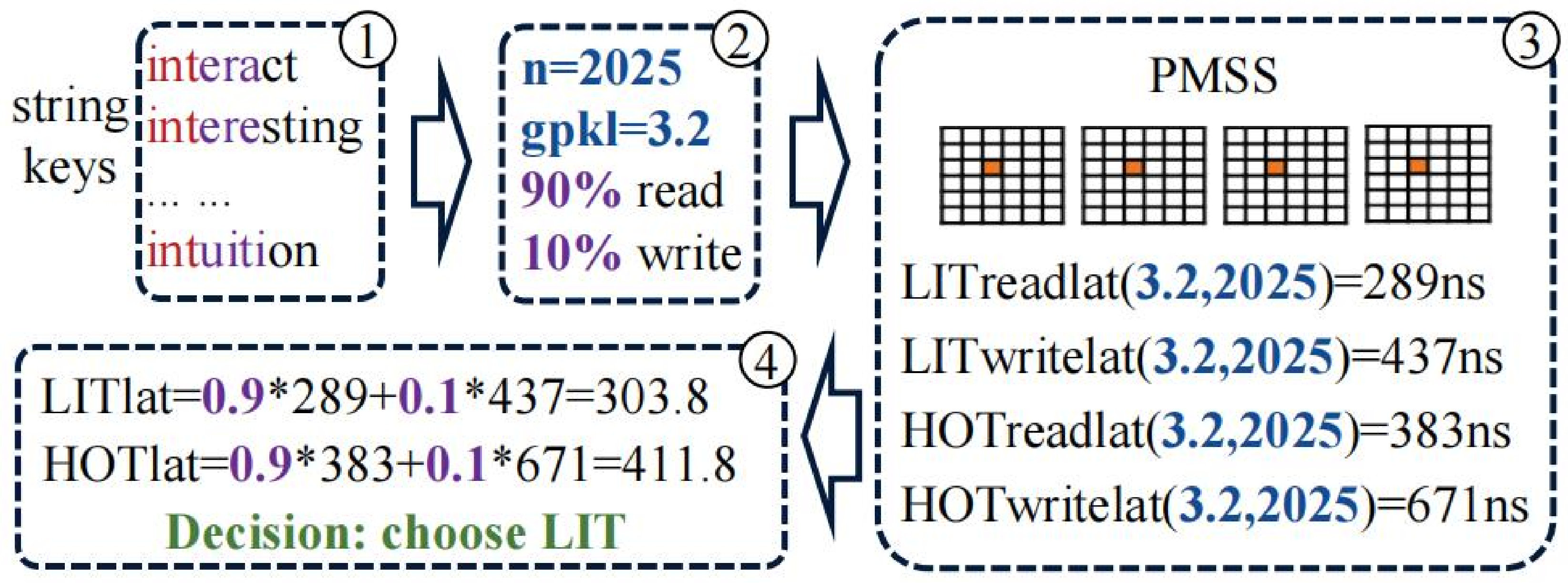
Fig. An illustration of the CDF computation using the HPT for string "bac".

Performance Model for Structure Selection

Hardness of String Data Set: Group Partial Key Length (GPKL)

- GPKL reflects the hardness of modeling a string data set
- GPKL can be computed efficiently by reading the strings in one pass

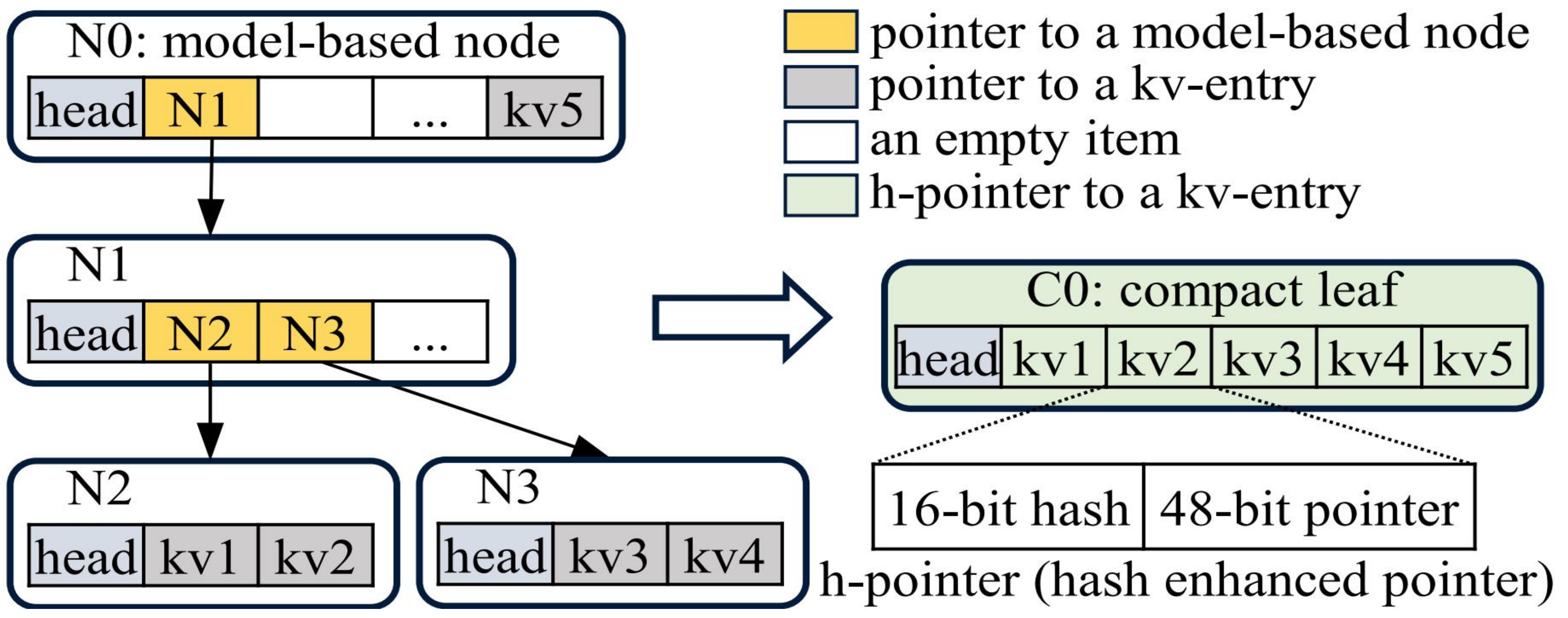
Idea of PMSS: select the optimal structure for a sub-trie based on the hardness of a subset and offline benchmark tests



LITreadlat(3.2,2025)=289ns
LITwritelat(3.2,2025)=437ns
HOTreadlat(3.2,2025)=383ns
HOTwritelat(3.2,2025)=671ns

LITlat=0.9*289+0.1*437=303.8
HOTlat=0.9*383+0.1*671=411.8

Compact Leaf Node

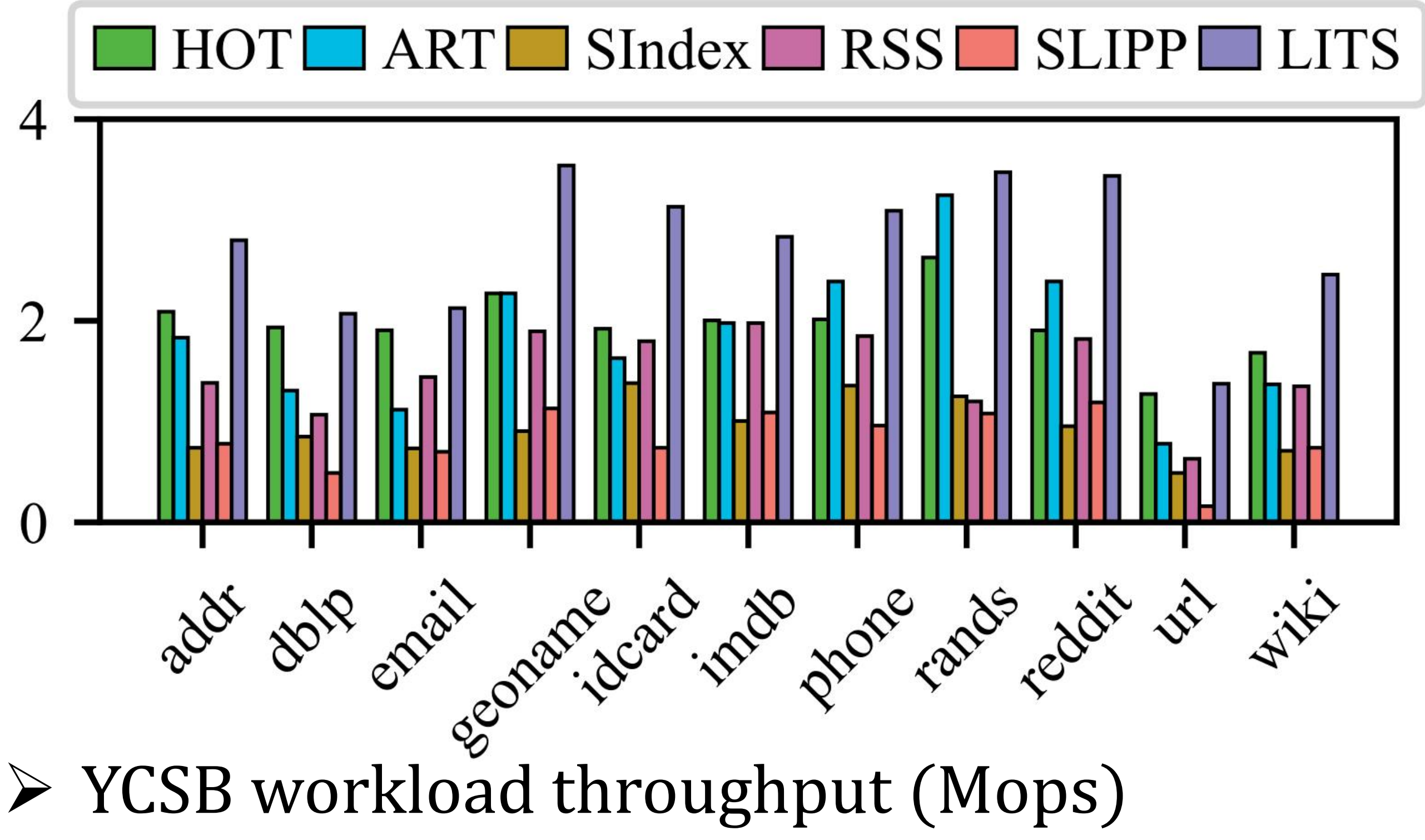


🟨 pointer to a model-based node
▨ pointer to a kv-entry
□ an empty item
🟩 h-pointer to a kv-entry

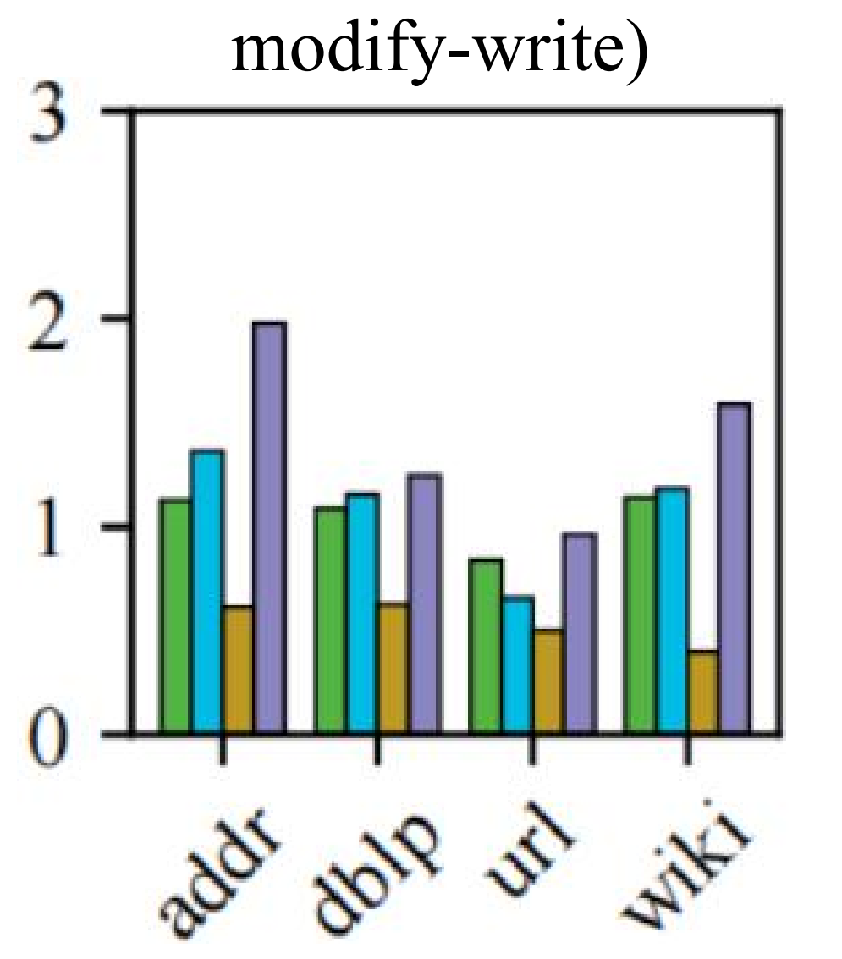
Evaluation

□ 7 real-world string data sets and 4 synthetic string data sets

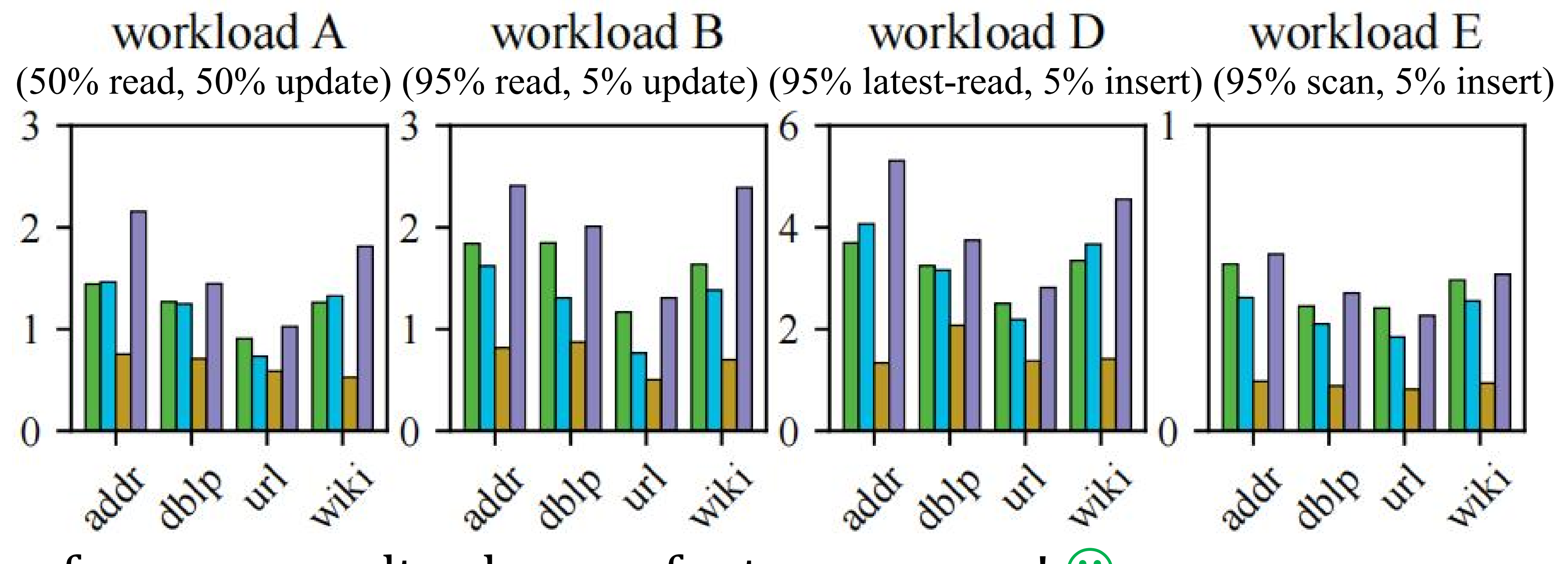
➢ read-only workload throughput (Mops)



workload F
(50% read, 50% read-modify-write)



➢ YCSB workload throughput (Mops)



for more results please refer to our paper! ☺

中国科学院大学 University of Chinese Academy of Sciences
ICT 中国科学院计算技术研究所 INSTITUTE OF COMPUTING TECHNOLOGY, CHINESE ACADEMY OF SCIENCES