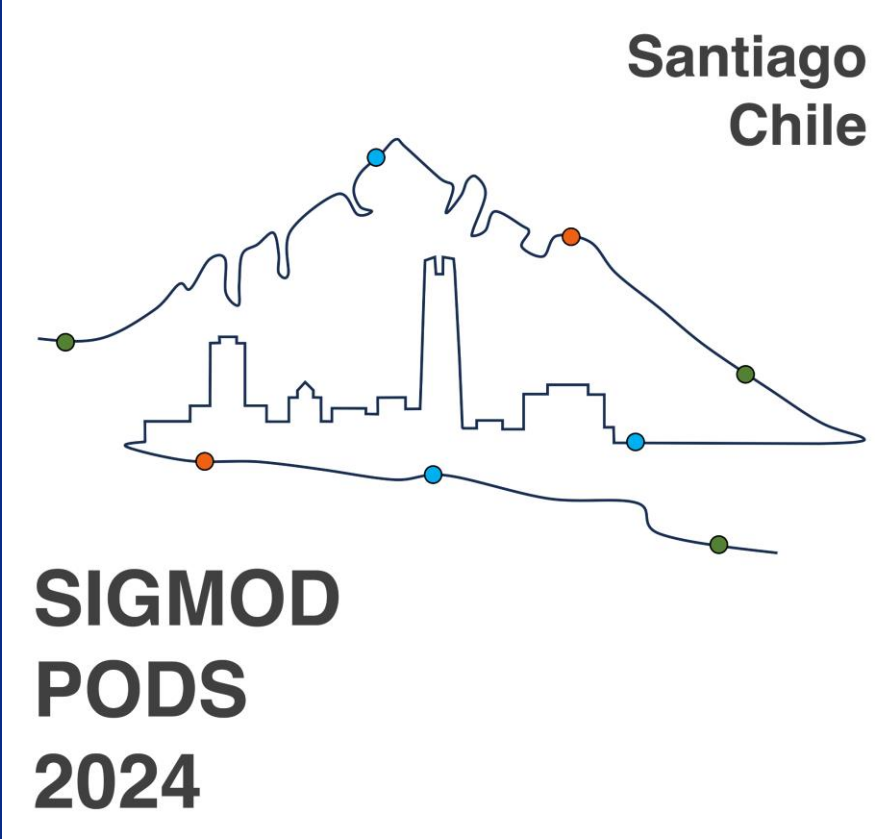




Can Learned Indexes be Built Efficiently? A Deep Dive into Sampling Trade-offs

Minguk Choi, Seehwan Yoo, and Jongmoo Choi
Dankook University, South Korea
{mgchoi, seehwan.yoo, jmchoi}@dankook.ac.kr



Seeking a Ph.D.
Position for 2025

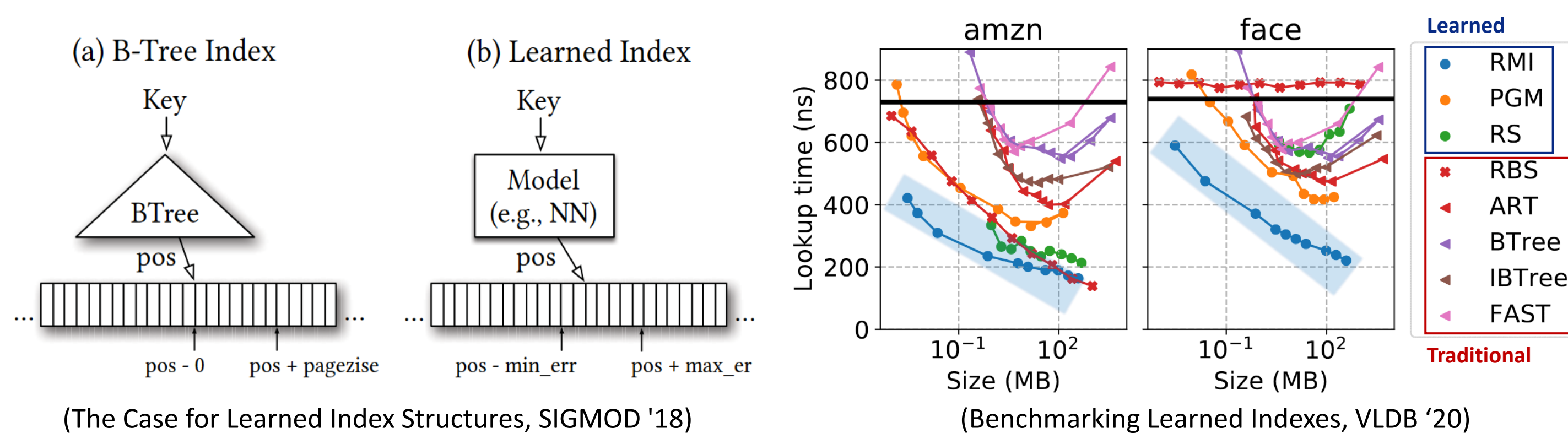
Background

Learned Index Structure

- Index structure employs machine learning techniques
- View the index as a model that predicts the position of a key

Performance of Learned Index: Space-efficient

- Pareto optimal in terms of index size and lookup latency in read-only
 - No alternative exists that has both a smaller size and lower latency



Motivation

Long Index Build Time

- Up to about 2,000x slower than traditional indexes
- But still there are application where index build time is crucial (e.g., LSM-tree)

Why Building the Learned Index is Slow?

Index build time = 1) Number of elements \times 2) Per - element overhead

- Complete traversal and training
- Higher per-element training overhead
 - Light-weight training model: RadixSpline (aiDM'20), Bourbon (OSDI'20)
 - But it's still longer than traditional indexes

This study began with the question ...

Since the learned index uses the model,
Can't it learn efficiently even with less data?

Design

Our Approach: Sampling

Challenges

- Losing the error-bound property due to sampling loss
- Complex trade-offs in terms of model, index, and micro-architecture
- Absence of benchmark for sampling applied indexes

1. Error-bound Preserving Sample Learning Algorithm

- EB-PLA (Error-bounded Piece-wise Linear Approximation) Model

(a) Train all keys with error-bound ϵ
 $\rightarrow \forall k, Error(k) \leq \epsilon$

(b) Train sample I^{th} keys with the error-bound ϵ
 $\rightarrow \forall k, Error(k) \leq \epsilon$

Sample EB-PLA Algorithm

(c) Refine the error-bound due to sampling loss
 $\rightarrow \forall k, Error(k) \leq \epsilon' (= \epsilon + I - 1)$

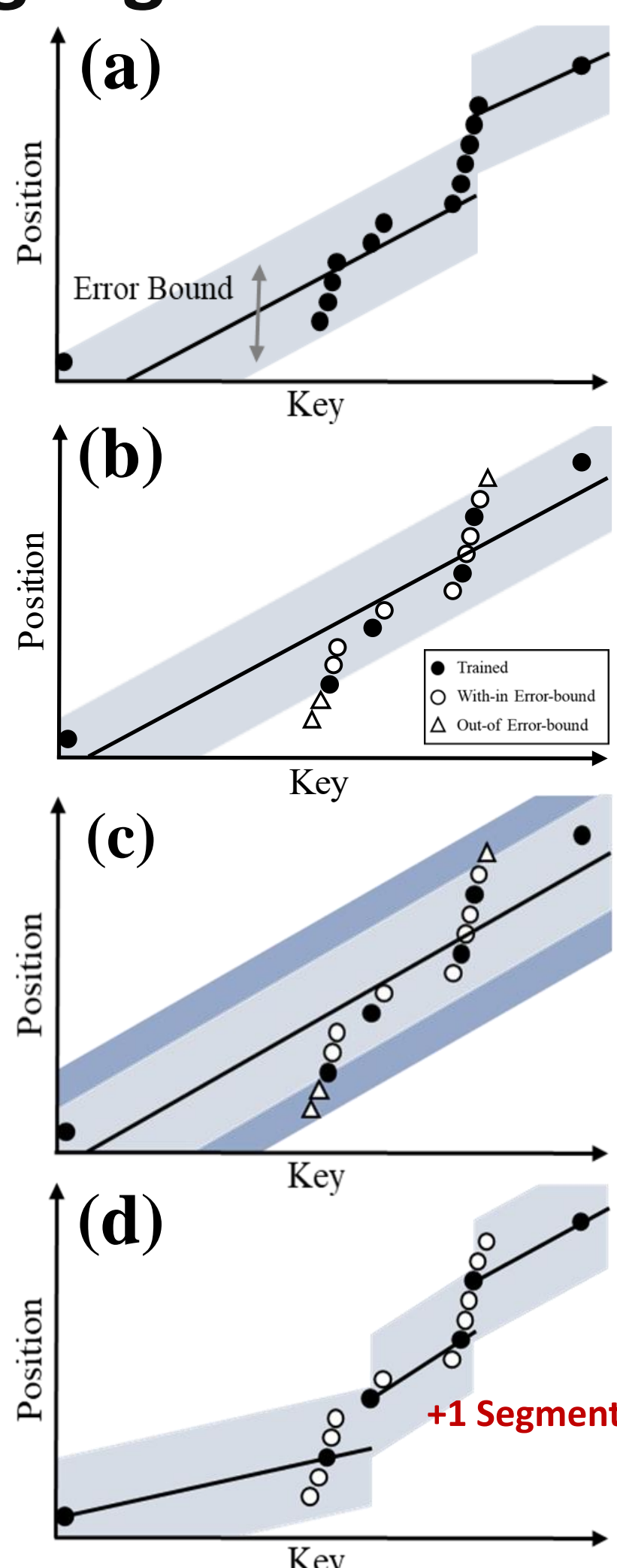
➢ Preserve the error-bound property

(d) Replace the sample learning error-bound to $\delta (= \epsilon - I + 1) \rightarrow \forall k, Error(k) \leq \delta$

➢ Preserve the error-bound (ϵ) by learning less data with smaller & stricter error-bound (δ)

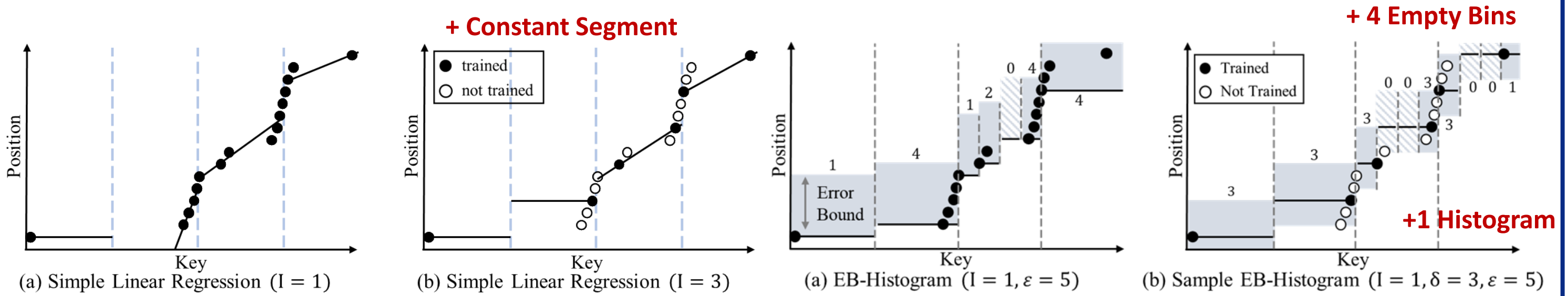
Sample EB-Histogram

- PLR with Simple Linear Regression



2. Internal Changes due to Sampling

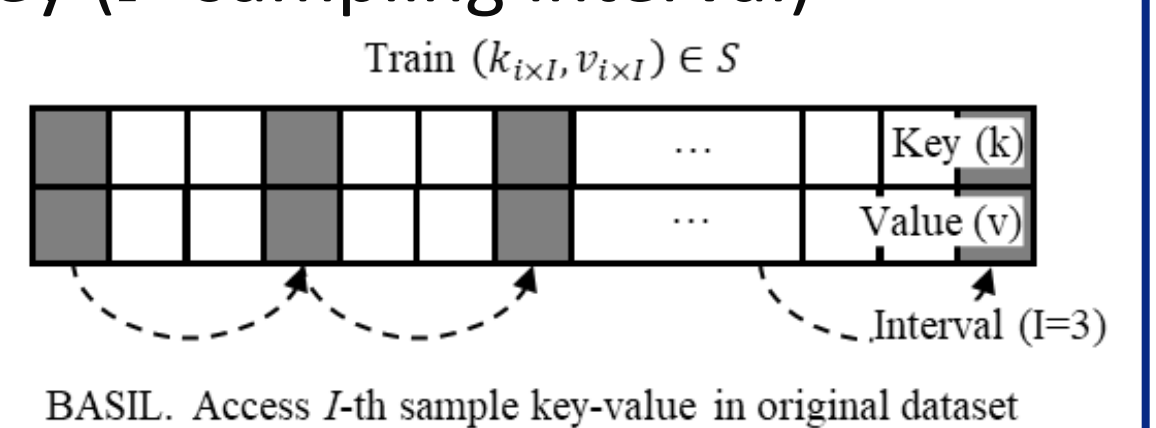
- Dynamic Segmentation (Key range of each segment is different)
 - Aggressive sampling can increase the number of segments
- Fixed Segmentation (Key range of each segment is equal)
 - Aggressive sampling can increase the number of under-fitting segments



3. Unified Sampling Algorithm & Implementation

BASIL (Benchmark of Sampling Applied Learned Indexes)

- Unified Sampling Algorithm: Systematic Sampling
 - Extract every I^{th} key from first to last key (I =sampling interval)
- Unified Sampling Implementation
 - Index access and train only sample key-value data from entire dataset



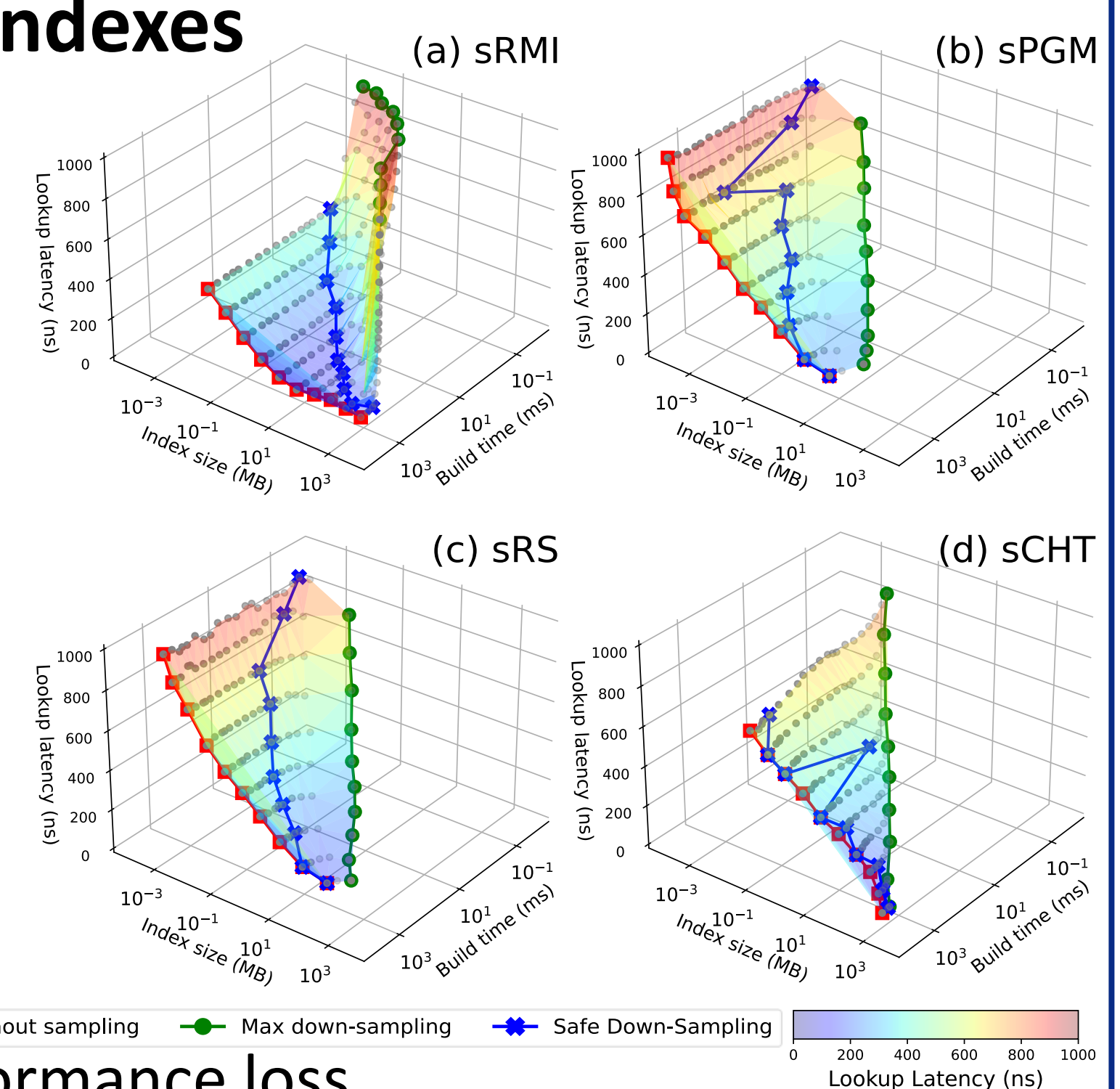
Evaluation

1. Sampling Trade-offs

- Sampling interval (I) \uparrow
 \rightarrow (a) build time \downarrow
- Each error-bound (ϵ) has threshold interval (I^{TH})
- Until I^{TH} , (b-i) rest of metrics remain consistent
- After I^{TH} ,
 - # of segments \uparrow
 \rightarrow (b) Size \uparrow , (d) Height \uparrow
 \rightarrow (e) Pred. cache miss \uparrow , (f) Pred. latency \uparrow
 - After I^{TH} ,
 - # of segments $\uparrow \rightarrow$ (g) MSE $\downarrow \rightarrow$ (h) Corr. cache miss \downarrow , (i) Corr. latency \downarrow

2. Design Space of Learned Indexes

- Without sampling, absence of trade-offs between build, size, and lookup
- Sampling introduce trade-offs between build, size, and lookup
- Broaden design space of learned indexes from 2D to 3D



3. Build Speed-up

- Explore Safe down-sampling, where size & lookup latency increased by less than 5%
- Max build speedup without performance loss
 - sRMI: 1/44,514, sPGM: 1/40,781, sRS: 1/14,479

4. Pareto Analysis

- Can learned indexes be built more efficiently in terms of build time and lookup latency than traditional indexes through sampling?
- To the best of our knowledge, this is first to show that learned indexes are also Pareto optimal in terms of build time and (average and tail) lookup latency

