A Practical Theory of Generalization in Selectivity Learning

Peizhi Wu
University of Pennsylvania
pagewu@cis.upenn.edu

Haoshu Xu
University of Pennsylvania
haoshuxu@sas.upenn.edu

Ryan Marcus
University of Pennsylvania
rcmarcus@cis.upenn.edu

Zachary G. Ives
University of Pennsylvania
zives@cis.upenn.edu

ABSTRACT

Query-driven machine learning models have emerged as a promising estimation technique for query selectivities. Yet, surprisingly little is known about the efficacy of these techniques from a theoretical perspective, as there exist substantial gaps between practical solutions and state-of-the-art (SOTA) theory based on the Probably Approximately Correct (PAC) learning framework. In this paper, we aim to bridge the gaps between theory and practice. First, we demonstrate that selectivity predictors induced by *signed measures* are learnable, which relaxes the reliance on *probability measures* in SOTA theory. More importantly, beyond the PAC learning framework (which only allows us to characterize how the model behaves when both training and test workloads are drawn from the *same* distribution), we establish, under mild assumptions, that selectivity predictors from this class exhibit favorable *out-of-distribution* (OOD) generalization error bounds.

These theoretical advances provide us with a better understanding of both the in-distribution and OOD generalization capabilities of query-driven selectivity learning, and facilitate the design of two general strategies to improve OOD generalization for existing query-driven selectivity models. We empirically verify that our techniques help query-driven selectivity models generalize significantly better to OOD queries both in terms of prediction accuracy and query latency performance, while maintaining their superior in-distribution generalization performance.

1 INTRODUCTION

We study the learning of selectivity functions for selection queries in database management systems (DBMSes). As the key to effective query optimization, selectivity estimation has continued to be one of the most important problems in DBMSes since the 1980s [36, 47]. The earliest approach was to collect basic statistics (such as histograms) for selectivity estimation, and then to make uniformity (within a bucket) and independence (among columns) assumptions.

Although widely adopted in real DBMSes due to its simplicity, this approach is prone to large estimation errors [26, 32].

More recently, selectivity estimation has been formulated as a machine learning (ML) problem, where the system learns from observed samples (data or queries) to make selectivity predictions for incoming queries. Proposals for learning-based selectivity estimation can be broadly categorized into *data-driven* and *query-driven* models (with a few exceptions in the form of hybrid models). Data-driven techniques [15, 22, 24, 30, 39, 56, 58, 60] build models of the data distribution by scanning the underlying data. Conversely, query-driven techniques either learn a regression model from query features to selectivity [13, 31], or model the data distribution from a set of observed queries and their selectivities [4].

In this paper, we focus on query-driven models [19, 25, 31, 33, 43, 45, 51] as they enjoy a smaller model size, faster training, and possibly faster inference (for example, regression models [19, 31, 33]) compared to data-driven models. In addition, they can also achieve much better performance than traditional histograms [31].
Importance of theoretical understanding of generalization. In machine learning, *generalization* ("a central goal in pattern recognition [10]") refers to a model's ability to perform well on new, unseen data that was not part of the training set. With respect to query-driven selectivity learning, the large variability in queries seen in practice means that any training workload can represent only a *tiny* subset of all possible queries. Therefore, it is crucial to accurately characterize the generalization ability of selectivity models, specifically how they perform on queries that were *not* seen during training. This understanding is essential to ensure reliable predictions in real-world applications. Yet, surprisingly, there is limited theoretical analysis of the generalizability of query-driven models. An initial and promising step towards such understanding [25] proves that selectivity functions are learnable using the Probably Approximately Correct (PAC) learning framework [28]. However, significant gaps remain in our understanding.
Limitations of prior results. The current SOTA result [25] assumes that *every selectivity predictor in the hypothesis class is induced by a probability measure*. Consequently, *learnability* (in-distribution generalization, to be formally introduced in § 3.4) results can be applied *only* to a small fraction of existing query-driven models (*e.g.,* those that build histograms from queries [4]). Indeed, as we will see later in the paper, predictors from regression-based query-driven models, which achieve impressive empirical performance, are not induced by a probability measure. Therefore, existing learnability results [25] cannot be applied to these practical approaches. Given this gap between theory and practice, a natural question arises:

> *Question 1: Is it feasible to reduce the reliance on probability measures, thereby broadening our theoretical understanding of selectivity learning models?*

Another challenge in applying the theoretical results to practical scenarios is that PAC learning, as a framework, *only* allows us

to quantify the *in-distribution* generalization error, where both training and test queries are drawn from the *same* distribution. This means that previous theory [25] based on PAC learning is *not* able to characterize generalization error for OOD scenarios. Nevertheless, in the real world, query workloads may shift constantly [42, 55]. This raises another, perhaps more challenging, question:

> **Question 2:** *Given mild assumptions, is it feasible to quantify* **OOD generalization error** *in selectivity learning, thereby enhancing the practical relevance of theoretical results?*

Our first goal in this paper is to answer these two questions theoretically. Thereafter, based on the new generalization results, we design new learning paradigms/frameworks for improving selectivity estimation in practice, which leverage the theoretical results to provide formal guarantees.

A sketch of our results. The paper delivers two *positive and encouraging* theoretical results toward answering the two questions:

- Addressing Question 1, we introduce a new theoretical result of learnability (*i.e.,* in-distribution generalization) that applies to selectivity functions/models whose predictions are induced by a *signed measure*, removing the positivity and sum-to-unity constraints that are required by prior work.
- More interestingly, under mild assumptions, we establish *nontrivial* OOD generalization error bound for selectivity predictors that are induced by a signed measure. The new result, *beyond the PAC learning framework*, quantifies the generalization error when training and testing workloads do not follow the same distribution, hence answering Question 2. For a taste of our theory, our main theorem (Theorem 4.2) is simplified below.

> **Simplified Theorem 4.2.** *For any selectivity estimator $\hat{S}$ that is induced by a signed measure, if $\hat{S}$ is trained under distribution Q with in-distribution generalization error $\text{er}_Q(\hat{S})$ upper bounded by ϵ with probability at least $1 - \delta$, then under a different testing distribution P, the out-of-distribution generalization error $\text{er}_P(\hat{S})$ satisfies*
>
> $$\text{er}_P(\hat{S}) \leq O(\sqrt{\epsilon})$$
>
> *with probability at least $1 - \delta$, under mild assumptions on distribution P and Q (see Theorem 4.2 for details).*

A key implication of our result is that, for any class of selectivity predictors that is induced by signed measures, both our in-distribution and OOD generalization results apply *immediately*.

Improvement strategies inspired by our theory. From this aspect of our theory, we propose novel and practical methodologies for *improving existing query-driven selectivity learning models*.

- We propose a new modeling paradigm for query-driven selectivity learning, NEUROCDF, which models the underlying cumulative distribution functions (CDFs) using a neural network. NEUROCDF is proved to be induced by signed measures, and thus enjoys the theoretical guarantees of our theory, and enjoys the superior empirical performance of deep learning. Although challenging to optimize with relative error metrics like Qerror, NEUROCDF *provably* offers better generalization performance for OOD queries, compared to the common paradigm for selectivity estimation that targets the query selectivity directly.

- Inspired by the lessons learned from our theory and NEUROCDF, we propose a general training methodology for enhancing existing query-driven selectivity models. SECONCDF incorporates the idea of CDF modeling of NEUROCDF into query-driven models by enforcing model **Se**lf-**Con**sistency with the learned **C**umulative **D**istribution **F**unctions. However, unlike NEUROCDF, SECONCDF keeps the original loss functions (Qerror or RMSE) of existing query-driven models, which allows for good in-distribution generalization with either relative or absolute loss functions. Moreover, the CDF self-consistency training of SECONCDF significantly enhances model OOD generalization ability.

Takeaways from the experiments. The proposed improvement strategies are *orthogonal* to selectivity model architectures, making them applicable to various existing models. Our primary goal is *not* to outperform current SOTA query-driven selectivity learning models, but to validate the practicality of our theory by designing algorithms that improve the OOD generalization capabilities of existing models *with theoretical guarantees*. Thus, we focus our experimental evaluation on aspects in which our strategies are expected to provide improvements. Indeed, this focused approach has yielded clear, compelling results: across both single- and multitable datasets, our strategies can *significantly* improve the OOD generalization of existing selectivity learning models, in terms of both estimation accuracy (*i.e.,* smaller Qerror and RMSE) and query running time performance (*i.e.,* lower query latency).

Organization. § 2 reviews prior work. § 3 outlines preliminaries and the problem setup. In § 4, we introduce our new theory, followed by two improvement strategies in § 5 and § 6. Our algorithms are evaluated in § 7, and we conclude in § 8.

2 PRIOR WORK

Selectivity estimation dates back to the beginning of query processing [47], where rather than computing intermediate results and then finding query plans [52], System-R instead used histograms and independence assumptions. Such techniques were refined to use queries themselves to compute histograms [4, 12, 34], query expression statistics [11] and adjustments to correlated predicates [38]. More recent *learned data-driven methods* [24, 58] do offline computation over samples of existing database instances to build models of data distributions in the presence of skew and correlations. *Learned query-driven database systems* can learn or improve an ML model for a variety of database components, by using the execution log of a query workload [6–8, 33, 57]. More recently, there is active work on workload-aware cardinality predictors [31, 53, 54]. In this paper, we consider several families of selectivity estimation techniques.

Parametric Functions [13]. The early approach fits a parametric function (*e.g.,* linear and polynomial) to observed queries. These functions take a query as input and produce a selectivity estimate. However, the performance of parametric functions is not as good as more recent approaches due to the limited model capacity.

Histograms [4, 37]. Histogram-based models, widely studied in database literature, build histograms from query workloads by adjusting bucket frequencies to correct prior errors or by aligning with a maximum entropy distribution with observed queries. They assume uniformity within buckets and independence across columns (or features), which could lead to large estimation errors.

LEO [49]. Intuitively, LEO can be seen as a combination of parametric functions and histograms — it learns the adjustment factors from observed queries to correct incorrect statistics such as histograms. Specifically, LEO collects a set of previous ratios $r = \frac{act\_sel}{stat\_set}$ of actual selectivity (act_set) and statistics estimate ($stat\_set$) from past queries. To estimate an incoming query, LEO uses the ratios to adjust the statistics estimate by multiplying it by a chosen adjustment ratio r. For example, consider a query asks for the range $\{x < 1\}$ and the selectivity estimate of the histogram for the query is $\hat{Hist}(x < 1)$. LEO produces the adjusted estimate by $adjusted\_sel = \hat{Hist}(x < 1) * r(x < 1)$, where $r(x < 1)$ is the collected adjustment factor at $x = 1$. If there is no adjustment factor for $x = 1$, LEO computes the factor by linear interpolation.

Deep Learning Models [19, 31]. More recently, deep learning models have been proposed to learn the mapping from a query to its selectivity prediction. Deep learning models function as regression models in a way that is similar to parametric functions but has a larger model capacity and much better performance.

3 PRELIMINARIES AND PROBLEM SETUP

In this section, we define key concepts for selectivity estimation in § 3.1. We introduce measure theory in § 3.2 as it will shape our theory. We frame selectivity estimation as an ML problem in § 3.3, discuss the PAC learning framework in § 3.4, and review SOTA theoretical results in § 3.5. The section concludes with the gap between theory and practice, motivating the goals of this paper.

3.1 Selectivity Functions of Range Queries

Range Space. Consider a d-dimensional dataset D. A range space is defined as $\Sigma = (X, \mathcal{R})$. X is a set of objects (*e.g.,* tuples or data points in D). $\mathcal{R}$ is a collection of ranges R, which is *a subset of X*. For instance, $\mathcal{R}$ can be a set of all d-dimensional hyper-rectangles.

Range Queries. A range query q is defined as a query that retrieves tuples within the range R_q. Thus range query q and its querying range R_q are interchangeable. We focus on range selection queries, corresponding to d-*dimensional hyper-rectangles.* Join queries can be viewed as range selection queries over the join result.

Selectivity (Cardinality) Functions. For a dataset D, let P_D be the data probability distribution over D, we define the selectivity functions as $S_D(R) = P_{x \sim P_D}(x \in R)$, or equivalently,

$$S_D(R) = \sum_{x \in R} P_D(x) \tag{1}$$

Another related term is cardinality $C_D(R)$ (the output size of a query), *i.e.,* $C_D(R) = S_D(R) \cdot |T|$ where $|T|$ is the size of table T.

3.2 Measure Theory

Basic Concepts. We first formally introduce fundamental notations from measure theory that will be used to shape our theorems. A σ-algebra $\mathcal{M}$ of "measurable" sets is a non-empty collection of subsets of X closed under complements and countable unions and intersections. For all practical applications, it holds that $\mathcal{M} \supset \mathcal{R}$.

A function $\mu : \mathcal{M} \to \mathbb{R}$ is a *probability measure* on $(X, \mathcal{M})$ if it satisfies

C1. Countable additivity: if $E_1, E_2, \dots$ is a countable family of disjoint sets in $\mathcal{M}$, then $\mu\left(\bigcup_{n=1}^{\infty} E_n\right) = \sum_{n=1}^{\infty} \mu(E_n)$.
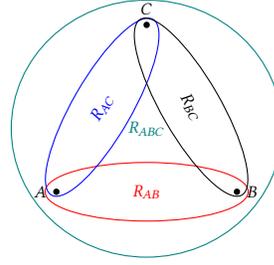
Figure 1: Left: three data points (A, B, C) and four ranges. Right Top: predictions from selectivity functions ($S_1 \sim S_4$) for the four ranges. Right Bottom: corresponding measures ($\mu_1 \sim \mu_4$) that induce $S_1 \sim S_4$, including their measure values on A, B, C (technically, the three sets $\{A\}, \{B\}, \{C\}$).

C2. Positivity: $\mu(E) \geq 0$ for any $E \in \mathcal{M}$.
C3. Sum to unity: $\mu(X) = 1$.

If μ only satisfies **C1** and **C2**, it is called a *measure*; if it only satisfies **C1**, then it is a *signed measure*. A signed measure is essentially the difference between two measures.

We now define ***induction*** for selectivity functions using measure theory. A selectivity estimate $\hat{S} : \mathcal{R} \to \mathbb{R}$ is said to be ***induced*** by a (probability or signed) measure if there exists a measure, denoted by $\mu_{\hat{S}}$, that satisfies $\hat{S}(R) = \mu_{\hat{S}}(R)$ for all $R \in \mathcal{R}$. Intuitively, **C1** implies the *finite additivity* of selectivity functions: $\hat{S}(R_1) = \hat{S}(R_2) + \hat{S}(R_3)$ if $R_1 = R_2 \cup R_3$ and $R_2 \cap R_3 = \emptyset$. Moreover, **C2** requires that $\hat{S}$ only outputs positive values; **C3** means that the values of $\hat{S}$ sum to 1 over the entire set of data points.

EXAMPLE 3.1. *Figure 1 (left) gives an illustration of three data points (A, B, C) and four possible ranges $(R_{AB}, R_{AC}, R_{BC}, R_{ABC})$, with four range functions and their selectivity predictions on the right. We also show the measures that induce the selectivity predictions of each range function (with their outputs on the three data points) in the bottom-right table. First, using Eq. 1 and basic linear algebra, one can see that S_1 is induced by a proper probability measure (e.g., $\mu_1(A) = 0.2, \mu_1(B) = 0.3, \mu_1(C) = 0.5$). However, this does not hold for the other selectivity functions. Specifically, S_2 does not satisfy **C1** as $S_2(R_{AB}) + S_2(R_{AC}) + S_2(R_{BC}) \neq 2 \cdot S_2(R_{ABC})$, which indicates that S_2 cannot be induced by a probability measure or a signed measure. Additionally, S_3, S_4 can only be induced from a signed measure — S_3 violates **C3** and S_4 violates **C2** of a probability measure.*

Advanced Concepts. We introduce two notions used only in proofs; readers may skip this subsection at first. For a signed measure μ, the *total variation* of μ, denoted by $|\mu|$, is defined by $|\mu|(E) = \sup \sum_{n=1}^{\infty} |\mu(E_n)|$ where the supremum is over all disjoint measurable partitions $E = \bigcup_{n=1}^{\infty} E_n$. Intuitively, $|\mu|$ measures how much μ "varies", and $|\mu|$ itself is a measure that dominates μ ($|\mu| \geq \mu$).

A signed measure μ is *absolutely continuous* w.r.t. the Lebesgue measure m if $\mu(E) = 0$ whenever $E \in \mathcal{M}$ and $m(E) = 0$. Absolute continuity can be interpreted as the *smoothness* of a measure. It guarantees the existence of a *signed density* $f : X \to \mathbb{R}$ such that $\mu(E) = \int_E f(x) dx$ for any $E \in \mathcal{M}$. Specifically, when μ is a

probability measure, then f must satisfy $f(x) \geq 0$ for any $x \in X$ and $\int_X f(x)dx = 1$. See [17] and [48] for details on measure theory.

3.3 ML Models as Selectivity Predictors

We formulate selectivity estimation as an ML problem. A learning algorithm $\mathcal{A}$ learns a model M to predict query selectivity from a training set $\mathcal{W} = \{z_i = (q_i, l_i)\}_{i=1}^n$, comprising observed queries/ranges and their selectivities. M minimizes the mean of loss ℓ over the dataset, where ℓ can be defined as the squared error (q_i, l_i): $\ell = (M(q_i) - l_i)^2$, or the absolute error: $\ell = |M(q_i) - l_i|$. Additionally, Qerror [40] $(\max(\frac{M(q_i)}{l_i}, \frac{l_i}{M(q_i)})$ and Squared Logarithmic Error (*e.g.,* SLE $= (\log M(q_i) - \log l_i)^2$, which is equivalent to optimizing Qerror), are prevalent in the literature as it better captures errors on selective queries.

3.4 PAC Learning Framework

Probably Approximately Correct (PAC) learning [28] is a framework for mathematical and rigorous analysis of *in-distribution* (In-Dist) generalization in machine learning. We first intuitively explain the high-level idea of PAC learnability. Readers who prefer a simpler explanation may directly refer to Table 1 for an intuitive summary.

PAC Learnability. Consider a learner $\mathcal{A}$ that receives training samples $\{z_i\}$ from an unknown distribution $Q(z)$ and picks a **hypothesis** (or a function) h from a **hypothesis space** or **function family** $\mathcal{H}$ (*i.e.,* a family/class of selectivity functions in our scenario). In the classical PAC framework, $\mathcal{A}$ is assumed to efficiently find the best h. We say a function family $\mathcal{H}$ is **learnable** if, given **enough training data**, then **with high probability** $(1 - \delta)$, the chosen function $h \in \mathcal{H}$ will have **low error** (no more than ϵ) on unseen data from $Q(z)$. Importantly, $\mathcal{A}$ must succeed for any distribution $Q(z)$ and any choice of δ, ϵ. We then introduce two key theoretical results that can *determine* whether a function family $\mathcal{H}$ (including $\{0, 1\}$-valued and real-valued functions) is PAC learnable.

(1) A class of $\{0, 1\}$-valued functions is PAC learnable **if and only if** its VC dimension (will be introduced later) is finite.
(2) A real-valued function class is γ-learnable **if and only if** its γ-fat dimension (will be introduced later) is finite.

In practice, finding the absolute best hypothesis in $\mathcal{H}$ can be challenging (*i.e.,* such efficient learner $\mathcal{A}$ may not exist); sometimes we only need a hypothesis whose true error is small, even if it is not the best. Fortunately, **uniform convergence** theory [5] (or specifically, Chernoff bound [23]) says that if $\mathcal{H}$ has a finite VC dimension (or γ-fat dimension), then with a **sufficiently large** training set, **every** hypothesis $h \in \mathcal{H}$ will have its **empirical error** close to its **true error** with high probability. Thus, we do not need the absolute best function; "what you see is what you get." As a result, if 1) $\mathcal{H}$ is learnable and 2) our optimization finds a hypothesis $h \in \mathcal{H}$ with small training error, it follows that h will have small true error. This paper adopts this definition of learnability.

Below is a short introduction to the *VC dimension* and the *fat-shattering dimension*, which are measures of complexity for classification and real-valued function classes, respectively. Since they appear only in our in-distribution generalization error theorem (Thm 4.1) and our paper primarily focuses on OOD generalization, first-time readers may skip the two definitions if desired.

Table 1: Key concepts and their intuitive explanations.

Concept	Intuitive Explanation
Generalization	The model's capability to perform well on unseen queries that are *not* in the training workload. We can predict the outcomes on unseen data based only on training samples.
Learnability/ In-Dist Generalization	Given sufficient training queries, the model's true error on unseen queries drawn from the *same distribution* with training queries is close to the training error.
OOD Generalization	Given sufficient training queries, the model's true error on unseen queries drawn from a *different distribution* from the training set is close to the training error.

Vapnik–Chervonenkis (VC) Dimension. $\mathcal{H}$ shatters a set of points if, for every possible way to assign 0/1 labels to those points, $\mathcal{H}$ contains at least one function that matches those labels exactly. The VC dimension is the size of the largest set of points that can be shattered by $\mathcal{H}$. Similarly, the VC dimension of a range space Σ is the size of the largest subset of X that can be shattered by Σ. Thus, if Σ is the set of d-dimensional hyper-rectangles, $\text{VC-dim}(\Sigma) = 2d$ [28].

Fat-Shattering Dimension. To handle real-valued functions (*e.g.,* selectivity functions), we use the fat-shattering dimension [27], which extends VC dimension. Informally, a set of points is "γ-shattered" if $\mathcal{H}$ can position those points above or below some target values by at least γ, matching any desired "above/below" pattern. The γ-fat dimension is how many points can be arranged this way. We define the γ-fat shattering dimension $\text{fat}_{\mathcal{H}}(\gamma)$ to be the size of the largest subset of X that can be γ-shattered by $\mathcal{H}$.

Limitation of the PAC Learning Framework. While PAC learnability can be used to quantify the generalization error for hypothesis spaces with finite VC (or fat-shattering) dimension, they are applicable *solely* to *in-distribution generalization* where both training and test queries are drawn from the same distribution $Q(z)$.

3.5 Existing Theoretical Results

For self-containment, in this section, we briefly review the main learnability results of selectivity functions from the literature [25], and point out the important assumption made by the paper.

Overview of [25]. Since selectivity functions are real-valued, to prove their learnability it suffices to show that their fat-shattering dimension is bounded. We cite the main Lemma in [25].

LEMMA 3.1. *Consider a range space $\Sigma = (X, \mathcal{R})$ and the hypothesis class $\mathcal{S}$ of range functions over input query ranges $R \in \mathcal{R}$. For any $\gamma \in (0, 1/2)$, the γ-fat shattering dimension of $\mathcal{S}$ is $\tilde{O}(\frac{1}{\gamma^{\lambda+1}})$[1], where λ is the $\text{VC-dim}(\Sigma)$ of the range space.*

Assumption. Note that the proof in [25] relies on an important condition on the hypothesis class: every range function $S \in \mathcal{S}$ is induced by a *probability measure* via (1).

3.6 The Gap Between Theory and Practice

Recall from Section 2 that there are four categories of query-driven approaches for learning selectivity functions: (1) linear and polynomial parametric functions (PFs), (2) histograms built from queries, (3) LEO, which learns to adjust histograms, (4) deep learning (DL) models such as Multi-Set Convolutional Network (MSCN) [31].

---

[1] $\tilde{O}(\cdot)$ hides polylogarihm dependencies on $1/\gamma$ for constant λ

Table 2: Theoretical Characteristics of query-driven methods.

	PFs	Histograms	LEO	DL Models
Monotonicity	✗	✓	✗	✗
Additivity	✗	✓	✓	✗

Among them, deep learning models achieve the best practical performance. In this section, we theoretically analyze whether the probability measure assumption holds for these methods.

Two necessary conditions. One can show that if a learned selectivity function $\hat{S}(R)$ (by a selectivity estimation model) is induced by a probability measure, it must satisfy **finite additivity** as well as **monotonicity** defined as follows.

- **Finite Additivity.** Implication of **C1**, defined in § 3.2.
- **Monotonicity.** Let R_1 and R_2 be two union-compatible ranges over schema Σ, such that $R_1 \subseteq R_2$ for any instance of Σ. We refer to this as a case of *query containment* [3]. Then finite additivity and positivity (**C2**) imply that $\hat{S}(R_1) \leq \hat{S}(R_2)$.

Theoretical Characteristics. First, histograms satisfy both monotonicity and additivity *by construction*. In contrast, PFs can violate both monotonicity and additivity due to *negative parameters* and *non-linear mappings* in the input and output of polynomial functions, respectively. For LEO, while it maintains the additivity of histograms through the *piecewise linear* form of its adjustment ratio r, it fails to ensure monotonicity because r is not strictly increasing. This non-monotonicity means LEO is not derived from a probability measure. Like PFs, DL models also break both monotonicity and additivity due to *negative weights* and *non-linear mappings*. We summarize the analysis results in Table 2.

Hence, unlike histograms (*i.e.,* data models built from queries), other three regression-like approaches learn a direct mapping from query ranges to selectivities and are not guaranteed to be induced by probability measures. Therefore, they (including the best-performing deep learning models) do *not* enjoy the theoretical results in § 3.5.

Problem Definition. We have shown that the selectivity functions learned by most query-driven models are not induced by probability measures, rendering the learnability results from previous work [25] inapplicable. Despite this, these models, such as MSCN, exhibit impressive practical performance, outperforming histograms on several benchmarks [31]. Additionally, the PAC learning framework fails to characterize generalization error for OOD test workloads, which are prevalent in real-world scenarios. Therefore, this paper aims to bridge the gap between theory and practice by ① relaxing the restrictions on the hypothesis class and deriving the corresponding PAC learnability results (**Goal 1**); ② exploring OOD generalization error beyond the PAC learning framework (**Goal 2**); ③ leveraging the theoretical results to design new strategies for improving existing selectivity learning models (**Goal 3**).

4 A NEW GENERALIZATION THEORY

In this section, we propose a new generalization theory that addresses the first two goals of the Problem Definition. Note that the proofs in § 4.1 and § 4.2 require advanced knowledge of measure theory and probability theory introduced in § 3. **Readers who prefer a simpler explanation may refer to § 4.3.**

4.1 Learnability Under Signed Measures

We first demonstrate the learnability of selectivity predictors induced by signed measures (*i.e.,* removing restrictions **C2** and **C3**). The results will be applied to NeuroCDF in Section 5 after showing that its hypothesis classes are induced by signed measures.

4.1.1 Learnability. Given a range space $\Sigma = (\mathcal{X}, \mathcal{R})$, let $\mathcal{S}_{sgn}$ denote the hypothesis class that consists of all functions $\hat{S} : \mathcal{R} \to \mathbb{R}$ that are induced by signed measures absolutely continuous *w.r.t.* the Lebesgue measure. Recall the definition of $\mu_{\hat{S}}$ in Section 3.2, and define the hypothesis class $\mathcal{S}_{sgn}(C)$ for any $C \geq 0$ as follows.

$$\mathcal{S}_{sgn}(C) := \left\{ \hat{S} \in \mathcal{S}_{sgn} : \left| \mu_{\hat{S}} \right| \leq C \right\}$$

Theorem 4.1 (In-Distribution Generalization Error Bound).
If VC-dim $(\Sigma) = \lambda$ *where* λ *is some constant, then the fat-shattering dimension of* $\mathcal{S}_{sgn}(C)$ *is finite and satisfies:*

$$fat\left(\mathcal{S}_{sgn}(C); \gamma\right) = \tilde{O}\left(C \cdot (1/\gamma)^{\lambda+1}\right) \qquad (2)$$

PROOF. The proof will be presented in a technical report. ☐

Theorem 4.1 implies that $\mathcal{S}_{sgn}(C)$ is ϵ-learnable with a polynomial number of training queries in $\frac{1}{\epsilon}$ and $\log \frac{1}{\delta}$, following the arguments in [9, 25]. By *uniform converge* theory, it means that given a sufficient number of training queries, the training and true errors remain *close, i.e.,* $|er(\hat{S}) - er^{train}(\hat{S})|$ is *bounded*.

4.1.2 Remark. Although Thm 4.1 extends prior work [25], it is key to a *practical* theory. It applies to a wider range of selectivity predictors beyond probability measures (see §5), and under mild assumptions, they have bounded OOD generalization errors (see §4.2). This allows performance prediction even when test queries come from a different distribution than training queries, a common real-world scenario. **More importantly, proving OOD generalization lies *outside* the PAC framework, so existing results (*e.g.,* fat-shattering bounds and [25]) do *not* apply.**

4.2 OOD Generalization Error

In this section, we target OOD generalization error beyond PAC learning. The main results appear in the callout for Theorem 4.2.

The theorem shows that under the realizable assumption, a predictor $\hat{S}$ trained with n i.i.d. samples from a training distribution Q to (ϵ, δ)-learn will have its generalization error on a different testing distribution P bounded above by $C\sqrt{\epsilon}$ with probability at least $1 - \delta$, provided that Assumptions 4.1 through 4.3 (introduced later in § 4.2.1) hold. As will be introduced in § 5, this result will theoretically demonstrate the potential advantage of modeling CDFs over selectivities in terms of out-of-distribution generalization error.

The assumptions are relatively mild. Intuitively, Assumption 4.1 requires only that $\hat{S}$ is bounded and that the *densities exist*; Assumption 4.2 stipulates that the region covered by the testing distribution P must be *contained within* the region covered by the training distribution Q; and Assumption 4.3 essentially requires sufficient *diversity* in the training ranges.

4.2.1 Main Theoretical Results. It is important to note that one cannot expect an algorithm trained on a distribution Q to generalize

well to an arbitrary testing distribution P. To ensure provable and robust generalization, we impose the following assumptions.

ASSUMPTION 4.1. *The learned selectivity $\hat{S}$ is bounded such that there exists a constant C_1 for which $\left|\hat{S}(R)\right| \leq C_1$ for any $R \in \mathcal{R}$.*

Before proceeding, we introduce some additional notations. Let $\mathcal{Z} = \mathcal{R} \times \mathbb{R}$. We use Q and P to denote the training and testing distribution of $Z = (R, W) \in \mathcal{Z}$, respectively. Given the training distribution Q, let Q_R be the *marginal distribution* of R and define $\mathcal{X}_Q := \bigcup_{R \in \text{supp } Q_R} R$, which is a subset of $\mathcal{X}$. The marginal distribution P_R and the set $\mathcal{X}_P$ are defined similarly for the testing distribution. We now introduce Assumption 4.2 and 4.3.

ASSUMPTION 4.2. *There exists a constant C_2 such that the marginal training and testing distributions Q_R and P_R satisfy*

$$P_{R \sim P_R}\left[x \in R\right] \leq C_2 \cdot P_{R \sim Q_R}\left[x \in R\right], \quad \forall x \in \mathcal{X}.$$

Remark. Assumption 4.2 requires that the probability $P_{R \sim P_R}\left[x \in R\right]$ (the likelihood of x being sampled during testing) is upper-bounded by the probability $P_{R \sim Q_R}\left[x \in R\right]$ (the likelihood of x being sampled during training) multiplied by a constant C_2. This implies that $\mathcal{X}_P \subset \mathcal{X}_Q$. The rationale is that if $\mathcal{X}_P$ includes some x that is not covered by any range during training, then one cannot expect to learn the selectivity around x accurately.

ASSUMPTION 4.3. *The true S_D and the learned selectivity $\hat{S}$ are induced by signed measures that are absolutely continuous, with corresponding signed densities $f_D, \hat{f}$. Additionally, there exists a constant $c_3 > 0$ such that Q_R and the signed density $\hat{f}$ satisfy*

$$\mathbb{E}_{R \sim Q_R}\left|\int_{\mathcal{X}}\left(\hat{f}(x) - f_D(x)\right) \mathbb{1}(x \in R) dx\right|$$
$$\geq c_3 \cdot \mathbb{E}_{R \sim Q_R} \int_{\mathcal{X}}\left|\hat{f}(x) - f_D(x)\right| \mathbb{1}(x \in R) dx$$

Remark. Assumption 4.3 presupposes the validity of interchanging the order of integration and the absolute value. Intuitively, it ensures that Q_R covers a diverse set of ranges rather than focusing on ranges where the error $\hat{S}(R) - S_D(R)$ happens to be relatively small. A simple example illustrating a situation where Assumption 4.3 holds is provided below in Example 4.1.

EXAMPLE 4.1. *For $\mathcal{X} = [-1/2, 1/2]$, suppose the densities f_D and $\hat{f}$ are defined as $f_D(x) = 1$ and $\hat{f}(x) = 1 + 2\delta_n x$, where δ_n is a parameter that quantifies how well $\hat{f}$ approximates f_D. If Q_R is uniformly distributed over intervals of length $1/4$ with centers located within the range $[-3/8, 3/8]$, then it can be verified by direct computation that Assumption 4.3 is satisfied with $c_3 = 1/2$ for any value of δ_n.*

Now, we are ready to present our OOD generalization error bound:

Theorem 4.2 (OOD Generalization Error Bound). *Suppose Assumption 4.1-4.3 hold. In addition, if the in-distribution generalization error of $\hat{S}$ can be bounded by*

$$P_{Z_1^n \sim Q^{\otimes n}}\left[\text{er}_Q(\hat{S}) < \epsilon\right] \geq 1 - \delta \tag{3}$$

then the out-of-distribution generalization error $\text{er}_P(\hat{S})$ satisfies

$$P_{Z_1^n \sim Q^{\otimes n}}\left[\text{er}_P(\hat{S}) < \frac{(C_1 + 1)C_2}{c_3}\sqrt{\epsilon}\right] \geq 1 - \delta \tag{4}$$

Informally, the Theorem states that if Assumptions 4.1-4.3 hold and $\hat{S}$ achieves bounded in-distribution generalization error, then $\hat{S}$ will also have bounded out-of-distribution generalization error.

We then present the full proof. To better understand the proof, consider the following sequence of inequalities:

$$\text{er}_P(\hat{S}) \overset{(a)}{\lesssim} \mathbb{E}_{R \sim P_R}\left|\hat{S}(R) - S_D(R)\right|$$
$$\overset{(b)}{\lesssim} \mathbb{E}_{R \sim Q_R}\left|\hat{S}(R) - S_D(R)\right| \overset{(c)}{\leq} \left[\text{er}_Q(\hat{S})\right]^{1/2}$$

Here $a_n \lesssim b_n$ means $a_n = O(b_n)$. Step (a) follows from the upper-bound Assumption 4.1; (b) involves a change of measure from P_R to Q_R and connects through Assumptions 4.2 and 4.3; and (c) is based on the Cauchy–Schwarz inequality. The complete proof provides a detailed justification for each of these inequalities.

PROOF. To bound $\text{er}_P(\hat{S})$ in (4), note that one has

$$\text{er}_P(\hat{S}) = \mathbb{E}_{R \sim P_R}\left(\hat{S}(R) - S_D(R)\right)^2$$
$$\overset{(i)}{\leq} (C_1 + 1) \cdot \mathbb{E}_{R \sim P_R}\left|\hat{S}(R) - S_D(R)\right|$$
$$\leq (C_1 + 1) \cdot \int_{\mathcal{X}}\left|\hat{f}(x) - f_D(x)\right| \cdot \left[\mathbb{E}_{R \sim P_R} \mathbb{1}(x \in R)\right] \cdot dx$$
$$\overset{(ii)}{\leq} (C_1 + 1) \cdot C_2 \underbrace{\int_{\mathcal{X}}\left|\hat{f}(x) - f_D(x)\right| \cdot \left[\mathbb{E}_{R \sim Q_R} \mathbb{1}(x \in R)\right] \cdot dx}_{(I)}$$

$$\tag{5}$$

Here (i) follows from Assumption 4.1; (ii) is due to Assumption 4.2. Meanwhile, one can obtain that

$$(I) \overset{(i)}{=} \mathbb{E}_{R \sim Q_R} \int_{\mathcal{X}}\left|\hat{f}(x) - f_D(x)\right| \mathbb{1}(x \in R) dx$$
$$\overset{(ii)}{\leq} c_3^{-1} \mathbb{E}_{R \sim Q_R}\left|\int_{\mathcal{X}}\left(\hat{f}(x) - f_D(x)\right) \mathbb{1}(x \in R) dx\right|$$
$$= c_3^{-1} \underbrace{\mathbb{E}_{R \sim Q_R}\left|\hat{S}(R) - S_D(R)\right|}_{(II)} \tag{6}$$

Here (i) follows from the Fubini's theorem, and (ii) is a result of Assumption 4.3.

Note that the Cauchy–Schwarz inequality implies that

$$(II) \leq \left[\mathbb{E}_{R \sim Q_R}\left(\hat{S}(R) - S_D(R)\right)^2\right]^{1/2} = \left[\text{er}_Q(\hat{S})\right]^{1/2}$$

Combine the above inequality with (3) implies

$$P\left\{(II) < \sqrt{\epsilon}\right\} \geq 1 - \delta \tag{7}$$

Finally, combining (5), (6) and (7) gives (4). □

4.2.2 OOD Scenarios.

We define three specific OOD scenarios which naturally arise in real-world applications.

Scenario 1: Query Center Move refers to a shift in the predominant focus of queries, characterized by a change in the attribute values around which the queries are *concentrated*.

Scenario 2: Query Granularity Shift refers to a change in the granularity of query selection predicates. Granularity pertains to the *specificity* or *broadness* of the data subsets accessed by queries. **Scenario 3: Query Structure Change.** When the join graph remains unchanged, adding or dropping predicates essentially changes the query granularity, therefore reducing to Scenario 2. Our theory does not support changes in the join graph as both Thm 4.1 and Thm 4.2 hold for each join graph *independently*.

4.2.3 **Point Queries**. Following [18, 19], we can treat point queries as range queries that cover only the corresponding data point. For instance, point query (t.production_year=1980) can be rewritten as (t.production_year>1979 ∧ t.production_year ≤ 1980). This approach makes our results applicable to point queries.

4.2.4 **Data Distribution Shifts**. Our theory assumes a static data distribution, and shifts in data distribution can introduce additional errors. Although handling data distribution shifts is not the primary focus of this paper, we provide a preliminary theoretical result that extends Theorem 4.2 to account for such shifts: Let $TV\,(Q_X, P_X) \coloneqq \sup_{x \in \mathcal{X}} |Q_X(x) - P_X(x)|$ be the *total variation distance* between the old data distribution Q_X and the new data distribution P_X.

PROPOSITION 4.1 (**OOD GENERALIZATION ERROR BOUND WITH DATA SHIFTS**). *Suppose assumptions in Theorem 4.2 hold. In addition, if $TV\,(Q_X, P_X) \leq \epsilon_{TV}$, and the in-distribution generalization error of $\hat{S}$ can be bounded by $P_{Z_1^n \sim Q^{\otimes n}} \left[\mathrm{er}_Q(\hat{S}) < \epsilon \right] \geq 1 - \delta$, then the out-of-distribution generalization error $\mathrm{er}_P(\hat{S})$ satisfies*

$$P_{Z_1^n \sim Q^{\otimes n}} \left[\mathrm{er}_P(\hat{S}) < (C_1 + 1)C_2 \left(c_3^{-1}\sqrt{\epsilon} + 2\epsilon_{TV} \right) \right] \geq 1 - \delta. \quad (8)$$

PROOF. The proof will be presented in a technical report. □

Intuitively, this proposition says that if P_X and Q_X are close in terms of the total variation distance, then the OOD generalization error for $\hat{S}$ remains bounded. However, we note that relying on TV as the measure of data distribution shifts is a strong assumption. Future research might explore more flexible metrics for studying how OOD generalization error behaves under diverse data shifts.

4.3 Summary and Discussion

Combining the results from § 4.1 and § 4.2:

> **Summary of our results**: If a selectivity learning model is induced by a *signed measure* and trained on a sufficient number of queries, both its in-distribution generalization error (when training and test queries are from the same distribution) and out-of-distribution generalization error (when training and test queries come from different distributions) are *bounded* (*i.e.,* close to training error), under mild assumptions (Assumption 4.1-4.3).

The summary provides insights for designing improvement strategies for query-driven models: if we can show a class of selectivity learning models that are *provably* induced by signed measures, then the favorable in-distribution (Theorem 4.1) and OOD (Theorem 4.2) generalization results are immediately applicable.

5 MODELING *CDF*S WITH *NEURAL NETS*

Building on the insights in § 4.3, this section introduces the first strategy for enhancing query-driven selectivity learning models.

We first consider this question: is it feasible to design a selectivity estimation paradigm that works well in practice *and* inherits our theoretical guarantees? Unfortunately, this is not easy. Although existing approaches developed from SOTA theory [25] achieve SOTA results *among selectivity predictors that are induced by probability measures*, they are not as effective (experimentally shown in § 7) as recent deep learning-based models in practice due to the limited model capacities of models induced by probability measures.

On the other hand, NNs, while lacking comprehensive theoretical backing, show remarkable efficacy in practice. Often, NNs achieve both very small training and test errors when queries are drawn from the same distribution, which cannot be fully explained by existing theories like the PAC framework. This aligns with extensive well-known empirical evidence in the ML literature [59].

Therefore, an ambitious goal is to combine the theoretical results and the practicality of NNs. To this end, we propose NEUROCDF, a novel selectivity estimation paradigm/framework that enjoys both theoretical guarantees and practical utility.

5.1 Overview

High-Level Idea. NEUROCDF leverages the fact *that the selectivity of a rectangular query can be computed as a linear combination of the CDFs evaluated at its vertices.* (will be discusses in § 5.2). CDFs, in statistical terms, measure the probability that a random variable takes a value less than or equal to a specific point. Therefore, the key idea of NEUROCDF is that, instead of directly modeling the ultimate selectivities of input queries, *we use a neural network as the model to parameterize the underlying CDFs.* The query selectivity can be estimated by multiple calls to the *CDF prediction model $\mathcal{M}$* and aggregating the results, as shown in Figure 2.

Theoretical Guarantees. As will be discussed in § 5.3, NEUROCDF, as a framework, can be proved to be induced by signed measures through its CDF modeling. Hence, both the in-distribution (Thm 4.1) and OOD (Thm 4.2) generalization error bounds directly apply to NEUROCDF. This means given sufficient training queries and under Assumption 4.1-4.3, the in-distribution and OOD generalization errors of NEUROCDF are both close to its training error — an advantage not present in existing methods that directly model selectivities.

Apart from the theoretical guarantees provided by NEUROCDF, NEUROCDF combines the empirical strengths of neural nets as NNs are known for achieving very low training error [21] due to their high model capacity. Note that NEUROCDF does not offer generalization theories for neural networks *per se*, but it leverages their empirical success (*e.g.,* low training error) alongside the formal guarantees of the CDF modeling paradigm permitted by our theory.

Workflow of NEUROCDF. In a d-dimensional data space, the CDF prediction model $\mathcal{M}$ of NEUROCDF takes as input a vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^\top$ of real-valued variables and outputs an estimated cumulative distribution function (CDF), $\hat{F}(\mathbf{x}) = \hat{P}(X \leq \mathbf{x})$. With a query workload $\mathcal{W} = \{(q, l)\}$, the NEUROCDF framework proceeds in four steps, beginning with two data preprocessing phases.
① **Normalization.** Each range query q is a d-dimensional hyper-rectangle $(\tilde{a}_1 < x_1 \leq \tilde{b}_1) \wedge \cdots \wedge (\tilde{a}_d < x_d \leq \tilde{b}_d)$. We apply min-max normalization to scale all unnormalized $\tilde{a}_i, \tilde{b}_i$ into normalized values $a_i, b_i \in [0, 1]$. Unqueried attributes/dimensions (*i.e.,* attribute not involved in the query) are set to $[0, 1]$.
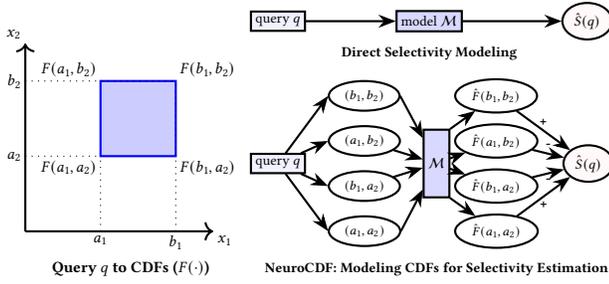
Figure 2: Left: relationship between a rectangle query and CDFs; Right: direct selectivity modeling v.s. NeuroCDF.

② **CDF Conversion (§ 5.2).** Each range query q is converted into a set of vectors v (at each vertex of q), which serve as inputs for training the CDF prediction model $\mathcal{M}$.

③ **Model Training.** Training of NeuroCDF uses forward–backward propagation with mean squared error (MSE) loss. Unlike existing query-driven models which directly predict the query selectivity using only one forward pass, NeuroCDF computes a query's selectivity $\hat{S}(q)$ by gathering multiple CDF values $\hat{F}(v)$ via $\mathcal{M}$. These values are combined using Equation (9). Because the entire procedure is fully *differentiable*, the loss $\sum(\hat{S}(q) - S(q))^2$ can be optimized via stochastic gradient descent (SGD) and batched training.

④ **Prediction.** Once trained, NeuroCDF uses the same multi-call forward process: for an incoming query q, it computes each required $\hat{F}(v)$, then aggregates them to derive $\hat{S}(q)$.

5.2 Converting Queries to CDFs

Consider the case of 2-dimensional data shown in Figure 2 (left), one can verify that the selectivity of a query $q : \{(a_1 < x_1 \le b_1) \land (a_2 < x_2 \le b_2)\}$ (represented by the rectangle in blue) can be computed by aggregating the CDF values at the four vertices (*i.e.,* $(b_1, b_2), (a_1, b_2), (b_1, a_2), (a_1, a_2)$) of the query rectangle,

$$S(q) = F(b_1, b_2) - F(a_1, b_2) - F(b_1, a_2) + F(a_1, a_2).$$

We extend the formula to $d-$dimensional data (outlined on page 197 of the book [17]). Let q be a range query (of hyper-rectangle) in the d dimensional space, *i.e.,* $q = (a_1, b_1] \times \ldots \times (a_d, b_d]$. The vertices V of this hyper-rectangle are $V = \{a_1, b_1\} \times \ldots \times \{a_d, b_d\}$. For any vertex $v \in V$, define $\#a(v)$ as the number of a's in v, indicating the count of left endpoints. For example, in Figure 2 (left), $\#a([a_1, b_2]) = 1$. The general case formula is subsequently provided for completeness.

Theorem 5.1. *Let* $sgn(v) = (-1)^{\#a(v)}$. *The selectivity of range query q in $d-$dimensional space is computed by aggregating the CDF values at all vertices of the query hyper-rectangle using the below formula,*

$$S(q) = \sum_{v \in V} sgn(v) F(v) \tag{9}$$

PROOF SKETCH. This is a direct application of the inclusion-exclusion principle [46]. See page 36 of the book [17] for details. □

Model Choice for CDF Prediction. NeuroCDF can incorporate any query-driven model architecture by viewing the input vector as a query (*i.e.,* we can interpret every CDF as the selectivity estimate of a one-sided query). For example, $F(b_1, b_2)$ is equivalent to the selectivity estimation of a legitimate query $q : \{(x_1 \le b_1) \land (x_2 \le b_2)\}$.

Thus, possible model choices include Multi-Set Convolution Networks [31], MLP with flattened query encoding [19], or more recent NN models [33, 45, 50]. Although non-NN methods such as XG-Boost [14] offer greater interpretability and could be used for CDF prediction NeuroCDF, we opt for NNs due to their *ease of optimization*. XGBoost relies on direct mappings from data points to their CDF values ($\mathbf{x} \mapsto F(\mathbf{x})$) and optimizes based on gradients between predictions and actual values. However, in selectivity learning, we *only* have mappings from queries to their selectivities ($q \mapsto S(q)$), lacking the direct data-to-CDF mappings required by XGBoost. This makes optimizing XGBoost challenging. NNs, on the other hand, can be trained *end-to-end* effectively using only $q \mapsto S(q)$ mappings without needing direct $\mathbf{x} \mapsto F(\mathbf{x})$ mappings.

Efficiency. Below result shows that in NeuroCDF, we do not have to estimate the CDF value for every possible vertex v. Only those *necessary* vertices require estimates from $\mathcal{M}$.

Theorem 5.2. *The number of calls to the CDF prediction model $\mathcal{M}$ for estimating a query selectivity is 2^{n_c}, where n_c is the number of attributes/columns involved in the query.*

PROOF. The proof will be provided in a technical report. □

5.3 Theoretical Analysis

Next, we prove that NeuroCDF, as a framework, is induced by a signed measure due to its CDF modeling paradigm. *This connects NeuroCDF to the two theoretical results in previous sections.* Surprisingly, this property applies to LEO as well.

Theorem 5.3. *Suppose $\mathcal{R}$ consists of axis-aligned hyper-rectangles. Given a function $\hat{S} : \mathcal{R} \to \mathbb{R}$, suppose there exists a function $F_{\hat{S}} : \mathcal{X} \to \mathbb{R}$ such that for any $R \in \mathcal{R}$, $\hat{S}(R) = \sum_{v \in V_R} sgn(v) F_{\hat{S}}(v)$ where V_R is the vertex set of R. Then $\hat{S}$ is induced by a signed measure.*

PROOF SKETCH. This can be shown by a simple modification of the proof of Theorem 1.1.11 in [17]. □

COROLLARY 5.4. *All predictions from NeuroCDF and LEO are induced by signed measures.*

PROOF SKETCH. NeuroCDF and LEO satisfy Theorem 5.3 by design, as they model CDFs (For LEO, $\hat{F}_{LEO}(x) = F_{\text{hist}}(x) \cdot g_{\text{adjust}}(x)$ where $F_{\text{hist}}(x)$ is the CDF of the histogram and $g_{\text{adjust}}(x)$ is the adjustment factor at x) and use Eq 9 to estimate range queries. □

With Corollary 5.4 in place, let $\mathcal{S}_{\text{NeuroCDF}}$ and $\mathcal{S}_{LEO}$ denote the hypothesis class of NeuroCDF and LEO when the inducing signed measures are all absolute continuous. Then the learnability results for NeuroCDF and LEO are given as follows.

Theorem 5.5. *Let $\Sigma = (\mathcal{X}, \mathcal{R})$ be a range space. If $\text{VC-dim}(\Sigma) = \lambda$ where λ is some constant, then the fat-shattering dimension of $\mathcal{S}$ is finite and satisfies: $fat(\mathcal{S}; \gamma) = \tilde{O}((1/\gamma)^{\lambda+1})$ for any $\mathcal{S} \in \{\mathcal{S}_{\text{NeuroCDF}}, \mathcal{S}_{LEO}\}$.*

PROOF SKETCH. One can show that the predictions of NeuroCDF and LEO are bounded, and hence $\mathcal{S} \subset \mathcal{S}_{\text{sgn}}(C)$ for some constant C. Then the theorem follows by applying Theorem 4.1. □

Limitation of NeuroCDF. Currently, NeuroCDF is not compatible with Qerror or MSLE because it can yield negative estimates where Qerror does not apply. This issue arises as NNs might fail to

Table 3: Generalization performances of different models

Model	In-Dis Generalization		OOD Generalization	
	RMSE	Qerror	RMSE	Qerror
LW-NN	★☆	★★	☆☆	☆☆
MSCN	★☆	★★	☆☆	☆☆
NeuroCDF (LW-NN)	★★	★☆	★☆	★☆
NeuroCDF (MSCN)	★★	★☆	★☆	★☆

produce a valid CDF, which can lead to negative values in estimates from (9). We attempted to address this issue by clipping negative estimates to $1/|D|$ or enforcing monotonicity [35]. Unfortunately, we observed significant performance degradation since 1) the clipping is not differentiable preventing the model from learning from queries with clipped estimates; 2) the enforcement of monotonicity would reduce model capacity and introduce noises into training.

5.4 Preliminary Evaluation of NeuroCDF

We implement NeuroCDF with LW-NN [19] and MSCN [31]. We intentionally exclude data information to concentrate on the modeling paradigm. Despite that NeuroCDF is not compatible with Qerror, we observe significant improvement in OOD generalization on both models, which further inspires us to design a more general improvement strategy in the next section.

We generate a collection of training queries on a synthetic dataset sampled from a 10-dimensional highly correlated Gaussian distribution. Moreover, we use two types (In-distribution and OOD) of test queries to assess the model generalization capabilities.

NeuroCDF *v.s.* Direct Selectivity Modeling? We summarize the generalization performance of different models w.r.t two popular measures (RMSE and Qerror) in Table 3. We define three qualitative levels of generalization performance on test sets — (★★): RMSE < 0.05 or median Qerror < 2; (★☆): 0.05 < RMSE < 0.2 or 2 < median Qerror < 10; (☆☆): RMSE > 0.2 or median Qerror > 10. From the table, we observe two important findings.

F1. All four models achieve very good in-distribution generalization performance w.r.t the metric they are optimized for. Specifically, both LW-NN and MSCN are optimized for Qerror, but after using the NeuroCDF paradigm, they are optimized for RMSE.

F2. LW-NN and MSCN perform poorly on OOD queries both in terms of Qerror and RMSE. Importantly, NeuroCDF improves their OOD generalization performance even with Qerror. This matches the theoretical result regarding OOD generalization error in § 4.2.

6 TRAINING WITH CDF SELF-CONSISTENCY

Motivated by the theoretical results and the limitation observed in NeuroCDF, this section introduces a new training framework, SeConCDF, for query-driven selectivity models.

6.1 High-level Idea

In the previous section, we noted that direct query selectivity modeling is effective for in-distribution generalization with respect to arbitrary measures or loss functions. However, the CDF modeling paradigm used in NeuroCDF provides superior OOD generalization because it enforces a *hard* constraint on a signed measure, ensuring that all predictions from NeuroCDF are *coherently* induced by a signed measure. However, it does not support arbitrary
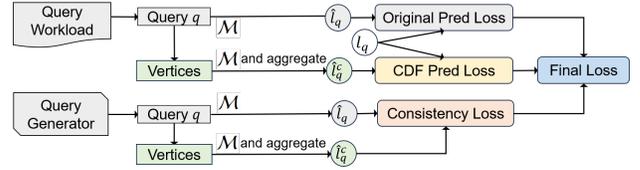
Figure 3: Training a selectivity model $\mathcal{M}$ with SeConCDF.

loss functions, such as Qerror. This raises a key question: *can we combine the advantages of direct query selectivity modeling and CDF modeling to achieve both strong in-distribution generalization with arbitrary loss functions and improved OOD generalization?*

SeConCDF addresses this limitation by adopting direct query selectivity modeling (which avoids the negative estimate issue) and introducing a **soft** constraint on the signed measure, unlike the *hard* constraint used in NeuroCDF. Specifically, SeConCDF operates on selectivity learning model $\mathcal{M}$ that targets the query selectivity directly (instead of NeuroCDF that requires $\mathcal{M}$ to model the CDFs), and applies a *soft* constraint through *CDF self-consistency regularization* during training. Recall that as discussed in § 5.2, *each CDF corresponds to the selectivity estimate of a one-sided rectangle query*, thus we can extract the CDFs learned by the selectivity model $\mathcal{M}$ from these queries. We then utilize appropriate loss functions to maintain consistency between the learned CDFs and the learned selectivity function. The intuition is that better *alignment* between the learned selectivity functions and the extracted CDFs indicates that $\mathcal{M}$ is more closely induced by a signed measure, thereby being more likely to achieve bounded OOD generalization error.

This approach combines the benefits of both paradigms, providing robust OOD generalization and allowing flexibility in the choice of loss functions. Although SeConCDF is inspired by both the theoretical and empirical analyses of NeuroCDF, it does not come with a theoretical guarantee because it cannot be confirmed as being entirely induced by signed measures. Despite this, SeConCDF shows significant practical effectiveness in our experiments.

6.2 CDF Self-Consistency Regularization

Figure 3 illustrates the training workflow of $\mathcal{M}$ using SeConCDF. SeConCDF processes a *query workload* $\mathcal{W} = \{(q, l)\}$ (same as NeuroCDF), and utilizes a *query generator* $\mathcal{G}$. They collectively contribute to the final loss optimized by $\mathcal{M}$. The operation of SeConCDF within each query batch is described step-by-step.

① **Loss Computation with Query Workload $\mathcal{W}$.** SeConCDF initiates with two preprocessing steps analogous to NeuroCDF: *normalization* and *CDF conversion* (which extracts the set $\{v\}$ of vectors for each query q at its vertices). With all information needed, SeConCDF then computes two types of losses: **Original Prediction Loss** $\mathcal{L}_{\text{OriPred}}$ and **CDF Prediction Loss** $\mathcal{L}_{\text{CDFPred}}$.

$\mathcal{L}_{\text{OriPred}}$ is calculated as the discrepancy between $\mathcal{M}$'s direct selectivity prediction for a query q, denoted $\hat{l}_q$, and the true label $\hat{l}_q$. Typically, the loss function involves Qerror or MSLE, consistent with current methods in query-driven selectivity learning.

$\mathcal{L}_{\text{CDFPred}}$ aligns with the procedures of NeuroCDF. For each query q, SeConCDF transforms its vertex set $\{v\}$ into corresponding one-sided queries and extracts the CDFs as predicted by $\mathcal{M}$.

Using the formula (9), the selectivity estimate (denoted $\hat{l}_q^c$) from the learned CDFs is calculated, and $\mathcal{L}_{\text{CDFPred}}$ is then defined as the RMSE between $\hat{l}_q^c$ and the actual label l_q. This loss forces the model $\mathcal{M}$ to learn the underlying CDFs from the *training workload*, aside from the direct mapping from queries to selectivities.

② **Loss Computation with Query Generator** $\mathcal{G}$. The query generator samples queries from a distribution, using random sampling for this paper, although other sampling methods are compatible within SeConCDF. Each sampled query q undergoes the same *normalization* and *CDF conversion* steps as in ① to produce a set of vertex vectors for q. These vectors are used to compute the selectivity estimate $\hat{l}_q^c$ from the learned CDFs. We then introduce a third type of loss, **Consistency Loss** $\mathcal{L}_{\text{Consistent}}$, defined as the RMSE between $\hat{l}_q^c$ and $\mathcal{M}$'s direct selectivity estimate of q, $\hat{l}_q$. This loss enforces *consistency* between $\mathcal{M}$'s direct selectivity predictions and its learned CDFs across diverse queries. This step can be implemented synchronously with ① to enhance training efficiency.

③ **Model Training.** The final loss is defined as

$$\mathcal{L} = \mathcal{L}_{\text{OriPred}} + \omega_1 \mathcal{L}_{\text{CDFPred}} + \omega_2 \mathcal{L}_{\text{Consistent}}, \qquad (10)$$

where ω_1, ω_2 are hyper-parameters controlling the balance among the three losses. We empirically tune them from four candidate values $\{0.1, 1, 10, 100\}$. $\mathcal{M}$ is optimized to minimize $\mathcal{L}$ using SGD.

④ **Prediction.** Once trained, $\mathcal{M}$ can directly predict the selectivities for incoming queries without needing CDF conversion.

Remark. SeConCDF does *not* change the *model architecture* or *inference procedure* of existing selectivity models $\mathcal{M}$ that directly target selectivities. The two losses $\mathcal{L}_{\text{CDFPred}}$ and $\mathcal{L}_{\text{Consistent}}$ serve as the key to incorporating CDF self-consistency regularization into $\mathcal{M}$. Furthermore, computing these losses does not require new actual query executions to obtain selectivities.

7 EXPERIMENTS OF SeConCDF

In this section, we implement SeConCDF and integrate it into two recent NN-based query-driven proposals — LW-NN [19] and MSCN [31]. Both models utilize an MLP; however, they adopt distinct methodologies for query encoding. LW-NN employs a flattened query encoding mechanism, while MSCN stands for multi-set convolutional network. We aim to answer two research questions as follows. 1) While existing query-driven models perform well for in-distribution generalization, are they robust to OOD generalization? 2) Can SeConCDF improve their OOD generalization performance while maintaining their in-distribution performance, in terms of both prediction accuracy and query latency performance? Note that while we implement SeConCDF with two query-driven models, it is general and applicable to any loss-based deep learning models.

7.1 Experimental Setup

Datasets. We conducted experiments using one single-table dataset, Census, and three multi-table datasets: IMDb-small, DSB [16] and CEB [41]. Census comprises the basic population characteristics in US, with approximately 49K tuples across 13 attributes. We use Census for prediction accuracy experiments since a few relevant approaches only support single-table queries. The IMDb [32] dataset is derived from the Internet Movie Database. Previous studies [32] show that IMDb is highly correlated and skewed. IMDb-small and

CEB use 6 and 15 tables of the original IMDb, respectively. DSB is as an extension of the TPC-DS benchmark [44], characterized by more complex data distributions and demanding query templates. We populated a DSB database with a scale factor 50 using the default physical design configuration, and use 5 tables in our experiments.

Workloads. Since the primary goal of this section is to assess both the in-distribution (In-Dist) and OOD generalization capabilities of query-driven models, we focus on the first two OOD scenarios as outlined in § 4.2.2. Specifically, We train models on specific query distributions and assess their performance on unseen queries both within the same distribution (In-Dist generalization) and from different distributions (OOD generalization). To generate such workloads, for each dataset, we initially create a set of candidate queries. For IMDb-small, we directly leverage the training queries from [29] with up to 5 joins and diversified join graphs. For DSB and Census, we create candidate queries by randomly sampling join graphs and filter conditions. However, IMDb-small and DSB are limited to 5- and 4-way star join queries, respectively. To explore the scalability of SeConCDF, we extend our analysis to more complex join queries using template 1a of CEB, which includes 9-way joins with star, chain, and self-joins. Due to the limited range variation in predicate values, such as the 14 different ranges for `t.production_year`, which does not satisfy Assumption 4.3, we have generated new candidate queries from existing CEB-1a queries while enriching the diversity of the `t.production_year` ranges. We denote the new 9-way join workload **CEB-1a-varied**. After this, we obtained 50K, 60K, 70K and 43K [2] candidate queries for Census, DSB, IMDb-small and CEB-1a-varied, respectively. From the candidate queries, we simulate training and test workloads for both two OOD scenarios.

To simulate OOD scenarios, we designate a shifting attribute a, for each dataset: age for the Census dataset, `t.production_year` for IMDb-small and CEB-1a-varied, and `ss.ss_list_price` for DSB. In both OOD scenarios, models are trained on queries with the attribute a normalized within specific bounds (c_a for query centers and l_a for range lengths). For in-distribution generalization, models are evaluated on queries matching training conditions. For OOD generalization, they are tested on queries where c_a (for center move) or l_a (for granularity shift) falls outside these bounds. Training and test queries are kept strictly non-overlapping.

Compared Approaches. We implemented LW-NN [19] ourselves. For MSCN, we used the code from [1]. We evaluate MSCN and LW-NN trained with SeConCDF, referred to as MSCN+CDF and LW-NN+CDF. We include two query-driven approaches, PtsHist and QuadHist (code from [2]), which are based on SOTA theory [25], to demonstrate the limitations of PAC learning. We also include Quicksel [43] in our comparison. The three query-driven models are induced from *probability measures* where our OOD generalization result (Thm 4.2) is applicable. For data-driven approaches, we use PostgreSQL (multi-dimensional histograms) and uniform sampling (Sampling) as baselines. PtsHist, QuadHist, Quicksel, and the LW-NN we implemented do not support joins, so we evaluate them on Census. For a fair comparison, we exclude bitmaps from LW-NN or MSCN, since other query-driven models only utilize query information. We turn on the bitmaps in multi-table experiments. We also compare against Robust-MSCN [42] (join bitmaps

---

[2]We include all subqueries of CEB-1a-varied queries in the accuracy experiment

Table 4: Prediction accuracy on IMDb-small (Left) and DSB (Right) w.r.t. in-distribution queries/out-of-distribution queries.

Model	Query Center Move			Query Granularity Shift			Model	Query Center Move			Query Granularity Shift		
	RMSE	Qerror		RMSE	Qerror			RMSE	Qerror		RMSE	Qerror	
		Median	90%		Median	90%			Median	90%		Median	90%
PostgreSQL	0.042/0.086	6.3/4.2	669/549	0.045/0.124	6.1/3.7	921/297	PostgreSQL	0.033/0.068	1.6/1.9	6.6/14	0.050/0.098	1.6/2.8	5.2/15
Sampling	0.175/0.196	31/35	$10^3/10^3$	0.180/0.197	29/21	$10^3/10^3$	Sampling	0.121/0.186	3.7/9.2	38/82	0.143/0.194	4.7/12	75/64
MSCN	0.020/0.700	1.6/10^5	6.5/10^6	0.021/0.763	1.5/10^3	8.6/10^6	MSCN	0.057/0.283	1.4/5.0	3.6/78	0.027/0.345	1.2/10^3	1.8/10^6
MSCN + CDF	0.022/0.035	1.9/2.0	7.3/10	0.024/0.047	1.8/1.7	11/7.0	**MSCN + CDF**	0.061/0.158	1.6/2.1	1.6/2.1	5.3/17	1.3/2.5	1.8/73

and query masking), and its variant, Robust-MSCN*, which excludes query masking. Our experiments show that removing query masking improves the performance of Robust-MSCN in the two OOD scenarios (which is likely because we do not include PostgreSQL estimates in the query encoding). We report their results on CEB-1a-varied (which contains the most complex joins).

The goal of the experiments is *not* to beat the SOTA query-driven models but to validate the *practicality* of our theory. Specifically, we aim to show that SECONCDF, which is designed based on our theory, *reliably* improves upon existing NN-based query-driven models, and *consistently* outperforms the models derived from SOTA theory.

Evaluation Metrics. For accuracy, we use both RMSE and Qerror as the metrics. While our theory assumes absolute error as the loss function, we also evaluate Qerror (which is more critical in query optimization [40]) to demonstrate the effectiveness of SECONCDF. For query latency performance, we report the query running time.

Hardware. We train all NN models on an Amazon SageMaker ml.g4dn.xlarge node, and conduct latency experiments on an EC2 r5d.2xlarge node (8 core CPUs, 3.1GHz, 64G memory) for IMDb-small and CEB, and on an EC2 c5.9xlarge node (36 core CPUs, 3.1GHz, 72G memory) for DSB.

7.2 Accuracy

Figure 4 and Table 4 present the prediction accuracy on single-table and multi-table datasets, respectively. First, deep query-driven models (MSCN and LW-NN) demonstrate superior performance for In-Dist generalization across all datasets and consistently outperform all compared data-driven approaches on multi-table datasets. They perform comparably to PostgreSQL on single-table queries, where PostgreSQL is already effective. For PTSHIST and QUADHIST, despite that they outperform QUICKSEL and theoretically benefit from the SOTA theory, they may fail to match the empirical performance of the two deep query-driven models due to their limited model capacity, especially for Qerror (Figure 4b) since they are optimized specifically for RMSE. *These findings confirm the In-Dist generalization capability of deep learning-based query-driven models.*

However, deep learning-based models show limited robustness to OOD queries, especially in multi-table datasets with intricate joins and skewed distributions. For example, MSCN achieves strong In-Dist accuracy on the three multi-table datasets, but struggles with OOD generalization on IMDb-small, where it significantly underperforms compared to PostgreSQL and Sampling. On DSB, MSCN shows less vulnerability to query center shifts. This is likely due to the less skewed data distributions of DSB, allowing easier adaptation of selectivity functions across different data regions. PTSHIST, QUADHIST, and QUICKSEL do not show such drastic drops in OOD performance because they are induced by probability measures. This supports our OOD generalization theory, as signed measures

(a) RMSE   (b) Median Qerror
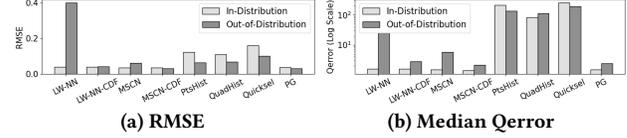
Figure 4: Accuracy on Census with granularity shifts.

Model	Query Center Move			Query Granularity Shift		
	RMSE	Qerror		RMSE	Qerror	
		Median	90%		Median	90%
PostgreSQL	0.054/0.062	22/6.8	$10^3/213$	0.088/0.038	23/8.3	$10^3/269$
Sampling	0.144/0.089	$10^3/10^4$	$10^5/10^5$	0.191/0.051	$10^4/10^3$	$10^5/10^5$
MSCN	0.012/0.045	1.1/1.9	1.5/42	0.014/0.117	1.1/5.8	1.3/53
Robust-MSCN	0.019/0.050	1.2/2.7	1.7/25	0.022/0.152	1.2/7.1	1.6/81
Robust-MSCN*	0.011/0.045	1.1/2.1	1.3/15	0.015/0.082	1.1/3.7	1.4/31
MSCN + CDF	0.010/0.019	1.1/1.6	1.4/8.7	0.012/0.012	1.1/1.5	1.3/5.4

Table 5: Accuracy on CEB-1a-varied (In-Dist/OOD).

are a superset of probability measures and thus fall within the scope of our theory.

The integration of SECONCDF significantly enhances the OOD generalization of query-driven models like MSCN, without compromising their In-Dist generalization. For instance, under query center moves, SECONCDF training reduces MSCN's median and 90-percentile Qerrors by 2-3 orders of magnitude.

More Joins. Table 5 shows the accuracy over CEB-1a-varied (featuring 9-way joins). We observe similar trends in the previous two multi-table datasets: SECONCDF significantly enhances MSCN's OOD performance despite the increased complexity.

Comparison with Robust-MSCN. Table 5 reveals that while Robust-MSCN* marginally improves upon MSCN, they are surpassed by SECONCDF. This is because Robust-MSCN is tailored for *different OOD scenarios* like new join templates and missing tables/columns, which are not the primary focus of this paper.

Point Queries. Figure 6 presents the OOD performance of MSCN and MSCN+CDF on IMDb-small for point queries. The results indicate that, by treating point queries as range queries, SECONCDF still enhances the OOD robustness of MSCN in these cases.

7.3 Query Latency Performance

In this subsection, we showcase the improved generalization capabilities from SECONCDF can result in a better end-to-end performance. All end-to-end experiments are conducted with a modified PostgreSQL 13.1 that can accept injected cardinalities estimates [20]. We exclude Sampling in the experiments since it is much worse than others. We compare MSCN+CDF to the original MSCN, PostgreSQL (an important baseline upon which learned cardinality estimation

Figure 5: Per-query latency performance on CEB-1a-varied under granularity shift. Left: In-Dist queries; Right: OOD queries.

(a) RMSE

(b) Median Qerror
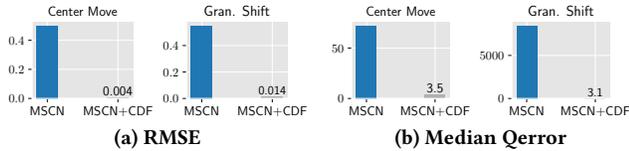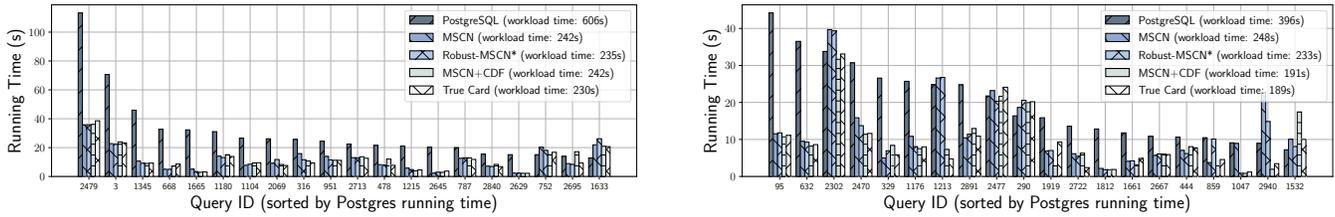
Figure 6: Accuracy of OOD point queries on IMDb-small.

(a) center move

(b) granularity shift

(c) center move

(d) granularity shift
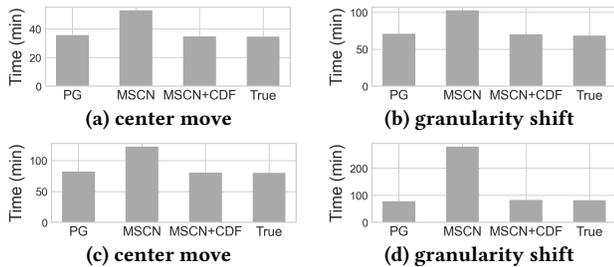
Figure 7: OOD query latency performance on IMDb-small (top two subfigures) and DSB (bottom two subfigures).

should improve), and True cardinalities. For each OOD scenario, we randomly sample 30 queries each from In-Dist and OOD test queries to conduct the latency experiments. The results for OOD queries are shown in Figure 7. Due to space constraints, we exclude In-Dist performance results, but we note that both the MSCN and MSCN+CDF demonstrate notably efficient running times for In-Dist queries, significantly surpassing PostgreSQL on IMDb-small and matching its performance on DSB. Indeed, they are close to True cardinalities on both datasets.

The OOD results yield two key insights. First, *the inaccurate cardinality estimates by MSCN for OOD queries lead to considerably poorer query latency performance compared to In-Dist queries.* Notably, MSCN's latency performance is significantly worse than PostgreSQL for both IMDb-small OOD queries. Second, the integration of SeConCDF significantly enhances MSCN+CDF's OOD latency performance, bringing it on par with PostgreSQL. This demonstrates that the improved accuracy from SeConCDF for OOD generalization can translate into enhanced runtime performance.

More Joins. To assess scalability, we conducted latency experiments on CEB-1a-varied, using the same approach to construct workloads of 20 In-Dist and OOD test queries (with 9-way joins) each (we observed consistent results across various sampled workloads). Figure 5 presents per-query latency with workload times indicated in the legend. All MSCN models significantly outperform

PostgreSQL, as traditional methods struggle with larger numbers of joins. Moreover, MSCN+CDF outperforms MSCN in OOD scenarios. Notably, SeConCDF reduces MSCN's running time significantly (by at least a factor of two) in 4 of the 20 OOD queries, with no substantial regressions. These results confirm that SeConCDF scales effectively to more joins. Additionally, consistent with the observations in § 7.2, SeConCDF outperforms Robust-MSCN in enhancing MSCN's query latency for the OOD scenarios discussed in the paper.

7.4 Efficiency

Training. SeConCDF uses pre-loading and asynchronous query sampling (parallelizing two loss computations) to minimize idle time during training. Training times per epoch are 50s for IMDb-small, 26s for DSB, and 63s for CEB. MSCN+CDF converges within 80 epochs for all datasets. While the training overheads are higher than MSCN, they are not costly. Additionally, since the training is performed offline, it does not impact real query performance.

Inference. Inference time is crucial for real query performance (including planning and execution). Since SeConCDF *does not change the model architecture or inference procedure*, the inference process remains efficient. On a CPU, the average processing time for each query (including subqueries) is 1ms for IMDb-small and DSB, and 14ms for CEB-1a-varied, negligible compared to execution times.

8 CONCLUSIONS AND OPEN PROBLEMS

In this paper, we proved the theory: selectivity predictors induced by a signed measure are learnable, and under mild assumptions, they exhibit **bounded OOD generalization error**. Based on the theory, we propose a new selectivity estimation paradigm NeuroCDF, and a principled training framework SeConCDF to enhance OOD generalization capabilities for *any* NN-based existing query-driven selectivity models. We empirically demonstrate that SeConCDF improves query-driven models' OOD generalization performance in terms of accuracy and query latency performance.

This work opens up many promising research directions. First, extending our theory beyond signed measures could provide new insights. Second, substituting the error function in our theory with Qerror presents an intriguing challenge. Furthermore, developing a unified generalization theory for all join graphs is also interesting.

ACKNOWLEDGMENTS

REFERENCES

[1] MSCN code. https://github.com/andreaskipf/learnedcardinalities. Last Accessed Date: 2025-04-05.

[2] PtsHist and QuadHist code. https://github.com/huxiao2010/Selectivity/. Last Accessed Date: 2025-04-05.

[3] ABITEBOUL, S., HULL, R., AND VIANU, V. *Foundations of databases*, vol. 8. Addison-Wesley Reading, 1995.

[4] ABOULNAGA, A., AND CHAUDHURI, S. Self-tuning histograms: building histograms without looking at data. In *Proceedings of the 1999 ACM SIGMOD international conference on Management of data* (1999), ACM, pp. 181–192.

[5] ALON, N., BEN-DAVID, S., CESA-BIANCHI, N., AND HAUSSLER, D. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM 44*, 4 (1997), 615–631.

[6] ANAGNOSTOPOULOS, C., AND TRIANTAFILLOU, P. Learning Set Cardinality in Distance Nearest Neighbours. In *2015 IEEE International Conference on Data Mining* (2015), IEEE, pp. 691–696.

[7] ANAGNOSTOPOULOS, C., AND TRIANTAFILLOU, P. Learning to accurately COUNT with query-driven predictive analytics. In *2015 IEEE International Conference on Big Data (Big Data)* (2015), IEEE, pp. 14–23.

[8] ANAGNOSTOPOULOS, C., AND TRIANTAFILLOU, P. Query-Driven Learning for Predictive Analytics of Data Subspace Cardinality. *ACM Transactions on Knowledge Discovery from Data 11*, 4 (2017), 1–46.

[9] BARTLETT, P. L., AND LONG, P. M. More theorems about scale-sensitive dimensions and learning. In *Proceedings of the eighth annual conference on Computational learning theory* (1995), ACM Press, pp. 392–401.

[10] BISHOP, C. M. *Pattern recognition and machine learning*. Information science and statistics. Springer, 2006.

[11] BRUNO, N., AND CHAUDHURI, S. Exploiting statistics on query expressions for optimization. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* (2002), ACM, pp. 263–274.

[12] BRUNO, N., CHAUDHURI, S., AND GRAVANO, L. STHoles: a multidimensional workload-aware histogram. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data* (2001), ACM, pp. 211–222.

[13] CHEN, C. M., AND ROUSSOPOULOS, N. Adaptive selectivity estimation using query feedback. In *Proceedings of the 1994 ACM SIGMOD international conference on Management of data* (1994), ACM, pp. 161–172.

[14] CHEN, T., AND GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), ACM, pp. 785–794.

[15] DEEDS, K. B., SUCIU, D., AND BALAZINSKA, M. SafeBound: A Practical System for Generating Cardinality Bounds. *Proceedings of the ACM on Management of Data 1*, 1 (2023), 1–26.

[16] DING, B., CHAUDHURI, S., GEHRKE, J., AND NARASAYYA, V. DSB: a decision support benchmark for workload-driven and traditional database systems. *Proceedings of the VLDB Endowment 14*, 13 (2021), 3376–3388.

[17] DURRETT, R. *Probability: theory and examples*, fifth edition ed. No. 49 in Cambridge series in statistical and probabilistic mathematics. Cambridge University Press, Cambridge ; New York, NY, 2019.

[18] DUTT, A., WANG, C., NARASAYYA, V., AND CHAUDHURI, S. Efficiently approximating selectivity functions using low overhead regression models. *Proceedings of the VLDB Endowment 13*, 12 (2020), 2215–2228.

[19] DUTT, A., WANG, C., NAZI, A., KANDULA, S., NARASAYYA, V., AND CHAUDHURI, S. Selectivity estimation for range predicates using lightweight models. *Proceedings of the VLDB Endowment 12*, 9 (2019), 1044–1057.

[20] HAN, Y., WU, Z., WU, P., ZHU, R., YANG, J., TAN, L. W., ZENG, K., CONG, G., QIN, Y., PFADLER, A., QIAN, Z., ZHOU, J., LI, J., AND CUI, B. Cardinality estimation in DBMS: a comprehensive benchmark evaluation. *Proceedings of the VLDB Endowment 15*, 4 (2021), 752–765.

[21] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), IEEE, pp. 770–778.

[22] HEDDES, M., NUNES, I., GIVARGIS, T., AND NICOLAU, A. Convolution and Cross-Correlation of Count Sketches Enables Fast Cardinality Estimation of Multi-Join Queries. *Proceedings of the ACM on Management of Data 2*, 3 (2024), 1–26.

[23] HELLMAN, M., AND RAVIV, J. Probability of error, equivocation, and the Chernoff bound. *IEEE Transactions on Information Theory 16*, 4 (1970), 368–372.

[24] HILPRECHT, B., SCHMIDT, A., KULESSA, M., MOLINA, A., KERSTING, K., AND BINNIG, C. DeepDB: learn from data, not from queries! *Proceedings of the VLDB Endowment 13*, 7 (2020), 992–1005.

[25] HU, X., LIU, Y., XIU, H., AGARWAL, P. K., PANIGRAHI, D., ROY, S., AND YANG, J. Selectivity Functions of Range Queries are Learnable. In *Proceedings of the 2022 ACM SIGMOD International Conference on Management of Data* (2022), ACM, pp. 959–972.

[26] IOANNIDIS, Y. E., AND CHRISTODOULAKIS, S. On the propagation of errors in the size of join results. In *Proceedings of the 1991 ACM SIGMOD international conference on Management of data* (1991), ACM Press, pp. 268–277.

[27] KEARNS, M. J., AND SCHAPIRE, R. E. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences 48*, 3 (1994), 464–497.

[28] KEARNS, M. J., VAZIRANI, U. V., AND VAZIRANI, U. V. *An introduction to computational learning theory*, 2. print ed. MIT Press, Cambridge, Mass., 1997.

[29] KIM, K., JUNG, J., SEO, I., HAN, W.-S., CHOI, K., AND CHONG, J. Learned Cardinality Estimation: An In-depth Study. In *Proceedings of the 2022 ACM SIGMOD International Conference on Management of Data* (2022), ACM, pp. 1214–1227.

[30] KIM, K., LEE, S., KIM, I., AND HAN, W.-S. ASM: Harmonizing Autoregressive Model, Sampling, and Multi-dimensional Statistics Merging for Cardinality Estimation. *Proceedings of the ACM on Management of Data 2*, 1 (2024), 1–27.

[31] KIPF, A., KIPF, T., RADKE, B., LEIS, V., BONCZ, P. A., AND KEMPER, A. Learned cardinalities: Estimating correlated joins with deep learning. In *9th Biennial Conference on Innovative Data Systems Research, CIDR 2019* (2019).

[32] LEIS, V., GUBICHEV, A., MIRCHEV, A., BONCZ, P., KEMPER, A., AND NEUMANN, T. How good are query optimizers, really? *Proceedings of the VLDB Endowment 9*, 3 (2015), 204–215.

[33] LI, P., WEI, W., ZHU, R., DING, B., ZHOU, J., AND LU, H. ALECE: An Attention-based Learned Cardinality Estimator for SPJ Queries on Dynamic Workloads. *Proceedings of the VLDB Endowment 17*, 2 (2023), 197–210.

[34] LIM, L., WANG, M., AND VITTER, J. S. SASH: A self-adaptive histogram set for dynamically changing workloads. In *Proceedings of 29th International Conference on Very Large Data Bases, VLDB 2003, Berlin, Germany, September 9-12, 2003* (2003), pp. 369–380.

[35] LIU, X., HAN, X., ZHANG, N., AND LIU, Q. Certified monotonic neural networks. In *Advances in Neural Information Processing Systems 33* (2020).

[36] LYNCH, C. A. Selectivity estimation and query optimization in large databases with highly skewed distribution of column values. In *Proceedings of the VLDB Endowment* (1988), pp. 240–251.

[37] MARKL, V., HAAS, P. J., KUTSCH, M., MEGIDDO, N., SRIVASTAVA, U., AND TRAN, T. M. Consistent selectivity estimation via maximum entropy. *The VLDB Journal 16*, 1 (2006), 55–76.

[38] MARKL, V., LOHMAN, G. M., AND RAMAN, V. LEO: An autonomic query optimizer for DB2. *IBM Systems Journal 42*, 1 (2003), 98–106.

[39] MENG, Z., CAO, X., AND CONG, G. Selectivity Estimation for Queries Containing Predicates over Set-Valued Attributes. *Proceedings of the ACM on Management of Data 1*, 4 (2023), 1–26.

[40] MOERKOTTE, G., NEUMANN, T., AND STEIDL, G. Preventing bad plans by bounding the impact of cardinality estimation errors. *Proceedings of the VLDB Endowment 2*, 1 (2009), 982–993.

[41] NEGI, P., MARCUS, R., KIPF, A., MAO, H., TATBUL, N., KRASKA, T., AND ALIZADEH, M. Flow-loss: learning cardinality estimates that matter. *Proceedings of the VLDB Endowment 14*, 11 (2021), 2019–2032.

[42] NEGI, P., WU, Z., KIPF, A., TATBUL, N., MARCUS, R., MADDEN, S., KRASKA, T., AND ALIZADEH, M. Robust Query Driven Cardinality Estimation under Changing Workloads. *Proceedings of the VLDB Endowment 16*, 6 (2023), 1520–1533.

[43] PARK, Y., ZHONG, S., AND MOZAFARI, B. QuickSel: Quick Selectivity Learning with Mixture Models. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data* (2020), ACM, pp. 1017–1033.

[44] POESS, M., SMITH, B., KOLLAR, L., AND LARSON, P. TPC-DS, taking decision support benchmarking to the next level. In *Proceedings of the 2002 ACM SIGMOD international conference on Management of data* (2002), ACM, pp. 582–587.

[45] REINER, S., AND GROSSNIKLAUS, M. Sample-Efficient Cardinality Estimation Using Geometric Deep Learning. *Proceedings of the VLDB Endowment 17*, 4 (2023), 740–752.

[46] ROBERTS, F. S., AND TESMAN, B. *Applied combinatorics*, second edition ed. CRC Press, 2009.

[47] SELINGER, P. G., ASTRAHAN, M. M., CHAMBERLIN, D. D., LORIE, R. A., AND PRICE, T. G. Access path selection in a relational database management system. In *Proceedings of the 1979 ACM SIGMOD international conference on Management of data* (1979), ACM Press, p. 23.

[48] STEIN, E. M. *Real Analysis: Measure Theory, Integration, and Hilbert Spaces*, 1st ed ed. Princeton University Press, 2009.

[49] STILLGER, M., LOHMAN, G. M., MARKL, V., AND KANDIL, M. LEO - db2's learning optimizer. In *Proceedings of the VLDB Endowment* (2001), pp. 19–28.

[50] SUN, J., AND LI, G. An end-to-end learning-based cost estimator. *Proceedings of the VLDB Endowment 13*, 3 (2019), 307–319.

[51] WANG, F., YAN, X., YIU, M. L., LI, S., MAO, Z., AND TANG, B. Speeding Up End-to-end Query Execution via Learning-based Progressive Cardinality Estimation. *Proceedings of the ACM on Management of Data 1*, 1 (2023), 1–25.

[52] WONG, E., AND YOUSSEFI, K. Decomposition—a strategy for query processing. *ACM Transactions on Database Systems 1*, 3 (1976), 223–241.

[53] WU, C., JINDAL, A., AMIZADEH, S., PATEL, H., LE, W., QIAO, S., AND RAO, S. Towards a learning optimizer for shared clouds. *Proceedings of the VLDB Endowment 12*, 3 (2018), 210–222.

[54] WU, P., AND CONG, G. A Unified Deep Model of Learning from both Data and Queries for Cardinality Estimation. In *Proceedings of the 2021 International Conference on Management of Data* (2021), ACM, pp. 2009–2022.

[55] WU, P., AND IVES, Z. G. Modeling Shifting Workloads for Learned Database

Systems. *Proceedings of the ACM on Management of Data 2*, 1 (2024), 1–27.

[56] Wu, Z., Negi, P., Alizadeh, M., Kraska, T., and Madden, S. FactorJoin: A New Cardinality Estimation Framework for Join Queries. *Proceedings of the ACM on Management of Data 1*, 1 (2023), 1–27.

[57] Xiu, H., Agarwal, P. K., and Yang, J. PARQO: Penalty-Aware Robust Plan Selection in Query Optimization. *Proceedings of the VLDB Endowment 17*, 13 (2024), 4627–4640.

[58] Yang, Z., Kamsetty, A., Luan, S., Liang, E., Duan, Y., Chen, X., and Stoica, I. NeuroCard: one cardinality estimator for all tables. *Proceedings of the VLDB Endowment 14*, 1 (2020), 61–73.

[59] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM 64*, 3 (2021), 107–115.

[60] Zhang, K., Wang, H., Lu, Y., Li, Z., Shu, C., Yan, Y., and Yang, D. Duet: Efficient and Scalable Hybrid Neural Relation Understanding. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)* (2024), IEEE, pp. 56–69.