Coresets for Robust Query Optimization*

RAHUL RAYCHAUDHURY, Department of Computer Science, Duke University, USA
HAIBO XIU, Department of Computer Science, Duke University, USA
PANKAJ K. AGARWAL, Department of Computer Science, Duke University, USA
STAVROS SINTOS, Department of Computer Science, University of Illinois Chicago, USA
JUN YANG, Department of Computer Science, Duke University, USA

Query optimizers must efficiently choose a query plan using noisy estimates of cardinalities. We study robust query optimization, where the optimizer is made aware of the uncertainty in cardinality estimates and needs to select the most "robust" plan. A key empirical observation is that a small set of plans often suffices as robust plan candidates for all queries following a given template. We formalize this phenomenon by introducing coresets of plans and extend this notion to robust coresets, which incorporate uncertainty. We prove positive and negative results on the sizes of such coresets for common cost-function classes. Because our lower bounds suggest that the size of a coreset can be large in the worst case, we exploit the fact that in practice queries arise from workload distributions, rather than being arbitrary. We present algorithms that construct workload-aware coresets whose size and performance closely match those of the optimal workload-specific coresets.

CCS Concepts: • **Theory of computation → Database query processing and optimization (theory)**.

Additional Key Words and Phrases: coreset, query optimization, query planning, robustness

ACM Reference Format:
Rahul Raychaudhury, Haibo Xiu, Pankaj K. Agarwal, Stavros Sintos, and Jun Yang. 2026. Coresets for Robust Query Optimization. *Proc. ACM Manag. Data* 4, 2 (PODS), Article 100 (May 2026), 23 pages. https://doi.org/10.1145/3801896

1 Introduction

Query optimization is a foundational problem in database systems, where the goal is to compile a declarative query into an efficient execution plan [40]. This optimization is guided by a cost function that predicts the runtime cost of a candidate plan using estimates of the cardinalities of intermediate results produced by various subplans (or, equivalently, their selectivities over the cross products of their input tables). A central and long-standing challenge in query optimization is coping with inaccurate estimates, which can lead the optimizer to choose plans whose execution costs are orders of magnitude higher than optimal. While improving estimator accuracy remains an active area of research, estimation uncertainty is inherent, motivating the study of robust optimization techniques that select plans resilient to such uncertainty.

*This work was supported by NSF grants CCF-2223870, IIS-2402823, IIS-2348919, and a US-Israel Binational Science Foundation Grant 2022131.

Authors' Contact Information: Rahul Raychaudhury, Department of Computer Science, Duke University, Durham, USA, rahul.raychaudhury@duke.edu; Haibo Xiu, Department of Computer Science, Duke University, Durham, USA, haibo.xiu@duke.edu; Pankaj K. Agarwal, Department of Computer Science, Duke University, Durham, USA, pankaj@cs.duke.edu; Stavros Sintos, Department of Computer Science, University of Illinois Chicago, Chicago, USA, stavros@uic.edu; Jun Yang, Department of Computer Science, Duke University, Durham, USA, junyang@cs.duke.edu.

ACM 2836-6573/2026/5-ART100
https://doi.org/10.1145/3801896

Instead of using point estimates from an estimator, we can leverage modern learned estimators' ability to provide measures of uncertainty (e.g., [34]) or profile the errors produced by the estimator (e.g., [49, 50]). Such information about uncertainty can be modeled as a predictive distributions over the cardinality space, where each dimension corresponds to the cardinality of a subquery relevant to the query. This capability enables a principled stochastic formulation of "robustness," in which we seek plans that minimize objectives such as the expected competitive ratio or the probability of exceeding the point-wise optimal cost. However, finding robust plans comes at a computational cost. Evaluating plans under these robust objectives requires integration over a high-dimensional distribution, and searching for the optimal robust plan is a challenging stochastic optimization problem over an exponentially large plan space. A classical strategy to mitigate online optimization cost is *parametric query optimization* (*PQO*) [30], which precomputes a compact coreset of plans designed to cover the cardinality space. This raises several foundational theoretical questions: *Do small, general-purpose coresets exist for modern robustness objectives? If so, how can we compute them efficiently, and how can they enable fast, provably robust plan selection at runtime?*

To address these questions, we initiate a systematic study of coresets for robust query optimization. We model plan costs as functions of cardinalities and represent uncertainty in cardinality estimates as distributions over the cardinality space. Within this framework, we consider two robustness objectives: minimizing risk (measured as the probability of exceeding the optimal cost by a prescribed factor) and minimizing expected (competitive) penalty (measured as the relative slowdown compared with point-wise optimal plans).[1] We formalize the notion of coresets under the aforementioned robustness objectives and establish combinatorial bounds on their worst-case complexity. Our primary focus, however, is on a more realistic setting in which queries arise from an underlying workload distribution. In this setting, we present efficient algorithms for computing near-optimal workload-dependent coresets and for inferring robust plans from these coresets at runtime.

Our algorithms incorporate and build on several heuristics from query optimization that were previously evaluated primarily empirically (e.g., [20, 43, 49–51]), and a key aspect of our work is that it provides mathematical foundations for these approaches and establishes provable guarantees on their performance.

2 Framework and Contributions

In this section, we present our framework for robust query optimization and coresets, and summarize our main results.

2.1 Our Framework

Consider a query template Γ with parameter set $\mathsf{params}(\Gamma)$, whose elements serve as placeholders for query constants drawn from appropriate domains. A query Q following Γ assigns specific values to the parameters in $\mathsf{params}(\Gamma)$. We call the set $\mathbb{Q}$ of possible queries following Γ the *query space* of Γ.

Let Π denote the set of all possible plans for queries in $\mathbb{Q}$. In database systems, the cost of a plan $\pi \in \Pi$ for a given query $Q \in \mathbb{Q}$ is modeled as a function of the cardinalities of intermediate results required to compute Q [17]. Define the *cardinality space* with respect to Γ by $\mathbb{S} := \mathbb{R}^d_{\geq 0}$, where each dimension corresponds to an intermediate result (defined by parameterized subqueries under Γ) that can be produced by executing some plan in Π. Each plan $\pi \in \Pi$ has an associated *cost function* $\mathsf{Cost}(\pi, \mathbf{s}) : \Pi \times \mathbb{S} \to \mathbb{R}^+$ that returns the cost of executing plan π when the sizes of the intermediate results are given by $\mathbf{s} \in \mathbb{S}$. For a cardinality vector $\mathbf{s}$, let $\pi^\star(\mathbf{s}) := \arg\min_{\pi \in \Pi} \mathsf{Cost}(\pi, \mathbf{s})$ denote the best plan w.r.t. $\mathbf{s}$, and let $\mathsf{Cost}^\star(\mathbf{s}) := \mathsf{Cost}(\pi^\star(\mathbf{s}), \mathbf{s})$ denote its cost.

[1]Although we focus on these two objectives, our framework applies more broadly to other related robustness objectives.

For a query $Q \in \mathbb{Q}$, we define its *true cardinality vector* $\mathbf{s}^\star_Q \in \mathbb{S}$ as the vector of cardinalities of all intermediate results when Q is evaluated on the underlying database. The *true cost* of executing plan $\pi \in \Pi$ for Q is $\mathsf{Cost}(\pi, \mathbf{s}^\star_Q)$. In query optimization, given a query Q, we would like to compute the optimal plan $\pi^\star(\mathbf{s}^\star_Q)$. However, in practice, $\mathbf{s}^\star_Q$ is unknown when optimizing Q, and the database optimizer uses an *estimated cardinality vector* $\hat{\mathbf{s}}_Q$. A major challenge in query optimization is that when $\hat{\mathbf{s}}_Q$ differs from $\mathbf{s}^\star_Q$, the resulting cost $\mathsf{Cost}(\pi^\star(\hat{\mathbf{s}}_Q), \mathbf{s}^\star_Q)$—that is, the cost of the plan chosen using the estimated cardinalities when evaluated under the true cardinalities—can significantly exceed $\mathsf{Cost}^\star(\mathbf{s}^\star_Q) = \mathsf{Cost}(\pi^\star(\mathbf{s}^\star_Q), \mathbf{s}^\star_Q)$, the optimal cost under the true cardinalities.

Robust Query Optimization. We assume that the cardinality estimator returns not merely a point estimate $\hat{\mathbf{s}}_Q$ for a given query Q, but instead a probability distribution U_Q over the cardinality space $\mathbb{S}$ that captures the uncertainty about the true cardinality vector $\mathbf{s}^\star_Q$. We call U_Q an *uncertainty profile*. We do not assume any specific structure for U_Q, nor do we require uncertainty profiles to be identical across queries; we only assume that *samples can be drawn from* U_Q. Such uncertainty profiles are widely used [1, 8, 16, 47–50] and increasingly available from learned cardinality estimators that output predictive distributions rather than point estimates [38, 52].

In robust query optimization (RQO), we aim to identify plans that perform well across possible cardinality realizations using the additional available information from the uncertainty profile. We first define the *(competitive) penalty* of a plan $\pi \in \Pi$ at cardinality vector $\mathbf{s} \in \mathbb{S}$ as

$$\mathsf{Pnl}(\pi, \mathbf{s}) := \frac{\mathsf{Cost}(\pi, \mathbf{s})}{\mathsf{Cost}^\star(\mathbf{s})}.$$

We then consider two principal notions of robustness.

(1) **Risk**: Given some threshold $\tau \geq 1$, the τ*-risk* of a plan π for query Q is defined as

$$\mathsf{Risk}(\pi, Q\,;\,\tau) := \Pr_{\mathbf{s}' \sim \mathrm{U}_Q}[\mathsf{Pnl}(\pi, \mathbf{s}') > \tau].$$

In other words, it is the probability that π incurs a penalty exceeding τ under the uncertainty profile U_Q. We denote by $\mathsf{Risk}^\star(Q\,;\,\tau) := \min_{\pi \in \Pi} \mathsf{Risk}(\pi, Q\,;\,\tau)$ the optimal risk value, and by $\pi^\star_{\mathsf{Risk}}(Q\,;\,\tau) := \arg\min_{\pi \in \Pi} \mathsf{Risk}(\pi, Q\,;\,\tau)$ the plan achieving it. When the threshold τ is clear from the context, we drop the parameter τ and simply write $\mathsf{Risk}(\pi, Q)$, $\mathsf{Risk}^\star(Q)$, and $\pi^\star_{\mathsf{Risk}}(Q)$.

(2) **Expected penalty**: The *expected penalty* of a plan π for a query Q is defined by

$$\mathsf{EPnl}(\pi, Q) := \mathrm{E}_{\mathbf{s}' \sim \mathrm{U}_Q}[\mathsf{Pnl}(\pi, \mathbf{s}')].$$

Given a query Q, we write $\mathsf{EPnl}^\star(Q) := \min_{\pi \in \Pi} \mathsf{EPnl}(\pi, Q)$ for the optimal expected penalty under U_Q, and by $\pi^\star_{\mathsf{EPnl}}(Q) := \arg\min_{\pi \in \Pi} \mathsf{EPnl}(\pi, Q)$ the plan achieving it.

Given a query Q, our goal is to efficiently compute the plan that minimizes the objectives above.

Coresets of plans. Query optimization is NP-hard even without robustness considerations [13, 17, 29], motivating extensive prior work in *parametric query optimization* [14, 20, 22, 24, 30, 50]. PQO precomputes and caches a small set of candidate plans per query template; at runtime, a plan is selected from this cache, avoiding costly optimizer invocations.

We formalize this cache selection problem through *coresets of plans*, a concept that has been used broadly in optimization problems [3, 37] and query processing [4, 6, 54]. For a query template Γ, a coreset $\mathsf{M} \subseteq \Pi$ is a small subset of plans that contains a near-optimal plan for any query $Q \in \mathbb{Q}$ under a chosen objective function. Adapting the classical coreset notion to query plans, we say that for $\alpha \geq 1$, a subset $\mathsf{M} \subseteq \Pi$ is an *α-coreset for cost* if for any cardinality vector $\mathbf{s} \in \mathbb{S}$,[2]

$$\min_{\pi \in \mathsf{M}} \mathsf{Cost}(\pi, \mathbf{s}) \leq \alpha \, \mathsf{Cost}^{\star}(\mathbf{s}).$$

Our focus is on coresets for robust optimization. We study two variants corresponding to our two robustness objectives.

(1) **Coresets for risk.** For $\tau \geq 1$ and $\varepsilon \in (0, 1)$, a subset $\mathsf{M} \subseteq \Pi$ is an *(ε, τ)-coreset for risk* if, for any query $Q \in \mathbb{Q}$, $\min_{\pi \in \mathsf{M}} \mathsf{Risk}(\pi, Q\,;\, \tau) \;\leq\; \mathsf{Risk}^{\star}(Q\,;\, \tau) + \varepsilon$.
(2) **Coresets for expected penalty.** For $\alpha \geq 1$, a subset of plans $\mathsf{M} \subseteq \Pi$ is an *α-coreset for expected penalty* if, for any query $Q \in \mathbb{Q}$, $\min_{\pi \in \mathsf{M}} \mathsf{EPnl}(\pi, Q) \;\leq\; \alpha \, \mathsf{EPnl}^{\star}(Q)$.

The above notions of coreset are strong as they require worst-case guarantees for all queries in $\mathbb{Q}$. In practice, real queries are often drawn from a subset within $\mathbb{Q}$, governed by the query workload. We therefore focus on *workload-dependent* coresets, defined below. We call a probability distribution over the query space $\mathbb{Q}$ a *workload distribution.*

Let W be a workload distribution and let $\delta \in (0, 1)$ be a *confidence* parameter. A set of plans $\mathsf{M} \subseteq \Pi$ is a *workload-dependent $(\varepsilon, \tau, \delta)$-coreset for risk* or a *workload-dependent (α, δ)-coreset for expected penalty* w.r.t. W if, for any random $Q \sim \mathsf{W}$, with probability at least $1 - \delta$,

$$\min_{\pi \in \mathsf{M}} \mathsf{Risk}(\pi, Q\,;\, \tau) \;\leq\; \mathsf{Risk}^{\star}(Q\,;\, \tau) + \varepsilon \quad \text{and} \quad \min_{\pi \in \mathsf{M}} \mathsf{EPnl}(\pi, Q) \;\leq\; \alpha \, \mathsf{EPnl}^{\star}(Q).$$

As a shorthand, we denote them as $(\varepsilon, \tau, \delta|\mathsf{W})$-coreset for risk and $(\alpha, \delta|\mathsf{W})$-coreset for expected penalty, respectively.

2.2 Our Contributions

We begin by proving worst-case bounds on the size of coresets. However, our main contribution is the design of algorithms for computing workload-dependent coresets of plans with respect to robustness objectives, and for inferring robust plans at runtime using these coresets. We divide our results into two categories: *worst-case results* and *workload-dependent results.* In the main body of the paper, we focus on the workload-dependent results, while presenting the worst-case results in the appendix.

Worst-case results. Here we focus on cost coresets, as they are simpler to analyze than coresets for robustness objectives. In general, the minimum size of a coreset of plans with respect to an objective function depends on the query template Γ and the cost model, making general bounds difficult without template-specific assumptions. To obtain template-agnostic worst-case guarantees, we instead focus on families of cost functions commonly used to model plan costs. In practice, these are typically low-degree polynomials, with linear functions being particularly common [28, 47, 48].

We consider a simplified setting where the cost function of a plan is a linear function over $\mathbb{S} = \mathbb{R}^{d}_{\geq 0}$ with coefficients in $[1, M]$. We construct a set of cost functions such that, in the worst case, any α-coreset for cost must have size at least $\Omega\left(\frac{1}{\sqrt{d}} \log^{d-1}_{\alpha \cdot d} M\right)$ in the worst case, showing that the curse of dimensionality is unavoidable even in this simplified setting. We also prove a nearly matching upper bound, showing that there always exists an α-coreset for cost of size $O(\log^{d}_{\alpha} M)$, and extend the upper bound to polynomial cost functions with bounded degree (Theorem A.1, Appendix A). Moreover, given any set Π of plans whose cost functions are linear with coefficients in $[1, M]$, we design an algorithm that computes a coreset for cost whose size is comparable to the instance-optimal coreset (Theorem A.5, Appendix A).

[2]We note that we define the coreset in terms of all possible cardinality vectors overall, which is a stronger condition than requiring that M approximates the costs for all $Q \in \mathbb{Q}$ and all possible $\mathbf{s}_Q$. This stronger notion is convenient in that we do not need to explicitly require that every selectivity vector $\mathbf{s}$ sampled from U_Q is realizable.

Workload-dependent results. The above worst-case bounds contrast with experimental studies showing that small coresets often exist in practice, even for robust objectives [18–20, 50]. This gap arises for several reasons. In practice, the query template significantly restricts the space of feasible plans. Moreover, the arbitrary cost functions used in the lower-bound constructions may not correspond to legitimate plans. Most importantly, queries typically arise from an underlying workload distribution rather than from the entire space of possibilities. Consequently, an effective coreset only needs to cover the region of the query space induced by this workload distribution, which can be much smaller than the worst-case space. This raises the following question: *suppose there exists a small workload-dependent coreset for expected penalty or risk; can we efficiently compute a robust coreset of comparable size and use it to select robust plans?*

Robust coreset construction. Our main algorithmic contribution is a technique for computing near-optimal-size workload-dependent (robust) coresets. We present the algorithm for the risk objective, but the approach is general and extends with minor modifications to cost, expected penalty, and other robustness notions. The algorithm is also oblivious to the form of the cost function, requiring only non-negativity and no structural assumptions such as linearity.

Specifically, let $\mathbb{Q}$ denote the query space and Π the corresponding set of plans for a query template. Let $|\Pi| = N$. Let W be a workload distribution over $\mathbb{Q}$. For parameters $\varepsilon \in (0, 1]$, $\tau \geq 1$, and $\delta \in (0, 1)$, let κ^* denote the size of the smallest $(\varepsilon, \tau, \delta|\mathsf{W})$-coreset for risk. We present (Section 3) an algorithm, AlgRiskCoreset, that observes $\widetilde{O}(\kappa^*)$ queries sampled from W and computes a set of plans $C \subseteq \Pi$ of size $\widetilde{O}(\kappa^*)$ such that, with high probability, C is a $(3\varepsilon, \tau, 3\delta|\mathsf{W})$-coreset for risk (Theorem 3.5).[3] For clarity, we present the algorithm in the setting where the plan set Π and their cost functions are explicitly available, and $\mathsf{Cost}(\pi, \mathbf{s})$ can be evaluated in constant time for any plan π. This corresponds to scenarios where plans are enumerated offline for a query template or obtained from historical workloads [22, 24, 43]. In this setting the algorithm runs in $\widetilde{O}(\kappa^* N)$ time. We also extend our algorithm to a setting where plans are not explicitly enumerated, but instead are obtained by solving a Mixed-Integer Linear Programming (MILP) optimization problem, which implicitly defines the feasible set of plans (Section 3.3).

Robust plan selection using coresets. We present (Section 4) an inference procedure LocalInference that, given a set of plans C and a query Q, picks a plan among C. We show that when C is an $(\varepsilon, \tau, \delta|\mathsf{W})$-coreset for risk and $Q \sim \mathsf{W}$, the procedure returns a plan $\pi \in C$ such that $\mathsf{Risk}(\pi, Q\,;\,\tau^2) \leq 2\,\mathsf{Risk}^\star(Q\,;\,\tau) + O(\varepsilon)$ with probability at least $1 - O(\delta)$ (Theorem 4.1). In other words, if the optimal τ-risk is $p = \mathsf{Risk}^\star(Q\,;\,\tau)$, then the procedure returns a plan whose τ^2-risk is at most $2p + O(\varepsilon)$. The procedure requires near-linear time in the size of C, i.e., $\tilde{O}(\kappa^*)$.

We also show how to augment a given $(\varepsilon, \tau, \delta|\mathsf{W})$-coreset C for risk with an additional set of plans C' so as to strictly improve the quality of inference. Specifically, we present a procedure that, given parameters $\alpha > 1$ and $\delta \in (0, 1)$, produces a set of plans C' such that for any query $Q \sim \mathsf{W}$, LocalInference$(C \cup C', Q)$ returns a plan $\pi \in C \cup C'$ such that $\mathsf{Risk}(\pi, Q\,;\,\tau\alpha) \leq \mathsf{Risk}^\star(Q\,;\,\tau) + O(\varepsilon)$ with probability at least $1 - O(\delta)$ (Theorem 4.5, Lemma 4.7 and Theorem 4.8). To achieve this, we introduce the notion of a *neighborhood-cost coreset* and show that C' is a small instance of such a coreset, keeping the augmented set compact (Section 4.3).

3 Algorithm for Computing Coresets for Risk

In this section, we present an algorithm, called AlgRiskCoreset, for learning a workload-dependent coreset for risk from samples of the query workload. Throughout this section, we assume a fixed query template Γ. Let $\mathbb{Q}$ denote the query space and Π the set of plans corresponding to Γ. Let $|\Pi| = N$. We assume queries are drawn from a workload distribution W over $\mathbb{Q}$.

[3] The $\widetilde{O}(\cdot)$ notation suppresses polylogarithmic factors in N and constants depending on parameters such as ε, τ, and δ.

The algorithm takes as input a confidence parameter $\delta \in (0,1)$, a penalty threshold $\tau \geq 1$, and an accuracy parameter $\varepsilon \in (0,1)$. It does not require explicit knowledge of the workload distribution W, but instead draws samples from W. The algorithm returns a $(c\varepsilon, \tau, c'\delta \mid \mathrm{W})$-coreset for risk of Π, of size $O(\kappa^* \log \delta^{-1})$, where κ^* is the optimal size of a $(\varepsilon, \tau, \delta \mid \mathrm{W})$-coreset for risk, and $c, c' > 1$ are constants that can be chosen arbitrarily close to 1.

We first present AlgRiskCoreset under the *explicit setting*, in which the algorithm is given the set Π of all plans and the cost function $\mathsf{Cost}(\pi, \cdot)$ for every plan $\pi \in \Pi$ (Sections 3.1 and 3.2). Next, we discuss how to adapt AlgRiskCoreset to the *implicit setting*, where the plan set Π is defined via an MILP (Section 3.3).

3.1 AlgRiskCoreset

Let κ^* denote the size of the smallest $(\varepsilon, \tau, \delta|\mathrm{W})$−coreset for risk. Here is the main idea of the algorithm. We choose a random sample $\widetilde{\mathcal{W}}$ of $\widetilde{O}(\kappa^*/\varepsilon^2)$ queries from the workload distribution W. We say that a query $Q \in \widetilde{\mathcal{W}}$ is *covered* by a plan $\pi \in \Pi$ if $\mathsf{Risk}(\pi, Q) \leq \mathsf{Risk}^\star(Q) + \varepsilon$. We use a greedy maximum-coverage algorithm to construct the desired coreset C of size $O(\kappa^* \ln \delta^{-1})$. At each step, it chooses a plan $\pi \in \Pi$ that covers the maximum number of "uncovered queries" of $\widetilde{\mathcal{W}}$. We repeat this step $O(\kappa^* \ln \delta^{-1})$ times. There are, however, two challenges in implementing this algorithm: we do not know the value of κ^*, and computing $\mathsf{Risk}(\pi, Q)$ is infeasible since we do not know U_Q precisely. We therefore guess the value of κ^* and estimate $\mathsf{Risk}(\pi, Q)$ by drawing samples from U_Q. We now describe the algorithm in detail.

The algorithm proceeds in rounds. At the beginning of round r, it sets a parameter $k_r := 2^{r-1}$, which is the current guess for κ^*. A round works as follows:

(1) **[Workload sampling]** Set $\mathfrak{n}_r = (c_1 k_r/\delta^2)\ln(N)\ln(\delta^{-1})$ for some constant $c_1 > 0$. We choose a random set $\widetilde{\mathcal{W}}_r = \{Q_1, \ldots, Q_{\mathfrak{n}_r}\}$ of $\mathfrak{n}_r$ queries, where each Q_i is drawn independently and identically.

(2) **[Risk estimation]** Set $\mathfrak{m}_r = (c_2/\varepsilon^2)\ln(N)\ln(\mathfrak{n}_r)$ for some constant $c_2 > 0$. For each query $Q_i \in \widetilde{\mathcal{W}}_r$, draw a set $\mathcal{Z}_i = \{\mathbf{s}_{i,1}, \ldots, \mathbf{s}_{i,\mathfrak{m}_r}\}$ cardinality vectors according to its uncertainty profile U_{Q_i}. Next, for each Q_i and $\pi \in \Pi$, compute the *proxy risk*, $\widetilde{\mathsf{Risk}}(\pi, Q_i)$, which is the fraction of cardinality vectors in $\mathcal{Z}_i$ for which the penalty of π exceeds τ, i.e.,

$$\widetilde{\mathsf{Risk}}(\pi, Q_i) := \frac{1}{\mathfrak{m}_r}\sum_{j=1}^{\mathfrak{m}_r} \mathbf{1}\{\mathsf{Pnl}(\pi, \mathbf{s}_{i,j}) > \tau\}. \tag{1}$$

The algorithm makes two passes over Π to compute $\mathsf{Pnl}(\pi, \mathbf{s}_{i,j})$. First, it computes $\mathsf{Cost}(\pi, \mathbf{s}_{i,j})$ for every plan π, which also gives $\mathsf{Cost}^\star(\mathbf{s}_{i,j})$. In the second pass, it computes $\mathsf{Pnl}(\pi, \mathbf{s}_{i,j})$ for every π.

(3) **[Greedy cover]** For each plan $\pi \in \Pi$, we define its cover to be the set of queries Q_i for which $\widetilde{\mathsf{Risk}}(\pi, Q_i)$ exceeds the optimal proxy risk by at most 2ε. More precisely,

$$\mathsf{Cov}(\pi) := \left\{Q_i \in \widetilde{\mathcal{W}}_r : \widetilde{\mathsf{Risk}}(\pi, Q_i) \leq \min_{\pi' \in \Pi} \widetilde{\mathsf{Risk}}(\pi', Q_i) + 2\varepsilon\right\}.$$

We now run the greedy maximum coverage algorithm to choose the coreset C. Set the *budget* $\mathbf{b}_r := c_3 k_r \ln \delta^{-1}$ for a suitable constant $c_3 > 0$, and initialize $C = \emptyset$. Iteratively select the plan $\pi^* = \arg\max_{\pi \in \Pi} \mathsf{Cov}(\pi)$ with the largest coverage, add π^* to C and set $\mathsf{Cov}(\pi) = \mathsf{Cov}(\pi) \setminus \mathsf{Cov}(\pi^*)$ for all $\pi \in \Pi$. We continue until either all of $\widetilde{\mathcal{W}}_r$ is covered or $|C| = \mathbf{b}_r$.

(4) **[Validation]** If $|\bigcup_{\pi \in C} \mathsf{Cov}(\pi)| \geq (1 - 2\delta)\,\mathfrak{n}_r$, accept C and return; otherwise, set $r \leftarrow r + 1$ and return to Step (1).

3.2 Analysis of AlgRiskCoreset

In this subsection, we prove that with probability $1 - N^{-O(1)}$, AlgRiskCoreset outputs a $(3\varepsilon, \tau, 3\delta \mid \mathrm{W})$−coreset for risk.

Recall that for the algorithm to terminate in a round r, it must return a set of plans $C \subseteq \Pi$ of size $|C| \le \mathbf{b}_r = c_3 k_r \ln \delta^{-1}$. For any $\varepsilon' \in (0,1)$ and for a set of plans $\mathcal{F} \subseteq \Pi$, let $\mathrm{UC}(\varepsilon', \mathcal{F}) \subseteq \mathbb{Q}$ denote the set of *uncovered* queries by $\mathcal{F}$, i.e., $Q \in \mathbb{Q}$ for which $\mathcal{F}$ does not contain a plan whose risk is within ε' of $\mathsf{Risk}^\star(Q)$. Formally,

$$\mathrm{UC}(\varepsilon', \mathcal{F}) := \{Q \in \mathbb{Q} \mid \min_{\pi \in \mathcal{F}} \mathsf{Risk}(\pi, Q) > \mathsf{Risk}^\star(\mathrm{U}) + \varepsilon'\}.$$

Define the *ε'-failure probability* of $\mathcal{F}$, denoted by $p(\varepsilon', \mathcal{F})$, as

$$p(\varepsilon', \mathcal{F}) = \Pr_{Q \sim \mathrm{W}}[Q \in \mathrm{UC}(\varepsilon', \mathcal{F})]$$

Recall that at the beginning of round r, the algorithm samples a set $\widetilde{\mathcal{W}}_r$ of queries from W. Define the random variable

$$\widetilde{p}_r(\varepsilon', \mathcal{F}) := \frac{1}{\mathfrak{n}_r} \sum_{Q \in \widetilde{\mathcal{W}}_r} \mathbf{1}\big(Q \in \mathrm{UC}(\varepsilon', \mathcal{F})\big)$$

as the *proxy ε'-failure probability.*

Overview of the analysis. Fix a round r, the analysis proceeds as follows. Lemmas 3.1 and 3.2 show that $\widetilde{p}_r$ and $\widetilde{\mathsf{Risk}}$ are accurate estimates of failure probability and risk, respectively. The latter implies that the greedy cover procedure defined using proxy risks behaves almost as if it had access to the true risks on the sampled queries.

Using these two lemmas, we prove two structural claims about the algorithm. In Lemma 3.3, we show that whenever the algorithm terminates in round r, the set C returned in that round satisfies $p(3\varepsilon, C) \le 3\delta$, implying that C is a valid $(3\varepsilon, \tau, 3\delta|\mathrm{W})$-coreset. Finally, Lemma 3.4 shows that once the guess k_r is at least κ^*, the greedy maximum-coverage step succeeds within budget and the algorithm terminates.

Lemma 3.1. *Let $\varepsilon' \in (0,1)$ and $r \ge 1$. Then $\left|\widetilde{p}_r(\varepsilon', \mathcal{F}) - p(\varepsilon', \mathcal{F})\right| \le \delta/4$ holds simultaneously for all subsets $\mathcal{F} \subseteq \Pi$ of size at most $\mathbf{b}_r$ with probability at least $1 - N^{-O(1)}$.*

Proof. Observe that for any set of plans $\mathcal{F} \subseteq \Pi$, the random variable $\widetilde{p}_r(\varepsilon', \mathcal{F})$ is an unbiased estimator of $p(\varepsilon', \mathcal{F})$. Since each Q_i is drawn i.i.d. from W, by Hoeffding's inequality [35],

$$\Pr\left[\left|\widetilde{p}_r(\varepsilon', \mathcal{F}) - p(\varepsilon', \mathcal{F})\right| > \delta/4\right] \;\le\; 2\exp\left(-2\mathfrak{n}_r\,(\delta/4)^2\right) = 2\exp\left(-\delta^2\mathfrak{n}_r/8\right).$$

Taking a union bound over all sets of plans of size up to $\mathbf{b}_r$ and using the fact that the number of subsets of plans of size at most $\mathbf{b}_r$ is bounded by $(\mathbf{b}_r+1)N^{\mathbf{b}_r}$, we obtain

$$\Pr\left[\exists\, \mathcal{F} \subseteq \Pi \text{ with } |\mathcal{F}| \le \mathbf{b}_r : \left|\widetilde{p}_r(\varepsilon', \mathcal{F}) - p(\varepsilon', \mathcal{F})\right| > \delta/4\right] \;\le\; 2(\mathbf{b}+1)N^{\mathbf{b}_r}\exp\left(-\delta^2\,\mathfrak{n}_r/8\right),$$

where $\mathbf{b}_r = c_3 k_r \ln(\delta^{-1})$ and $\mathfrak{n}_r = (c_1 k_r/\delta^2)\,\ln(N)\,\ln(\delta^{-1})$. By a suitable choice of the constants c_1 and c_3, the right-hand side is N^{-c} for any $c > 0$.

□

Lemma 3.2. *For any round r, any $\pi \in \Pi$ and any query $Q_i \in \widetilde{\mathcal{W}}_r$, $\left|\widetilde{\mathsf{Risk}}(\pi, Q_i) - \mathsf{Risk}(\pi, Q_i)\right| \le \varepsilon/10$ with probability at least $1 - N^{-O(1)}$.*

Proof. For a fixed plan $\pi \in \Pi$ and $Q_i \in \widetilde{\mathcal{W}}_r$, let $Y_{i,j} = \mathbf{1}\{\mathsf{Pnl}(\pi, Q_{i,j}) > \tau\}$. Then $\widetilde{\mathsf{Risk}}(\pi, Q_i) = \frac{1}{\mathfrak{m}_r}\sum_{j=1}^{\mathfrak{m}_r} Y_{i,j}$ and $\mathbb{E}[Y_{i,1}] = \mathsf{Risk}(\pi, Q_i)$. By Hoeffding's inequality [35],

$$\Pr\left[\left|\widetilde{\mathsf{Risk}}(\pi, Q_i) - \mathsf{Risk}(\pi, Q_i)\right| > \varepsilon/10\right] \le 2\exp\left(-\mathfrak{m}_r\,\varepsilon^2/50\right).$$

By plugging in $\mathfrak{m}_r = (c_2/\varepsilon^2)\ln(N)\ln(\mathfrak{n}_r)$ from the algorithm and taking a union bound over all $|\Pi| \cdot \mathfrak{n}_r = N \cdot \mathfrak{n}_r$ pairs $(\pi, Q_i) \in \Pi \times \widetilde{\mathcal{W}}_r$, the failure probability is at most

$$2N\mathfrak{n}_r \exp(-c_2 \ln(N)\ln(\mathfrak{n}_r)/50) = 2N\mathfrak{n}_r\, N^{-(c_2/50)\ln(\mathfrak{n}_r)}.$$

This is at most $N^{-O(1)}$ for a sufficiently large constant c_2. □

Since $k_r \leq |\Pi| = N$ always holds and k_r increases geometrically, the algorithm performs at most $\log N$ rounds. In any round, either Lemma 3.1 or Lemma 3.2 fails with probability at most $N^{-O(1)}$, and therefore (by union bound) all rounds simultaneously satisfy the conclusions of Lemmas 3.1 and 3.2 with probability at least $1 - N^{-O(1)}$.

We now show that if the algorithm terminates in a round, the returned set C satisfies the desired coreset guarantee.

Lemma 3.3. *Assuming the conditions of Lemmas 3.1 and 3.2 hold in every round, the algorithm terminates with $p(3\varepsilon, C) \leq 3\delta$.*

Proof. Fix an arbitrary terminating round r and let $C \subseteq \Pi$ denote the set returned in that round, with $|C| \leq \mathbf{b}_r$. For any $Q_i \in \widetilde{\mathcal{W}}_r$ and let $\pi_i := \arg\min_{\pi \in C} \widetilde{\mathsf{Risk}}(Q_i, C)$. In other words, π_i is the best plan in C in terms of the proxy risk.

By the termination condition in Step (4), when the algorithm terminates, there must be a subset of queries $\widetilde{\mathcal{W}}'_r \subseteq \widetilde{\mathcal{W}}_r$ of size at least $(1-2\delta)\mathfrak{n}_r$ such that for every $Q_i \in \widetilde{\mathcal{W}}'_r$,

$$\widetilde{\mathsf{Risk}}(\pi_i, Q_i) \;\leq\; \min_{\pi \in \Pi} \widetilde{\mathsf{Risk}}(\pi, Q_i) + 2\varepsilon.$$

Applying Lemma 3.2 to both sides yields,

$$\mathsf{Risk}(\pi_i, Q_i) - \varepsilon/10 \leq \; \mathsf{Risk}^\star(Q_i) + \tfrac{2\varepsilon}{10} + 2\varepsilon.$$

Hence, for all $Q_i \in \widetilde{\mathcal{W}}'_r$,

$$\min_{\pi \in C} \mathsf{Risk}(\pi, Q_i) \;\leq\; \mathsf{Risk}^\star(Q_i) + 3\varepsilon.$$

Therefore, by the definition of $\widetilde{p}_r$, we have $\widetilde{p}_r(3\varepsilon, C) \leq 1 - (1-2\delta) \leq 2\delta$. Invoking Lemma 3.1 with $\varepsilon' = 3\varepsilon$, this implies that $p(3\varepsilon, C) \leq 2\delta + \delta/4 < 3\delta$.

□

Finally, we show that once $k_r \geq \kappa^*$, the algorithm necessarily terminates.

Lemma 3.4. *Assuming the conditions of Lemmas 3.1 and 3.2 hold in every round, the algorithm terminates with $k_r \leq 2\kappa^*$.*

Proof. By the definition of a $(\varepsilon, \tau, \delta|\mathsf{W})$-coreset for risk, there exists $\mathsf{M} \subseteq \Pi$ of size κ^* such that $p(\varepsilon, \mathsf{M}) \leq \delta$. Invoking Lemma 3.1 with $\varepsilon' = \varepsilon$, $\widetilde{p}_r(\varepsilon, \mathsf{M}) \leq \delta + \delta/4$. Therefore, there is a subset $\widetilde{\mathcal{W}}'_r$ of size at least $(1 - \frac{5\delta}{4})\mathfrak{n}_r$ such that for each $Q_i \in \widetilde{\mathcal{W}}'_r$,

$$\min_{\pi \in \mathsf{M}} \mathsf{Risk}(\pi, Q_i) \leq \mathsf{Risk}^\star(Q_i) + \varepsilon.$$

Applying Lemma 3.2, for every $Q_i \in \widetilde{\mathcal{W}}'_r$,

$$\min_{\pi \in \mathsf{M}} \widetilde{\mathsf{Risk}}(\pi, Q_i) \;\leq\; \min_{\pi \in \mathsf{M}} \mathsf{Risk}(\pi, Q_i) + \varepsilon/10 \;\leq\; \mathsf{Risk}^\star(Q_i) + \varepsilon + \varepsilon/10 \;\leq\; \mathsf{Risk}^\star(Q_i) + 2\varepsilon.$$

Thus, there exists a subset of κ^* plans (with M being a specific example) that *covers* (as defined in Step (3)) at least $(1 - \frac{3\delta}{2})\mathfrak{n}_r \geq (1 - \frac{5\delta}{4})\mathfrak{n}_r$ queries in $\widetilde{\mathcal{W}}_r$.

Let $n' := (1 - \frac{3\delta}{2})\mathfrak{n}_r$. For $t > 0$, let $C_t \subseteq \widetilde{\mathcal{W}}_r$ be the set of profiles covered after t rounds of the greedy cover algorithm. Set $D_t := n' - |C_t|$. Since there exists a family of κ^* plans covering n' profiles, among those κ^* plans there is one whose marginal gain at round $t+1$ is at least D_t/κ^*. Greedy picks a plan with maximal marginal gain, hence

$$D_{t+1} \leq D_t - \frac{D_t}{\kappa^*} = \left(1 - \frac{1}{\kappa^*}\right) D_t \leq \left(1 - \frac{1}{\kappa^*}\right)^t n' \leq \exp(-t/\kappa^*) n',$$

so the number of covered points satisfies

$$|C_t| = n' - D_t \;\geq\; n'\big(1 - \exp(-t/\kappa^*)\big) \;\geq\; \left(1 - \frac{3\delta}{2}\right)\mathfrak{n}_r \big(1 - \exp(-t/\kappa^*)\big).$$

For $t \geq c_3 \kappa^* \ln \delta^{-1}$,

$$\left(1 - \frac{3\delta}{2}\right)\big(1 - \exp(-t/\kappa^*)\big) \;\geq\; 1 - 2\delta,$$

assuming $c_3 > 0$ is chosen appropriately and setting $\delta \leq 1/2$. Hence, $|C_t| \geq (1 - 2\delta)\mathfrak{n}_r$, implying that the algorithm terminates once $k_r \geq \kappa^*$.

□

Finally, we analyze the sample complexity and time complexity of the algorithm. Since k_r doubles each round, the last round dominates the sample complexity. Thus, the total number of samples drawn from W is $O((\kappa^*/\delta^2)\ \ln(N) \ln(\delta^{-1}))$.

In each round, the algorithm draws $\mathfrak{n}_r = (c_1 k_r/\delta^2) \cdot \ln(N) \cdot \ln(\delta^{-1})$ queries from W and $\mathfrak{m}_r = O((1/\varepsilon^2)\ln(N)\ln(\mathfrak{n}_r))$ cardinality vectors per query. Thus, the total number of sampled cardinality vectors is $O(\mathfrak{n}_r \mathfrak{m}_r)$. Evaluating the penalties for all $\pi \in \Pi$ for each sampled cardinality vector takes $O(|\Pi|) = O(N)$ time, giving a per-round cost of $O(\mathfrak{n}_r \mathfrak{m}_r N)$. Constructing all coverage sets and executing the greedy cover procedure each require $O(\mathfrak{n}_r\, N)$ time. Since the last round dominates, the total running time is $O(\mathfrak{n}_r \mathfrak{m}_r N)$ in the final round, i.e., $O((\kappa^*/(\varepsilon^2\delta^2))\, N\ \ln^3(N)\ \ln(\delta^{-1}))$.

Putting everything together, we get the following.

Theorem 3.5. *Let* W *be a workload distribution. Given a set of plans* Π *of size* $|\Pi| = N$ *and parameters* $\varepsilon \in (0,1]$, $\tau \geq 1$, *and* $\delta \in (0,1)$, *let* κ^* *denote the size of the smallest* $(\varepsilon, \tau, \delta|\mathrm{W})$*–coreset for risk. Fix any pair of constants* $c_1, c_2 > 1$. *There exists an algorithm that samples* $O\big((\kappa^*/\delta^2)\ \ln(N)\ \ln(\delta^{-1})\big)$ *queries from* W *and computes in time* $O\big((\kappa^*/(\varepsilon^2\delta^2))\, N\ \ln^3(N)\ \ln(\delta^{-1})\big)$, *a set* $C \subseteq \Pi$ *of size* $O\big(\kappa^* \ln \delta^{-1}\big)$ *such that, with high probability,* C *is a* $(c_1\varepsilon, \tau, c_2\delta|\mathrm{W})$*–coreset for risk.*

3.3 AlgRiskCoreset in the implicit-setting

In many query optimization settings, the plan set is not explicitly enumerated but is defined *implicitly* via a structured optimization problem. We focus on query optimization based on Mixed-Integer Linear Programming (MILP) as in [39, 41].

Let Γ be a query template and let $\mathbb{Q}$ denote the corresponding query space. Instead of the cardinality vector corresponding to a query in $\mathbb{Q}$, we work with the corresponding *selectivity vector*, as required by MILP-based methods. Recall that each dimension of the cardinality vector corresponds to the size of an intermediate result; the selectivity vector is obtained by normalizing the cardinalities of intermediate results with the size of the corresponding table. Accordingly, we define cost functions and penalties in terms of selectivity vectors rather than cardinality vectors. We note that AlgRiskCoreset is oblivious to this distinction and works unchanged.

Let $\mathbb{S}$ denote the selectivity space (overloading notation) corresponding to Γ. For any query $Q \in \mathbb{Q}$, the optimizer maps Q to an MILP instance. The instance is defined by *binary decision variables* encoding structural choices of the plan (e.g. if table t is in the outer/inner operand of the j-th join, if predicate p is evaluated at the j-th join), together with *feasibility constraints* ensuring that these choices define a valid plan (e.g. the tables in the join operands cannot overlap for the same join). These variables and constraints are identical across all queries as long as the query template is fixed. An instance also includes *cost-accounting variables and constraints*, which capture the sizes of intermediate results (e.g. the output of the j-th join). The coefficients in these constraints are defined by the vector $\mathbf{s}_Q \in \mathbb{S}$ corresponding to Q.[4] The *objective* is to minimize the total cost, expressed as a sum of intermediate result sizes.

The key observation is that the MILP structure remains unchanged across queries for a fixed query template; only the coefficients in the cost-accounting constraints vary with $\mathbf{s}_Q$. This allows us to emulate the steps of AlgRiskCoreset by combining MILP instances corresponding to different selectivity vectors, sharing the structural decision variables and duplicating only the cost-accounting constraints. We illustrate this for computing the plan that minimizes the *proxy risk* of a query.

Computing a plan minimizing proxy risk via MILP. Fix a query Q. Let $\mathcal{Z} = \{\mathbf{s}_1, \ldots, \mathbf{s}_m\} \subseteq \mathbb{S}$ be a collection of m selectivity vectors, where each $\mathbf{s}$ is sampled from the uncertainty profile U_Q. The goal is to compute $\arg\min_{\pi\in\Pi}(1/m)\sum_{j=1}^{m}\mathbf{1}\{\mathsf{Pnl}(\pi,\mathbf{s}_j) > \tau\}$.

We first compute $\mathsf{Cost}^\star(\mathbf{s}_j)$ for each $\mathbf{s}_j \in \mathcal{Z}$ using the MILP. We then construct a single MILP instance that combines the m instances corresponding to the samples. The combined instance contains only one copy of the structural decision variables and feasibility constraints, i.e., the binary variables encoding structural choices of the plan (such as join ordering) together with constraints ensuring these choices form a valid plan. Let π denote the plan represented by these shared decision variables.

For each $\mathbf{s}_j$, however, we duplicate the cost-accounting variables and constraints using distinct variables, so that the MILP separately accounts for the cost of π with respect to each selectivity vector in $\mathcal{Z}$. We ensure that this cost, $\mathsf{Cost}(\pi, \mathbf{s}_j)$, is captured by a variable c_j. We also introduce a binary variable $y_j \in \{0, 1\}$ indicating whether $\mathsf{Pnl}(\pi, \mathbf{s}_j) > \tau$, enforced via the constraint

$$c_j \;\le\; \tau \cdot \mathsf{Cost}^\star(\mathbf{s}_j) + \beta \cdot y_j,$$

where β is a sufficiently large constant. Thus, if $c_j > \tau \cdot \mathsf{Cost}^\star(\mathbf{s}_j)$, the solver must set $y_j = 1$. Minimizing $\sum_{j=1}^{m} y_j$ yields a plan π that minimizes the proxy risk.

Adapting AlgRiskCoreset. We now briefly discuss how to use the MILP to emulate different steps of AlgRiskCoreset. Recall that in Step (2), AlgRiskCoreset computes the proxy risk of every plan in Π with respect to each query Q in a collection $\widetilde{\mathcal{W}} \subseteq \mathbb{Q}$. These values are then used in Step (3) to define the covering problem. However, to define the covering problem, it is not necessary to know the proxy risk of every plan with respect to each query. It only suffices to know $r_Q = \min_{\pi\in\Pi}\widetilde{\mathsf{Risk}}(\pi, Q)$ for each $Q \in \widetilde{\mathcal{W}}$. We can compute r_Q for each $Q \in \widetilde{\mathcal{W}}$ by invoking the previous procedure for computing a plan minimizing the proxy risk via MILP. To solve the covering problem in Step (3), we can emulate each step of the greedy-cover procedure by encoding a single MILP instance (similarly to the above procedure) that selects the plan covering the maximum number of queries. To encode this instance we use the previously computed r_Q values; we omit details for brevity.

4 Inferring a low-risk plan using a coreset

In this section, we study the problem of inference: given a $(\varepsilon, \tau, \delta \mid \mathrm{W})$-coreset $C \subseteq \Pi$ for risk and a new query Q, how should an appropriate plan $\pi \in C$ be selected such that $\mathsf{Risk}(\pi, Q)$ is not much larger than $\mathsf{Risk}^\star(Q)$. A difficulty in computing $\mathsf{Risk}(\pi, Q)$ for a plan $\pi \in C$ is that for a cardinality vector $\mathbf{s}$, $\mathsf{Pnl}(\pi, \mathbf{s})$ depends on $\mathsf{Cost}^\star(\mathbf{s})$, the optimal cost with respect to all plans in Π. Instead, we

[4] In practice, the optimizer uses the estimated selectivity vector $\hat{\mathbf{s}}_Q$, although the "true" MILP instance is defined by the true selectivity vector $\mathbf{s}^\star_Q$.

would like to compute penalty using the optimal cost with respect to the plans in C. We describe such a "local" inference procedure in Section 4.1 and show (in Section 4.2) that this only gives a weaker result (Theorem 4.1). We then show that in order to do strengthen the guarantees, we need to augment C with another "cost based" coreset C'. We define this coreset in Section 4.3, describe how to adapt AlgRiskCoreset to compute C', and analyze the performance of local inference w.r.t. $C \cup C'$ (Theorems 4.5 and 4.8).

Before proceeding further, we present some essential definitions.

Local performance measures. For any set of plans $\mathcal{F} \subseteq \Pi$, a plan $\pi \in \Pi$, and cardinality vector $\mathbf{s} \in \mathbb{S}$, the *local penalty* of π with respect to $\mathcal{F}$ measures how π compares to the best plan in $\mathcal{F}$, i.e.,

$$\mathsf{L\text{-}Pnl}_{\mathcal{F}}(\pi, \mathbf{s}) := \frac{\mathsf{Cost}(\pi, \mathbf{s})}{\min_{\pi' \in \mathcal{F}} \mathsf{Cost}(\pi', \mathbf{s})}.$$

For a query $Q \in \mathbb{Q}$, the corresponding *local τ-risk* of a plan π w.r.t. $\mathcal{F}$ is denoted by $\mathsf{L\text{-}Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau)$. It is is defined by replacing $\mathsf{Pnl}(\pi, \cdot)$ with $\mathsf{L\text{-}Pnl}_{\mathcal{F}}(\pi, \cdot)$ in the definition of $\mathsf{Risk}(\pi, Q\,;\,\tau)$.

We make two observations that will be useful throughout the rest of this section. First, for any $\mathbf{s} \in \mathbb{S}$ and $\pi \in \Pi$, we have $\mathsf{L\text{-}Pnl}_{\mathcal{F}}(\pi, \mathbf{s}) \le \mathsf{Pnl}(\pi, \mathbf{s})$, since the denominator in $\mathsf{L\text{-}Pnl}_{\mathcal{F}}(\pi, \mathbf{s})$ is minimized over a subset of plans. Consequently, for any query $Q \in \mathbb{Q}$, plan $\pi \in \Pi$, and threshold τ, we have $\mathsf{L\text{-}Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) \le \mathsf{Risk}(\pi, Q\,;\,\tau)$.

4.1 LocalInference

The procedure is initialized with parameters $\tau \ge 1$ and $\varepsilon, \delta \in (0, 1)$, and a subset of plans $\mathcal{F} \subseteq \Pi$. Given a new query Q, the goal is to select the plan with the smallest local risk for Q among the plans in $\mathcal{F}$. But the local risk cannot be computed exactly, so the procedure draws samples from U_Q and uses them to estimate the local risk of each plan in $\mathcal{F}$ and selects the plan with the smallest estimated local risk. More precisely, we set a parameter $\mathfrak{m} = \Theta((\ln|\mathcal{F}| + \ln(1/\delta))/\varepsilon^2)$.

Given a query Q, draw a set $\mathcal{Z} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{\mathfrak{m}}\}$ of $\mathfrak{m}$ cardinality vectors such that each $\mathbf{s}_i \sim \mathrm{U}_Q$. For each $\pi \in \mathcal{F}$, compute the *proxy local risk*

$$\widetilde{\mathsf{L\text{-}Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau) := \frac{1}{\mathfrak{m}} \sum_{\mathbf{s}_j \in \mathcal{Z}} \mathbf{1}\{\mathsf{L\text{-}Pnl}_{\mathcal{F}}(\pi, \mathbf{s}_j) > \tau\}.$$

Finally, return

$$\hat{\pi} := \arg\min_{\pi \in \mathcal{F}} \widetilde{\mathsf{L\text{-}Risk}}_{\mathcal{F}}(\pi, \mathrm{U}\,;\,\tau).$$

We refer to this procedure as LocalInference$(Q, \mathcal{F}, \varepsilon, \delta, \tau)$. If the parameters ε, δ, and τ are fixed or follow from the context, we drop them and simply use LocalInference$(Q, \mathcal{F})$.

4.2 Analysis of LocalInference on a workload-dependent coreset

The following theorem states the main result of this subsection.

Theorem 4.1. *Let $\varepsilon, \delta \in (0, 1)$ and $\tau \ge 1$ be parameters. Let* W *be a workload distribution over $\mathbb{Q}$, and let $\mathcal{F} \subseteq \Pi$ be a $(\varepsilon, \tau, \delta \mid \mathrm{W})$-coreset for risk. Let $Q \sim \mathrm{W}$. Suppose LocalInference$(Q, \mathcal{F})$ returns a plan $\hat{\pi}$. Then, with probability at least $1 - 2\delta$,*

$$\mathsf{Risk}(\hat{\pi}, Q\,;\,\tau^2) \;\le\; 2\,\mathsf{Risk}^{\star}(Q\,;\,\tau) + 4\varepsilon.$$

Moreover, the algorithm uses $\Theta(|\mathcal{F}|(\ln|\mathcal{F}| + \ln(1/\delta))/\varepsilon^2)$ time.

Here is a high-level overview of the proof of Theorem 4.1. Let $\mathcal{F} \subseteq \Pi$ be an arbitrary set of plans. For a query Q, let $r = \min_{\pi \in \mathcal{F}} \mathsf{Risk}(\pi, Q\,;\,\tau)$ denote the optimal (global) τ-risk among plans in $\mathcal{F}$. We first relate global and local risk for arbitrary plan sets. In particular, Lemma 4.2 shows that if we could select the plan in $\mathcal{F}$ with the minimum local τ-risk, its global τ^2-risk would be at most $2r$. However, LocalInference estimates the local risk of a plan from samples drawn from the uncertainty profile U_Q. Lemma 4.3 shows that the empirical quantity $\text{L-}\widetilde{\mathsf{Risk}}_{\mathcal{F}}$ uniformly approximates the true local risk $\text{L-}\mathsf{Risk}_{\mathcal{F}}$. Next, Lemma 4.4 combines the guarantees of Lemmas 4.2 and 4.3 to show that LocalInference returns a plan whose global τ^2-risk is at most $2r + 2\varepsilon$. Finally, we consider the case when $\mathcal{F}$ is an $(\varepsilon, \tau, \delta \mid \mathrm{W})$-coreset for risk. In this case, with probability at least $1 - \delta$, we have $r \leq \mathsf{Risk}^\star(Q\,;\,\tau) + \varepsilon$. Combining this bound with the guarantee in Lemma 4.4 yields Theorem 4.1. We now give the proof in detail through a sequence of lemmas.

Lemma 4.2. *Let $\mathcal{F} \subseteq \Pi$ be a set of plans. For any plan $\pi \in \mathcal{F}$ and query $Q \in \mathbb{Q}$,*

$$\mathsf{Risk}(\pi, Q\,;\,\tau^2) \;\leq\; \text{L-}\mathsf{Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) + \min_{\pi' \in \mathcal{F}} \mathsf{Risk}(\pi', Q\,;\,\tau).$$

It follows that

$$\min_{\pi \in \mathcal{F}} \mathsf{Risk}(\pi, Q\,;\,\tau^2) \;\leq\; 2 \min_{\pi \in \mathcal{F}} \mathsf{Risk}(\pi, Q\,;\,\tau).$$

Proof. Fix a plan $\pi \in \mathcal{F}$ and consider any plan $\pi' \in \mathcal{F}$. We have

$$\begin{aligned}
\mathsf{Pnl}(\pi, \mathbf{s}) &= \frac{\mathsf{Cost}(\pi, \mathbf{s})}{\min_{\pi'' \in \Pi} \mathsf{Cost}(\pi'', \mathbf{s})} \cdot \frac{\mathsf{Cost}(\pi', \mathbf{s})}{\mathsf{Cost}(\pi', \mathbf{s})} \\
&\leq \frac{\mathsf{Cost}(\pi, \mathbf{s})}{\mathsf{Cost}(\pi', \mathbf{s})}\, \mathsf{Pnl}(\pi', \mathbf{s}) \\
&\leq \text{L-}\mathsf{Pnl}_{\mathcal{F}}(\pi, \mathbf{s})\, \mathsf{Pnl}(\pi', \mathbf{s}).
\end{aligned} \tag{2}$$

The first inequality follows from the definition of the penalty, and the second inequality follows from the fact that $\min_{\pi'' \in \mathcal{F}} \mathsf{Cost}(\pi'', \mathbf{s}) \leq \mathsf{Cost}(\pi', \mathbf{s})$, which implies $\mathsf{Cost}(\pi, \mathbf{s})/\mathsf{Cost}(\pi', \mathbf{s}) \leq \text{L-}\mathsf{Pnl}_{\mathcal{F}}(\pi, \mathbf{s})$. Consider the sets

$$\begin{aligned}
E_1 &= \{\mathbf{s} \in \mathbb{S} \mid \mathsf{Pnl}(\pi, \mathbf{s}) > \tau^2\}, \\
E_2 &= \{\mathbf{s} \in \mathbb{S} \mid \text{L-}\mathsf{Pnl}_{\mathcal{F}}(\pi, \mathbf{s}) > \tau\}, \\
E_3 &= \{\mathbf{s} \in \mathbb{S} \mid \mathsf{Pnl}(\pi', \mathbf{s}) > \tau\}.
\end{aligned}$$

It follows from (2) that if $\mathbf{s} \in E_1$, then either $\mathbf{s} \in E_2$ or $\mathbf{s} \in E_3$.

Suppose we draw a selectivity vector $\mathbf{s} \sim \mathrm{U}_Q$. The preceding argument implies that

$$\Pr_{\mathbf{s} \sim \mathrm{U}_Q}[\mathbf{s} \in E_1] \leq \Pr_{\mathbf{s} \sim \mathrm{U}_Q}[\mathbf{s} \in E_2 \cup E_3] \leq \Pr_{\mathbf{s} \sim \mathrm{U}_Q}[\mathbf{s} \in E_2] + \Pr_{\mathbf{s} \sim \mathrm{U}_Q}[\mathbf{s} \in E_3].$$

By definition of τ-risk,

$$\begin{aligned}
\Pr_{\mathbf{s} \sim \mathrm{U}_Q}[E_1] &= \mathsf{Risk}(\pi, Q\,;\,\tau^2), \\
\Pr_{\mathbf{s} \sim \mathrm{U}_Q}[E_2] &= \text{L-}\mathsf{Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau), \\
\Pr_{\mathbf{s} \sim \mathrm{U}_Q}[E_3] &= \mathsf{Risk}(\pi', Q\,;\,\tau).
\end{aligned}$$

Substituting these expressions yields

$$\mathsf{Risk}(\pi, Q\,;\,\tau^2) \leq \text{L-}\mathsf{Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) + \mathsf{Risk}(\pi', Q\,;\,\tau).$$

Minimizing over $\pi' \in \mathcal{F}$ yields the claim. □

Lemma 4.3. *Let $\varepsilon, \delta \in (0, 1)$, and $\tau \geq 1$ be parameters, let $\mathcal{F} \subseteq \Pi$ be a set of plans and let $Q \in \mathbb{Q}$. If LocalInference$(Q, \mathcal{F})$ returns a plan $\hat{\pi}$. Then, with probability at least $1 - \delta$, for every plan $\pi \in \mathcal{F}$,*

$$\left|\text{L-}\widetilde{\mathsf{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau') - \text{L-}\mathsf{Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau')\right| \leq \varepsilon.$$

Proof. For a fixed $\pi \in \mathcal{F}$, the random variables $Y_j := \mathbf{1}\{\text{L-Pnl}_{\mathcal{F}}(\pi, \mathbf{s}_j) > \tau'\}$ are i.i.d. in $[0,1]$ with $\mathcal{E}[Y_j] = \text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau')$ and $\text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau') = \frac{1}{\mathfrak{m}} \sum_{j=1}^{\mathfrak{m}} Y_j$. Hoeffding's inequality [35] gives

$$\Pr\left[\,\left|\text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau') - \text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau')\right| > \varepsilon\right] \;\leq\; 2\exp(-2\mathfrak{m}\varepsilon^2).$$

A union bound over $\pi \in \mathcal{F}$, combined with the choice of $\mathfrak{m} = \Theta((\ln|\mathcal{F}| + \ln(1/\delta))/\varepsilon^2)$, completes the proof. □

Lemma 4.4. *Let $\varepsilon, \delta \in (0,1)$ and $\tau \geq 1$ be parameters, let $\mathcal{F} \subseteq \Pi$ be a set of plans and let $Q \in \mathbb{Q}$. Suppose* LocalInference$(Q, \mathcal{F})$ *returns a plan $\hat{\pi}$. Then, with probability at least $1 - \delta$,*

$$\mathit{Risk}(\hat{\pi}, Q\,;\,\tau^2) \leq 2\min_{\pi\in\mathcal{F}} \mathit{Risk}(\pi, Q\,;\,\tau) + 2\varepsilon.$$

Proof. By Lemma 4.3, with probability at least $1 - \delta$, for every plan $\pi \in \mathcal{F}$,

$$|\text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau) - \text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau)| \leq \varepsilon.$$

Fix this event. By design of LocalInference, $\hat{\pi}$ minimizes $\text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau)$ over $\mathcal{F}$, and hence for every $\pi \in \mathcal{F}$ we have $\text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\hat{\pi}, Q\,;\,\tau) \leq \text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau)$. Therefore, for every $\pi \in \mathcal{F}$,

$$\text{L-Risk}_{\mathcal{F}}(\hat{\pi}, Q\,;\,\tau) \leq \text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\hat{\pi}, Q\,;\,\tau) + \varepsilon \leq \text{L-}\widetilde{\text{Risk}}_{\mathcal{F}}(\pi, Q\,;\,\tau) + \varepsilon \leq \text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) + 2\varepsilon.$$

Taking the minimum over $\pi \in \mathcal{F}$ gives

$$\text{L-Risk}_{\mathcal{F}}(\hat{\pi}, Q\,;\,\tau) \leq \min_{\pi\in\mathcal{F}} \text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) + 2\varepsilon. \tag{3}$$

Applying Lemma 4.2 with $\pi = \hat{\pi}$ gives

$$\text{Risk}(\hat{\pi}, Q\,;\,\tau^2) \leq \text{L-Risk}_{\mathcal{F}}(\hat{\pi}, Q\,;\,\tau) + \min_{\pi\in\mathcal{F}} \text{Risk}(\pi, Q\,;\,\tau). \tag{4}$$

Substituting the bound from (3) for $\text{L-Risk}_{\mathcal{F}}(\hat{\pi}, Q\,;\,\tau)$ in (4), we obtain

$$\text{Risk}(\hat{\pi}, Q\,;\,\tau^2) \leq \min_{\pi\in\mathcal{F}} \text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) + \min_{\pi\in\mathcal{F}} \text{Risk}(\pi, Q\,;\,\tau) + 2\varepsilon.$$

Finally, since $\text{L-Risk}_{\mathcal{F}}(\pi, Q\,;\,\tau) \leq \text{Risk}(\pi, Q\,;\,\tau)$ for every $\pi \in \mathcal{F}$, the lemma follows. □

Putting everthing together. Let W be a workload distribution over $\mathbb{Q}$, let $\mathcal{F} \subseteq \Pi$ be a $(\varepsilon, \tau, \delta \mid \mathrm{W})$-coreset for risk, and let $Q \sim \mathrm{W}$. Suppose LocalInference$(Q, \mathcal{F})$ returns a plan $\hat{\pi}$. By definition of risk coresets, with probability $1 - \delta$,

$$\min_{\pi\in\mathcal{F}} \text{Risk}(\pi, Q\,;\,\tau) \leq \text{Risk}^\star(Q\,;\,\tau) + \varepsilon.$$

Let E_1 denote this event. Applying Lemma 4.4 with the same set $\mathcal{F}$, with probability at least $1 - \delta$, we have

$$\text{Risk}(\hat{\pi}, Q\,;\,\tau^2) \leq 2\min_{\pi\in\mathcal{F}} \text{Risk}(\pi, Q\,;\,\tau) + 2\varepsilon.$$

Let E_2 denote this event. Therefore $\Pr[E_1 \cap E_2] \geq 1 - 2\delta$. On $E_1 \cap E_2$ the two bounds hold simultaneously, which implies

$$\text{Risk}(\hat{\pi}, Q\,;\,\tau^2) \leq 2(\text{Risk}^\star(Q\,;\,\tau) + \varepsilon) + 2\varepsilon = 2\,\text{Risk}^\star(Q\,;\,\tau) + 4\varepsilon.$$

Finally, an easy calculation shows that the time complexity is bounded by $\Theta(|\mathcal{F}|(\ln|\mathcal{F}| + \ln(1/\delta))/\varepsilon^2)$. This completes the proof of Theorem 4.1.

4.3 Analysis of LocalInference on an augmented workload-dependent coreset

In this section, we first introduce the notion of the *coreset for neighborhood-cost*. Next, we show how, when a coreset for neighborhood-cost is augmented to a risk-based coreset, the resulting set C provides stronger inference guarantees for LocalInference than those established in Section 4.2. Finally, we briefly discuss how to construct a neighborhood-cost coreset by using a simpler variant of AlgRiskCoreset.

$(\alpha, \delta \mid \mathrm{W})$-coreset for neighborhood-cost. Let W be a workload distribution. For $\alpha \geq 1$ and $\delta \in (0,1)$, a set of plans $C \subseteq \Pi$ is an $(\alpha, \delta \mid \mathrm{W})$-*coreset for neighborhood-cost* if

$$\Pr_{Q \sim \mathrm{W},\, \mathbf{s} \sim \mathrm{U}_Q}\left[\min_{\pi \in C} \mathsf{Cost}(\pi, \mathbf{s}) > \alpha \cdot \mathsf{Cost}^\star(\mathbf{s})\right] \leq \delta.$$

In other words, for a random query $Q \sim \mathrm{W}$ and a cardinality vector $\mathbf{s} \sim \mathrm{U}_Q$, the set C contains a plan whose cost is within a factor α of the optimal cost with probability at least $1 - \delta$.

Sample Complexity of LocalInference. Given an error margin $\varepsilon \in (0,1)$, a failure probability $\delta \in (0,1)$, and an integer $n \in \mathbb{N}$, let $\mathfrak{m}(\varepsilon, \delta, n)$ denote the sample complexity of LocalInference, i.e., this is the exact number of independent cardinality vectors drawn from U_Q when executing LocalInference$(Q, \mathcal{F}, \varepsilon, \delta, \tau)$ to infer the best plan from a set of n plans for a query Q. As the notation implies, this sample size depends exclusively on ε, δ, and $n = |\mathcal{F}|$.

The following theorem states the main result of this subsection.

Theorem 4.5. *Let $\varepsilon, \delta \in (0,1)$ and $\alpha, \tau \geq 1$ be given parameters. Let W be a workload distribution and let $Q \sim \mathrm{W}$. Let $C = C_1 \cup C_2 \subseteq \Pi$ be a set of plans such that C_1 is an $(\varepsilon, \tau, \delta \mid \mathrm{W})$−coreset for risk, and C_2 is an $\left(\alpha, \frac{\delta^2}{\mathfrak{m}(\varepsilon, \delta, |C|)} \mid \mathrm{W}\right)$-coreset for neighborhood-cost. If LocalInference(Q, C) returns $\hat{\pi}$, then with probability at least $1 - 5\delta$,*

$$\mathit{Risk}(\hat{\pi}, Q\,;\, \tau\alpha) \leq \mathit{Risk}^\star(Q\,;\, \tau) + 5\varepsilon.$$

Before giving the proof of Theorem 4.5, we first introduce the notion of *neighborhood error-rate* for a set of plans $\mathcal{F}$. For a fixed $\alpha \geq 1$ and a plan set $\mathcal{F} \subseteq \Pi$, the neighborhood error-rate on Q measures the probability that, for a random cardinality vector $\mathbf{s} \sim \mathrm{U}_Q$, the set $\mathcal{F}$ fails to provide a plan whose cost is within a factor α of the optimal cost. Formally, we define the *neighborhood error-rate function* $\mathcal{N}_{\alpha,\mathcal{F}} : \mathbb{Q} \to [0,1]$ as:

$$\mathcal{N}_{\alpha,\mathcal{F}}(Q) = \Pr_{\mathbf{s} \sim \mathrm{U}_Q}\left[\min_{\pi \in \mathcal{F}} \mathsf{Cost}(\pi, \mathbf{s}) > \alpha \cdot \mathsf{Cost}^\star(\mathbf{s})\right].$$

Let $\mathfrak{m} = \mathfrak{m}(\varepsilon, \delta, |C|)$. In order to prove Theorem 4.5, we first show in Lemma 4.6 that for a random query $Q \sim \mathrm{W}$, the probability that the neighborhood error-rate $\mathcal{N}_{\alpha,C_2}(Q) > \delta/\mathfrak{m}$ is at most δ. Specifically, this lemma allows us to translate a global coreset guarantee into a high-probability bound on the local error-rate for the uncertainty profile of Q.

Lemma 4.6. *Let W be a workload distribution, let $\delta_1, \delta_2 \in (0,1)$ be two parameters, and let $C \subseteq \Pi$ be an $(\alpha, \delta_1\delta_2 \mid \mathrm{W})$-coreset for neighborhood-cost. Then*

$$\Pr_{Q \sim \mathrm{W}}\left[\mathcal{N}_{\alpha,C}(Q) > \delta_1\right] \leq \delta_2.$$

Proof. Consider the set $A = \left\{\mathbf{s} \in \mathbb{S} \mid \min_{\pi \in C} \mathsf{Cost}(\pi, \mathbf{s}) > \alpha \cdot \mathsf{Cost}^\star(\mathbf{s})\right\}$. By definition of C:

$$\Pr_{Q \sim \mathrm{W},\, \mathbf{s} \sim \mathrm{U}_Q}[\mathbf{s} \in A] \leq \delta_1\delta_2.$$

Using an indicator function, we can rewrite this probability as an expectation over the joint distribution:

$$\mathsf{E}_{Q\sim \mathrm{W},\, \mathbf{s}\sim \mathrm{U}_Q}[\mathbf{1}\{\mathbf{s} \in A\}] \le \delta_1\delta_2.$$

By the Law of Total Expectation, we can condition this on Q:

$$\mathsf{E}_{Q\sim \mathrm{W}}\left[\mathsf{E}_{\mathbf{s}\sim \mathrm{U}_Q}[\mathbf{1}\{\mathbf{s} \in A\} \mid Q]\right] \le \delta_1\delta_2.$$

The inner conditional expectation is precisely $\mathcal{N}_{\alpha,C}(Q)$. Therefore:

$$\mathsf{E}_{Q\sim \mathrm{W}}[\mathcal{N}_{\alpha,C}(Q)] \le \delta_1\delta_2.$$

Since $\mathcal{N}_{\alpha,C}(Q)$ is a probability, $\mathcal{N}_{\alpha,C}(Q) \ge 0$. Applying Markov's inequality:

$$\Pr_{Q\sim \mathrm{W}}\left[\mathcal{N}_{\alpha,C}(Q) > \delta_1\right] \le \frac{\mathsf{E}_{Q\sim \mathrm{W}}[\mathcal{N}_{\alpha,C}(Q)]}{\delta_1} \le \frac{\delta_1\delta_2}{\delta_1} = \delta_2.$$

□

Proof of Theorem 4.5. Let $C = C_1 \cup C_2 \subseteq \Pi$ be a set of plans. Let $\varepsilon, \delta \in (0,1)$ and let $\alpha, \tau \ge 1$. Let W be a workload distribution. Suppose C_1 is an $(\varepsilon, \tau, \delta \mid \mathrm{W})$–coreset for risk and C_2 is an $(\alpha, \delta^2/\mathfrak{m} \mid \mathrm{W})$-coreset for neighborhood-cost, where $\mathfrak{m} = \mathfrak{m}(\varepsilon, \delta, |C|)$ is the sample complexity of LocalInference. Suppose we draw a query $Q \sim \mathrm{W}$ and run LocalInference(Q, C).

By a simple union bound, with probability $1 - 2\delta$, both of the following hold:

$$\min_{\pi\in C_1} \mathsf{Risk}(\pi, Q\,;\, \tau) \;\le\; \mathsf{Risk}^\star(Q\,;\, \tau) + \varepsilon \tag{5}$$

$$\mathcal{N}_{\alpha,C_2}(Q) \le \frac{\delta}{\mathfrak{m}}. \tag{6}$$

Here (5) follows from the fact that C_1 is an $(\varepsilon, \tau, \delta \mid \mathrm{W})$–coreset for risk. Equation (6) follows from applying Lemma 4.6 to the $(\alpha, \delta \cdot (\delta/\mathfrak{m}) \mid \mathrm{W})$-coreset C_2 (by setting the lemma's parameters to $\delta_1 = \delta$ and $\delta_2 = \delta/\mathfrak{m}$).

Let $\mathcal{Z}$ be the set of cardinality vectors sampled by LocalInference from U_Q. Then assuming (6) holds, with probability at least $1 - \left(\mathfrak{m} \cdot \frac{\delta}{\mathfrak{m}}\right) = 1 - \delta$, the following event holds:

$$\forall\, \mathbf{s}_j \in \mathcal{Z},\ \exists\, \pi_j \in C_2 \text{ such that } \mathsf{Cost}(\pi_j, \mathbf{s}_j) \le \alpha \cdot \mathsf{Cost}^\star(\mathbf{s}_j). \tag{7}$$

Following the definition in (1), let $\widetilde{\mathsf{Risk}}(\pi, Q\,;\, \tau')$ denote the proxy τ'-risk of a plan π for a query Q, where threshold $\tau' \ge 1$, i.e.,

$$\widetilde{\mathsf{Risk}}(\pi, Q\,;\, \tau') := \frac{1}{\mathfrak{m}} \sum_{j=1}^{\mathfrak{m}} \mathbf{1}\{\mathsf{Pnl}(\pi, \mathbf{s}_j) > \tau'\}$$

Since the sample set $\mathcal{Z}$ is drawn independently, standard application of Hoeffding's inequality (similar to Lemma 4.3) implies that, with probability at least $1 - 2\delta$, the following empirical approximations hold simultaneously for every plan $\pi \in C$:

$$\left|\widetilde{\mathsf{Risk}}(\pi, Q\,;\, \tau\alpha) - \mathsf{Risk}(\pi, Q\,;\, \tau\alpha)\right| \le \varepsilon. \tag{8}$$

$$\left|\text{L-}\widetilde{\mathsf{Risk}}_C(\pi, Q\,;\, \tau) - \text{L-}\mathsf{Risk}_C(\pi, Q\,;\, \tau)\right| \le \varepsilon. \tag{9}$$

Thus, for a random query $Q \sim \mathrm{W}$ and $\mathcal{Z} \sim \mathrm{U}_Q^{\mathfrak{m}}$ drawn by LocalInference, all of the above events—(5), (6), (7), (8), and (9)—hold simultaneously with probability at least $1 - 5\delta$.

Since $\hat{\pi}$ minimizes the local proxy τ-risk over C, (9) implies:

$$\text{L-}\mathsf{Risk}_C(\hat{\pi}, Q\,;\, \tau) \;\le\; \min_{\pi\in C} \text{L-}\mathsf{Risk}_C(\pi, Q\,;\, \tau) + 2\varepsilon.$$

Using the fact that the minimum local risk over C is upper-bounded by the minimum global risk over C_1 (since $C_1 \subseteq C$), along with (5), we obtain:

$$\text{L-Risk}_C(\hat{\pi}, Q\,;\,\tau) \;\leq\; \min_{\pi \in C_1} \mathsf{Risk}(\pi, Q\,;\,\tau) + 2\varepsilon \;\leq\; \mathsf{Risk}^{\star}(Q\,;\,\tau) + 3\varepsilon. \tag{10}$$

For each $\mathbf{s}_j \in \mathcal{Z}$, condition (7) ensures that the optimal plan in C is at most an α factor worse than the true optimal. Thus, whenever $\mathsf{Pnl}(\hat{\pi}, \mathbf{s}_j) > \tau\alpha$, it must follow that $\text{L-Pnl}_C(\hat{\pi}, \mathbf{s}_j) > \tau$. Therefore:

$$\widetilde{\mathsf{Risk}}(\pi, Q\,;\,\tau\alpha) \;\leq\; \text{L-}\widetilde{\mathsf{Risk}}_C(\hat{\pi}, Q\,;\,\tau). \tag{11}$$

Applying (8) to the left-hand side of (11), we obtain:

$$\mathsf{Risk}(\hat{\pi}, Q\,;\,\tau\alpha) \;\leq\; \text{L-}\widetilde{\mathsf{Risk}}_C(\hat{\pi}, Q\,;\,\tau) \;+\; \varepsilon. \tag{12}$$

Applying (9) to the right-hand side of (12), we get:

$$\mathsf{Risk}(\hat{\pi}, Q\,;\,\tau\alpha) \;\leq\; \text{L-Risk}_C(\hat{\pi}, Q\,;\,\tau) \;+\; 2\varepsilon. \tag{13}$$

Finally, substituting (10) into (13) yields the ultimate bound:

$$\mathsf{Risk}(\hat{\pi}, Q\,;\,\tau\alpha) \;\leq\; \mathsf{Risk}^{\star}(Q\,;\,\tau) + 5\varepsilon. \tag{14}$$

This completes the proof of Theorem 4.5.

Constructing a neighborhood-cost coreset. Recall that $N = |\Pi|$ denotes the size of the set of all plans. Let $\hat{\kappa}$ denote the size of the smallest $(\alpha, \delta \mid \mathbb{W})$-coreset for neighborhood-cost. We use a simpler variant of the algorithm in AlgRiskCoreset described in Section 3.1 to compute a neighborhood-cost based coreset. For the sake of completeness, we describe the entire algorithm.

Specifically, the algorithm proceeds in rounds, until it finds a small set of plans covering a $(1-2\delta)$ fraction of the vectors. In each round r, we maintain a guess $k_r := 2^{r-1}$ for the value of $\hat{\kappa}$ and the round works as follows:

(1) **[Workload Sampling]** Set $\mathfrak{n}_r = (c_1 k_r/\delta^2)\ln(N)\ln(\delta^{-1})$ for a constant $c_1 > 0$. We draw a random set $\widetilde{\mathbb{S}}_r = \{\mathbf{s}_1, \ldots, \mathbf{s}_{\mathfrak{n}_r}\}$ of cardinality vectors, where each $\mathbf{s}_i$ is sampled independently according to the joint distribution $Q \sim \mathbb{W}$ and $\mathbf{s} \sim \mathrm{U}_Q$.

(2) **[Greedy Cover]** For each plan $\pi \in \Pi$, we define its coverage set:

$$\mathsf{Cov}(\pi) := \{\mathbf{s}_i \in \widetilde{\mathbb{S}}_r : \mathsf{Cost}(\pi, \mathbf{s}_i) \leq \alpha \cdot \mathsf{Cost}^{\star}(\mathbf{s}_i)\}.$$

Initialize $C = \emptyset$ and set a budget $\mathbf{b}_r = c_2 k_r \ln \delta^{-1}$ for a constant $c_2 > 0$. Iteratively select $\pi^* = \arg\max_{\pi \in \Pi} |\mathsf{Cov}(\pi)|$, add it to C, and remove covered vectors from all candidate sets. Continue until either $\widetilde{\mathbb{S}}$ is fully covered or $|C| = \mathbf{b}_r$.

(3) **[Validation]** If C covers at least a $(1-2\delta)$ fraction of the samples in $\widetilde{\mathbb{S}}_r$, return C; otherwise, set $r \leftarrow r+1$ and return to Step 1.

The following lemma follows from a simplified version of the analysis in Section 3.2.

Lemma 4.7. *Let $\mathbb{W}$ be a workload distribution. Given a set of plans Π of size $|\Pi| = N$ and parameters $\alpha \geq 1$, and $\delta \in (0,1)$, let $\hat{\kappa}$ denote the size of the smallest $(\alpha, \delta|\mathbb{W})$–neighborhood-cost coreset. Fix any constant $c > 1$. There exists an algorithm that samples $O\big((\hat{\kappa}/\delta^2)\,\ln(N)\,\ln(\delta^{-1})\big)$ queries from $\mathbb{W}$ and outputs a set $C \subseteq \Pi$ of size $O\big(\hat{\kappa}\ln\delta^{-1}\big)$ such that, with high probability, C is a $(\alpha, c\delta|\mathbb{W})$–coreset for neighborhood-cost. The construction runs in time $O\big((\hat{\kappa}/\delta^2)\,N\,\ln^2(N)\,\ln(\delta^{-1})\big)$.*

Combining Theorems 3.5 and 4.5, and Lemma 4.7, we obtain the main result of this section.

Theorem 4.8. *Let $\varepsilon, \delta \in (0,1)$ and $\alpha, \tau \geq 1$ be given parameters. Let* W *be a workload distribution and let* Π *be a set of* $|\Pi| = N$ *plans. Define* $\kappa := \max\{\kappa^*, \hat{\kappa}\}$*, where* κ^* *and* $\hat{\kappa}$ *denote the sizes of the smallest* $(\varepsilon, \tau, \delta \mid \mathrm{W})$*-coreset for risk and the smallest* $(\alpha, \delta^2/\mathfrak{m}(\varepsilon, \delta, 2\kappa) \mid \mathrm{W})$*-coreset for neighborhood-cost, respectively. There exists an algorithm that draws* $O((\kappa/\delta)\ln N)$ *queries from* W *and outputs a set* $C \subseteq \Pi$ *of size* $O\big(\kappa \ln \delta^{-1}\big)$ *in* $O\big((\kappa/(\varepsilon^2\delta^2))\, N \ln^3(N) \ln(\delta^{-1})\big)$ *time, such that if* $Q \sim \mathrm{W}$ *and* LocalInference(Q, C) *returns* $\hat{\pi}$*, then*

$$\mathsf{Risk}(\hat{\pi}, Q\,;\,\tau\alpha) \;\leq\; \mathsf{Risk}^{\star}(Q\,;\,\tau) + 4\varepsilon$$

with probability at least $1 - 8\delta$*. Moreover, the inference step runs in time* $\Theta\big(\kappa(\ln\kappa + \ln\delta^{-1})/\varepsilon^2\big)$*.*

Size of neighborhood-cost coresets. Let $\bar{\kappa}$ denote the size of the smallest (worst-case) α-coreset for cost. Then, for any $\delta \in (0,1)$ and any workload distribution W, the size $\hat{\kappa}$ of the smallest $(\alpha, \delta \mid \mathrm{W})$-coreset for neighborhood-cost satisfies $\hat{\kappa} \leq \bar{\kappa}$. This follows because any α-coreset for cost guarantees that, for every cardinality vector $\mathbf{s} \in \mathbb{S}$, $\min_{\pi \in C} \mathsf{Cost}(\pi, \mathbf{s}) \leq \alpha \cdot \mathsf{Cost}^{\star}(\mathbf{s})$, and therefore the failure event never occurs.

Worst-case cost coresets thus provide a clean, distribution-independent upper bound on the size of neighborhood-cost coresets. However, this bound can be overly pessimistic in practice. A worst-case α-coreset for cost must provide a uniform guarantee over the entire cardinality space $\mathbb{S}$, whereas a workload-dependent neighborhood-cost coreset only needs to satisfy the guarantee for $Q \sim \mathrm{W}$ and $\mathbf{s} \sim \mathrm{U}_Q$. Consequently, we expect $\hat{\kappa}$ to be significantly smaller than $\bar{\kappa}$ in typical workloads.

5 Related work

Robust Query Optimization. Previous work has established that cost-based optimizers are highly sensitive to cardinality estimation errors [32], motivating numerous techniques for improving robustness [27, 53]. These approaches can be broadly categorized into robust query processing and robust plan selection; our work focuses on the latter. Various robustness definitions have been proposed in the literature, including worst-case sub-optimality [21, 36, 44], slope or integral of the cost function [45], near-optimality ranges [19, 46], predicted execution time distributions [33], and expected penalty or risk [49]. These definitions operate under different uncertainty models, which themselves vary in formulation from probability density functions [8, 16, 49] to uniform distributions over bounded spaces [1, 9, 23]. Robust plans are particularly valuable for parametric query optimization (PQO), which seeks to minimize optimizer invocations through plan caching while maintaining acceptable performance [7, 14, 19, 22, 28, 30]. PQO typically follows a two-step framework: candidate plan generation and online plan selection [20, 22, 43, 50]. In this context, robust plans serve as excellent caching candidates not only because they reduce optimizer invocations, but because they fundamentally outperform repeated re-optimization by providing consistently good performance across uncertainty scenarios. Xiu et al. [50] demonstrate that caching a compact set of robust plans can achieve superior overall query performance compared to per-query optimization, while simultaneously improving runtime planning efficiency.

Coresets. Coresets, originally developed in computational geometry, provide compact summaries of large optimization instances while preserving provable approximation guarantees [2, 3, 37]. A rich line of work establishes small coresets for geometric problems such as clustering, regression, and shape fitting, with coreset sizes typically depending on the dimension, the desired approximation factor, and sometimes logarithmically on the input size [2, 4, 10–12, 15, 26, 42]. Coresets have also been explored in query-processing settings [4, 6, 54], where a succinct subset of the data is selected during preprocessing and can subsequently be used to answer different types of queries, including top-k, k-regret minimization, and diversity queries.

6 Conclusion

We initiated a theoretical study of coresets for robust query optimization, introducing robust coresets for both expected penalty and risk under uncertainty-aware cost models. Several directions remain open. Interesting directions for future work include tightening the worst-case coreset bounds for broader classes of cost functions beyond the linear setting, as well as developing fully dynamic and online variants of robust coresets that adapt continuously to evolving workloads and feedback from query execution.

References

[1] M. Abhirama, S. Bhaumik, A. Dey, H. Shrimal, and J. R. Haritsa. On the stability of plan costs and the costs of plan stability. *Proc. VLDB Endow.*, 3(1):1137–1148, 2010.

[2] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *Journal of the ACM (JACM)*, 51(4):606–635, 2004.

[3] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52(1):1–30, 2005.

[4] P. K. Agarwal, N. Kumar, S. Sintos, and S. Suri. Efficient algorithms for k-regret minimizing sets. In *16th International Symposium on Experimental Algorithms (SEA 2017)*, pages 7–1, 2017.

[5] P. K. Agarwal and M. Sharir. Arrangements and their applications. In *Handbook of Computational Geometry (J.R. Sack, J. Urrutia, eds)*, pages 49–119. Elsevier, 2000.

[6] P. K. Agarwal, S. Sintos, and A. Steiger. Efficient indexes for diverse top-k range queries. In *Proceedings of the 39th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 213–227, 2020.

[7] G. Aluç, D. DeHaan, and I. T. Bowman. Parametric plan caching using density-based clustering. In *IEEE 28th International Conference on Data Engineering (ICDE 2012)*, pages 402–413, 2012.

[8] B. Babcock and S. Chaudhuri. Towards a robust query optimizer: A principled and practical approach. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 119–130, 2005.

[9] S. Babu, P. Bizarro, and D. J. DeWitt. Proactive re-optimization. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 107–118, 2005.

[10] D. Baker, V. Braverman, L. Huang, S. H.-C. Jiang, R. Krauthgamer, and X. Wu. Coresets for clustering in graphs of bounded treewidth. In *International Conference on Machine Learning*, pages 569–579, 2020.

[11] V. Braverman, D. Feldman, H. Lang, A. Statman, and S. Zhou. New frameworks for offline and streaming coreset constructions. *arXiv preprint arXiv:1612.00889*, 2016.

[12] V. Braverman, S. Jiang, R. Krauthgamer, and X. Wu. Coresets for clustering with missing values. *Advances in Neural Information Processing Systems*, 34:17360–17372, 2021.

[13] S. Chatterji, S. S. K. Evani, S. Ganguly, and M. D. Yemmanuru. On the complexity of approximate query optimization. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 282–292. ACM, 2002.

[14] S. Chaudhuri, H. Lee, and V. R. Narasayya. Variance aware optimization of parameterized queries. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 531–542, 2010.

[15] K. Chen. On coresets for k-median and k-means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.

[16] F. C. Chu, J. Y. Halpern, and P. Seshadri. Least expected cost query optimization: An exercise in utility. In *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pages 138–147, 1999.

[17] S. Cluet and G. Moerkotte. On the complexity of generating optimal left-deep processing trees with cross products. In *Proceedings of the 5th International Conference on Database Theory*, ICDT '95, page 54–67, 1995.

[18] H. Doraiswamy, P. N. Darera, and J. R. Haritsa. On the production of anorexic plan diagrams. In *Proceedings of the 33rd International Conference on Very Large Data Bases*, pages 1081–1092, 2007.

[19] H. Doraiswamy, P. N. Darera, and J. R. Haritsa. Identifying robust plans through plan diagram reduction. *Proc. VLDB Endow.*, 1(1):1124–1140, 2008.

[20] L. Doshi, V. Zhuang, G. Jain, R. Marcus, H. Huang, D. Altinbüken, E. Brevdo, and C. Fraser. Kepler: Robust learning for parametric query optimization. *Proc. ACM Manag. Data*, 1(1):109:1–109:25, 2023.

[21] A. Dutt and J. R. Haritsa. Plan bouquets: query processing without selectivity estimation. In *Proceedings of the ACM SIGMOD International Conference on Management of data*, pages 1039–1050, 2014.

[22] A. Dutt, V. R. Narasayya, and S. Chaudhuri. Leveraging re-costing for online optimization of parameterized queries with guarantees. In *Proceedings of the ACM International Conference on Management of Data*, pages 1539–1554, 2017.

[23] B. Ergenc, F. Morvan, and A. Hameurlain. Robust placement of mobile relational operators for large scale distributed query optimization. In *Proceedings of the 8th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007)*, pages 227–235, 2007.

[24] A. Ghosh, J. Parikh, V. S. Sengar, and J. R. Haritsa. Plan selection based on query clustering. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 179–190, 2002.

[25] M. Goodrich and H. Brönnimann. Almost optimal set covers in finite vc-dimension. *Discrete & Computational Geometry*, 14(4):463–479, 1995.

[26] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing*, pages 291–300, 2004.

[27] J. R. Haritsa. Robust query processing: A survey. *Found. Trends Databases*, 15(1):1–114, 2024.

[28] A. Hulgeri and S. Sudarshan. Parametric query optimization for linear and piecewise linear cost functions. In *Proceedings of 28th International Conference on Very Large Data Bases*, pages 167–178, 2002.

[29] T. Ibaraki and T. Kameda. On the optimal nesting order for computing n-relational joins. *ACM Trans. Database Syst.*, 9(3):482–502, Sept. 1984.

[30] Y. E. Ioannidis, R. T. Ng, K. Shim, and T. K. Sellis. Parametric query optimization. *The VLDB Journal*, 6(2):132–151, 1997.

[31] N. Kumar and S. Sintos. Faster approximation algorithm for the k-regret minimizing set and related problems. In *Proceedings of the 20th Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 62–74. SIAM, 2018.

[32] V. Leis, B. Radke, A. Gubichev, A. Mirchev, P. Boncz, A. Kemper, and T. Neumann. Query optimization through the looking glass, and what we found running the join order benchmark. *VLDB J.*, 27(5):643–668, 2018.

[33] Y. Li, X. Yu, N. Koudas, S. Lin, C. Sun, and C. Chen. dbet: Execution time distribution-based plan selection. *Proceedings of the ACM on Management of Data*, 1(1):1–26, 2023.

[34] J. Liu, W. Dong, D. Li, and Q. Zhou. Fauce: Fast and accurate deep ensembles with uncertainty for cardinality estimation. *Proc. VLDB Endow.*, 14(11):1950–1963, 2021.

[35] M. Mitzenmacher and E. Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis.* Cambridge University Press, 2005.

[36] H. Q. Ngo, E. Porat, C. Ré, and A. Rudra. Worst-case optimal join algorithms: [extended abstract]. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 37–48, 2012.

[37] J. M. Phillips. Coresets and sketches. In *Handbook of Discrete and Computational Geometry (J. E. Goodman, J. O'Rourke, C. D. Tóth eds.)*, pages 1269–1288. Chapman and Hall/CRC, 2017.

[38] C. Ré and D. Suciu. Understanding cardinality estimation using entropy maximization. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 53–64, 2010.

[39] M. Schönberger, I. Trummer, and W. Mauerer. Hybrid mixed integer linear programming for large-scale join order optimisation. *CoRR*, abs/2510.20308, 2025.

[40] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access path selection in a relational database management system. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 23–34, 1979.

[41] I. Trummer and C. Koch. Solving the join ordering problem via mixed integer linear programming. In *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017, Chicago, IL, USA, May 14-19, 2017*, pages 1025–1040. ACM, 2017.

[42] M. Tukan, X. Wu, S. Zhou, V. Braverman, and D. Feldman. New coresets for projective clustering and applications. In *International Conference on Artificial Intelligence and Statistics*, pages 5391–5415, 2022.

[43] K. Vaidya, A. Dutt, V. R. Narasayya, and S. Chaudhuri. Leveraging query logs and machine learning for parametric query optimization. *Proc. VLDB Endow.*, 15(3):401–413, 2021.

[44] S. K. Venkatesh, J. R. Haritsa, S. Kenkre, and V. Pandit. A concave path to low-overhead robust query processing. *Proc. VLDB Endow.*, 11(13):2183–2195, 2018.

[45] F. Wolf, M. Brendle, N. May, P. R. Willems, K. Sattler, and M. Grossniklaus. Robustness metrics for relational query execution plans. *Proc. VLDB Endow.*, 11(11):1360–1372, 2018.

[46] F. Wolf, N. May, P. R. Willems, and K. Sattler. On the calculation of optimality ranges for relational query execution plans. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 663–675, 2018.

[47] W. Wu, Y. Chi, S. Zhu, J. Tatemura, H. Hacigümüs, and J. F. Naughton. Predicting query execution time: Are optimizer cost models really unusable? In *Proceedings of the 29th IEEE International Conference on Data Engineering*, pages 1081–1092, 2013.

[48] W. Wu, X. Wu, H. Hacigümüs, and J. F. Naughton. Uncertainty aware query execution time prediction. *Proc. VLDB Endow.*, 7(14):1857–1868, 2014.

[49] H. Xiu, P. K. Agarwal, and J. Yang. PARQO: penalty-aware robust plan selection in query optimization. *Proc. VLDB Endow.*, 17(13):4627–4640, 2024.

[50] H. Xiu, Y. Li, Q. Yang, P. Agarwal, and J. Yang. PAR2QO: parametric penalty-aware robust query optimization. *Proc. VLDB Endow.*, 18(11):4532–4545, 2025.
[51] H. Xiu, Y. Li, Q. Yang, W. Guo, Y. Liu, S. Roy, P. K. Agarwal, and J. Yang. Hint-qpt: Hints for robust query performance tuning. *Proc. VLDB Endow.*, 18(12):5327–5330, 2025.
[52] Z. Yang, E. Liang, A. Kamsetty, C. Wu, Y. Duan, X. Chen, P. Abbeel, J. M. Hellerstein, S. Krishnan, and I. Stoica. Deep unsupervised cardinality estimation. *Proc. VLDB Endow.*, 13(3):279–292, 2019.
[53] S. Yin, A. Hameurlain, and F. Morvan. Robust query optimization methods with respect to estimation errors: A survey. *SIGMOD Rec.*, 44(3):25–36, 2015.
[54] A. Yu, P. K. Agarwal, and J. Yang. Processing a large number of continuous preference top-k queries. In *Proceedings of ACM SIGMOD International Conference on Management of Data*, pages 397–408, 2012.

A Worst-case Coresets

In general, the minimum size of a coreset of plans with respect to an objective function depends on the query template Γ and the cost model, making general bounds difficult without template-specific assumptions. To obtain template-agnostic worst-case guarantees, we instead focus on families of cost functions commonly used to model plan costs. In practice, these are typically low-degree polynomials, with linear functions being particularly common [28, 47, 48].

In Section A.1, we consider a simplified setting where the cost function of a plan $\pi \in \Pi$ is a bounded-degree polynomial over $\mathbb{R}^d_{\geq 0}$ whose coefficients are integers in the bounded range $[1, M]$, where $M \geq 1$ is a real number. Our goal here is to understand, in this clean setting, how the size of an α-coreset for minimization fundamentally depends on d and M, and whether one can algorithmically approach the smallest possible coreset for a given instance. We first present upper bounds showing that, for any set of plans Π such that for every $\pi \in \Pi$, $\mathsf{Cost}(\pi, \cdot)$ is a polynomial over $\mathbb{R}^d_{\geq 0}$ of degree g with integer coefficients in $[1, M]$, there exists an α-coreset of size $O\big((\log_\alpha M)^{\mathsf{dim}}\big)$ and it can be constructed in $O\big(\mathsf{dim} \cdot |\Pi| + (\log_\alpha M)^{\mathsf{dim}}\big)$ time, where $\mathsf{dim} = O\big((d+g)^{\min\{d,g\}}\big)$. We also establish a nearly tight lower bound when Π is restricted to d-dimensional linear cost functions with coefficients in $[1, M]$. Here we show that one can pick a collection of bounded such cost functions such that every α-coreset for minimization must have size at least polynomial in $\log M$ and exponential in d. This shows that the curse of dimensionality is unavoidable even in this simplified model. Both the lower and upper bounds are shown in Theorem A.1.

Next, in Section A.2, we move beyond worst-case existence and consider the problem of constructing an *instance-optimal* coreset given a set of plans Π with linear cost functions whose coefficients are integers in the range $[1, M]$. Let $\overline{\mathsf{M}} \subseteq \Pi$ denote the smallest α-coreset for the minimization objective. We present an algorithm that constructs an α^3-coreset whose size is within a polylogarithmic factor of $|\overline{\mathsf{M}}|$ in time $O\left(d \cdot |\Pi| + d^{O(d)} \left(\log_\alpha M\right)^{2d^2+d} \log^2 \left(\log_\alpha M\right)\right)$. This result shows that it is possible to compute coresets that are near–instance optimal in time polynomial in $|\Pi|$. The construction of such near-optimal coresets is given in Theorem A.5.

A.1 Worst-case bounds

Theorem A.1. *Let Π be a set of plans with linear cost functions, such that for every $\pi \in \Pi$, $\mathsf{Cost}(\pi, \mathbf{s}) = \langle c_\pi, \mathbf{s} \rangle$ over $\mathbb{R}^d_{\geq 0}$, where each coefficient is an integer in the range $[1, M]$. For any $\alpha \geq 1$, there exists an α-coreset $\mathsf{M} \subseteq \Pi$ of size $O\left(\left(\log_\alpha M\right)^d\right)$ that can be constructed in $O\left(d \cdot |\Pi| + \left(\log_\alpha M\right)^d\right)$ time. Moreover, this bound is almost tight: for every d and $M \geq 1$ there exists a family Π of linear functions for which every α-coreset has size $\Omega\left(\frac{1}{\sqrt{d}} \cdot \left(\log_{\alpha \cdot d} M\right)^{d-1}\right)$. The upper bound extends to polynomial functions: If Π is a set of plans represented by polynomial functions of dimension d and* *degree g, then for any $\alpha \geq 1$, there exists an α-coreset of size $O\left(\left(\log_\alpha M\right)^{\mathsf{dim}}\right)$ that can be constructed in $O\left(\mathsf{dim} \cdot |\Pi| + \left(\log_\alpha M\right)^{\mathsf{dim}}\right)$ time, where $\mathsf{dim} = O\left((d+g)^{\min\{d,g\}}\right)$.*

Upper bound. We first show the upper bound. We directly show the stronger result, proving that the upper bound holds for every set of plans that are represented by polynomial functions.

For a cardinality vector $\mathbf{s} \in \mathbb{R}^d_{\geq 0}$, let $\mathbf{s} = (s_1, \ldots, s_d)$. By definition of Π, for every plan $\pi \in \Pi$,

$$\mathsf{Cost}(\pi, \mathbf{s}) = c_{\pi,0} + \sum_{\vec{r} \in \mathbb{N}^d, 1 \leq ||\vec{r}||_1 \leq g} c_{\pi,\vec{r}} \cdot s_1^{r_1} \cdot \ldots \cdot s_d^{r_d},$$

where $1 \leq c_{\pi,\vec{r}} \leq M$ and $\vec{r} = (r_1, \ldots, r_d)$ is a vector in $\mathbb{N}^d$ with $1 \leq ||\vec{r}||_1 \leq g$. We use $\mathbb{N}$ to denote the set of natural numbers. Let $\mathcal{R} = \{\vec{r} \in \mathbb{N}^d \mid 1 \leq ||\vec{r}||_1 \leq g\}$. Furthermore, for any positive integer x, we define $[x] = \{1, \ldots, x\}$.

Algorithm. For every $\vec{r} \in \mathcal{R}$, we partition the polynomials in Π into $\lambda = \lceil \log_\alpha M \rceil = O(\log_\alpha M)$ buckets. For each $j \in [\lambda]$, we set

$$B_{\vec{r}}[j] = \{\pi \in \Pi \mid \alpha^{j-1} \leq c_{\pi,\vec{r}} < \alpha^j\}.$$

Let $\mathsf{dim} = \binom{d+g}{\min\{d,g\}} - 1 = O\left((d+g)^{\min\{d,g\}}\right)$. Notice that $|\mathcal{R}| = \mathsf{dim}$, so let $\mathcal{R} = \{\vec{r}^{(1)}, \ldots, \vec{r}^{(\mathsf{dim})}\}$. For every $h \in [\mathsf{dim}]$, we have $\vec{r}^{(h)} = (r_1^{(h)}, \ldots, r_d^{(h)})$.

Initially $\mathsf{M} = \emptyset$. For every $j_1, \ldots, j_{\mathsf{dim}} \in [\lambda]^{\mathsf{dim}}$ we add in M the polynomial

$$\underset{\pi \in \Pi : \forall \vec{r}^{(h)} \in \mathcal{R}, \pi \in B_{\vec{r}^{(h)}}[j_h]}{\arg\min} c_{\pi,0},$$

i.e., the plan π with the minimum coefficient $c_{\pi,0}$ among plans that belong in all buckets $B_{\vec{r}^{(h)}}[j_h]$ for $\vec{r}^{(h)} \in \mathcal{R}$.

Lemma A.2. $|\mathsf{M}| = O\left(\left(\log_\alpha M\right)^{\mathsf{dim}}\right)$.

Proof. By definition $|\mathcal{R}| = \mathsf{dim}$. For every $\vec{r}^{(h)} \in \mathcal{R}$ there are $\lambda = O(\log_\alpha M)$ buckets so in total $|\mathsf{M}| = \lambda^{\mathsf{dim}} = O(\log_\alpha^{\mathsf{dim}} M)$. □

Lemma A.3. *M is an α-coreset.*

Proof. Let $\mathbf{s} \in \mathbb{R}^d_{\geq 0}$ be a cardinality vector. Recall that $\pi^\star(\mathbf{s})$ is the optimum plan in Π with respect to $\mathbf{s}$. Notice that by definition, there exist indices $j_1^*, \ldots, j_{\mathsf{dim}}^*$ such that for every $\vec{r}^{(h)} \in \mathcal{R}$, $\pi^\star(\mathbf{s}) \in B_{\vec{r}^{(h)}}[j_h^*]$. Let $\Pi^* \subseteq \Pi$ be the set of polynomials defined as $\Pi^* = \{\pi \in \Pi \mid \forall \vec{r}^{(h)} \in \mathcal{R}, \pi \in B_{\vec{r}^{(h)}}[j_h^*]\}$, thus $\pi^\star(\mathbf{s}) \in \Pi^*$. Let $\bar{\pi} = \arg\min_{\pi \in \Pi^*} c_{\pi,0}$. By the construction of M, it follows that $\bar{\pi} \in \mathsf{M}$. Furthermore, since $\bar{\pi}, \pi^\star(\mathbf{s}) \in \Pi^*$, for every $\vec{r}^{(h)} \in \mathcal{R}$, it holds that

$$c_{\bar{\pi},\vec{r}^{(h)}} \leq \alpha \cdot c_{\pi^\star(\mathbf{s}),\vec{r}^{(h)}}.$$

Hence,

$$\begin{aligned}\mathsf{Cost}(\bar{\pi}, \mathbf{s}) &= c_{\bar{\pi},0} + \sum_{\vec{r}^{(h)} \in \mathcal{R}} c_{\bar{\pi},\vec{r}^{(h)}} \cdot s_1^{r_1^{(h)}} \cdot \ldots \cdot s_d^{r_d^{(h)}} \leq c_{\pi^\star(\mathbf{s}),0} + \alpha \sum_{\vec{r}^{(h)} \in \mathcal{R}} c_{\pi^\star(\mathbf{s}),\vec{r}^{(h)}} \cdot s_1^{r_1^{(h)}} \cdot \ldots \cdot s_d^{r_d^{(h)}} \\ &\leq \alpha \cdot \mathsf{Cost}^\star(\mathbf{s}).\end{aligned}$$

□

The running time of the algorithm for constructing M is $O\left(\mathsf{dim}\cdot|\Pi| + \left(\log_\alpha M\right)^{\mathsf{dim}}\right)$, since we construct all buckets in λ^{dim} time and for every $r \in \mathcal{R}$ we make a single pass over all plans to assign each plan to its corresponding bucket.

We note that for linear functions, $g = 1$ and thus $\mathsf{dim} = d$. Therefore, the upper bound on the size of M and the running time follow when Π consists of plans with linear cost functions.

Lower bound. We construct a set Π of n distinct linear functions in dimension d, where the value of n will be specified later. Specifically, Π consists of all distinct linear functions $\pi = \sum_{j\in[d]} c_{\pi,j}\, s_j$ whose coefficients are defined as $c_{\pi,j} = (d\cdot\alpha)^{i_{\pi,j}}$ for every $j \in [d]$, and whose exponents $i_{\pi,1}, \ldots, i_{\pi,d}$ satisfy $\sum_{j\in[d]} i_{\pi,j} = \left\lceil \frac{d\log_{d\cdot\alpha} M}{2} \right\rceil$ and $0 \le i_{\pi,j} \le \lceil \log_{d\cdot\alpha} M \rceil$.

We will show that every linear function in Π constructed as described above must be included in any α-coreset of Π.

Lemma A.4. *Π is the unique α-coreset of Π.*

Proof. For each $\pi \in \Pi$ let $\mathbf{s}_\pi \in \mathbb{R}^d_{\ge 0}$ be a cardinality vector defined as $\mathbf{s}_\pi = (s_{\pi,1}, \ldots, s_{\pi,d})$, where $s_{\pi,j} = \frac{1}{c_{\pi,j}} = \frac{1}{(d\cdot\alpha)^{i_{\pi,j}}}$.

We have

$$\langle \mathbf{s}_\pi, \pi \rangle = d.$$

For any other $\pi' \in \Pi \setminus \{\pi\}$, notice that $\sum_{j\in[d]} i_{\pi,j} = \sum_{j\in[d]} i_{\pi',j} = \left\lceil \frac{d\log_{d\cdot\alpha} M}{2} \right\rceil$. Since π, π' are two different linear functions in Π, there are at least two indices $j_1, j_2 \in [d]$ such that $i_{\pi,j_1} < i_{\pi',j_1}$ and $i_{\pi,j_2} > i_{\pi',j_2}$. Equivalently, we have $i_{\pi',j_1} - i_{\pi,j_1} \ge 1$ and $i_{\pi,j_2} - i_{\pi',j_2} \ge 1$. Based on these observations, we have

$$\langle \mathbf{s}_\pi, \pi' \rangle > \frac{(d\cdot\alpha)^{i_{\pi',j_1}}}{(d\cdot\alpha)^{i_{\pi,j_1}}} = (d\cdot\alpha)^{i_{\pi',j_1} - i_{\pi,j_1}} \ge \alpha\cdot d.$$

Hence, π must be in every α-coreset of Π. The same argument is repeated for every $\pi \in \Pi$ so Π is the unique α-coreset of Π. □

It remains to bound n, i.e., what is the number of linear functions in Π. This is equivalent to the following instance from discrete mathematics. Given d boxes, where each box can fit at most $\lceil \log_{d\cdot\alpha} M \rceil$ balls, and given $\left\lceil \frac{d\log_{d\cdot\alpha} M}{2} \right\rceil$ (identical) balls, the goal is to compute the number of ways (combinations) that the balls can be placed in the boxes satisfying the constraints. A lower bound for this quantity is $\Omega\left(\frac{1}{\sqrt{d}}\cdot\left(\log_{\alpha\cdot d} M\right)^{d-1}\right)$. Thus, we can construct $n = \Theta\left(\frac{1}{\sqrt{d}}\cdot\left(\log_{\alpha\cdot d} M\right)^{d-1}\right)$ linear functions in Π such that every α-coreset of Π has size n.

A.2 Instance-optimal construction

Theorem A.5. *Let Π be a set of linear plans, where all coefficients are integers in the range $[1, M]$. Given a parameter $\alpha > 1$, there exists an algorithm that computes an α^3-coreset $\widehat{\mathsf{M}} \subseteq \Pi$ of Π in*

$$O\left(d\cdot|\Pi| + d^{O(d)}\left(\log_\alpha M\right)^{2d^2+d}\log^2\left(\log_\alpha M\right)\right)$$

time such that $|\widehat{\mathsf{M}}| \le d^{O(1)}\cdot\log(|\overline{\mathsf{M}}|)\cdot|\overline{\mathsf{M}}|$, where $\overline{\mathsf{M}} \subseteq \Pi$ is the α-coreset of Π with the minimum size.

Proof. Let M be an α-coreset on Π constructed as described in Theorem A.1. By the construction and the proof of Theorem A.1, it holds that for each $\pi \in \Pi$, there exists $\pi_{\mathsf{M}} \in \mathsf{M}$ such that

$$\mathsf{Cost}(\pi_{\mathsf{M}}, \mathbf{s}) \le \alpha\cdot\mathsf{Cost}(\pi, \mathbf{s}), \tag{15}$$

for every cardinality vector $\mathbf{s}$. Recall that for every $\pi \in \Pi$, $\mathsf{Cost}(\pi, \mathbf{s}) = c_{\pi,0} + \langle \pi, \mathbf{s} \rangle = c_{\pi,0} + \sum_{j\in[d]} c_{\pi,j} s_j$.

For every $\pi \in \mathsf{M}$ we construct a plan π' with coefficients $c_{\pi',j} = \alpha^2 \cdot c_{\pi,j}$, for each $j \in [d]$, and $c_{\pi',0} = \alpha^2 \cdot c_{\pi,0}$. Hence, for every $\mathbf{s}$,

$$\mathsf{Cost}(\pi', \mathbf{s}) = \alpha^2 \cdot \mathsf{Cost}(\pi, \mathbf{s}).$$

We set $\Pi' = \mathsf{M} \cup \{\pi' \mid \pi \in \mathsf{M}\}$.

We construct the full arrangement $\mathcal{A}(\Pi')$ of hyperplanes in Π' [5]. For every $\mathbf{s} \in \mathbb{R}^d_{\geq 0}$, let $\mathrm{Order}_\mathbf{s}$ be an ordering of the plans in Π' such that $\pi_1 < \pi_2$ in $\mathrm{Order}_\mathbf{s}$ if and only if $\mathsf{Cost}(\pi_1, \mathbf{s}) < \mathsf{Cost}(\pi_2, \mathbf{s})$. Ties are broken arbitrarily. Similarly to [4], using the arrangement $\mathcal{A}(\Pi')$, we identify η convex regions $X_1, \ldots, X_\eta \in \mathbb{R}^d$, for $\eta = O((2|\mathsf{M}|)^{2d})$, such that for each fixed X_i, and for every pair of cardinality vectors $\mathbf{s}^{(1)}, \mathbf{s}^{(2)} \in X_i$, the ordering of plans with respect to $\mathbf{s}^{(1)}$ is the same as the ordering of the plans with respect to $\mathbf{s}^{(2)}$, i.e., $\mathrm{Order}_{\mathbf{s}^{(1)}} = \mathrm{Order}_{\mathbf{s}^{(2)}}$.

For each $i \in [\eta]$, let $\pi_{(i)}$ be the plan in the lower envelope of M in X_i. By definition, $\pi_{(i)}$ is the first plan in the ordering $\mathrm{Order}_\mathbf{s}$, for every $\mathbf{s} \in X_i$. Recall that $\pi'_{(i)} \in \Pi'$ with $\mathsf{Cost}(\pi'_{(i)}, \mathbf{s}) = \alpha^2 \cdot \mathsf{Cost}(\pi_{(i)}, \mathbf{s})$, for every $\mathbf{s} \in X_i$. Let

$$\mathsf{S}_i = \{\pi \in \mathsf{M} \mid \forall \mathbf{s} \in X_i, \mathsf{Cost}(\pi, \mathbf{s}) \leq \mathsf{Cost}(\pi'_{(i)}, \mathbf{s})\}$$

be the subset of plans in M that precede $\pi'_{(i)}$ in the ordering $\mathrm{Order}_\mathbf{s}$, and let S be the family of sets $\{\mathsf{S}_1, \ldots, \mathsf{S}_\eta\}$. Notice that (M, S) define a set system. We compute a hitting set $\widehat{\mathsf{M}} \subseteq \mathsf{M}$ for (M, S) using the algorithm in [25]; see also [4, 31].

Coreset. We show that $\widehat{\mathsf{M}}$ is an α^3-coreset of Π. Let $\mathbf{s}$ be a cardinality vector such that, without loss of generality, $\mathbf{s} \in X_i$. We observed that for $\pi_{(i)} \in \mathsf{M}$, it holds $\mathsf{Cost}(\pi_{(i)}, \mathbf{s}) \leq \alpha \cdot \mathsf{Cost}^\star(\mathbf{s})$. By definition, $\pi_{(i)} \in \mathsf{S}_i$ and by the solution of the hitting set instance for $(\mathsf{M}, \mathcal{S})$, there exists $\hat{\pi} \in \widehat{\mathsf{M}} \cap \mathsf{S}_i$ such that

$$\mathsf{Cost}(\hat{\pi}, \mathbf{s}) \leq \alpha^2 \cdot \mathsf{Cost}(\pi_{(i)}, \mathbf{s}) \leq \alpha^3 \cdot \mathsf{Cost}^\star(\mathbf{s}).$$

Repeating the same argument for every cardinality vector $\mathbf{s}$, we conclude that $\widehat{\mathsf{M}}$ is an α^3-coreset of Π.

Running time. Next, we analyze the running time. We construct M in $O\left(d \cdot |\Pi| + \left(\log_\alpha M\right)^d\right)$ time, we construct the arrangement $\mathcal{A}(\Pi')$ in $O((2|\mathsf{M}|)^{2d})$ time and identify the regions $X_1, \ldots, X_\eta$ in $d^{O(d)}(2|\mathsf{M}|)^{2d}$ time [4, 5]. We note that the set system (M, S) has $|\mathsf{M}|$ items, and η sets. Using the same algorithm as in [4] we compute $\widehat{\mathsf{M}}$ in $O(|\mathsf{M}| \cdot \eta \cdot \log|\mathsf{M}| \cdot \log \eta)$ time. The running time follows.

Approximation factor on the size of $\widehat{\mathsf{M}}$. Finally, we bound the size of $\widehat{\mathsf{M}}$. First we argue that the hitting set instance (M, S) has an optimum solution of size at most $|\overline{\mathsf{M}}|$. For every $\pi \in \overline{\mathsf{M}}$, by Equation (15), there exists $\pi_\mathsf{M} \in \mathsf{M}$ such that

$$\mathsf{Cost}(\pi_\mathsf{M}, \mathbf{s}) \leq \alpha \cdot \mathsf{Cost}(\pi, \mathbf{s})$$

for every cardinality vector $\mathbf{s}$. Let $\mathbf{s}$ be a cardinality vector (and without loss of generality assume that $\mathbf{s} \in X_i$) and let $\pi \in \overline{\mathsf{M}}$ be the plan such that $\pi = \arg\min_{\pi' \in \overline{\mathsf{M}}} \mathsf{Cost}(\pi', \mathbf{s})$. We have that

$$\mathsf{Cost}(\pi_\mathsf{M}, \mathbf{s}) \leq \alpha \cdot \mathsf{Cost}(\pi, \mathbf{s}) \leq \alpha^2 \cdot \mathsf{Cost}^\star(\mathbf{s}) \leq \mathsf{Cost}(\pi'_{(i)}, \mathbf{s}),$$

hence $\pi_\mathsf{M} \in \mathsf{S}_i$. The last inequality follows because $\mathsf{Cost}(\pi'_{(i)}, \mathbf{s}) = \alpha^2 \cdot \mathsf{Cost}(\pi_{(i)}, \mathbf{s}) \geq \alpha^2 \cdot \mathsf{Cost}^\star(\mathbf{s})$. Thus the set $\{\pi_\mathsf{M} \mid \pi \in \overline{\mathsf{M}}\}$ is a hitting set for (M, S) of size at most $|\overline{\mathsf{M}}|$. The approximation factor follows from [25]. □

Received December 2025; accepted February 2026