Lower Bounds for the Algorithmic Complexity of Learned Indexes

Luis Alberto Croquevielle
Imperial College London, UK

Roman Sokolovskii
Imperial College London, UK

Thomas Heinis
Imperial College London, UK

Abstract

Learned index structures aim to accelerate queries by training machine learning models to approximate the rank function associated with a database attribute. While effective in practice, their theoretical limitations are not fully understood. We present a framework for proving lower bounds on query time for learned indexes, expressed in terms of their space overhead and parameterized by the model class used for approximation. Our formulation captures a broad family of one-dimensional learned indexes, including most existing designs, as piecewise model-based predictors.

We solve the problem of lower bounding query time in two steps: first, we use probabilistic tools to control the effect of sampling when the database attribute is drawn from a probability distribution. Then, we analyze the approximation-theoretic problem of how to optimally represent a cumulative distribution function with approximators from a given model class. Within this framework, we derive lower bounds under a range of modeling and distributional assumptions, paying particular attention to the case of piecewise linear and piecewise constant model classes, which are common in practical implementations.

Our analysis shows how tools from approximation theory, such as quantization and Kolmogorov widths, can be leveraged to formalize the space-time trade-offs inherent to learned index structures. The resulting bounds illuminate core limitations of these methods.

2012 ACM Subject Classification Theory of computation → Predecessor queries

Keywords and phrases Learned Indexes, Stochastic Processes, Approximation Theory

Digital Object Identifier 10.4230/LIPIcs.ICDT.2026.14

Related Version *Full Version*: `https://arxiv.org/abs/2601.06629` [3]

Funding This work is funded by DNAMIC (grant 101115389), NEO (grant 101115317), and ANID Chile through the Scholarship Program (DOCTORADO BECAS CHILE/2023 – 72230222).

1 Introduction

Efficient query processing is a fundamental challenge in database systems. For relational databases, key areas of study include join optimization techniques [28, 16], cardinality estimation methods [2, 24], and spatial indexing for multidimensional data [9, 13]. A standard tool in this context is the use of *indexes*, data structures that improve query time at the cost of additional space overhead or higher number of I/O operations.

One of the most widely studied settings is *one-dimensional indexing*, which generally aims to answer both point and range queries over a single, ordered attribute. Typically, the attribute values (or *keys*) are stored in a sorted structure A, and queries are expressed as search operations over A. In particular, a point query for a value q requires locating the position of q in A, while a range query over an interval $[q, q']$ requires identifying the first key $\geq q$ and scanning forward until a key $> q'$ is reached.

In the context of one-dimensional indexing, tree-based structures such as B+trees [11] are the predominant choice in practice, and can generally handle both point and range queries. From an algorithmic perspective, this reduces to computing the *rank* function. Once the rank (or position) of a query key is known, range queries can be answered efficiently by scanning contiguous data stored at the leaves. This idea underlies not only B+trees [11], but also several new developments in database indexing, as we elaborate below.

In recent years, machine learning has emerged as a promising tool for query processing. A prominent example for one-dimensional indexing is the Recursive Model Index [17], which maintains a hierarchical structure similar to B+trees but embeds predictive models in internal nodes to estimate key positions within subsets of sorted data. This work has inspired a broad line of research into "learned indexes" [10, 15, 14, 5], which extend this idea to support dynamic workloads [5, 26], streaming data [27, 19], and multi-dimensional queries [22, 6, 18].

While empirical studies have highlighted their potential advantages over traditional methods [21, 25], learned indexes generally lack formal guarantees. Their performance has also been observed to vary across different datasets and query workloads, in contrast to the consistent behavior of well-established methods such as B+trees. Proving algorithmic guarantees for learned indexes and characterizing the data-dependent behavior of these methods are all important theoretical questions that have been explored only partially.

In this work, we introduce a formal framework that captures a broad class of one-dimensional learned index structures. Within this framework, we establish lower bounds on query time, explicitly characterizing trade-offs between time and space complexity. These bounds hold under general assumptions, covering a wide range of data-generating processes. As an interesting application, these lower bounds allow us to identify a parameter regime where this class of learned indexes offer no asymptotic advantage over traditional methods.

The paper is organized as follows: in Section 2, we present a more detailed account of related work and our own contributions. Section 3 introduces the mathematical setting and notation. Section 4 develops our general theoretical approach, while Sections 5 and 6 instantiate this framework by applying it to specific function approximation scenarios. Finally, Section 7 contains concluding remarks and a discussion of future work.

2 Related Work and Contributions

The next section introduces our full notation, but we provide some preliminary definitions here. We model an attribute A in a database relation as formed by a sequence $X_1, \ldots, X_n$ of numeric values. Our focus is on one-dimensional indexing, where queries are posed over a single, ordered attribute and both point and range queries are supported. The central object of study in our analysis is the **rank** function, defined as $\mathbf{rank}(q) = \sum_{i=1}^{n} \mathbf{1}_{(-\infty, X_i]}(q)$, which counts how many keys are less than or equal to a given query value q. In this context, query time is typically understood as the time required to evaluate $\mathbf{rank}(q)$.

Classical methods such as B+trees are agnostic to the process by which data is generated. In contrast, learned indexes derive performance guarantees by making assumptions about it, e.g., that the $\{X_i\}$ are i.i.d. with respect to some distribution. This is to be expected in light of related areas of research, in particular statistical learning theory, where generalization guarantees usually rely on i.i.d. assumptions for both the training and testing data.

As a baseline, B+trees support search, insertion, and deletion in $O(\log n)$ time using $O(n)$ space, with worst-case guarantees that hold independently of the data-generating process. Initial theoretical analysis of learned indexes, beginning with the PGM-Index [8], showed performance comparable to B+trees in the static setting, but with $O(\log^2 n)$ search complexity for dynamic workloads. Under (restrictive) assumptions on the data-generating process, a constant improvement in space complexity relative to B+trees was shown [7].

More recent work has established that specific learned indexes can asymptotically outperform B+trees in expectation, for the case of static databases. Zeighami and Shahabi [29] proved that $O(\log(\log n))$ expected search time is achievable with linear space. Croquevielle et al. [4] further improved this to $O(1)$ expected time. For dynamic databases, Zeighami and Shahabi [30] considered a model where the data distribution can vary over time, with a parameter δ capturing the amount of distribution shift. They introduce a learned index that supports insertions and queries in $O(\log(\log n))$ expected time when $\delta = 0$ (i.e., no distribution shift), an asymptotic improvement over B+trees. For $\delta > 0$, an extra $O(\log(\delta n))$ cost is incurred, so that asymptotic complexity remains the same as for B+trees.

A complementary line of work was recently introduced by Zeighami and Shahabi [31], who study lower bounds for learned database operations from an information-theoretic perspective. In their framework, a learned index is modeled as a parameterized function f_θ where the parameters θ are encoded using a total budget of σ bits. The authors derive lower bounds on the model size σ, showing that if it is too small, there exists at least one input sequence $X_1, \ldots, X_n$ for which a desired level of approximation accuracy cannot be satisfied.

Our work takes a fundamentally different approach. Instead of analyzing a general function f_θ, we focus on a structured class $\mathcal{I}$ of indexing algorithms that captures common design elements found in most learned index constructions for one-dimensional data. In particular, our scope consists of learned indexes that store keys in sorted order to support efficient point and range queries, which is the most common setting in the literature. While this narrows the scope of our work compared to the general setting of [31], it enables sharper lower bounds that are more directly relevant to learned indexes as commonly defined in the literature. Within this class, we establish lower bounds on query time as a function of the space overhead used by the indexing structure.

More specifically, we model learned indexes as directed acyclic graphs (DAGs), where internal nodes direct the search, and sink nodes encode predictive models and store the keys. This abstraction reflects a common structural pattern shared by many one-dimensional learned index designs. We develop a general framework for proving query time lower bounds as a function of the model class complexity, the number of sink nodes K, the size of the dataset n, and the choice of local search algorithm (e.g., linear or binary search).

For a broad class of learned indexes, capturing most known designs, we prove that query time is lower bounded by $\Omega(\log(n/K))$ in the worst case. A key implication is that, in order to achieve asymptotic improvement over traditional structures like B+trees (which guarantee $O(\log n)$ query time with linear space), a learned index must use nearly linear space. Specifically, if the number of sink nodes satisfies $K = O(n^\alpha)$ for any $0 < \alpha < 1$, the worst-case query time remains $\Omega(\log n)$–matching that of B+trees. Thus, learned indexes can only hope for worst-case asymptotic gains when space usage is close to linear. When the model class is restricted to piecewise constant functions, this same conclusion further applies to *average-case query time*, where the expectation is taken over the query distribution.

3 Preliminaries

3.1 Data as Stochastic Process

We consider a probabilistic model for a database attribute. Formally, we model the attribute as a stochastic process $X = \{X_i\}_{i \in \mathbb{N}}$, where each X_i is a real-valued random variable. At time $n \in \mathbb{N}$, the attribute consists of the first n values in the sequence, sorted in increasing order. We denote the sorted values by $X_{(1)} \leq \ldots \leq X_{(n)}$, so that the attribute is represented as an array A_n with $A_n[i] = X_{(i)}$ for all $i \in \{1, \ldots, n\}$. This framework accommodates

both worst-case and average-case analyses: we may consider a single sequence $X_1, \ldots, X_n$ to study worst-case behavior, or incorporate the probability measure $\mathbb{P}$ governing the stochastic process to analyze expected performance.

No complexity lower bounds will apply in full generality. For instance, if $X_i = i$, then the rank function **rank** can be computed in $O(1)$ time and space, and existing structures such as the PGM-Index easily attain that performance. In this work, we derive lower bounds under the assumption that the X_i are independent random variables drawn according to a common cumulative distribution function (CDF) F. We further assume that the query value q is generated independently from the X_i according to a probability measure μ which has support $\mathcal{Q} \subseteq \mathbb{R}$.[1] We emphasize that independence assumptions are only required for average-case analyses; our worst-case bounds hold unconditionally, though independence is used as a technical tool in their derivation.

Throughout the paper, we use $\mathbb{E}_X[\cdot]$ to denote expectation with respect to the measure $\mathbb{P}$ governing the key sequence $\{X_i\}_{i=1}^n$. The notation $\mathbb{E}_q[\cdot]$ denotes expectation with respect to the query distribution μ, treating the key values $\{X_i\}_{i=1}^n$ as fixed (i.e., conditioning on a realization of the data). When expectations are taken jointly, we write $\mathbb{E}_{X,q}[\cdot]$, corresponding to integration with respect to the product measure $\mathbb{P} \otimes \mu$.

3.2 Computational Model and Learning Problem

One-dimensional index structures that support point and range queries over a sorted attribute must be able to determine the relative position of a query value among the stored keys. This requirement naturally leads to the *rank* function, which maps a query value to the number of keys less than or equal to it, and serves as the central abstraction in our analysis. As introduced in Section 2, the **rank** function over the attribute A_n is given by

$$\mathbf{rank}_n(q) = \sum_{i=1}^{n} \mathbf{1}_{(-\infty, X_i]}(q)\,,$$

where $\mathbf{1}_U$ denotes the indicator function of a set U. Hence, $\mathbf{rank}_n(q)$ maps a query value $q \in \mathbb{R}$ to the number of keys less than or equal to it. Since it is usually clear from context, we omit the sub-index n and write **rank** instead.

Given access to $\mathbf{rank}(q)$, point queries and range queries can both be expressed in terms of positions within the array A_n. In most one-dimensional learned indexes, the goal is therefore to compute $\mathbf{rank}(q)$ while minimizing the required time and space overhead. We make these notions precise through the RAM model of computation.

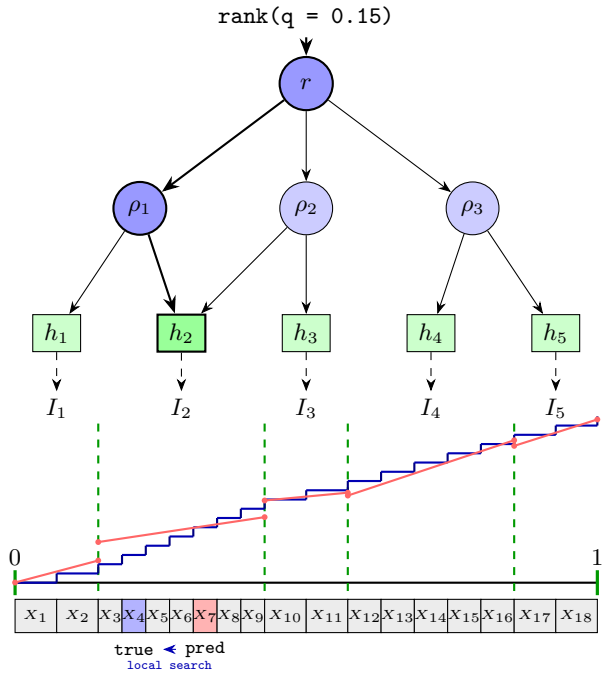
Specifically, we adopt the RAM model of computation under the uniform cost criterion [1], where each basic instruction requires one time unit and each register, storing an arbitrary integer, uses one space unit. *Time complexity* is defined as the number of basic instructions required to evaluate $\mathbf{rank}(q)$ using an index. *Space complexity* measures the redundant memory used by the index, excluding the space needed to store the attribute A_n itself.

Let I be an indexing procedure that builds a data structure $I(A_n)$ from the attribute A_n. For a query value q, let $T(I(A_n), q)$ denote the time required to compute $\mathbf{rank}(q)$ using the data structure $I(A_n)$, and let $S(I(A_n))$ denote the space overhead used by $I(A_n)$. Since it is always clear from context, we usually write $T(q)$ in place of $T(I(A_n), q)$.

[1] Formally, one must specify measurable spaces for the data and query distributions. For simplicity, we assume that all relevant spaces are equipped with the Borel σ-algebra and that all functions we integrate are measurable; these assumptions are standard and introduce no meaningful restrictions.

3.3 Learned Index Structures

We model a learned index as a directed acyclic graph (DAG), as shown in Figure 1. Internal nodes implement routing functions, and sink nodes contain predictive models and store the keys. Each query follows a path from the root to a sink node, where a prediction of its rank is refined via local search. In order to support efficient range queries, keys associated with each sink are stored contiguously (or in a small number of contiguous blocks), so that scanning incurs low overhead. This structure captures a wide range of learned indexes, most of which include hierarchical routing–such as RMI [17], ALEX [5], and PGM-Index [8].



Root node: All queries enter here and are routed through the DAG. An example route for $q = 0.15$ is highlighted with thicker traversal lines.

Internal nodes: Each internal node v applies a routing function $\rho_v(q)$ to direct the query to one of its children.

Sink nodes: Each sink node v contains a model $h_v(q)$ that estimates $\mathbf{rank}(q)$ within the subinterval I_v.

In this example, the true rank function is shown as a blue step function, and the predictive models h_v are represented as red linear segments.

Local search: The estimate $h_v(q)$ is corrected through a local search over a subarray stored at the sink node.

Figure 1 Illustration of a learned index modeled as a directed acyclic graph (DAG).

For the purpose of lower bounds, we focus on the *local correction step*. Regardless of how queries are routed, the exact rank of q is ultimately recovered by searching around the predicted value $h(q)$ in the sorted array. Since we aim for lower bounds that hold regardless of internal structure–which can vary substantially between different designs–the cost of this local search becomes the main driver of query complexity in our analysis. By isolating this step, our analysis applies independently of the routing mechanism and can only underestimate the true query time of any concrete index.

We therefore abstract the index as a piecewise function defined by its sink-node models, and let K denote the number of sinks. This corresponds to a partitioning of the query domain into K regions $\{I_v\}_{v=1}^{K}$, each associated with a model h_v. Although we do not analyze the routing structure, we include it in the model to emphasize that our lower bounds apply regardless of graph topology or the complexity of the routing functions. Understanding how this structure impacts query time remains an open question and may lead to sharper bounds.

This abstraction captures one-dimensional index structures designed to support both point and range queries by locating positions within sorted data. A canonical example is the PGM-Index, which partitions the query domain into disjoint intervals and assigns to

each interval a piecewise linear model that approximates the rank function over a contiguous subarray of keys. In contrast, data structures that only support membership or equality queries fall outside the scope of our analysis.

We denote by $\mathcal{I}$ the class of learned indexes captured by our DAG model, and by $\mathcal{I}_{K,\mathcal{L}}$ those whose sink-node models belong to a function class $\mathcal{L}$ (e.g., $\mathcal{L}$ could be the set of polynomial functions) and use at most K sinks. For the rest of the paper, we use K as a lower bound (and proxy) for space usage, and we focus on how the approximation power of $\mathcal{L}$ interacts with K to characterize performance.

4 General Framework for Lower Bounds

We now present a general framework for lower bounding the query time of learned indexes. We consider the query time $T(q)$ as the number of operations needed to compute $\mathbf{rank}(q)$ using the index built on the array A_n. The bounds are expressed in terms of the space overhead $S(R(A_n))$, via the number of sink nodes K. The analysis is based on three ideas:

(a) Relating the query time $T(q)$ to the prediction error $\varepsilon(q) = |h_v(q) - \mathbf{rank}(q)|$. Our analysis focuses on the final step of query execution: the local search after the model prediction. Since this step is always required to confirm or correct the predicted rank, its cost provides a valid lower bound on $T(q)$, which grows with the magnitude of $\varepsilon(q)$.

(b) Reducing the analysis of the prediction error $\varepsilon(q)$ to a function approximation problem for the CDF F, by using the approximation $\mathbf{rank}(q) \approx nF(q)$.

(c) A probabilistic analysis that formalizes the approximation $\mathbf{rank}(q) \approx nF(q)$ and allows us to derive lower bounds on $T(q)$ under various probabilistic assumptions.

We begin by describing components (b) and (c), and defer discussion of (a) to the end of this section, where we provide general lower bounds for $T(q)$. For clarity, most proofs are omitted during this exposition, and all omitted technical details are included in the appendix.

4.1 From Learned Indexes to Approximation Error

A learned index as defined in Section 3.3 can be seen as a structure that approximates $\mathbf{rank}(q)$ through a piecewise-defined prediction function

$$h(q) = \sum_{v=1}^{K} h_v(q)\mathbf{1}_{I_v}(q)\,, \tag{1}$$

where each h_v is a predictive model and $I_v \subseteq \mathcal{Q}$ its associated subdomain. For convenience, we identify an index I with its induced predictive model h; thus, we take $h \in \mathcal{I}_{K,\mathcal{L}}$ to mean that h is defined piecewise over K disjoint intervals using functions from the class $\mathcal{L}$.

Given a predictive function h, the prediction error is $\varepsilon(q) = |\mathbf{rank}(q) - h(q)|$. To analyze $\varepsilon(q)$ from the perspective of function approximation, we use the *empirical cumulative distribution function* (ECDF). Recall that our standing assumption is that the keys $X_1, \ldots, X_n$ are drawn i.i.d. from a distribution with CDF F. The ECDF is correspondingly defined as

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{(-\infty, X_i]}(x)\,.$$

The rank function can then be expressed as $\mathbf{rank}(q) = nF_n(q)$. Now, for both average and worst-case analysis, it is relevant to consider the expected error over all query values q, conditioned on $X_1, \ldots, X_n$. Since q is generated independently from $\{X_i\}$, we have

$$\mathbb{E}_q\big[\varepsilon(q)\big] = \int_{\mathcal{Q}} \varepsilon(q)d\mu = \int_{\mathcal{Q}} |\mathbf{rank}(q) - h(q)|d\mu = \int_{\mathcal{Q}} |nF_n(q) - h(q)|d\mu = \|nF_n - h\|_\mu\,, \tag{2}$$

where $\|\cdot\|_\mu$ denotes the absolute norm in $L^1(\mu)$, the space of μ-integrable functions over $\mathcal{Q}$. In other words, the best h function on average would be the function that minimizes the absolute distance to $\mathbf{rank} = nF_n$. This motivates the introduction of the *minimum approximation error*:

$$R_{K,\mathcal{L}}(G) = \inf_{h \in \mathcal{I}_{K,\mathcal{L}}} \|G - h\|_\mu \,,$$

for any $G \in L^1(\mu)$. When $G = \mathbf{rank} = nF_n$, this yields

$$R_{K,\mathcal{L}}(\mathbf{rank}) = R_{K,\mathcal{L}}(nF_n) = nR_{K,\mathcal{L}}(F_n) \,, \tag{3}$$

by the linear scaling property of the approximation error (see Appendix A in the extended version of the paper [3] for the proof). Now, the analysis of $\mathbb{E}_q[\varepsilon(q)]$ reduces to an approximation problem. The key idea is that $\mathbb{E}_q[\varepsilon(q)]$ can be lower bounded by the optimal $L^1(\mu)$-approximation of F, up to a small deviation term arising from sampling randomness. This deviation is captured by the function $\delta_n(x) = F(x) - F_n(x)$, where F_n is the ECDF. The following result makes this relationship precise. The proof is straightforward and can be found in Appendix A of the extended version of the paper [3].

▶ **Proposition 1.** *Let $X_1, \ldots, X_n$ be i.i.d. samples from a distribution with CDF F, and let $q \sim \mu$ independently from the $\{X_i\}$. Then for any predictive model $h \in \mathcal{I}_{K,\mathcal{L}}$ it holds that*

$$\mathbb{E}_q\big[\varepsilon(q)\big] = \|nF_n - h\|_\mu \geq n\left[R_{K,\mathcal{L}}(F) - \|\delta_n\|_\mu\right] \,,$$

provided $\mathcal{L}$ is invariant under scalar multiplication.

This shows that the expected prediction error is lower bounded by a deterministic approximation term $nR_{K,\mathcal{L}}(F)$, minus a data-dependent fluctuation term $n\|\delta_n\|_\mu$. Controlling this fluctuation is the main probabilistic step in our analysis and will be addressed in the next subsection. In addition to $\mathbb{E}_q\big[\varepsilon(q)\big]$, we also need bounds on $\mathbb{E}_q\big[\log_2 \varepsilon(q)\big]$, which arises in the analysis of exponential search. To this end, we define a proxy for query time: $\widetilde{T}(q) = \log_2(\max\{2, \varepsilon(q)\})$, and establish a lower bound for its expectation. This requires a careful analysis and specific assumptions on F, μ, and $\mathcal{L}$. The following result illustrates this type of bound (for the proof, see the extended version of the paper [3]).

▶ **Proposition 2.** *Let $q \sim \mu$ independently from the keys $\{X_i\}_{i=1}^n$. Assume that the CDF F and μ have respective densities f and g, such that $0 < c_F \leq f(q), g(q) \leq C_F < \infty$ for all $q \in \mathcal{Q}$. Then, there exists a realization of $X_1, \ldots, X_n$ such that:*

$$\mathbb{E}_q\big[\widetilde{T}(q)\big] \geq C_1\left[\log_2\left(nR_{K,\mathcal{L}}(F)\right) - C_2\right] \,,$$

where $C_1, C_2 > 0$ are independent of n and K, and $\mathcal{L} = P_0$–the class of constant functions.

Proposition 2 provides a logarithmic analogue to Proposition 1, which is useful for analyzing exponential search. The use of $\max\{2, \cdot\}$ in $\widetilde{T}(q)$ ensures that the proxy for query time is at least 1. The constants C_1 and C_2 depend on c_F and C_F, and might be disadvantageous. Still, they are independent of n and K, and do not change the asymptotic analysis. Moreover, the condition $0 < c_F \leq f, g \leq C_F < \infty$ need not be satisfied on all of $\mathcal{Q}$; it suffices that it holds over a subinterval $\mathcal{Q}' \subseteq \mathcal{Q}$. The main limitation of this variant, in contrast to Proposition 1, is the constraint $\mathcal{L} = P_0$, and the fact that this bound only holds for a specific realization $X_1, \ldots, X_n$: it does not necessarily hold for all $\{X_i\}_{i=1}^n \sim \mathbb{P}$.

4.2 Controlling the Randomness: Concentration of F_n

The lower bounds in Propositions 1 and 2 depend on $R_{K,\mathcal{L}}(F)$ and the behavior of δ_n. On the one hand, $R_{K,\mathcal{L}}(F)$ is a deterministic quantity that depends only on F, the model class $\mathcal{L}$, and the measure μ, and can be studied using approximation theory. On the other hand, we can use probabilistic tools to control the term $\|\delta_n\|_\mu$, ensuring it is small enough that the dominant source of error stems from approximating F rather than from sampling variability.

Generally, we wish to say that δ_n behaves like $1/r(n)$ for some function $r(n)$ that quantifies the rate of convergence of F_n to F. This allows us to isolate all randomness in a single term, $1/r(n)$, and express lower bounds on $\mathbb{E}_q[\varepsilon(q)]$ in terms of the deterministic approximation error for F. To make this precise, we rely on two types of results: expected and worst-case concentration. All proofs are included in the extended version of the paper, Appendix D [3].

Expected concentration. The main result of this type is given by Lemma 3, which we will use in conjunction with Proposition 1.

▶ **Lemma 3** (Expected bounds). *Assume $X_1, \ldots, X_n$ are i.i.d. and let μ be an arbitrary probability measure over $\mathcal{Q}$, independent from the $\{X_i\}$. Then, it holds that:*
(a) $\mathbb{E}_X[\|\delta_n\|_\infty] \leq \sqrt{\frac{\pi}{2n}}$.
(b) *As a consequence,* $\mathbb{E}_X[\|\delta_n\|_\mu] \leq \sqrt{\frac{\pi}{2n}}$.

These bounds allow us to reason about the concentration of δ_n in expectation over the random samples $\{X_i\}$, providing an average-case view of the approximation error–again, without making any assumptions on the query distribution μ.

Worst-case concentration. The main result of this type is given by Lemma 4, which we use to analyze the worst-case algorithmic performance of learned indexes.

▶ **Lemma 4** (Existence of small-deviations). *Assume $X_1, \ldots, X_n$ are i.i.d. and let μ be an arbitrary probability measure over $\mathcal{Q}$, independent from the $\{X_i\}$. Then, it holds that:*
(a) *There exists a realization of $\{X_i\}_{i=1}^n$ such that $\|\delta_n\|_\mu \leq \sqrt{\frac{\pi}{2n}}$.*
(b) *If μ is equal to the data-generating distribution, i.e., $d\mu = dF$, then there exists a realization of $\{X_i\}_{i=1}^n$ such that $\|\delta_n\|_\mu \leq \frac{1}{\sqrt{6}\,n}$.*

This highlights how stronger assumptions on μ can yield sharper bounds–specifically, a $1/n$ rate instead of the general $1/\sqrt{n}$. In contrast, bounds that hold for arbitrary μ must remain more conservative to account for unfavorable scenarios: they may be loose in most cases but cannot be improved without additional assumptions. While a full characterization of how different choices of μ affect the convergence rate is outside the scope of this paper, we include the $d\mu = dF$ case as it reflects a common setting in the learned indexes literature.

4.3 From Prediction Error to Query Time

We now complete our framework by translating prediction error into actual query time. As discussed in Section 3.3, the final stage of query processing involves a local search to correct the predicted rank $h(q)$ into the true value $\mathbf{rank}(q)$. The cost of this step depends on the prediction error $\varepsilon(q) = |\mathbf{rank}(q) - h(q)|$, and varies depending on the algorithm used. We analyze the three most common search strategies: linear, exponential, and binary search.

▶ **Proposition 5** (Lower bounds for query time via prediction error). *Let $q \in \mathcal{Q}$ be a query with predicted rank $h(q)$ and actual rank $\mathbf{rank}(q)$, and let $\varepsilon(q) = |\mathbf{rank}(q) - h(q)|$ be the prediction error. Then, under the respective search strategies:*

(a) ***Linear search:*** $T(q) \geq \varepsilon(q)$.
(b) ***Exponential search:*** $T(q) \geq \log_2(\max\{2, \varepsilon(q)\})$.
(c) ***Binary search:*** $\sup_q T(q) \geq \log_2(\sup_q \varepsilon(q))$.

Proof. We analyze each strategy in turn.

(a) **Linear search.** This method scans sequentially from the predicted position $h(q)$ until it reaches the true position $\mathbf{rank}(q)$, as illustrated in Figure 2. In practice, the search may advance by 2, 3, or any constant number of positions per step, but this only affects the total number by a constant factor and does not change asymptotic bounds. We therefore assume 1 step per comparison. The total number is thus at least $\varepsilon(q) = |\mathbf{rank}(q) - h(q)|$, so we obtain the lower bound $T(q) \geq \varepsilon(q)$.

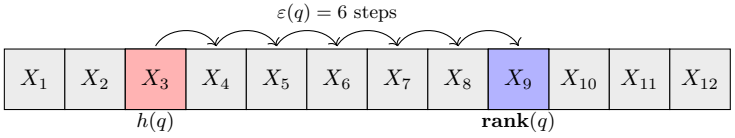


Figure 2 Linear search performs a step-by-step scan from the predicted position $h(q)$ to the true rank $\mathbf{rank}(q)$. The total cost is proportional to the prediction error $\varepsilon(q)$.

(b) **Exponential search.** Starting at $h(q)$, the algorithm expands geometrically (typically doubling the offset) until it brackets the true position $\mathbf{rank}(q)$ within a window, as visualized in Figure 3. It then performs binary search within that window. The number of expansion steps required to reach a position $\geq \mathbf{rank}(q)$ is $\lceil \log_2 \varepsilon(q) \rceil + 1 \geq \log_2 \varepsilon(q)$. The cost of the final binary search is also logarithmic in the window size. Since query time cannot involve less than 1 operation, we write:

$$T(q) \geq \widetilde{T}(q) = \log_2(\max\{2, \varepsilon(q)\}),$$

where $\widetilde{T}$ corresponds to the proxy function introduced earlier.

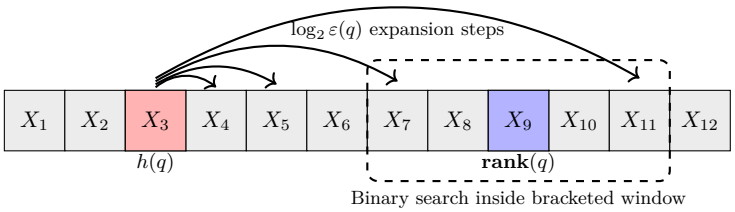


Figure 3 Exponential search expands a geometric window around the predicted position $h(q)$ until the true rank $\mathbf{rank}(q)$ is within range. The cost is logarithmic in the prediction error $\varepsilon(q)$.

(c) **Binary search.** In this case, the algorithm relies on a fixed window size that is guaranteed (by construction) to contain the true rank. The size of this window must be chosen to cover the *maximum possible prediction error* over all q, i.e., $\sup_q \varepsilon(q)$. Since binary search over a window of size M takes at least $\log_2 M$ steps in the worst case, we obtain:

$$\sup_q T(q) \geq \log_2(\sup_q \varepsilon(q)).$$

This is illustrated in Figure 4: although for this specific value of q the prediction error is smaller, the search range must still be wide enough to accommodate the worst-case error. ◀

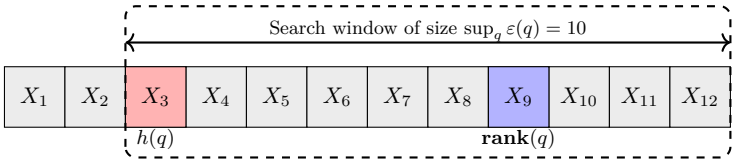


Figure 4 Binary search requires a search interval guaranteed to contain the true position $\mathbf{rank}(q)$. Therefore, the cost depends on the worst-case prediction error and is logarithmic in the interval size.

4.4 Lower Bounds on Query Time

We are now ready to combine the results from the previous subsections to derive concrete lower bounds on the query time $T(q)$ under the three search strategies described in Section 4.3. We consider three natural types of lower bounds, which differ in the strength of their guarantees and the assumptions they require. In increasing order of generality, we consider:

(A) Full expectation bounds (average-case):

$$\mathbb{E}_{X,q}[T(q)] \geq C(n, K, \mathcal{L}, F, \mu)\,.$$

This is the strongest type of result, holding in expectation over both the random dataset $\{X_i\}_{i=1}^n$ and the query value $q \sim \mu$. It typically requires strong assumptions (e.g., i.i.d. sampling) but yields average-case bounds across the entire system.

(B) Conditional expectation bounds (mixed-case):

$$\sup_{\{X_i\}_{i=1}^n} \mathbb{E}_q[T(q)] \geq C(n, K, \mathcal{L}, F, \mu)\,.$$

This corresponds to worst-case bound over data and average-case bound over queries.

(C) Pointwise bounds (worst-case):

$$\sup_{\{X_i\}_{i=1}^n, q \in \mathcal{Q}} T(q) \geq C(n, K, \mathcal{L}, F, \mu)\,.$$

This is a purely worst-case guarantee, asserting that there exists some dataset and query for which the query time is large. It requires the fewest assumptions.

These regimes form a natural hierarchy: a type (A) bound implies one of type (B), which in turn implies one of type (C). This implication structure follows directly from the probabilistic method and reflects a trade-off between generality and the strength of assumptions. Importantly, the bounds hold uniformly over the model class $\mathcal{I}_{K,\mathcal{L}}$, i.e., they apply to every learned index structure in $\mathcal{I}_{K,\mathcal{L}}$.

Table 1 summarizes the resulting lower bounds on query time $T(q)$ under different search strategies, assumptions on the query distribution μ, and the three types of lower bounds (A)–(C) discussed above. These results show how the strongest bounds are attainable for linear search, followed by exponential search, and finally binary search, where we can only derive worst-case guarantees. All proofs are included in the extended version of the paper [3]; most follow directly from the general framework developed earlier in this section.

Notice that the type (B) bound for exponential search is written asymptotically, without the constant C_1 from Proposition 2, in order to highlight its structural similarity to the rest.

These results highlight the trade-offs between model complexity, dataset size, search strategy, and query performance in learned indexes. Note that Table 1 considers two representative regimes for the query distribution: (*i*) arbitrary μ, and (*ii*) the matched case $d\mu = dF$, where queries are drawn from the same distribution as the dataset. The latter is especially relevant in practice, as it is widely used in evaluations of learned indexes [8, 21, 19].

Table 1 Lower bounds on query time $T(q)$ for various search strategies, bound types, and assumptions on μ. Type (A) bounds further assume that the $\{X_i\}$ are i.i.d. and that q is independent from the $\{X_i\}$. Type (B) bounds assume $\mathcal{L} = P_0$, that q is independent from the $\{X_i\}$, and the condition $0 < c_F \leq f, g \leq C_F < \infty$.

Bound	Search	Bound Type	Assumption	Lower Bound
L1	Linear	(A) (avg/avg)	Any μ	$n\left[R_{K,\mathcal{L}}(F) - \sqrt{\frac{\pi}{2n}}\right]$
L2	Linear	(B) (worst/avg)	$d\mu = dF$	$n\left[R_{K,\mathcal{L}}(F) - \frac{1}{\sqrt{6}\,n}\right]$
E1	Exponential	(B) (worst/avg)	Any μ	$\log_2\left(n\ R_{K,\mathcal{L}}(F)\right) - C_2$
E2	Exponential	(C) (worst/worst)	$d\mu = dF$	$\log_2\left(n\left[R_{K,\mathcal{L}}(F) - \frac{1}{\sqrt{6}\,n}\right]\right)$
B1	Binary	(C) (worst/worst)	Any μ	$\log_2\left(n\left[R_{K,\mathcal{L}}(F) - \sqrt{\frac{\pi}{2n}}\right]\right)$
B2	Binary	(C) (worst/worst)	$d\mu = dF$	$\log_2\left(n\left[R_{K,\mathcal{L}}(F) - \frac{1}{\sqrt{6}\,n}\right]\right)$

Altogether, this framework provides a general and flexible method for deriving query time lower bounds. In the following sections, we analyze the error term $R_{K,\mathcal{L}}(F)$ for concrete model classes commonly used in learned index structures. Section 5 focuses on $\mathcal{L} = P_0$, corresponding to piecewise constant models [29, 4], while Section 6 considers $\mathcal{L} = P_1$, corresponding to piecewise linear approximators, the most widely adopted choice in practice.

5 Lower Bounds for Piecewise Constant Models

The general lower bounds developed in Section 4 gain full significance once we derive explicit formulas for the approximation error $R_{K,\mathcal{L}}(F)$. This requires lower bounding this term for specific choices of the model class $\mathcal{L}$.

Virtually all learned index structures rely on polynomial models, as they have few parameters to store, and are fast to train and evaluate. Accordingly, we focus on model classes of the form $\mathcal{L} = P_m$, where P_m denotes the class of polynomials of degree at most m. In this section, we analyze the case $\mathcal{L} = P_0$, which corresponds to piecewise constant models. Next, in Section 6, we address the most widely used case in practice: $\mathcal{L} = P_1$, corresponding to piecewise linear models.

The key idea in our analysis is that, as we show below, for piecewise constant models the problem of approximating the CDF F reduces to finding the optimal quantization of a random variable distributed over $[0, 1]$. This allows us to borrow fundamental results from quantization theory to characterize the approximation error $R_{K,P_0}(F)$.

Our results in this section are derived for a broad class of distributions F and μ–specifically, the case where both are characterized by density functions with respect to the Lebesgue measure. We denote these densities f and g, respectively, and consider both the case $f = g$ (Section 5.1) and the general case (relegated to Appendix A). In both cases, we show that the approximation error decreases linearly with K; the difference is that for $f = g$ this can be shown for any K, whereas for $f \neq g$ the result holds asymptotically as $K \to \infty$.

More specifically, throughout Sections 5 and 6, we focus on the case where F is Lipschitz continuous. This regularity assumption is relatively mild, and it encompasses virtually all common continuous distributions (e.g., Gaussian, exponential, uniform). At the same time, it guarantees the existence of a density f, and it ensures that lower bounds reflect the limitations of the model class (as governed by K and $\mathcal{L}$), rather than artifacts of highly pathological target functions.

5.1 Same dataset and query distributions $f = g$

For the relatively simple case of piecewise constant approximating functions $h \in \mathcal{I}_{K,P_0}$ and $f = g$, the minimum approximation error $R_{K,P_0}(F)$ can be calculated exactly for any K. We achieve this by first showing that this problem setup reduces to optimal quantization of a uniformly distributed random variable, and then relying on a well-known result in quantization theory to calculate the error of the associated optimal quantizer.

Let $h \in \mathcal{I}_{K,P_0}$. Since F is a CDF with range $[0,1]$, without loss of generality we can assume $0 \le h(q) \le 1$ for all q. Therefore, h can be written as

$$h(q) = \textstyle\sum_{k=1}^{K} c_k \mathbf{1}_{I_k}(q)\,,$$

where the K intervals $\{I_k\}$ partition $\mathcal{Q}$, and $c_k \in [0,1]$ are the corresponding approximating constants. This means that when $f = g$, the absolute error $\|F - h\|_\mu$ can be written as

$$\|F-h\|_\mu = \int_{\mathcal{Q}} |F(q)-h(q)|\, f(q)dq = \sum_{k=1}^{K} \int_{I_k} |F(q)-c_k|\, f(q)dq = \sum_{k=1}^{K} \int_{J_k} |u-c_k|\, du\,, \quad (4)$$

where $J_k = F(I_k) = \{F(q) : q \in I_k\}$ (which is a continuous interval), and the last equality comes from substituting $u = F(q)$ and $du = f(q)dq$ inside the integrals. Since F is a CDF we know that $\{J_k\}$ is a partition of $[0,1]$. If we now define $Q_K(u) = \sum_{k=1}^{K} c_k \mathbf{1}_{J_k}(u)$, then (4) can be expressed as

$$\|F-h\|_\mu = \mathbb{E}\big[|U - Q_K(U)|\big]\,, \quad (5)$$

where Q_K is a piecewise constant function over $[0,1]$ and the expectation is taken over a Uniform($[0,1]$) random variable U. Note that (4)–(5) reflect the fact that $F(X)$ is uniformly distributed when $X \sim F$.

In other words, minimizing $\|F-h\|_\mu$ is equivalent to choosing a set of intervals $\{J_k\}$ and the associated points $\{c_k\}$ so as to minimize the expected distance between a realization of U and its corresponding value c_k. We recognize this as the problem of optimal quantization of U under the absolute error.

It is a well-established result in quantization theory that uniform quantization is optimal for Uniform($[0,1]$). Specifically, the quantization points should be equally spaced as $\{c_k = \frac{2k-1}{2K} : k = 1,\dots,K\}$, and the quantization intervals $\{J_k\}$ should split the domain $[0,1]$ evenly. The associated minimum quantization error is equal to [12, Example 5.5]

$$T_{K,r}(U) = \inf_{Q_K} \{\mathbb{E}\,[|U - Q_K(U)|^r] : Q_K \text{ has } K \text{ quantization points}\} = \frac{1}{K^r(1+r)2^r}\,;$$

this means that, in our notation,

$$R_{K,P_0}(F) = T_{K,1}(U) = \frac{1}{4K}\,. \quad (6)$$

It is interesting to note that the minimum approximation error (6) and the optimal $\{c_k\}$ and $\{J_k\}$ are the same for any F; the only dependence on F is via the regions $\{I_k\}$ that should split $\mathcal{Q}$ at the K-quantiles of F to yield equal-sized $\{J_k\}$.

We illustrate this setting graphically in Figure 5, where we show an example CDF curve $F(x)$ (blue), optimally approximated by a piecewise constant function h (red) with $K = 5$ segments. The approximation points $\{c_k\}$ are evenly spaced along the y-axis, and the mid-points between them, when reflected onto the x-axis via F^{-1}, mark the quintiles of F, which split $\mathcal{Q}$ into $K = 5$ equiprobable regions $\{I_k\}$. The "uniformization" of a distribution by its own CDF that we used in (4)–(5) is illustrated by the green dotted lines–their density below the blue CDF curve is proportional to f, whereas above the CDF curve, when reflected off it onto the y-axis, it becomes uniform.

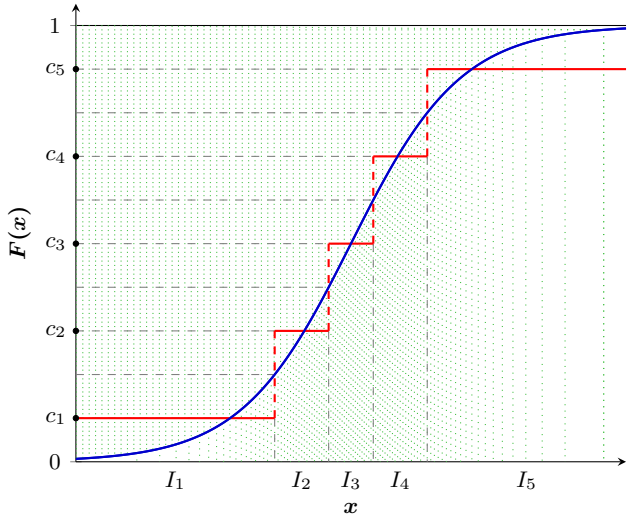


Figure 5 Optimal approximation of the CDF F (blue) via a piecewise constant function h (red) with $K = 5$ segments when $f = g$–i.e., when the dataset and queries are drawn from the same distribution. In this case, the optimal approximating values $\{c_k\}$ are evenly spaced (marked on the y-axis), and approximating the CDF is equivalent to quantizing Uniform($[0, 1]$) along the y-axis. Quantization boundaries (unmarked black horizontal dashed lines) are reflected onto the x-axis via F^{-1} to split $\mathcal{Q}$ into $K = 5$ equiprobable regions $\{I_k\}$ along the quintiles of F (black vertical dashed lines). The green dotted lines illustrate how $X \sim F$ becomes uniform when transformed via $F(X)$.

6 Lower Bounds for Piecewise Linear Models

As shown in Section 5, learned indexes based on piecewise constant approximations admit a clean lower bound analysis via quantization theory. There, the approximation error $R_{K,\mathcal{L}}(F)$ was analyzed for individual CDFs F. In this section, we extend our analysis to more expressive model classes, such as piecewise linear functions. Our central object of study will be $\sup_{F\in\mathcal{F}} R_{K,\mathcal{L}}(F)$, the worst-case approximation error over a function class $\mathcal{F}$.

By lower bounding this quantity, we can show that there exist functions within $\mathcal{F}$ which are not well approximated by any $h \in \mathcal{I}_{K,\mathcal{L}}$. In line with the rest of the paper, we focus on the class of Lipschitz continuous CDFs, and show that for $\mathcal{L} = P_1$, the lower bound $\Omega(1/K)$ still holds in this worst-case sense, matching classical upper bounds (for a recap of these upper bounds, see Appendix G.1 in the extended version of the paper [3]).

We establish this result through two complementary approaches. In Section 6.1, we describe an explicit family of Lipschitz continuous CDFs $\{F_M\}$ for which $R_{K,P_1}(F) = \Omega(1/K)$. The construction works for a broad class of measures μ, and shows that the class $\mathcal{I}_{K,P_1}$ cannot guarantee $o(1/K)$ approximation error uniformly across all Lipschitz continuous functions.

In Section 6.2, we approach the problem through *Kolmogorov widths*, an important tool from approximation theory. This approach typically assumes μ as the Lebesgue measure, and only allows for existential (rather than constructive) proofs. Nonetheless, it is a very flexible framework: it allows us to analyze broader model classes beyond P_1 and to derive sharper bounds for more regular function spaces. Together, these two methods highlight complementary aspects of the approximation limits faced by learned index structures.

6.1 An Adversarial CDF Construction

To illustrate the limitations of piecewise linear approximators, we construct a family of CDFs $\{F_M\}_{M\in\mathbb{N}}$ with the following property: for any $M > (1+\alpha)K$ with $\alpha > 0$, the approximation error satisfies $R_{K,P_1}(F_M) = \Omega(1/K)$. The construction, detailed in Appendix B, is based on stitching together rescaled copies of the quadratic function x^2 over M disjoint intervals. While this is presented under the assumption that μ is the uniform distribution, the argument naturally extends to a broader class: specifically, any measure μ with a density g satisfying $g(q) \geq \lambda > 0$ over an interval. In such cases, the lower bound continues to hold up to constants depending on λ.

Thus, this construction provides a worst-case lower bound for $R_{K,P_1}(F)$ within the class of Lipschitz continuous CDFs. It also highlights a fundamental aspect of learned indexes: the approximation error depends on both the location and the number of difficult-to-approximate regions. Hence, the choice of partition and local model must be adapted to the data. Fixing the model class and budget K in advance cannot guarantee uniformly good approximation.

6.2 Kolmogorov Widths and Approximation Lower Bounds

To further understand the limitations of learned index models beyond explicit constructions, we turn to a more general framework from approximation theory: Kolmogorov widths. This concept provides a way to quantify how well an entire function class $\mathcal{F}$ can be approximated by models of bounded complexity. Formally, the Kolmogorov width $d_\kappa(\mathcal{F})$ captures the best possible worst-case approximation error over $\mathcal{F}$:

$$d_\kappa(\mathcal{F}) = \inf_{H\in\mathcal{H}_\kappa} \sup_{F\in\mathcal{F}} \inf_{h\in H} \|F - h\|_\mu \,,$$

where the outermost infimum is taken over the set $\mathcal{H}_\kappa$ of all linear subspaces of dimension κ.

We relate $d_\kappa(\mathcal{F})$ to our context in the following way. Throughout, we have considered approximation by functions in $\mathcal{I}_{K,\mathcal{L}}$. This class is not necessarily a linear subspace of $L^1(\mu)$, since the sum of two K-piecewise functions is not necessarily K-piecewise. Nonetheless, if we fix a partition $\mathcal{P} = \{I_k\}_{k=1}^K$ of $\mathcal{Q}$, the subclass

$$\mathcal{I}_{\mathcal{P},\mathcal{L}} = \left\{h : h(q) = \textstyle\sum_k h_k(q)\mathbf{1}_{I_k}(q),\ h_k \in \mathcal{L}\right\}$$

does form a linear subspace, provided that $\mathcal{L}$ is itself a linear subspace of $L^1(\mu)$. In such cases–e.g., when $\mathcal{L}$ consists of polynomials of bounded degree–any function class $\mathcal{I}_{\mathcal{P},\mathcal{L}}$ has dimension at most $\kappa = K \cdot \dim(\mathcal{L})$. This characterization leads to the following inequality.

▶ **Lemma 6.** *For any $\mathcal{F}, \mathcal{L} \subseteq L^1(\mu)$ where $\mathcal{L}$ is a linear subspace of finite dimension, it holds*

$$\sup_{F\in\mathcal{F}} R_{K,\mathcal{L}}(F) \geq d_\kappa(\mathcal{F}), \qquad \textit{where } \kappa = K \cdot \dim(\mathcal{L})\,.$$

The proof can be found in Appendix G of the extended version of the paper [3]. We now focus on the case where $\mathcal{F}$ is the class of Lipschitz continuous functions, denoted by $W^{1,\infty}$. This notation comes from Sobolev space theory, which provides a convenient formalism for characterizing function smoothness, and within which most Kolmogorov width results are typically stated. When μ is the uniform distribution over a compact interval $[a, b]$, it holds that (see [20, Chapter 14, Theorem 3.8]):

$$d_\kappa(W^{1,\infty}) = \Omega(1/\kappa)\,. \tag{7}$$

Notice that for $\mathcal{L} = P_m$, the class of degree-m polynomials, we have $\kappa = (m+1)K$, since each piece has $m+1$ parameters. Piecewise linear models correspond to $m = 1$ and thus $\kappa = 2K$. Hence, for fixed m, Lemma 6 and equation (7) imply $\sup_{F \in W^{1,\infty}} R_{K,P_m}(F) = \Omega(1/K)$.

This has a direct implication for learned index structures: even the best possible model using K adaptive segments and bounded-degree polynomial predictors cannot achieve approximation error $o(1/K)$ for general Lipschitz continuous CDFs.[2]

These results match the $\Omega(1/K)$ result obtained in Section 6.1. While the explicit construction from Section 6.1 can accommodate a wide class of distributions μ, including non-uniform densities, the Kolmogorov width approach offers a wider perspective in regards to the approximating functions. In particular, it applies to any piecewise polynomial class $\mathcal{I}_{K,P_m}$ and, more generally, to any linear span of κ functions, such as truncated Fourier series.

The Kolmogorov width framework makes it explicit that the $\Omega(1/K)$ bound is inherently a worst-case result. For smoother subclasses $\mathcal{F} \subseteq W^{1,\infty}$, the theory yields sharper rates: for example, for $W^{2,\infty}$ (i.e., functions with Lipschitz continuous derivatives), one obtains $d_\kappa(W^{2,\infty}) = \Theta(1/\kappa^2)$. This highlights how stronger regularity assumptions can enable faster approximation. In contrast, the lower bound for piecewise constant models (Section 5) applies uniformly to all Lipschitz continuous CDFs and does not improve with additional regularity.

Concluding Remarks. A key takeaway from this section is that for the class $\mathcal{F} = W^{1,\infty}$ of Lipschitz continuous functions, the worst-case approximation error under learned index models in the class $\mathcal{I}_{K,P_m}$ satisfies $\Omega(1/K)$ for any fixed m. This resembles the $\Omega(1/K)$ rate derived in Section 5 for piecewise constant approximation, but here it holds only in a worst-case sense over $\mathcal{F}$, not uniformly for all $F \in \mathcal{F}$. In particular, the result does not extend to smoother subclasses such as $W^{2,\infty}$, where approximation can be better, but it still constitutes a meaningful lower bound for the general performance of learned indexes.

7 Discussion and Conclusions

This paper establishes an analytical framework for lower bounding query time using learned index structures (Sections 3–4). We consider two classes of approximating functions–piecewise constant and piecewise linear approximators–that are most commonly used in practice, and relate query time to the error of approximating the CDF of the data-generating distribution. For piecewise constant functions, we show that the approximation error decreases linearly with the number of segments K for a broad class of CDFs (Section 5). For piecewise linear approximators, the situation is more subtle: we show that the approximation error depends on the regularity of the CDF, and that in general using a richer approximating model does not guarantee an improved lower bound on the approximation error (Section 6). However, under additional assumptions on the smoothness of the data-generating CDF, the bound on the approximation error can be lowered further.

More specifically, Sections 5 and 6 establish a lower bound of $\Omega(1/K)$ on approximation error that, in general, cannot be improved for the class of Lipschitz continuous CDFs. For piecewise constant models, this holds uniformly (Section 5); for piecewise linear models, the

[2] This conclusion requires some care. While the Kolmogorov width bound is stated for the worst-case over the class $W^{1,\infty}$, it is not immediate that the worst-case functions realizing the $\Omega(1/\kappa)$ bound include any CDFs. In principle, one might worry that these worst-case functions do not include any valid CDFs. However, this is not the case. In the extended version of the paper [3], we provide a rigorous argument showing that any hard-to-approximate function $F \in W^{1,\infty}$ can be mapped to a CDF that remains equally hard to approximate.

bound applies in a worst-case sense over the choice of CDF (Section 6). In either case, the $\Omega(1/K)$ behavior governs the worst-case regime. Combining these results with the analysis summarized in Table 1 (e.g., see rows E2 and B2) we obtain an $\Omega(\log(n/K))$ lower bound on worst-case query time. As discussed in Section 2, this implies that the number of segments K (and therefore, space usage) must be nearly linear in n to yield an asymptotic advantage over traditional methods, a conclusion that holds for a broad class of learned indexes.

It is instructive to compare our bounds with the information-theoretic results in [31]. Their worst-case bounds require a finite query domain $\mathcal{Q}$, a restriction not present in our framework. Moreover, our analysis yields sharper and more applicable insights for typical learned index designs, particularly those based on piecewise constant or linear models. In the average-case setting, their results imply that achieving expected error η requires $\log(1/\eta)$ bits of model capacity. In contrast, our bounds–under independence assumptions–show a $1/\eta$ scaling, which is larger when $\eta \to 0$. Additionally, when the query and data distributions are the same–i.e., $d\mu = dF$–we show that the number of parameters can scale as $\Omega(n)$, compared to $\Omega(\sqrt{n})$ in their framework.

Aside from their theoretical importance, our results offer some practical insights for the design of learned indexes: first, they show that the number of segments K should not be chosen without regard to the underlying data, as there exist "adversarial" distributions for which the approximation error can be poor for a given K (see Section 6.1). Second, the results of Section 6.2 show that approximation capacity depends not only on the expressiveness of the model class but also on the smoothness of the CDF. This suggests that selecting the model class in a data-dependent manner–adapted to the regularity of the CDF–could lead to more efficient index designs, an idea that warrants further exploration.

We identify several open directions for future research. On the one hand, more general assumptions on the data-generating distribution can be explored, either by relaxing the independence assumptions or by considering wider classes of CDFs. Related, a taxonomy of rank functions could be developed by identifying meaningful classes for which learned indexing is provably easy or hard. An additional open problem is to extend average-case analysis to binary search and to richer approximation classes.

Another important direction is to extend our analysis to dynamic workloads. Many updatable index designs rely on additional algorithmic mechanisms, such as gaps, to support insertions. Including such features in our framework would enable the study of learned indexes (and their associated space–time trade-offs) under updates. We hope this work offers some of the foundational tools needed to explore these ideas further.

References

1 Alfred V. Aho and John E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1974.

2 Albert Atserias, Martin Grohe, and Dániel Marx. Size bounds and query plans for relational joins. *SIAM Journal on Computing*, 42(4):1737–1767, 2013. doi:10.1137/110859440.

3 Luis Alberto Croquevielle, Roman Sokolovskii, and Thomas Heinis. Lower bounds for the algorithmic complexity of learned indexes. *arXiv preprint*, 2026. arXiv:2601.06629.

4 Luis Alberto Croquevielle, Guang Yang, Liang Liang, Ali Hadian, and Thomas Heinis. Beyond Logarithmic Bounds: Querying in Constant Expected Time with Learned Indexes. In *28th International Conference on Database Theory (ICDT 2025)*, volume 328 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:21, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICDT.2025.19.

5 Jialin Ding, Umar Farooq Minhas, Jia Yu, Chi Wang, Jaeyoung Do, Yinan Li, Hantian Zhang, Badrish Chandramouli, Johannes Gehrke, Donald Kossmann, David Lomet, and Tim Kraska. Alex: An updatable adaptive learned index. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pages 969–984, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3318464.3389711.

6 Jialin Ding, Vikram Nathan, Mohammad Alizadeh, and Tim Kraska. Tsunami: a learned multi-dimensional index for correlated data and skewed workloads. *Proc. VLDB Endow.*, 14(2):74–86, 2020. doi:10.14778/3425879.3425880.

7 Paolo Ferragina, Fabrizio Lillo, and Giorgio Vinciguerra. Why are learned indexes so effective? In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3123–3132, Virtual, 13–18 July 2020. PMLR. URL: http://proceedings.mlr.press/v119/ferragina20a.html.

8 Paolo Ferragina and Giorgio Vinciguerra. The pgm-index: A fully-dynamic compressed learned index with provable worst-case bounds. *Proc. VLDB Endow.*, 13(8):1162–1175, April 2020. doi:10.14778/3389133.3389135.

9 R. A. Finkel and J. L. Bentley. Quad trees A data structure for retrieval on composite keys. *Acta Inf.*, 4(1):1–9, 1974. doi:10.1007/BF00288933.

10 Alex Galakatos, Michael Markovitch, Carsten Binnig, Rodrigo Fonseca, and Tim Kraska. Fiting-tree: A data-aware index structure. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, pages 1189–1206, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3299869.3319860.

11 Goetz Graefe et al. Modern B-tree techniques. *Foundations and Trends in Databases*, 3(4):203–402, 2011. doi:10.1561/1900000028.

12 Siegfried Graf and Harald Luschgy. *Foundations of quantization for probability distributions*. Springer Science & Business Media, 2000.

13 Antonin Guttman. R-trees: a dynamic index structure for spatial searching. In *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, pages 47–57, New York, NY, USA, 1984. Association for Computing Machinery. doi:10.1145/602259.602266.

14 Ali Hadian and Thomas Heinis. Shift-table: A low-latency learned index for range queries using model correction. In *Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021*, pages 253–264. OpenProceedings.org, 2021. doi:10.5441/002/edbt.2021.23.

15 Andreas Kipf, Ryan Marcus, Alexander van Renen, Mihail Stoian, Alfons Kemper, Tim Kraska, and Thomas Neumann. Radixspline: A single-pass learned index. In *Proceedings of the Third International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, aiDM '20, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3401071.3401659.

16 Paraschos Koutris, Shaleen Deep, Austen Fan, and Hangdong Zhao. The Quest for Faster Join Algorithms. In *28th International Conference on Database Theory (ICDT 2025)*, volume 328 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:12, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICDT.2025.1.

17 Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 489–504, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3183713.3196909.

18 Pengfei Li, Hua Lu, Qian Zheng, Long Yang, and Gang Pan. Lisa: A learned index structure for spatial data. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pages 2119–2133, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3318464.3389703.

19 Liang Liang, Guang Yang, Ali Hadian, Luis Alberto Croquevielle, and Thomas Heinis. Swix: A memory-efficient sliding window learned index. *Proc. ACM Manag. Data*, 2(1), 2024. doi:10.1145/3639296.

20 G. Lorentz, George and Manfred V. Golitschek. *Constructive approximation : advanced problems*. Springer-Verlag, Berlin, 1996.

21 Ryan Marcus, Andreas Kipf, Alexander van Renen, Mihail Stoian, Sanchit Misra, Alfons Kemper, Thomas Neumann, and Tim Kraska. Benchmarking learned indexes. *Proc. VLDB Endow.*, 14(1):1–13, September 2020. doi:10.14778/3421424.3421425.

22 Vikram Nathan, Jialin Ding, Mohammad Alizadeh, and Tim Kraska. Learning multi-dimensional indexes. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pages 985–1000, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3318464.3380579.

23 Theodore J Rivlin. *An introduction to the approximation of functions*. Dover, New York, 1981.

24 Dan Suciu. Applications of information inequalities to database theory problems. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–30. IEEE, 2023. doi:10.1109/LICS56636.2023.10175769.

25 Zhaoyan Sun, Xuanhe Zhou, and Guoliang Li. Learned index: A comprehensive experimental evaluation. *Proc. VLDB Endow.*, 16(8):1992–2004, 2023. doi:10.14778/3594512.3594528.

26 Jiacheng Wu, Yong Zhang, Shimin Chen, Jin Wang, Yu Chen, and Chunxiao Xing. Updatable learned index with precise positions. *Proc. VLDB Endow.*, 14(8):1276–1288, 2021. doi:10.14778/3457390.3457393.

27 Guang Yang, Liang Liang, Ali Hadian, and Thomas Heinis. FLIRT: A fast learned index for rolling time frames. In *Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023*, pages 234–246, Ioannina, Greece, 2023. OpenProceedings.org. doi:10.48786/edbt.2023.19.

28 Mihalis Yannakakis. Algorithms for acyclic database schemes. In *Proceedings of the Seventh International Conference on Very Large Data Bases - Volume 7*, VLDB '81, pages 82–94. VLDB Endowment, 1981.

29 Sepanta Zeighami and Cyrus Shahabi. On distribution dependent sub-logarithmic query time of learned indexing. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

30 Sepanta Zeighami and Cyrus Shahabi. Theoretical analysis of learned database operations under distribution shift through distribution learnability. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

31 Sepanta Zeighami and Cyrus Shahabi. Towards establishing guaranteed error for learned database operations. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL: https://openreview.net/forum?id=6tqgL8VluV.

A Piecewise Constant Approximation: Mismatched Dataset and Query Distributions

In contrast to matching dataset and query distributions $f = g$ considered in Section 5.1, the case $f \neq g$ is relatively more involved; here too, however, the problem of approximating F can be reduced to optimal quantization. The only difference is that the quantized random variable is no longer uniformly distributed, and our results hold asymptotically for $K \to \infty$ as opposed to any given K as in Section 5.1.

The results in this section assume that F is invertible. This is a relatively mild condition, which holds for most standard examples of continuous distributions. Moreover, this assumption can be relaxed: if F is only invertible on a subinterval $[a, b] \subseteq \mathcal{Q}$, the analysis still

applies by conditioning the query distribution to $[a, b]$. This affects only the scaling constants, through changes in the conditional query distribution, but preserves the asymptotic behavior in K. For clarity of exposition, we focus on the case where F is invertible over all of $\mathcal{Q}$.

▶ **Proposition 7** (Reduction to Quantization). *Let $h \in \mathcal{I}_{K,P_0}$ and $Y = F(Q)$ with $Q \sim \mu$ and $d\mu = g\,dq$. Then*

$$\|F - h\|_{\mu,r} \stackrel{\text{def}}{=\!=} \int_{\mathcal{Q}} |F(q) - h(q)|^r g(q) dq = \mathbb{E}[|Y - Q_K(Y)|^r]\,.$$

Proof. As in (4), we expand $\|F - h\|_{\mu,r}$ as

$$\|F - h\|_{\mu,r} = \sum_{k=1}^{K} \int_{I_k} |F(q) - c_k|^r g(q) dq\,.$$

Introducing $y = F(q),\ dy = f(q)dq,\ q = F^{-1}(y)$, and $dq = 1/f(q)\,dy$, we rewrite this as

$$\sum_{k=1}^{K} \int_{J_k} |y - c_k|^r \cdot \frac{g\left(F^{-1}(y)\right)}{f\left(F^{-1}(y)\right)} dy = \mathbb{E}[|Y - Q_K(Y)|^r]\,,$$

where the last equality follows because the second term in the integral is the probability density function (PDF) of $Y = F(Q)$ when Q distributes according to the PDF g:

$$f_Y(y) = g\left(F^{-1}(y)\right) \cdot \left|\frac{d}{dy} F^{-1}(y)\right| = g\left(F^{-1}(y)\right) \cdot \left|\frac{1}{F'\left(F^{-1}(y)\right)}\right| = \frac{g\left(F^{-1}(y)\right)}{f\left(F^{-1}(y)\right)}\,; \tag{8}$$

the first equality expands the density of a function of a random variable, and the second the derivative of an inverse function. We remark that this requires the CDF F to be invertible. ◀

Proposition 7 implies that approximating F reduces to quantizing a random variable $Y = F(Q)$ distributed according to (8) over $[0, 1]$. When $f \neq g$, this distribution is no longer uniform, and uniform quantization is no longer optimal. To carry on with our visual analogy in Figure 5, one can imagine the density of the green dashed lines below F no longer proportional to f, and therefore the ones above F no longer uniform, in which case the approximating values $\{c_k\}$ would have to be shifted accordingly.

We are now in a position to leverage an important result in quantization theory. The asymptotic quantization error is known to satisfy [12, Theorem 6.2]

$$\lim_{K\to\infty} K^r\ T_{K,r}(Y) = \frac{1}{(1+r)2^r} \left(\int f_Y(y)^{1/(1+r)} dy\right)^{1+r}\,.$$

The right-hand side of the equality is known as the r-th quantization coefficient and is a fundamental property of a probability distribution, similar to its moments; quantization coefficients play an important role in quantization theory. Therefore, for the absolute error

$$\lim_{K\to\infty} K\ R_{K,P_0}(F) = \lim_{K\to\infty} K\ T_{K,1}(Y) = \frac{1}{4} \left(\int \sqrt{f_Y(y)} dy\right)^2\,.$$

In other words, asymptotically, the absolute error decreases linearly with K, just like in the case of the matching distribution considered in Section 5.1, yielding $R_{K,\mathcal{L}}(F) = \Omega\left(1/K\right)$ when $\mathcal{L} = P_0$.

B Construction of Adversarial CDF with Respect to K

We illustrate the case $\mathcal{Q} = [0, 1]$, with μ being the uniform distribution. This will allow us to perform some exact calculations, but the asymptotic properties we derive in this section, and the argument we use to prove them, apply for very general choices of $\mathcal{Q}$ and μ. Consider the function $F(x) = x^2$ and the optimal approximation error $R_{1,P_1}(F)$ by a single linear function. Notice that x^2 over $[0, 1]$ is a valid CDF. From [23, Corollary 3.4.1], it is easy to deduce that

$$R_{1,P_1}(F) = 1/16\,, \tag{9}$$

which is attained by $h(x) = x - 3/16$. Now, take a positive integer M. The intuition is that we can construct a staircase-type function F_M, with M total steps and each step F_M^i being equal to $F(x) = x^2$ after appropriate translation and scaling, such that the full F_M constitutes a valid CDF and is hard to approximate. For any $i = 0, \ldots, M-1$ define F_M^i as:

$$F_M^i(x) = \frac{1}{M}\left[M\left(x - \frac{i}{M}\right)\right]^2 + \frac{i}{M} \quad \text{for} \quad x \in I_i = \left[\frac{i}{M}, \frac{i+1}{M}\right].$$

Now, the full F_M is defined as

$$F_M(x) = \sum_{i=0}^{M-1} F_M^i(x)\mathbf{1}_{I_i}(x) \quad \text{for} \quad x \in [0, 1]\,.$$

It is easy to check that F_M is continuous, increasing, and satisfies $F(0) = 0$, $F(1) = 1$, thus being a valid CDF on $[0, 1]$. Consider now any linear approximation $L_i \in P_1$ for the function F_M^i over the interval I_i. The approximation error is then given by:

$$E_i = \int_{\frac{i}{M}}^{\frac{i+1}{M}} |F_M^i(x) - L_i(x)|dx = \int_{\frac{i}{M}}^{\frac{i+1}{M}} \left|\frac{1}{M}\left[M\left(x - \frac{i}{M}\right)\right]^2 + \frac{i}{M} - L_i(x)\right| dx\,.$$

Substituting $u = M(x - i/M),\ du = Mdx$:

$$= \frac{1}{M}\int_0^1 \left|\frac{u^2}{M} + \frac{i}{M} - L_i\left(\frac{u+i}{M}\right)\right| du = \frac{1}{M^2}\int_0^1 \left|u^2 + i - ML_i\left(\frac{u+i}{M}\right)\right| du\,.$$

We focus now on the integral term. Since $i - ML_i((u+1)/M) \in P_1$, it cannot induce lower error when approximating u^2 than the optimal approximator $h(u) = u - 3/16$, hence by equation (9):

$$E_i \geq \frac{1}{16M^2}\,. \tag{10}$$

We now use this to derive a lower bound on $R_{K,P_1}(F_M)$. Since F_M consists of M disjoint quadratic steps, each over an interval of length $1/M$, a K-piecewise linear function can assign at most K linear segments across all M sub-parabolas. Therefore, at least $M - K + 1$ sub-parabolas must be covered using only one linear piece each, and on each corresponding interval, bound (10) applies. Summing over these $M - K + 1$ intervals, we obtain

$$R_{K,P_1}(F_M) \geq \frac{M - K + 1}{16M^2}\,.$$

This expression is maximized when $M = 2(K - 1)$, yielding

$$R_{K,P_1}(F_{2(K-1)}) \geq \frac{1}{64(K-1)} = \Omega\left(\frac{1}{K}\right).$$

This construction shows that even when F is a valid CDF, it is possible to exhibit $\Omega(1/K)$ lower bounds for approximation using K-piecewise linear functions. In particular, the worst-case behavior of $R_{K,P_1}(F)$ is no better than that of piecewise constant models, despite the greater expressive power of linear segments. This highlights the critical role of regularity assumptions on F: without sufficient smoothness, more expressive models do not necessarily yield improved approximation rates.