Suna: Scalable Causal Confounder Discovery over Relational Data

Jiaxiang Liu
jl6235@columbia.edu
Columbia University

Siyuan Xia
stevenxia@uchicago.edu
University of Chicago

Daniel Alabi
alabid@illinois.edu
University of Illinois Urbana-Champaign

Eugene Wu
ewu@cs.columbia.edu
Columbia University

ABSTRACT

Understanding the causal relationships between treatments and outcomes is fundamental in fields such as healthcare and finance. Causal inference aims to estimate the effect of one variable on another, and critically relies on access to those variables as well as the critical confounders. Unfortunately, data analysts often start with datasets lacking these columns, leading to incorrect estimations. Relational data repositories hold significant potential to augment such datasets with an admissible set of confounders necessary for causal analysis. While recent work has advocated for this potential, these approaches face notable limitations. They either assume the existence of a complete causal diagram over all datasets in the repository, which is impractical; rely on computationally infeasible techniques that do not scale to large data repositories with many features; or can only detect confounders in the absence of causal relations, and are thus ineffective when a causal effect exists.

We observe that the asymmetry between causes and effects used in causal discovery can be exploited to directly identify confounders for a given causal query. In this paper, we establish a connection between the existence of confounders and the presence of unconfounded ancestors of the treatment variable in the underlying causal diagram—without requiring access to the diagram itself. This makes it feasible to iteratively discover confounders until an admissible set is constructed. We propose Suna, a highly optimized, GPU-compatible system that implements a novel end-to-end algorithm for discovering confounders within large relational data repositories. Experiments on both real-world and synthetic datasets demonstrate that our system effectively discovers high-quality confounders. Furthermore, Suna employs algorithmic optimizations to accelerate confounder estimation without materializing joins. Our experiments show that Suna finds high-quality confounders while running >100x faster than existing confounder discovery systems.

1 INTRODUCTION

The goal of observational causal study is to understand the causal effect between a pair of treatment (T) and outcome (O). For example, medical researchers aim to understand to what extent smoking (T) elevates blood sugar levels (O), and advertisers try to determine whether a new ad campaign (T) leads to higher customer purchase rates (O). Understanding these causal relationships facilitates both knowledge discovery and decision-making. Due to the impossibility of observing the same subject under multiple treatment conditions simultaneously – known as the *fundamental problem of causal inference* – researchers must group data into comparable cohorts to effectively study these effects.

Despite the abundance of observational data, determining the causal effect from observational studies remains a challenging task. Researchers cannot actively form similar groups of individuals and assign treatments accordingly. A naive comparison of outcomes between individuals receiving different treatments is likely to yield a mixture of causal effect and spurious correlation. For example, smokers having a higher probability of blood sugar might be due to a specific type of gene that makes people more likely to smoke and have higher levels of blood sugar. Without controlling for these genetic factors, a naive causal analysis could mistakenly attribute the effect of these genetic factors to smoking, leading to a complete reversal of the true causal effect. To properly isolate spurious correlations from the causal effect, it is essential to identify key attributes of the data that forms an admissible adjustment set. By holding these attributes constant, different treatment groups become comparable in terms of their outcomes.

The identification of an admissible adjustment set presents significant practical challenges, because key attributes to control for confounding are often inaccessible. For example, data analysts struggle with picking an adjustment set without the aid of domain experts; more commonly, they start with datasets lacking critical variables to form an admissible adjustment set for a treatment and outcome pair of interest [21]. Without access to these key variables, the issue of *non-identifiability* arises, making it impossible to determine whether the correlation between the treatment and outcome is due to causal or non-causal effects. This can lead to incorrect conclusions regarding the causal query [31, 44].

Existing relational data repositories offer significant potential for enhancing data-oriented tasks, and prior work has applied this idea to machine learning tasks [13, 19]. In addition, past work [48] has explored the overlap between causal inference and large data repositories by leveraging knowledge graphs to discover attributes that offset spurious correlation between a treatment and outcome pair. However, this work assumes that there is no causal relationship between the pair, making it unable to find non-zero causal relations.

In this paper, we present the first work that discovers an admissible adjustment set for an arbitrary causal query $Q = (T, O)$. We use relational data repositories as external knowledge sources to identify an admissible adjustment set. Since these repositories can be large and contain many attributes, our approach needs to discover joinable datasets, integrate them, and identify a set of variables as an admissible adjustment set from a substantial search space. While much existing work focuses on discovering and integrating [8, 25, 26], we tackle the challenge of efficiently constructing an admissible adjustment set for a causal query.

This problem is easy if there exists an entire *causal directed acyclic diagram* (causal DAG) over attributes of all datasets, where each edge represents a causal relationships between two attributes. The *back-door criterion* [31] provides an algorithmic approach to read off one admissible adjustment set directly for a given $Q = (T, O)$ from the causal DAG. However, causal DAGs do not exist for most datasets, not to mention entire data repositories. Without the causal DAG, this problem is challenging for several reasons. Firstly, given an arbitrary candidate adjustment set, it is difficult to verify whether it is admissible for a causal query Q. Secondly, there are many variables in the knowledge source relevant to a given causal query, and any combination of these variables can be a potential candidate, making the number of possible sets exponential. Lastly, assessing each candidate adjustment set requires an integration of *all* participating datasets, which is impractically expensive.

In response, causal discovery methods aim to learn the underlying causal DAG from data. However, because these approaches target single tables, they do not scale to even modestly sized data repositories. For instance, Shimizu et al. [38] does not terminate within 10 hours on a data repository with 10K rows and 500 attributes. Some recent approaches rely on LLMs to construct the causal DAG [49], but it is not established whether LLMs have causal reasoning capabilities [9]. An additional consideration for causal discovery systems is that the adjustment sets they find must remain a modest size so that analysts can interpret the output. While controlling for thousands of variables might theoretically ensure validity, such expansive sets become computationally prohibitive and impossible to analyze.

To address the challenges above, we present a scalable confounder discovery system, Suna, that automatically searches for a minimal, yet sufficient, set of relevant attributes over large data repositories. Given a causal query, Suna (1) identifies an admissible set of variables as confounders with minimal size; (2) returns results within interactive timescales and (3) supports scaling to repositories with thousands or millions of variables. The user submits a dataset and a causal query over a pair of attributes in the dataset, and Suna returns an admissible adjustment set (with links to the original datasets), an estimation of the average treatment effect, and the relevant tables. Internally, Suna iteratively finds variables from datasets in the repository that are both unconfounded with the treatment variable and a *causal confounder* of the causal query. We prove that this iterative approach can construct an admissible adjustment set without the need to build a causal DAG. To make Suna even faster and more scalable, we leverage factorized learning technique to avoid explicitly joining datasets and implement a GPU-friendly architecture that significantly accelerates execution.

Our key innovation is a novel algorithm that leverages *bivariate causal discovery*, a building block for parametric causal discovery, as an oracle to iteratively identify confounders. The main challenge is selecting a variable that removes spurious correlations from a large pool of candidates; by doing this iteratively, we can find an admissible adjustment set. The novel insight from our main theorem shows that this step can be reduced to finding an unconfounded ancestor of the treatment variable, that is not part of the adjustment set found so far. This insight also helps us determine when the current adjustment set becomes admissible, which avoids making the adjustment set unmanageably large. Experiments on real-world repositories show that Suna identifies meaningful confounders for causal queries that are consistent with prior studies. Synthetic experiments show that Suna scales to 1 million variables, and answers the user's causal query with an r^2 accuracy of >0.99, compared to the ground truth. Our contributions include:

- We connect the existence of confounders to the existence of unconfounded ancestors of the treatment variable in an underlying causal diagram.
- We propose a novel algorithm that iteratively discovers confounders for causal queries based on bivariate causal discovery.
- We design a novel data structure that leverages ideas from factorized learning to avoid explicitly materializing joins during confounder discovery, and implement a GPU-compatible system to achieve interactive search latency.
- We evaluate the quality of Suna on a data corpus containing 346 datasets collected from NYC open data [6] and Kaggle datasets [24]. Results show that Suna discovers semantically meaning confounders complying with existing studies, while achieving orders of magnitude runtime improvements over existing baselines. We further use synthetic datasets to quantitatively study the accuracy of Suna.

Note: The paper is self-contained. References to appendices can be disregarded or located in the technical report.

2 RELATED WORK

Confounder Discovery. Several existing work [13, 14, 33, 48, 49] studied the problem of discovering confounders for a causal query $Q = (T, O)$ without an available causal diagram. HypDB aims to identify all parents of the treatment variable as an admissible adjustment set and evaluate various treatment effects through the *backdoor criteria* [33]. However, HypDB assumes that the set of parents must exist in the input dataset, an assumption that is unrealistic when the dataset contains limited features. In our problem, applying this algorithm would require a data integration step, requiring joining and materializing a large dataset with all relevant columns for the treatment and target variables, which is impractical in even modest dataset repositories.

MESA [48] assumes it is known a priori that there is no causal effect from T to O, and aims to find a set of attributes $\mathbf{Z}$ such that $T \perp\!\!\!\perp O \mid \mathbf{Z}$. Despite a similar formulation, this solution is inapplicable when the no-causal-effect prior is removed because the proposed algorithm may find variables that will block causal paths from T to O. Metam [13] proposes a multi-armed bandit algorithm to balance exploring and exploiting datasets in the data corpus to handle a wide range of data tasks, including causal inference. They use existing techniques to answer causal inference queries, which is orthogonal to our approach. Nexus [14] focuses on discovering correlations within a data corpus and inferring hidden confounders based on the discovered results. Finally, a vision is proposed in [49] to answer causal queries by leveraging an external data source. They achieve so by integrating treatment, outcome, and confounders into a unified dataset, followed by adjusting on integrating confounders. However, they rely on constructing a reliable causal diagram from an existing knowledge graph. Lastly, all approaches above involve materializing a single dataset that

includes the treatment, outcome, and all candidate confounders, which is impractical for large relational data repositories.

Causal Discovery. The causal discovery literature focuses on relaxing assumptions for parametric and non-parametric approaches to learn the causal diagram. For example, one popular non-parametric causal discovery algorithm is the PC algorithm [40]. The PC algorithm makes the *causal sufficiency* assumption that the underlying causal diagram does not contain any unobserved confounders. It begins with a complete undirected graph and iteratively prunes edges if two variables are conditionally independent given some other variables. After pruning, the algorithm directs the remaining edges based on the properties of colliders and some heuristics. The FCI algorithm [39] relaxes the *causal sufficiency* assumption by allowing unobserved confounders in the underlying causal diagram. As a tradeoff, FCI can only identify a set of causal DAGs, called a partial ancestral graph (PAG), that are compatible with the dataset. However, not all causal diagrams corresponding to a single PAG evaluate a causal query to the same answers. In such cases, the problem of *non-identifiability* rises again, where the given information is insufficient to estimate the causal query. Consequently, only a limited number of causal queries can be estimated over PAGs [23]. In general, most non-parametric approaches are prohibitively expensive because (1) the number of conditional independence tests is exponential in the number of covariates, and (2) conditional independence tests are both sample inefficient and computationally expensive [36].

On the other hand, parametric causal discovery approaches make assumptions about the data generation process (e.g., additive noise). For instance, LiNGAM [37, 38, 46] assumes a linear causal mechanism with non-Gaussian additive noise, while Hoyer et al. [18] address non-linear causal mechanisms. Under these assumptions, these methods exploit asymmetries between causes and effects to infer causal relationships and topologically sort the variables in the underlying DAG. A final maximum likelihood estimation step learns a causal structure consistent with the topological ordering. Yet, the sorting and the estimation step are still impractically expensive and do not scale to corpora with many attributes, as is the case in data repositories.

Task-based Dataset Search Systems There has been a number of recent work [8, 13, 19, 20, 34] that broadly fall under the category of task-based dataset search systems. On a high level, given a particular task (e.g. machine learning), these systems make use of external data to augment the performance on that task. Suna is one such system that is designed to optimize support for efficient and scalable causal queries.

Deconfounders. Deconfounders [17, 45] infer synthetic latent variables as substitutes for actual confounders in causal analysis. However, these synthetic variables are typically less interpretable than their real counterparts, and thus we consider these approaches orthogonal to our focus on dataset search.

3 BACKGROUND

Data Model. For a given a relation R, we use uppercase letter X to denote an attribute, $R[X]$ be its corresponding column, $\mathsf{dom}(X)$ be its domain, and $S_R = [X_1, \cdots, X_m]$ be its schema. A tuple $t \in R$ denotes a tuple in R, with $t[X]$ representing the value of attribute

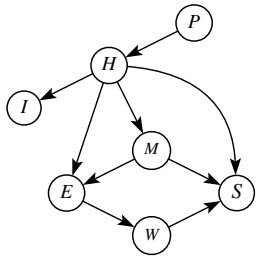


Figure 1: Example of a causal DAG $\mathcal{G}$.

A in tuple t. The domain of R is the Cartesian product of attribute domains $\mathsf{dom}(R) = \mathsf{dom}(X_1) \times \cdots \times \mathsf{dom}(X_m)$. Following conventions in statistics literature, we abuse notation by using the same X to denote the random variable representing attribute X. The annotated relational model [15] maps $t \in R$ to a commutative semi-ring $(D, \oplus, \otimes, 0, 1)$ where D is a set, $\oplus$ and $\otimes$ are commutative binary operators closed over D, and $0/1$ are zero/unit elements.

Causal Inference Foundations. The goal of causal inference is to infer the effect of a treatment attribute T on an outcome attribute O in some dataset about a study population. In the simplest case, the treatment received by each unit in the population is bivariate, consisting of two types of treatments: t_0 and t_1. There are two corresponding potential outcomes, denoted as $O(t_0)$ and $O(t_1)$, that represent the outcome O had the treatment being assigned to t_0 and t_1. From this point, we will use variables to denote attributes. One popular measurement to assess the causal effect for binary treatment variables is the *average treatment effect* (ATE); the ATE of T on O can be defined as

$$\mathsf{ATE}(T, O) = \mathbb{E}[O(t_1) - O(t_0)]$$

The expectation of the potential outcome, $\mathbb{E}[O(t_0)]$, is not equivalent to $\mathbb{E}[O \mid T = t_0]$, the expectation of the outcome given treatment t_0, because the former represents the expected outcome *if all units had been assigned the treatment* t_0, which is a hypothetical scenario that cannot be observed. The *fundamental problem of causal inference* arises because each unit can only receive either treatment t_0 or t_1. As a result, $\mathbb{E}[O(t_0)]$ cannot be trivially estimated from observational data [31]. To avoid bias and accurately estimate the potential outcome $\mathbb{E}[O(t_1)]$, a key step is to identify a set of variables $\mathbf{Z}$ such that, when conditioned upon, the treatment T is independent of the potential outcomes $O(t_0)$ and $O(t_1)$ (i.e., $O(t_0), O(t_1) \perp\!\!\!\perp T \mid \mathbf{Z}$). Then, the definition of the average treatment effect can be rewritten using probability axioms:

$$\mathbb{E}[O(t_i)] = \mathbb{E}[O \mid T = t_i, \mathbf{Z} = \mathbf{z}] \Pr(\mathbf{Z} = \mathbf{z})$$
$$\mathsf{ATE}(T, O) = (\mathbb{E}[O \mid T = t_0, \mathbf{Z} = \mathbf{z}] - \mathbb{E}[O \mid T = t_1, \mathbf{Z} = \mathbf{z}]) \Pr(\mathbf{Z} = \mathbf{z})$$

Causal DAG. A causal DAG $\mathcal{G}$ is a direct acyclic graph where each node in $\mathbf{V}(\mathcal{G})$ represents a variable, and each directed edge in $\mathbf{E}(\mathcal{G})$ indicates a potential causal relationship between two variables with the head node as cause and the tail node as effect. For example, Figure 1 (adopted from [47]) encodes potential causal relationships between 7 variables – large population size of a country (P) has a negative effect on the HDI of that country (H), which affects the internet penetration rate (I), mean year of schooling (M) in the country; a software developer's education level (E) is affected by

both M and H, and affects his length of work experience (W), which finally is causal factor of the developer's salary (S).

For each node V_i in the causal diagram, all nodes having a directed arrow towards it are regarded as the parents of V_i, denoted as $\mathrm{Pa}(V_i)$. A path p is a sequence of variables $(V_1, \ldots, V_k)$ such that each (V_i, V_{i+1}) is an edge in the undirected $\mathcal{G}$; p is a directed path if each $(V_i, V_{i+1}) \in \mathbf{E}(\mathcal{G})$. A collider V_i in p is when both V_{i-1} and V_{i+1} are parents of V_i. For concreteness, consider variable H in Figure 1, $\mathrm{Pa}(H) = \{P\}$ and S is a collider in the path $p = (W, S, H)$. A node V_i is a descendant of V_j if there exists a directed path $(V_j, \ldots, V_i)$. We use $\mathrm{De}(V_i)$ and $\mathrm{ND}(V_i)$ to denote the set of descendants and non-descendants of V_i. Again, consider M in Figure 1, $\mathrm{De}(M) = \{E, S, W\}$ and $\mathrm{ND}(I) = \{I, H, P\}$

D-separation and Causal Paths. D-separation offers a graphical criteria based on *open* and *closed* paths to determine (in)dependencies between pairs of variables without examining the data. For a causal diagram $\mathcal{G}$ over variables $\mathbf{V}(\mathcal{G})$ and any pair of variables (V_i, V_j) such that $V_i \neq V_j$, a path $p = (V_1, \ldots, V_k)$ in $\mathcal{G}$ is *open* conditioned on a set of variables $\mathbf{Z}$ if, for each triplet (V_{i-1}, V_i, V_{i+1}) in p

- if V_i is not a collider, then $V_i \notin \mathbf{Z}$.
- if V_i is a collider, then there exists $V_j \in \mathrm{De}(V_i) \cup \{V_i\}$ such that $V_j \in \mathbf{Z}$.

Otherwise, p is *closed* conditioned on $\mathbf{Z}$. V_i is dependent on V_j conditioned on $\mathbf{Z}$ if there exists a path $p = (V_i, \ldots, V_j)$ that is *open* conditioned on $\mathbf{Z}$. A causal path between the treatment and outcome variable is a directed path from T to O in the causal DAG.

Causal Confounders and Admissible Adjustment Sets. A *causal confounder* [16] for Q is a variable that, when conditioned upon, reduces spurious correlations in estimating the causal effect. Similarly, we say a variable Z is a causal confounder relative to a set of variables $\mathbf{Z}$ if conditioning on Z in addition to $\mathbf{Z}$ further reduces spurious correlations compared to conditioning on $\mathbf{Z}$ alone. From this point, we will use confounders to refer to causal confounders.

An admissible adjustment set is a set of variables $\mathbf{Z}$ such that when conditioned on, all non-causal paths from T to O are closed, and all causal paths from T to O remain open. Semantically, after condition on $\mathbf{Z}$, the dependency between T and O is only due to causal paths. Once an admissible adjustment set $\mathbf{Z}$ has been identified, conventional machine learning techniques can estimate the treatment effect. For example, we can train a linear regression model $O \sim \beta\mathbf{Z} + \alpha T$; the coefficient of the treatment T, α, estimates the average treatment effect of T on O.

Back-door Criterion. The back-door criterion [31] in the causal inference literature characterizes admissible adjustment sets using graphical conditions.

Definition 3.1 (Back-door Criterion). *Given a causal diagram $\mathcal{G}$, a treatment variable T and an outcome variable O. A set of variables $\boldsymbol{Z}$ satisfies the back-door criterion if*

- *all back-door paths*[1] *with respect to Q are closed conditioned on $\boldsymbol{Z}$.*
- *$V \notin \boldsymbol{Z}$ if $V \in \mathrm{De}(T)$.*

[1] a backdoor path with respect to Q is a path $(T, V_i, \ldots, O)$ where the first directed edge is $T \leftarrow V_i$.

Additive Noise Model. Following the definition of *Structural Causal Model (SCM)* [31], each variable in the causal model can be expressed as

$$V_i = f_i(\mathrm{Pa}(V_i), \epsilon_i)$$

where the causal mechanism f_i is an arbitrary function and ϵ_i is a random noise variable independent from any variable $V \in \mathrm{Pa}(V_i)$. The *Additive Noise Model* (**ANM**) [40] is a special case of SCM in which the independent noise variable is additive

$$V_i = f_i(\mathrm{Pa}(V_i)) + \epsilon_i$$

Causal Discovery for LiNGAM. Linear Non-Gaussian Acyclic Model (LiNGAM) is a type of **ANM** where each causal mechanism f_i is linear, and the random noise ϵ_i is non-Gaussian. Under LiNGAM assumptions, the correct causal direction between any two unconfounded variables with an unknown causal relationship can be determined. This is a special case applying the contrapositive of the Darmois-Skitovitch theorem [12]. In practice, one determines the causal direction between a pair of unconfounded variables A and B by training two linear regression models $M_{A\to B}$ (A predicts B) and $M_{B\to A}$, followed by a measure of independence [38]. The direction where there is a statistically significant dependency between the explanatory variable (e.g., A) and the prediction's residual (e.g., $B - M_{A\to B}$) is rejected. The unconfoundedness between A and B is crucial, because otherwise the explanatory variable and the prediction residuals could be dependent even in the correct causal direction. We refer to this building block as *bivariate causal discovery* (BCD). By assuming *causal sufficiency*, a source node S always exists and can be identified from the causal DAG; it is unconfounded with its descendants and independent of its non-descendants. Iteratively, a source node can be discovered and removed to sort $\mathbf{V}(\mathcal{G})$ in topological order.

4 PROBLEM DEFINITION

Problem Formulation. Let $\mathcal{R} = \{R_1, R_2, \ldots\}$ be a data corpus with a set of relations. We denote the set of features for each relation R_i as $\mathbf{A}_{R_i}$, and let $\mathbf{A} = \cup_i \mathbf{A}_{R_i}$ denote the collection of all features across all relations. A data analyst submits a causal query $Q = (T, O)$ in an input relation R, and aims to discover a set of variables as an admissible adjustment set to study the causal effect of T on O.

Problem 1 (Confounder Discovery for Causal Query). *Given a causal query $Q = (T, O)$ in input relation R, let $\mathcal{G}$ be an underlying causal diagram with $\boldsymbol{V}(\mathcal{G}) = \boldsymbol{A} \cup \{T, O\}$, find the set of variables $\boldsymbol{Z}^*$ as attributes in relations that is join-compatible with R such that*

$$\begin{aligned} \boldsymbol{Z}^* = \quad & \operatorname*{argmin}_{\boldsymbol{Z}} |\boldsymbol{Z}| \\ s.t. \quad & T \perp\!\!\!\perp O \mid_{\mathcal{G}_{\underline{T}}} \boldsymbol{Z} && \textit{unconfoundedness} \\ & \boldsymbol{Z} \cap \mathrm{De}(T) = \emptyset && \textit{no disturbance} \end{aligned}$$

$\mathcal{G}_{\underline{T}}$ is the causal diagram generated by removing all edges where T is a head vertex. The two constraints correspond to the back-door criterion in Definition 3.1.

Assumptions. We assume that all relations in $\mathcal{R}$ are join-compatible with R where $\mathbf{A} \cup \{T, O\}$ can be modeled with a causal DAG $\mathcal{G}$ satisfying *causal sufficiency* and LiNGAM. While the non-Gaussianity property from LiNGAM may seem counterintuitive, this property is frequently observed in real-world data, such as financial and

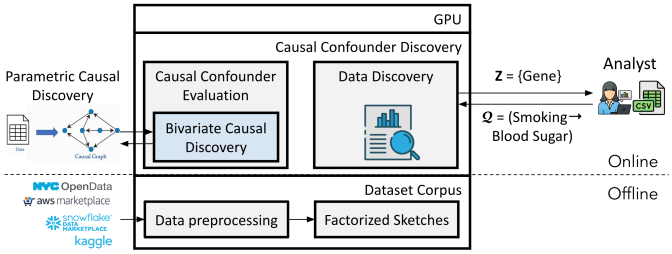


Figure 2: Overview of Suna's architecture. Suna has an offline and an online component. The offline component preprocesses and computes sketches containing sufficient statistics from each dataset. The online component uses the computed sufficient statistics to iteratively discover causal confounders.

sensor data [10, 42]. In fact, the join-compatibility of **all** relations in $\mathcal{R}$ with R can be relax to handle multiple causal queries; for each causal query Q_i, the subset of all relations join-compatible with R is modeled with LiNGAM.

5 SYSTEM ARCHITECTURE

The primary contribution of this work is our algorithm for iteratively identifying confounders without the need to construct the full causal diagram or materializing $R \bowtie \{R_i\}_{i=1}^{n}$. Thus, we design the Suna system to support an efficient and scalable implementation of the algorithm.

Figure 2 gives an overview of the system architecture. Suna has an offline and an online phase. The offline phase involves collecting tabular datasets from open data repositories [6, 24], preprocessing them by aggregating and removing outliers [29], then transforming them into semi-ring sketches that will be used as input to the confounder discovery algorithm in the online phase. A crucial feature of these sketches is that they allow the confounder discovery algorithm to be run without materializing joins, leading to large performance gains. Suna's online phase processes analysts' causal queries. Upon receiving a query, Suna's *data discovery component* leverages existing techniques to find datasets that are join-compatible with the query input. The query is then processed by the *causal confounder evaluation component*, which employs BCD as an oracle to directly find confounders. Moreover, the *causal confounder evaluation component*'s algorithm is designed to be parallelizable, making it possible for us to implement GPU acceleration. These components work together to let Suna achieve interactive latency in processing, even for large-scale datasets.

6 CAUSAL CONFOUNDER DISCOVERY

We now describe the theoretical and algorithmic foundations to solving Problem 1. We refer to attributes in $\mathbf{A}$ as variables, and describe how to select a subset of $\mathbf{A}$ as an adjustment set. The main challenge is that *no method exists for validating the correctness of an arbitrary $\mathbf{Z}$ without referring to a causal diagram.* We present a novel algorithm that uses BCD to identify the topological order for a superset of the adjustment set in the underlying (but not accessible) causal diagram, and iteratively select an adjustment set that satisfies *unconfoundedness* and *no disturbance* in Problem 1.

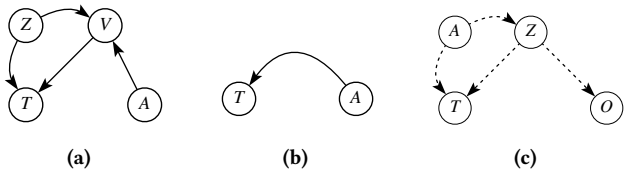


Figure 3: Figure 3a is a example causal diagram $\mathcal{G}$ where $\mathbf{V}(\mathcal{G}) = \{A, T, V, Z\}$, Figure 3b is a causal diagram representing the projection of $\mathcal{G}$ onto $\{T, A\}$. Figure 3c is a snippet of a causal diagram with $\{A, T, Z, O\} \subseteq \mathbf{V}(\mathcal{G})$, dotted directed edges indicate a directed path from the head node to the tail node in $\mathcal{G}$, i.e., there exists a directed path from Z to T in $\mathcal{G}$.

This section is organized into three parts. Firstly, we present a baseline that extends the LiNGAM causal discovery algorithm to take the causal query $Q = (T, O)$ into account, and analyze its limitations. Secondly, we prove the main theory that bridges bivariate causal discovery with confounder discovery and enables our iterative discovery method. Lastly, we describe an end-to-end algorithm that iteratively discovers confounders for Q until the *unconfoundedness* constraint is satisfied.

6.1 Baseline

The back-door criterion implies that the set of all non-descendants of the treatment variable $\mathrm{ND}(T)$ forms an admissible adjustment set for Q because it automatically closes all back-door paths. As a result, one simple extension of LiNGAM's causal discovery algorithm is to build the adjustment set by starting with an empty adjustment set $\mathbf{Z} = \emptyset$, and iteratively add the source node of $\mathbf{A}$, as discovered by the vanilla LiNGAM algorithm, to $\mathbf{Z}$. Since variables are added to $\mathbf{Z}$ in topological order in $\mathcal{G}$, we can terminate the vanilla LiNGAM algorithm and conclude that the current $\mathbf{Z}$ is admissible if the current source node is T. It is guaranteed that $\mathbf{Z} \subseteq \mathrm{ND}(T)$, so $\mathbf{Z}$ forms an admissible adjustment set.

However, this approach falls short in two ways. First, the size of $\mathrm{ND}(T)$ can be arbitrarily large, leading to a huge adjustment set that contains attributes spanning over many datasets, which is both sample-inefficient and analytically challenging. Second, the iterative process of obtaining a topological order of $\mathbf{V}(\mathcal{G})$ requires calling BCD $O(|\mathbf{Z}||\mathbf{A}|^2)$ times where $|\mathbf{A}|$ can be large.

Example 1. *Consider $Q = (E, S)$ with Figure 1 as the underlying causal DAG. (P, H, I, M, E, W, S) is a valid topological order over $\mathbf{V}(\mathcal{G})$, suggesting $\mathbf{Z} = \{P, H, I, M\}$ as an adjustment set. However, P and I do not close any backdoor path for Q, so $\mathbf{Z}$ is still admissible even if we remove them.*

The main problem with the approach above is that it does not guarantee the variable discovered in each iteration to be a confounder. We now study how to address this problem.

6.2 The Main Theorem

In this section, we first introduce causal diagram projection, which defines the causal relationship on a subset of variables $\mathbf{V}' \in \mathbf{V}(\mathcal{G})$. We then present our insights on this projection and state our main theorem, which provides the theoretical support for an iterative algorithm that only identifies confounders.

6.2.1 Connection between Causal Confounder Discovery and Bivariate Causal Discovery. We adopt the definition of causal diagram projection from [43]. At a high level, this definition characterizes the causal relationships among a subset of variables within a larger causal diagram. This approach allows us to focus on the causal structures between variables of interest. Given two variables V_1 and V_2 in $\mathbf{V}(\mathcal{G})$ where (1) there are no open non-causal paths between V_1 and V_2 (*unconfounded*), and (2) there exists a causal path from V_1 to V_2 (*causally related*), any projection of $\mathbf{V}(\mathcal{G})$ onto a set $\{V_1, V_2\} \subseteq \mathbf{V}'$ will include a directed edge $V_1 \to V_2$ and will not contain a bi-directed edge $V_1 \leftrightarrow V_2$.

Example 2. *Consider variables T and A in the causal DAG in Figure 3a, the non-causal path, (A, V, Z, T), between A and T is closed, while the causal path (A, V, T) is open. Thus, the projection of this causal DAG onto $\{A, T\}$ shows a simple cause-and-effect relationship as shown in Figure 3b.*

Under the LiNGAM assumption, the projection of a causal DAG onto two unconfounded and causally related variables will always exhibit a simple cause-and-effect relationship satisfying LiNGAM. With this observation, we now introduce our main theorem.

Theorem 6.1. *Fix a causal diagram $\mathcal{G}$, a causal query $\mathcal{Q} = (T, O)$, and a set of variables $\mathbf{Z}$ where $\mathbf{Z} \cap \mathrm{De}(T) = \emptyset$. If there exists an open non-causal path for $\mathcal{Q}$ conditioned on $\mathbf{Z}$ in $\mathcal{G}$, there must exist a variable $Z \in \mathbf{V}(\mathcal{G})$ satisfying (1) there is an open causal path from Z to T, and all non-causal paths between Z and T are closed conditioned on $\mathbf{Z}$ in the projection of $\mathcal{G}$ onto $\{T, Z\} \cup \mathbf{Z}$; and (2) Z is a confounder for $\mathcal{Q}$ conditioned on $\mathbf{Z}$.*

Due to space constraints, we defer the proof in Appendix A. To illustrate, consider $\mathbf{Z} = \emptyset$ and an open non-causal path $p = (T, \ldots, Z, \ldots, O)$ in Figure 3c, where Z has a directed path to both T and O. Z violates condition (1) due to confounding with T via the back-door path $p' = (T, \ldots, A, \ldots, Z)$, resulting in an open non-causal path $(T, \ldots, A, \ldots, Z, \ldots, O)$ for $\mathcal{Q}$. A satisfies both conditions, as required. If A violates condition (1), we iterate this procedure. Termination is guaranteed since each iteration identifies variables higher in the topological ordering, eventually yielding a variable satisfying both conditions.

One direct implication of Theorem 6.1 is that, when $\mathbf{Z} = \emptyset$, we can detect the presence of open non-causal paths in the causal diagram for $\mathcal{Q}$, by applying BCD on each (T, Z) pair where $Z \in \mathbf{V}(\mathcal{G})$. Specifically, if such a path exists, there will be at least one variable Z for which regressing T on Z results in a significant dependency between T and the residuals of this regression. Conversely, if such a Z does not exist, we can safely conclude that there is no open non-causal path for $\mathcal{Q}$ in $\mathcal{G}$ conditioned on $\mathbf{Z}$; hence $\mathbf{Z}$ forms an admissible adjustment set. However, Theorem 6.1 does not immediately translate to an algorithm because (1) empirical estimation error can lead to false positives and false negatives that violates the condition $\mathbf{Z} \cap \mathrm{De}(T) = \emptyset$, (2) it only guarantees to find a superset C that contains a confounder, without giving a strategy to select the confounder from C, and (3) it does not address how to account for a non-empty $\mathbf{Z}$ in BCD after the first iteration. We now present our solution to these challenges.

Algorithm 1 Iterative Confounder Discovery

```
1:  Input: R, (T, O) ∈ S_R, τ
2:  Return: Z
3:  Z ← {}
4:  while True do
5:      C ← {}
6:      for all R_i ∈ R do
7:          B ← gen-bootstrap(R, R_i)        ▷ generate bootstrap samples
8:          for all Z ∈ R_i do               ▷ augment R w/ R_i's attrs
9:              for all b ∈ B do
10:                 p_b^{Z+}, p_b^{Z−} ← bivariate-CD(b, Z)
11:                 CI ← CI ∪ {p_b^{Z−} − p_b^{Z+}}
12:             end for
13:             C ← C ∪ Z if q_τ(CI) significant
14:         end for
15:     end for
16:     break if C = {}
17:     Z_opt ← heuristic-search(C)
18:     Z ← Z ∪ Z_opt
19:     update-corpus(Z_opt, R)
20: end while
```

6.3 Algorithm

This section presents the full algorithm that iteratively discovers confounders for $\mathcal{Q}$ in Algorithm 1, while accounting for the three challenges explained previously. We first give the algorithm overview, then explain individual components in depth. The input of the algorithm is the data corpus $\mathcal{R}$ and an input dataset containing the treatment and outcome pair (T, O). Each iteration tests whether $Z \in R_i$ is a unconfounded ancestor of T (L6-15), with respect to the current adjustment set $\mathbf{Z}$. `heuristic-search` then explores these confounder candidates and identifies the optimal confounder Z_{opt} (L17). Lastly, we update $\mathbf{Z}$ (L18) and $\mathcal{R}$ (L19) to facilitate confounder discovery in future iterations.

6.3.1 Bootstrap Resampling. To enhance estimation robustness and reduce estimation error, `gen-bootstrap` (L7) generates resampled datasets with replacement for conducting hypothesis tests across multiple samples. `bivariate-CD` (L10) implements bivariate causal discovery, returning dependency measures between the explanatory variable and prediction residuals for both forward ($Z \to T$) and backward ($T \to Z$) causal directions. We employ mutual information scores to quantify these dependencies [27], defining $p^{Z+} = \mathrm{MI}(Z; \epsilon_T)$ and $p^{Z-} = \mathrm{MI}(T; \epsilon_Z)$, where ϵ_T and ϵ_Z represent prediction residuals in their respective models. Variables f for which the $100 \cdot (1 - \tau)$-percentile of the bootstrapped distribution ($q_\tau(\mathrm{CI})$) is significant are considered as candidates of confounders (L13), where τ is a tunable hyperparameter defaulted to 0.05. Semantically, this difference serves as a score indicating the likelihood of Z being an ancestor of T.

6.3.2 Heuristic Selection. Theorem 6.1 guarantees that there exists a variable $Z \in C$ such that conditioning on Z mitigate spurious correlation between T and O and reduces their dependency. As a result, one approach to selecting a confounder from C is to estimate the mutual information score between T and O with and without conditioning on Z for each $Z \in C$. A non-trivial difference in $\mathrm{MI}(T; O \mid \mathbf{Z}) - \mathrm{MI}(T; O \mid \mathbf{Z} \cup \{Z\})$ suggests that Z is a confounder. Therefore, once we have identified a superset of variables containing at least one confounder, we can select the confounder based

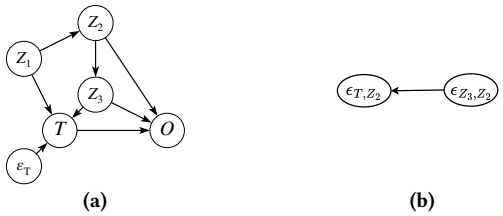


Figure 4: Figure 4a is a causal diagram consisting treatment T and outcome O, a latent variable ϵ_T representing the noise of T, and the three other variables Z_1, Z_2 and Z_3. Figure 4b is a causal diagram considering the residuals of T and Z_3 predicted by Z_2.

on a drop in mutual information when conditioned upon. Empirically, we can also use the absolute difference in coefficients of T regressing $T \cup \mathbf{Z}$ and $\{T, Z\} \cup \mathbf{Z}$ on O.

6.3.3 Update Residuals. In iterations where $\mathbf{Z}$ is non-empty, our algorithm must identify variables that close remaining open non-causal paths *conditioned on the current* $\mathbf{Z}$. For clarity of illustration, we assume the current adjustment set $\mathbf{Z}$ contains a single confounder, though the underlying methodology generalizes seamlessly to adjustment sets of higher cardinality.

Let $\mathbf{Z} = \{A\}$. Due to condition (1) in Theorem 6.1, A is an admissible adjustment set for the causal query $\boldsymbol{Q} = (Z, T)$. Hence, for linear causal mechanisms, we write

$$T = \alpha Z + \beta A + \epsilon_T$$

for arbitrary constants α and β where $A, Z \perp\!\!\!\perp \epsilon_T$. Let $\epsilon_{T,A}$ and $\epsilon_{Z,A}$ denote the prediction residuals of regressing A on T and regressing A on Z, then

$$\epsilon_{T,A} = \alpha \epsilon_{Z,A} + \epsilon_T$$

If we treat $\epsilon_{T,A}$ and $\epsilon_{Z,A}$ as variables, and construct a causal diagram between them, the causal diagram is $\epsilon_{Z,A} \to \epsilon_{T,A}$ that satisfies the LiNGAM property. Hence, a significant $p^{\epsilon_{Z,A}-} - p^{\epsilon_{Z,A}+}$ indicates that Z is a variable satisfying both conditions in Theorem 6.1.

Example 3. *Consider a causal diagram $\mathcal{G}$ in Figure 4a, where T is the treatment variable and $\mathbf{Z} = \{Z_2\}$. In this setup, the causal path from Z_3 to T is open, while the non-causal path (T, Z_1, Z_2, Z_3) is closed conditioned on Z_2. By computing the residuals ϵ_{T,Z_2} and ϵ_{Z_3,Z_2} after regressing Z_2 on T and Z_3, the causal diagram regarding ϵ_{T,Z_2} and ϵ_{Z_3,Z_2} is shown in Figure 4b.*

For each iteration, we always need to consider the current $\mathbf{Z}$ over BCD. That is, we work with $\epsilon_{T,\mathbf{Z}}$ and $\epsilon_{Z,\mathbf{Z}}$[2] for each $Z \in \mathbf{Z}$ in each iteration. By applying the Frisch-Waugh-Lovell theorem, those $\epsilon_{T,\mathbf{Z}}$ and $\epsilon_{f,\mathbf{Z}}$ can be maintained incrementally over the search.

Example 4. *Let Z_1 and Z_2 be the causal confounders discovered over the first 2 iterations. We want to compute* BCD *between T and Z_3 conditioned on $\mathbf{Z} = \{Z_1, Z_2\}$. Note that:*

$$T = \alpha_1 Z_1 + \alpha_2 Z_2 + \epsilon_{T,\mathbf{Z}}$$
$$\epsilon_{T,Z_1} = \alpha_2 \epsilon_{Z_2,Z_1} + \epsilon_{T,\mathbf{Z}}$$

[2]the residuals of T and Z after regressing on all variables in $\mathbf{Z}$, $\epsilon_{T,\mathbf{Z}} = T$ and $\epsilon_{Z,\mathbf{Z}} = Z$ when $\mathbf{Z} = \emptyset$.

Hence, we maintain ϵ_{T,Z_1} and ϵ_{Z_2,Z_1} at the end of the first iteration once Z_1 is discovered. Then, we can calculate $\epsilon_{T,\mathbf{Z}}$ by regressing ϵ_{Z_2,Z_1} on ϵ_{T,Z_1} and taking the prediction residual.

7 OPTIMIZATIONS

The major performance bottleneck is L6-15 in Algorithm 1, which runs BCD between the treatment variable and every other variable in $\mathbf{A}$, and update residuals in L19. Suna accelerates these steps by factorizing BCD to avoid join materialization and using GPU-acceleration to answer queries in seconds.

Main Bottlenecks. For every pair of treatment and potential confounder variables (T, Z), where $T \in S_R$ and $Z \in S_{R_i}$, `bivariate-CD` consists of four steps: (1) materialize $R \bowtie_{\mathcal{J}} R_i$; (2) train linear regression models $M_{T \to Z}$ to predict Z from T and $M_{Z \to T}$ to predict T from Z, and calculate the prediction residuals $\epsilon_Z = Z - M_{T \to Z}$ and $\epsilon_T = T - M_{Z \to T}$; (3) assess the dependency level between T and ϵ_Z and between Z and ϵ_T by computing the mutual information scores $\mathsf{MI}(T; \epsilon_Z)$ and $\mathsf{MI}(Z; \epsilon_T)$; and (4) use Z_{opt} to predict all variables in $\mathbf{A}$ and update them to their respective prediction residuals, once the optimal confounder Z_{opt} is discovered. The steps are computationally expensive because of the many (T, Z) pairs (due to potentially large numbers of candidate confounders), and the high cost of materializing joins, training models, computing residuals and estimating mutual information.

Factorized Learning. To address the challenges of training regression models and updating variables to their respective prediction residuals, Suna leverages *factorized learning* techniques [29] to distribute linear regression training and inference over joins by pre-computing sketches. These sketches contain semi-ring annotations that transform group-by and join operations into addition ($\oplus$) and multiplication ($\otimes$) operations within a semi-ring structure $(D, \oplus, \otimes, 0, 1)$. The annotation for group-by $\gamma_{\mathcal{J}} R$ is the sum of the annotations within the group, and the annotation of a tuple $t \in R_1 \bowtie R_2$ is the product of annotations from contributing tuples in R_1 and R_2. The sketches for regression models are called the variance semi-ring, which consists of the count, sum and sum of squares of attributes for each candidate confounder $Z \in \mathbf{A}$ over $\mathcal{J}$. The key optimization of *factorized learning* pushes aggregations γ ($\oplus$) before joins $\bowtie$ ($\otimes$) to avoid aggregating on the materialized join. During pre-computation, each tuple in $t \in R$ is lifted to a semi-ring annotation via a lift function $g(\cdot) : \mathrm{dom}(R) \to D$, and aggregated for each group defined by $\mathcal{J}$. Pre-aggregated sketches have several advantages: (a) they are smaller than the original dataset—linear in the join key's domain; (b) they are sufficient for training linear regression models over joins; and (c) they can be easily updated to a new variance semi-ring suitable for deriving prediction residuals. When combined with GPU acceleration, Suna can train of over one million linear regression models through joins in $\approx 1sec$.

Although factorized learning techniques address steps (1,2,4), they do not address (3), because of the inherent challenges in estimating mutual information. Estimating $\mathsf{MI}(T; \epsilon_Z)$ for continuous variables T and ϵ_Z requires estimating the differential entropies $h(T), h(\epsilon_Z)$ and joint differential entropy $h(T, \epsilon_Z)$, which in turn depends on estimating the joint pdf $\Pr(T, \epsilon_Z)$.

Due to the invariance property of mutual information under invertible transformations, `bivariate-CD`(T, Z) is equivalent to

bivariate-CD($\widetilde{T}, \widetilde{Z}$), where $\widetilde{T}$ and $\widetilde{Z}$ denote the normalized columns T and Z in $R_{\bowtie}$. Then, the significance level of a candidate confounder can be rewritten as

$$p^+ - p^- = \mathsf{MI}(\widetilde{T}; \widetilde{Z} - \beta \cdot \widetilde{T}) - \mathsf{MI}(\widetilde{Z}; \widetilde{T} - \beta \cdot \widetilde{Z}) = \mathsf{MI}(\widetilde{T}; \widetilde{\epsilon_Z}) - \mathsf{MI}(\widetilde{Z}; \widetilde{\epsilon_T})$$

As pointed out in [22], this is equivalent to estimating $h(\widetilde{T}) + h(\widetilde{\epsilon_Z}) - h(\widetilde{Z}) - h(\widetilde{\epsilon_T})$, which boils down to estimating the marginal pdfs $\Pr(\widetilde{T}), \Pr(\widetilde{Z}), \Pr(\widetilde{\epsilon_T})$ and $\Pr(\widetilde{\epsilon_Z})$. However, these distributions can only be computed after materializing $R_{\bowtie}$. For instance, in Figure 6a, the distribution of $R[T]$ differs from that of $R_{\bowtie}[T]$ in Figure 6b. Further, no semi-ring (sufficient statistics) exists for this estimation, which necessitates join materialization. To avoid this, we develop a two-phase solution: we first adopt equi-width histograms for marginal pdf estimation, then design a novel semi-ring structure that efficiently computes these histogram-based estimates.

7.1 Histogram-based BCD

7.1.1 Equi-width Histogram. For a relation R and an attribute $X \in S_R$, an equi-width histogram over $R[X]$ places $\{t[X]\}_{t \in R}$ into non-overlapping bins of the same width. To minimize the integrated mean squared error between the histogram and the underlying distribution, we follow Scott's rule [35] and set the bin width to be $w_x = (24\sqrt{\pi})^{1/3} \sigma_X n^{-1/3}$, where σ_X is the (empirical) standard deviation of $R[X]$ and $n = |R|$. In fact, any binning strategy based on σ_X and n is supported. We define the *bin assignment function* $f : \mathbb{R} \to \mathbb{Z}$ that maps each $t[X]$ to its corresponding bin as

$$f : x \mapsto \lfloor (x - \min\{t[X]\}_{t \in R}) / w_x \rfloor$$

Then, the final histogram is an aggregation query that counts the number of tuples within each bin, which can be expressed in relational algebra as $\gamma_{f(X), COUNT}(R)$, or simplified as $\gamma_{f(X)}(R)$. $\Pr(X)$ is then approximated by normalizing the bin counts by n. We aim to answer the queries $\gamma_{f(X)}(R_{\bowtie})$ for $X \in \{\widetilde{T}, \widetilde{Z}, \widetilde{\epsilon_T}, \widetilde{\epsilon_Z}\}$. Notably, σ_X and n in $R_{\bowtie}$ come free with the variance semi-ring, and $\min\{t[X]\}_{t \in R}$ can be computed using a tropical semi-ring [15]. For simplicity, we denote $\min\{t[X] \mid t \in R\}$ as α_X and treat these statistics as constants as they are not the focus of this section.

7.1.2 Histogram Semi-ring. We define the histogram semi-ring $(D, \oplus, \otimes, 0, 1)$, where D is the set of ordered pairs in $\mathbb{Z} \times \mathbb{Z}$[3]. To ease expression, we use a key-value pair (k, v) to denote an element of D with $(k, 0) \in D$ if the key k does not exist in D. For two semi-ring annotations $D_1, D_2 \in D$, we define the semi-ring addition $\oplus$ and semi-ring multiplication $\otimes$ as

Definition 7.1 (Semi-ring Addition $\oplus$). *The semi-ring addition $\oplus$ of $D_\oplus = D_1 \oplus D_2$ is defined as*

$$D_\oplus = \{(k, v_1 + v_2) \mid \forall (k, v_1) \in D_1, (k, v_2) \in D_2\}$$

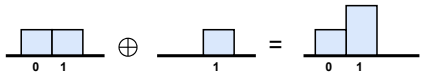
The additive identity (0) is defined as $\{(0, 0)\}$.

Definition 7.2 (Semi-ring Multiplication $\otimes$). *The semi-ring multiplication $\otimes$ of $D_\otimes = D_1 \otimes D_2$ can be defined as*

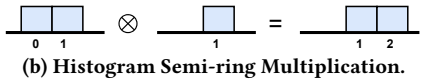
$$\{(k, \sum_{k_1 + k_2 = k} v_1 \cdot v_2) \mid (k_1, v_1) \in D_1, (k_2, v_2) \in D_2\}$$

The multiplicative identity (1) is defined as $\{(0, 1)\}$.

[3] In fact, D is a multiset over $\mathbb{Z}$.



(a) Histogram Semi-ring Addition.



(b) Histogram Semi-ring Multiplication.

Figure 5: Semantics for addition in Figure 5a and multiplication in Figure 5b over histogram semi-ring.

Proposition 7.1. *The semi-ring add ($\oplus$) and multiplication ($\otimes$) satisfy the commutative and distributive law.*

Intuitively, the histogram semi-ring represents local histograms within each join key. We illustrate the semi-ring operators through an example in Figure 5.

Example 5. *The $\oplus$ operator in Figure 5a corresponds to the bin-wise addition of histograms. The $\otimes$ operator computes new histogram bins through a multiplication-addition mechanism: consider the two histograms in Figure 5b, combining bin 0 of the left histogram with bin 1 of the right histogram creates a new bin whose key is the sum of the participating bins' keys (0 + 1), and value equal to the product of the participating bins' values ($1 \cdot 1$).*

Next, we use the histogram semi-ring to estimate $\Pr(\widetilde{T})$ and $\Pr(\widetilde{\epsilon_Z})$. The same approach applies to $\Pr(\widetilde{Z})$ and $\Pr(\widetilde{\epsilon_T})$.

7.1.3 Estimating pdf for $\widetilde{T}$. To compute $\gamma_{f(\widetilde{T})}(R_{\bowtie})$ without materializing $R_{\bowtie}$, we distribute the aggregation through joins using two steps: (1) compute per-join-key histograms of $\widetilde{T}$ in R, and (2) obtain per-join-key counts in R_i via $\gamma_{\mathcal{J}}(R_i)$. Leveraging the histogram semi-ring, we define the lift function $g_{\widetilde{T}}(\cdot) : \mathrm{dom}(R) \to D$ as

$$g_{\widetilde{T}}(t) = \begin{cases} (f(t[\widetilde{T}]), 1) & t \in R \\ (0, 1) & t \in R_i \end{cases}$$

Here, $g_{\widetilde{T}}(\cdot)$ maps tuples in R to single-bin histograms $(f(\widetilde{T}), 1)$ and tuples in R_i to the semi-ring multiplicative identity. We illustrate the whole procedure using a running example.

Example 6. *Consider relations R and R_1 in Figure 6a, with each tuple annotated by the histogram semi-ring for $g_{\widetilde{T}}(t)$; for illustration purpose, $T = \widetilde{T}, Z = \widetilde{Z}$. To avoid materializing $R \bowtie R_1$ in Figure 6b, we first compute $\gamma_{\mathcal{J}} R$ and $\gamma_{\mathcal{J}} R_1$ by aggregating semi-ring annotations within each join key. For join key a_1, the summation $g_{\widetilde{T}}(t)$ is $(0, 1) \oplus (1, 1) \oplus (1, 1) = (0, 2), (3, 1)$ as shown in Figure 6c. Next, we combine sketches using the semi-ring $\otimes$ operator – the product of the two semi-ring annotations across a_1 is $\{(0, 1), (1, 2)\} \otimes \{(0, 3)\} = \{(0 + 0, 1 \cdot 3), (0 + 1, 2 \cdot 3)\} = \{(0, 3), (1, 6)\}$. The final histogram $\gamma(\gamma_{\mathcal{J}}(R) \otimes \gamma_{\mathcal{J}}(R_1))$ (Figure 6d) equals the ground truth in Figure 6b.*

7.1.4 Estimating pdf for $\widetilde{\epsilon_Z}$. $f(\widetilde{T})$ is completely dependent on R, so it easily distribute through $R_{\bowtie}$. The same trick does not apply to calculating $f(\widetilde{\epsilon_Z})$, because it is unclear how R and R_1 contribute to $\widetilde{\epsilon_Z}$. To factorize $\gamma_{f(\widetilde{\epsilon_Z})}(R_{\bowtie})$, we first decompose $f(\widetilde{\epsilon_Z})$ by leveraging its linearity, and the fact that $|\lfloor a + b \rfloor - (\lfloor a \rfloor + \lfloor b \rfloor)| \in \{0, 1\}$.

R

$\mathcal{J}$	T	$g_{\widetilde{T}}(t)$	$g_{\widetilde{\epsilon_Z}}(t)$
a_1	-0.65	{(0, 1)}	{(1, 1)}
a_1	0	{(1, 1)}	{(0, 1)}
a_1	1.3	{(1, 1)}	{(0, 1)}
a_2	-1.94	{(0, 1)}	{(1, 1)}

R_1

$\mathcal{J}$	Z	$g_{\widetilde{T}}(t)$	$g_{\widetilde{\epsilon_Z}}(t)$
a_1	-0.49	{(0, 1)}	{(0, 1)}
a_1	0	{(0, 1)}	{(1, 1)}
a_1	0.81	{(0, 1)}	{(1, 1)}
a_2	-0.97	{(0, 1)}	{(0, 1)}

(a) Relations R and R_1 annotated with *lift functions* $g_{\widetilde{T}}(t)$ and $g_{\widetilde{\epsilon_Z}}(t)$ for $\gamma_{f(\widetilde{T})}(R_{\bowtie})$ and $\gamma_{f(\widetilde{\epsilon_Z})}(R_{\bowtie})$, respectively.

$R_{\bowtie}$

$\mathcal{J}$	T	Z	$\widetilde{\epsilon_Z}$	$f(\widetilde{T})$	$f(\widetilde{\epsilon_Z})$
a_1	-0.65	-0.49	-0.17	0	0
a_1	-0.65	0	0.33	0	1
a_1	-0.65	0.81	1.14	0	2
a_1	0	-0.49	-0.49	1	0
a_1	0	0	0	1	1
a_1	0	0.81	0.81	1	1
a_1	1.3	-0.49	-1.14	1	0
a_1	1.3	0	-0.65	1	0
a_1	1.3	0.81	0.16	1	1
a_2	-1.94	-0.97	0	0	1

$\gamma_{f(\widetilde{T})}(R_{\bowtie})$

$f(\widetilde{T})$	CNT
0	4
1	6

$\gamma_{f(\widetilde{\epsilon_Z})}(R_{\bowtie})$

$f(\widetilde{\epsilon_Z})$	CNT
0	4
1	5
2	1

(b) Naive join $R_{\bowtie} = R \bowtie_{\mathcal{J}} R_1$ including $\widetilde{\epsilon_Z}$, $f(\widetilde{T})$ and $f(\widetilde{\epsilon_Z})$, along with the query results.

$\gamma_{f(\widetilde{T})}(R_{\bowtie})$:

	$\gamma_{\mathcal{J}}(R)$		$\gamma_{\mathcal{J}}(R_1)$		$\gamma_{\mathcal{J}}(R) \otimes \gamma_{\mathcal{J}}(R_1)$
$\mathcal{J}$	Annotation	$\mathcal{J}$	Annotation	$\mathcal{J}$	Annotation
a_1	{(0, 1), (1, 2)}	a_1	{(0, 3)}	a_1	{(0, 3), (1, 6)}
a_2	{(0, 1)}	a_2	{(0, 1)}	a_2	{(0, 1)}

$\gamma_{f(\widetilde{\epsilon_Z})}(R_{\bowtie})$:

$\mathcal{J}$	Annotation	$\mathcal{J}$	Annotation	$\mathcal{J}$	Annotation
a_1	{(0, 2), (1, 1)}	a_1	{(0, 1), (1, 2)}	a_1	{(0, 2), (1, 5), (2, 2)}
a_2	{(1, 1)}	a_2	{(0, 1)}	a_2	{(1, 1)}

(c) Aggregation of semi-ring annotation as sketches and combining semi-rings between two sketches.

Est. $\gamma_{f(\widetilde{T})}(R_{\bowtie})$

$\mathcal{I}$	CNT
0	4
1	6

Est. $\gamma_{f(\widetilde{\epsilon_Z})}(R_{\bowtie})$

$\mathcal{I}$	CNT
0	2
1	6
2	2

(d) Exact estimation of $\gamma_{f(\widetilde{T})}(R_{\bowtie})$ and an approximation of $\gamma_{f(\widetilde{\epsilon_Z})}(R_{\bowtie})$.

Figure 6: Example of factorized histogram method.

For each $t \in R_{\bowtie}$, we rewrite $f(t[\widetilde{\epsilon_Z}])$ as

$$\lfloor \frac{t[\widetilde{Z}] - \beta \cdot t[\widetilde{T}] - \alpha_{\widetilde{\epsilon_Z}}}{w_{\widetilde{\epsilon_Z}}} \rfloor \approx \lfloor \frac{t[\widetilde{Z}] - \alpha_{\widetilde{\epsilon_Z}}}{w_{\widetilde{\epsilon_Z}}} \rfloor - \lfloor \frac{\beta \cdot t[\widetilde{T}]}{w_{\widetilde{\epsilon_Z}}} \rfloor$$

This decompose $f((t[\widetilde{Z}] - \beta \cdot t[\widetilde{T}])/\sigma_{\widetilde{\epsilon_Z}})$ into two parts where the first term (highlighted in red) only involves $t[\widetilde{Z}]$ in R_1 and the second term (highlighted in blue) only involves $t[\widetilde{T}]$ in R, such that they can be computed locally. We define the lift function $g_{\widetilde{\epsilon_Z}}(\cdot)$ as

$$g_{\widetilde{\epsilon_Z}}(t) = \begin{cases} \left(-\lfloor \frac{\beta \cdot t[\widetilde{T}]}{w_{\widetilde{\epsilon_Z}}} \rfloor,\ 1\right) & t \in R \\ \left(\lfloor \frac{(t[\widetilde{Z}] - \alpha_{\widetilde{\epsilon_Z}})}{w_{\widetilde{\epsilon_Z}}} \rfloor,\ 1\right) & t \in R_i \end{cases}$$

Here, $g_{\widetilde{\epsilon_Z}}(\cdot)$ maps tuples in R and R_i to their contributing factors in $f(\widetilde{\epsilon_Z})$. We will use the same running example in Figure 6 to illustrate this procedure.

Example 7. *Consider again relations R and R_1 in Figure 6a, we first obtain $\beta = 0.5$ using the variance semi-ring, and lift each tuple in R and R_1 with the histogram semi-ring through $g_{\widetilde{\epsilon_Z}}(t)$. Similar to Example 6, we aggregate the semi-ring annotations locally in R and R_1 and merge them using the semi-ring $\otimes$ operator as shown in the lower half of Figure 6c, the estimated histogram shown in Figure 6d approximates the ground truth in Figure 6b.*

7.1.5 Implementation. With Suna's algorithmic design, confounder discovery reduces to independent pair-wise operations between the treatment T and each variable in $\mathbf{A}$. Our semi-ring annotations framework enables rewriting these operations as matrix computations, making GPUs an ideal acceleration platform. Following the approach in [29], we cluster semi-ring annotations by join key domains, allowing batch processing of variables in $\mathbf{A}$. This batched execution maximizes GPU thread utilization and delivers substantial performance improvements.

7.2 Convergence Analysis

As noted in previous sections, we use histograms to estimate differential entropy and mutual information. In Lemma 7.3 and Theorem 7.2, we demonstrate that the histogram-based approach yields unbiased estimates for both differential entropy and conditional mutual information, which naturally extend to mutual information.

The differential entropy (which can be negative) applies to continuous random variables, while entropy (always non-negative) applies only to discrete random variables. Both definitions can be unified via measure theory; so we only need to focus on proving results for entropy convergence. In particular, probability mass functions (pmf) can be translated to probability density functions (pdf) by using the counting measure (instead of the Lebesgue measure). We show that the histogram semiring-based entropy estimation, over mixed random vectors, converges to the true entropy:

Lemma 7.3. *Given a random vector (X, Y, Z) that contains discrete-continuous mixture random variables, with semiring histogram bins $B = \underline{B} \cup \overline{B}$ and the resulting discretized random vector (X', Y', Z'), where the bins in $\overline{B}$ contain discrete data points (of which every dimension has a discrete value) and $\underline{B} = B \setminus \overline{B}$, we have*

$$\lim_{b \to 0} H(X', Y', Z') = H(X, Y, Z)\ ,$$

where $b = \max_{B_j \in \underline{B}}(\mu(B_j))$, and μ is the product measure.

Furthermore, the histogram-based mutual information estimation also converges to the true mutual information:

Theorem 7.2. *Given a mixed random vector (X, Y, Z),*

$$\lim_{b \to 0} \lim_{n \to \infty} \mathrm{MI}^h(X; Y|Z) = \mathrm{MI}(X; Y|Z)$$

almost surely, where n refers to the sample size and b refers to the maximum volume of the semiring histogram volumes for bins in $\underline{B}$. $\mathrm{MI}^h(X; Y|Z)$ is the mutual information computed via histograms, and $\mathrm{MI}(X; Y|Z)$ is the true mutual information.

Detailed discussion and proof of Lemma 7.3 and Theorem 7.2 can be found in Appendix B. Note that Theorem 7.2 that applies to conditional mutual information also applies to the mutual information, since the latter is a special case of the former. Furthermore,

it is straightforward to see that the results apply beyond mixed random vectors of size three.

8 EVALUATION

We now present evaluations of the quality and efficacy of Suna. We first evaluate Suna by constructing a large real-world data corpus consisting of datasets from NYC Open Data [6] and Kaggle [24] to determine if Suna can discover semantically meaningful confounders. Then, we generate synthetic datasets to quantitatively assess the quality of the adjustment sets discovered by Suna. Finally, we examine the effectiveness of our optimization proposed in Section 7.

8.1 Real-world Experiments

We use real-world datasets to evaluate Suna's ability to reason against spurious correlations. Our evaluation addresses two key questions: **Q1**: Does Suna discover semantically meaningful confounders? and **Q2**: Can Suna deliver causal query results with interactive latency? We cite existing studies in the social science literature to reason about semantic information. This is consistent the methodologies in prior work [33, 48] for evaluating the quality of causal queries over real-world datasets.

8.1.1 Data and Workload. We construct a large corpus of 346 datasets from NYC Open Data [6] and Kaggle datasets [24]. We construct 4 causal queries using the following datasets:

- **SO** [41] contains the results of annual surveys over 2023 and 2024 conducted by Stack Overflow that collect insights from developers worldwide. We convert the values in the *education level* column to numerical values that represent approximate years of formal schooling.
- **ELA** [4] integrates 2006-11 English Language Arts (ELA) test results with annual school progress reports [1] for each District Borough Number (DBN).
- **Ratio** [3] contains 2013-18 ELA test results over DBNs integrated with the annual pupil-to-teacher ratio report [5] for each DBN.
- **SAT** [2] contains 2012 college-bound seniors' mean SAT scores over DBNs.

8.1.2 Baselines. We compare Suna with the following approaches: (1) MESA [48] discovers attributes that explain confounding bias between two variables with no causal relationship, (2) HypDB [33] is a confounder detection system that discovers confounders for arbitrary treatment and outcome variables, and (3) Correlation identifies the attribute most strongly correlated with the treatment variable. Both MESA and HypDB require a single table as input, so they require a preprocessing step that materializes the join across all join-compatible datasets. Moreover, HypDB only accepts categorical attributes, necessitating a binning procedure for numerical attributes. However, it is optimized for binary features; with more than two bins per numerical attribute, it fails to terminate within 10 hours on each query. Consequently, we convert all attributes into binary features to run HypDB efficiently[4].

[4]Although this binning transformation discards some information, it preserves the underlying causal diagram.

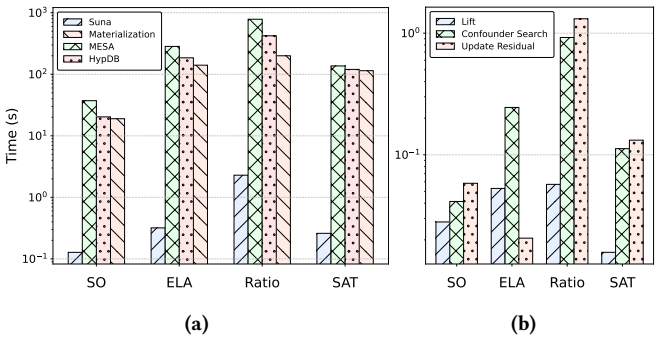


Figure 7: (a) Runtime performance of Suna vs other baselines for real-world causal queries. The x-axis includes all baselines for each query. The y-axis shows runtime. Suna is ≥2 orders of magnitude faster than baselines for causal queries. (b) Runtime breakdown of Suna's algorithmic components for each query.

8.1.3 Quality analysis. In this section, we aim to answer **Q1**. We report the confounders discovered by different methods for each query in Table 1. Next, we analyze the quality of the discovered confounders by referencing existing studies. We highlight confounders complying with existing studies in blue in Table 1. Suna discovers semantically meaningful confounders for each causal query.

SO. We examine potential confounders for causal analysis between years of formal schooling (treatment) and yearly salary (outcome). An economic study [30] shows that social factors—such as higher living costs in urban areas—can drive both greater educational attainment and increased salary. Both Suna and Correlation identify relevant confounders consistent with these findings.

ELA. We next investigate confounders for causal analysis between the additional score a school receives in its progress report (treatment), which represents extra points awarded beyond the core performance metrics, and the school's average ELA score (outcome). Research suggests socioeconomic status is a key confounder. For this query, only Suna discovers *% poverty*.

Ratio. Then, we explore potential confounders for causal analysis between the pupil-to-teacher ratio (treatment) on the school's average ELA score (outcome). Prior work indicates that students' socio-economic status impacts both class sizes and performance. In this query, Suna and Correlation identify *% Students with Disabilities*, while HypDB detects *Percent HRA Eligible* (a proxy for the socioeconomic needs of the school community), all in line with existing literature.

SAT. Finally, we examine confounders when studying how the number of SAT test takes (treatment) affect the average math score (outcome) for each high school. Previous research [28, 32] notes that school size and environment can shape test participation and performance. All methods (Suna, MESA, HypDB, and Correlation) identify plausible confounders.

8.1.4 Runtime analysis. We now evaluate the scalability of Suna. Figure 7a shows the cumulative runtime for Suna, MESA, and HypDB, along with the time required to materialize joins for each causal query. Importantly, Suna is ≥2 orders of magnitude faster than all baselines for every causal query. In fact, Suna's completion

Dataset	Query	Suna	MESA	HypDB	Correlation
SO	What is the effect of education level on salary?	Cost of Living & Rent Index	Unemployment Rate	Country	Rent Index
ELA	What is the effect of each school's extra credit performance score on students' ELA score?	Enrollment % Poverty	% of grads Performance Category Score Total Student Response Rate	School Overall Score ID Pct Level 3 and 4	School Overall Score
Ratio	What is the effect of each school's pupil-to-teacher ratio on student's ELA score?	Level 4: % % Students with Disabilities Minimum Class Size	Performance Category Score Math Proficiency ELA % Proficient	Percent HRA Eligible Average Class Size Pct Level 2	% Students with Disabilities
SAT	What is the effect of test takers numbers on SAT score?	# Safety Incidents Enrollment Total Regents #	Safety and Respect Score Graduation Rate # Ontrack at Year 1	Total Regents #	Total Regents #

Table 1: Report of discovered confounders for causal queries.

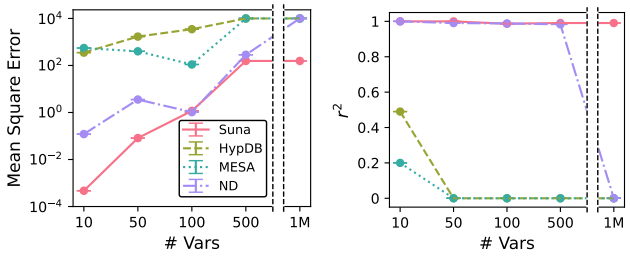
times of the queries are orders of magnitude faster than MESA and HypDB's join completion time already alone. This demonstrates the efficiency of Suna's optimizations. Figure 7b decomposes Suna's end-to-end runtime into its algorithmic components, including transforming the input dataset into semi-ring sketches (lift), iteratively searching for confounders with sketches (confounder search), and incrementally maintaining the sketches across iterations (update residual). For all queries, none of these components is a significant runtime bottleneck, as they all finish in $\leq$ one second, achieving interactive latency.

8.1.5 Takeaways: Suna discovers semantically meaningful confounders as validated by existing studies. With optimizations discussed in Section 7, Suna runs orders of magnitude (>100 times) faster than existing confounder discovery systems.

8.2 Synthetic Experiments

We further assess Suna's accuracy on synthetic datasets to address three key questions. **Q1**: Does Suna discover an admissible adjustment set that accurately estimates the ATE? **Q2**: How does the total number of variables in the causal diagram affect Suna's estimate of the average treatment effect, both with and without irrelevant variables as noise? **Q3**: How well do our approximations in Section 7.1 estimate $\Pr(\epsilon_Z)$ and $\Pr(\epsilon_T)$, and how this approximation implies to the estimation of $H(\epsilon_Z)$ and $H(\epsilon_T)$.

8.2.1 Setup. We construct a causal diagram using synthetic datasets following the LiNGAM model. The coefficients of linear mechanisms are drawn randomly within the range of $[1, 2]$; additive noises are sampled from a Uniform distribution $U(0, 1)$. To avoid overflow, we normalize the coefficients that determine each variable by dividing them by the number of parents of that variable and rounding the data to two digits. To construct causal queries, we generate a random pair of treatment and outcome $\mathcal{Q} = (T, O)$ where T is a non-descendant of O (i.e., $T \notin \text{ND}(O)$). We generate a relation $\mathcal{R}$ with n tuples and m variables. Then, we partition $\mathcal{R}$ vertically where the input dataset R contains $[\mathcal{J}, T, O]$, and the remaining columns are partitioned into $m-2$ relations, each contains a single attribute. To support joins, we include a join key $\mathcal{J}$, which serves as the primary key of $\mathcal{R}$, and distribute it over each of the $m-2$ relations, $\mathcal{R}$ can be recovered by joining all relations through one-to-one joins over $\mathcal{J}$.

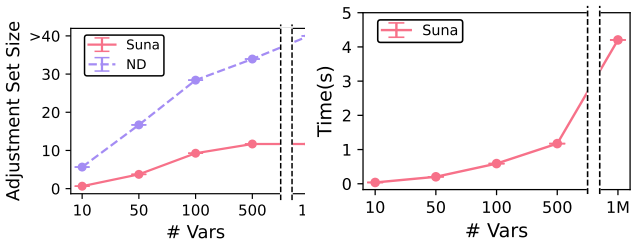


(a) **(b)**

Figure 8: MSE and r^2 (normalizing for variations in the ground truth ATE) of the estimated ATE by Suna and different baselines evaluated on over $m \in \{10, 50, 100, 500, 1M\}$. Suna consistently achieves high accuracy across different number of variables, while ND does not scale to large $|\mathbf{A}|$.

8.2.2 Accuracy Evaluation. We study **Q1** and **Q2** by randomly generating 100 causal queries with $m \in \{10, 50, 100, 500, 1M\}$. To answer **Q1**, we use the discovered set of confounders $\mathbf{Z}$ to train a linear regression model $O \sim \beta\mathbf{Z} + \alpha T$, and compare α against the ground-truth ATE in the pre-defined causal models. To evaluate **Q2**—Suna's resilience to irrelevant variables—we set $m = 1M$ to construct a causal diagram over 500 variables and fill the remaining ones with random noises. For baselines, we compare Suna with MESA, HypDB, and the simple extension we proposed in Section 6.1, referred to as ND. Since ND does not scale to moderately large m (the topological sort and early stop do not terminate over an hour for $m = 500$ for each causal query), we evaluate ND by using the set of all non-descendants of T obtained from the ground truth causal diagram as the adjustment set. This approach assumes ND will always estimate ND(T) accurately, an assumption that may not always hold.

Figure 8a reports the mean square error (MSE) between the estimated ATE versus the ground truth. To account for the difference in magnitude of the ground truth causal effect, we report the r^2 score in Figure 8b. Suna and ND has similar performance – lowest MSE and highest r^2 score of > 0.99 for each m. On the other hand, MESA and HypDB both fail to discover admissible adjustment sets with r^2 score degrades for larger m. This is because MESA assumes no causal effect between the treatment and outcome, which is easily violated for larger m. When this assumption is violated, MESA may mistakenly select mediators as confounders that leads to wrong



(a) (b)

Figure 9: Adjustment set size |Z| discovered by Suna and size of ND(T) in $\mathcal{G}$, along with Suna's runtime to iteratively construct Z. The adjustment set discovered by Suna is >3x smaller than |ND(T)|.

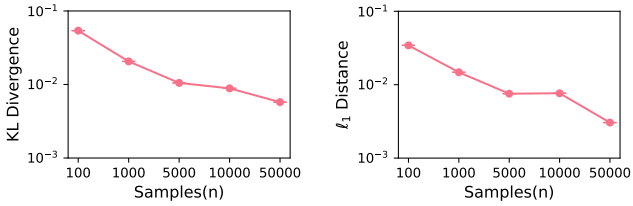
adjustment sets. On the other hand, HypDB 's reliance on binning leads to significant information loss, thus hindering its ability to find valid confounders. Interestingly, although ND guarantees to find an admissible adjustment set in theory under infinite sample sizes, it estimates the average treatment effect poorly for $m = 1M$. This is due to estimation error caused by large adjustment sets.

8.2.3 Scalability. We further study **Q1** and **Q2** by examining how the size of the adjustment set and the runtime varies as the number of variables increases. The settings remain the same as Section 8.2.2. Figure 9a shows that Suna discovers a smaller (less than one third of the size than ND), yet still admissible confounder set across all numbers of variables. This makes Suna a more interpretable system for analysts to use, especially when they seek to analyze causal relations in complex scenarios when the number of variables is large. Figure 9b shows that the mean runtime over 100 queries grows slightly as the number of variables increases, due to the preprocessing step to filter out irrelevant variables. By leveraging factorized learning techniques and GPU acceleration (following SET [29]), we manage to keep the overhead minimal (~3s for one million variables), even at a scale where most baselines would time out or take more than days to complete.

8.2.4 Histogram Semi-ring Approximation Evaluation. We investigate **Q3** by examining the divergence between the approximate histogram and the exact histogram (obtained via join materialization), as well as the resulting impact on entropy estimation.

We use the same setup from Section 8.2.1 for data generation. For robustness, we take 10 attributes $\{Z_1, \ldots, Z_{10}\}$ and conduct BCD for each pair of (T, Z_i). Due to symmetry in estimating $\Pr(\widetilde{\epsilon_Z})$ and $\Pr(\widetilde{\epsilon_T})$, we focus our study on the approximation of $\Pr(\widetilde{\epsilon_Z})$ and how such approximation affects $H(\widetilde{\epsilon_Z})$ - a core estimate in BCD-using methods described in Section 7.1. We measure the approximation accuracy of $\Pr(\widetilde{\epsilon_Z})$ using KL divergence from the exact histogram computed by materializing joins, and then quantify how this affects entropy estimates by computing the ℓ_1 distance between the approximated entropy and the ground truth entropy of each $\widetilde{\epsilon_{Z_i}}$.The sample size n is varied from $\{100, 1000, 5000, 10000, 50000\}$.

Figure 10a reports that the mean KL-divergence between the approximated and exact histograms converges to 0 as the number of samples increases, all (T, Z_i) pairs. Convergence is also observed in Figure 10b, which shows that as the number of samples increases, the ℓ_1 distance between the approximate and exact histograms



(a) The KL Divergence of the approximated histogram with respect to the ground truth histogram converges to 0 as the sample size n increases.

(b) The ℓ_1 distance of the entropy estimated using approximate histograms versus exact histograms converges to 0 as the sample size n increases.

Figure 10: Comparison of KL Divergence and entropy estimates as sample size n increases.

approaches 0. In fact, both the KL divergence and ℓ_1 distance fall below a negligible 0.01 at a small sample size ($n = 5000$).

8.2.5 Takeaways. Suna prunes away irrelevant variables and identifies high-quality confounders to accurately answer causal queries in just a few seconds, scaling to 1M variables. Further, empirical evaluations show that the histogram estimation technique reliably converges as the number of samples increases. Combined with the theoretical analysis in Section 7.2, these results validate our approach in Section 7.

9 CONCLUSIONS

Suna is a confounder discovery system that searches large corpora to find datasets to improve causal inference task estimation. We prove the connection between bivariate causal discovery, a key building block in parametric causal discovery that exploits the asymmetry between causes and effects, and confounder discovery. This theorem lets Suna directly search for causal confounders without the need for a full causal DAG. To improve scalability, Suna designs a novel data structure inspired by factorized learning to avoid materializing joins and implements a GPU-compatible system to further scale up performance. Suna achieves an r^2 score of >0.99 over a causal diagram of 500 variables where all existing confounder discovery systems fail, and scales to repositories with $1M$ variables while answering queries within a few seconds.

Discussion. Our main contribution is an algorithm that efficiently discovers confounders in an unobserved causal DAG from a relational data repository—all without materializing joins. In our current implementation, we leverage LiNGAM to analyze linear relationships among numerical attributes under the *causal sufficiency assumption*. Our approach lays a solid foundation for extending towards more general and complex causal settings. In particular, moving beyond linear models and incorporating latent confounders only requires the capability to perform bivariate causal discovery on the projected causal diagram that includes a variable pair (T, Z) from the current adjustment set.

Future work could extend Suna in several promising directions. One avenue is to explore more complex parametric causal mechanisms (e.g. non-linear) or design a hybrid approach capable to handle arbitrary data types. Additionally, relaxing the *causal sufficiency assumption*, improving runtime and estimation accuracy as users issue causal queries over time, and leveraging Suna's scalability for query-relevant data integration are important next steps.

REFERENCES

[1] [n. d.]. 2007-2008-School-Progress-Reports-All-Schools. https://data.cityofnewyork.us/Education/2007-2008-School-Progress-Reports-All-Schools/dj4e-3xrn.

[2] [n. d.]. 2012 SAT Results. https://data.cityofnewyork.us/Education/2012-SAT-Results/f9bf-2cp4.

[3] [n. d.]. 2013 - 2018 School ELA REsults. https://data.cityofnewyork.us/Education/2013-2018-School-ELA-REsults/qkpp-pbi8.

[4] [n. d.]. 2014-15 To 2016-17 School- Level NYC Regents Report For All Variables. https://data.cityofnewyork.us/Education/2014-15-To-2016-17-School-Level-NYC-Regents-Report/csps-2ne9/.

[5] [n. d.]. 2018 - 2019 Class Size Pupil to Teacher Ratio. https://data.cityofnewyork.us/Education/2018-2019-Class-Size-Pupil-to-Teacher-Ratio/axb2-9jkb.

[6] 2022. NYC Open Data. https://opendata.cityofnewyork.us/.

[7] András Antos and Ioannis Kontoyiannis. 2001. Convergence properties of functional estimates for discrete distributions. *Random Structures & Algorithms* 19 (2001). https://api.semanticscholar.org/CorpusID:55801502

[8] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David Karger. 2020. ARDA: automatic relational data augmentation for machine learning. *arXiv preprint arXiv:2003.09758* (2020).

[9] François Chollet. 2019. On the measure of intelligence. *arXiv preprint arXiv:1911.01547* (2019).

[10] Rama Cont. 2001. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative finance* 1, 2 (2001), 223.

[11] T.M. Cover and J.A. Thomas. 2012. *Elements of Information Theory*. Wiley. https://books.google.com/books?id=VWq5GG6ycxMC

[12] G. Darmois. 1953. Analyse générale des liaisons stochastiques: etude particulière de l'analyse factorielle linéaire. *Revue de l'Institut International de Statistique / Review of the International Statistical Institute* 21, 1/2 (1953), 2–8. http://www.jstor.org/stable/1401511

[13] Sainyam Galhotra, Yue Gong, and Raul Castro Fernandez. 2023. METAM: Goal-Oriented Data Discovery. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2780–2793.

[14] Yue Gong, Sainyam Galhotra, and Raul Castro Fernandez. 2024. Nexus: Correlation Discovery over Collections of Spatio-Temporal Tabular Data. *Proceedings of the ACM on Management of Data* 2, 3 (2024), 1–28.

[15] Todd J Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance semirings. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. 31–40.

[16] Sander Greenland, Judea Pearl, and James M Robins. 1999. Confounding and collapsibility in causal inference. *Statistical science* 14, 1 (1999), 29–46.

[17] Tobias Hatt and Stefan Feuerriegel. 2024. Sequential deconfounding for causal inference with unobserved confounders. In *Causal Learning and Reasoning*. PMLR, 934–956.

[18] Patrik Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. 2008. Nonlinear causal discovery with additive noise models. *Advances in neural information processing systems* 21 (2008).

[19] Zezhou Huang, Jiaxiang Liu, Daniel Alabi, Raul Castro Fernandez, and Eugene Wu. 2023. Saibot: A Differentially Private Data Search Platform. *arXiv preprint arXiv:2307.00432* (2023).

[20] Zezhou Huang, Jiaxiang Liu, Haonan Wang, and Eugene Wu. 2023. The Fast and the Private: Task-based Dataset Search. *arXiv preprint arXiv:2308.05637* (2023).

[21] Paul Hünermund, Jermain Kaminski, and Carla Schmitt. 2022. Causal machine learning and business decision making. *Available at SSRN 3867326* (2022).

[22] Aapo Hyvärinen and Stephen M Smith. 2013. Pairwise likelihood ratios for estimation of non-Gaussian structural equation models. *The Journal of Machine Learning Research* 14, 1 (2013), 111–152.

[23] Amin Jaber, Adele Ribeiro, Jiji Zhang, and Elias Bareinboim. 2022. Causal identification under Markov equivalence: calculus, algorithm, and completeness. *Advances in Neural Information Processing Systems* 35 (2022), 3679–3690.

[24] Kaggle. 2024. Kaggle Datasets Repository. https://www.kaggle.com/datasets Accessed: 2024-10-15.

[25] Aamod Khatiwada, Roee Shraga, Wolfgang Gatterbauer, and Renée J Miller. 2022. Integrating data lake tables. *Proceedings of the VLDB Endowment* 16, 4 (2022), 932–945.

[26] Aamod Khatiwada, Roee Shraga, and Renée J Miller. 2023. DIALITE: discover, align and integrate open data tables. In *Companion of the 2023 International Conference on Management of Data*. 187–190.

[27] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics* 69, 6 (2004), 066138.

[28] Valerie E Lee and Julia B Smith. 1997. High school size: Which works best and for whom? *Educational evaluation and policy analysis* 19, 3 (1997), 205–227.

[29] Jiaxiang Liu, Zezhou Huang, and Eugene Wu. 2024. SET: Searching Effective Supervised Learning Augmentations in Large Tabular Data Repositories. In *Proceedings of the Conference on Governance, Understanding and Integration of Data for Effective and Responsible AI*. 26–31.

[30] Enrico Moretti. 2013. Real wage inequality. *American Economic Journal: Applied Economics* 5, 1 (2013), 65–103.

[31] Judea Pearl. 2009. *Causality*. Cambridge university press.

[32] Sean F Reardon, Demetra Kalogrides, and Kenneth Shores. 2019. The geography of racial/ethnic test score gaps. *Amer. J. Sociology* 124, 4 (2019), 1164–1221.

[33] Babak Salimi, Johannes Gehrke, and Dan Suciu. 2018. Bias in OLAP queries: Detection, explanation, and removal. In *Proceedings of the 2018 International Conference on Management of Data*. 1021–1035.

[34] Aécio Santos, Aline Bessa, Christopher Musco, and Juliana Freire. 2022. A sketch-based index for correlated dataset search. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 2928–2941.

[35] David W Scott. 1979. On optimal and data-based histograms. *Biometrika* 66, 3 (1979), 605–610.

[36] Rajen D Shah and Jonas Peters. 2020. The hardness of conditional independence testing and the generalised covariance measure. (2020).

[37] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. 2006. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research* 7, 10 (2006).

[38] Shohei Shimizu, Takanori Inazumi, Yasuhiro Sogawa, Aapo Hyvarinen, Yoshinobu Kawahara, Takashi Washio, Patrik O Hoyer, Kenneth Bollen, and Patrik Hoyer. 2011. DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research-JMLR* 12, Apr (2011), 1225–1248.

[39] Peter Spirtes. 2001. An anytime algorithm for causal inference. In *International Workshop on Artificial Intelligence and Statistics*. PMLR, 278–285.

[40] Peter Spirtes, Clark Glymour, and Richard Scheines. 2001. *Causation, prediction, and search*. MIT press.

[41] Stack Overflow. 2024. Stack Overflow Developer Survey. https://insights.stackoverflow.com/survey/ Accessed: 2024-10-11.

[42] Hui Yie Teh, Andreas W Kempa-Liehr, and Kevin I-Kai Wang. 2020. Sensor data quality: A systematic review. *Journal of Big Data* 7, 1 (2020), 11.

[43] Jin Tian and Judea Pearl. 2002. A general identification condition for causal effects. In *Aaai/iaai*. 567–573.

[44] Clifford H Wagner. 1982. Simpson's paradox in real life. *The American Statistician* 36, 1 (1982), 46–48.

[45] Yixin Wang and David M Blei. 2019. The blessings of multiple causes. *J. Amer. Statist. Assoc.* 114, 528 (2019), 1574–1596.

[46] Y Samuel Wang and Mathias Drton. 2023. Causal discovery with unobserved confounding and non-Gaussian data. *Journal of Machine Learning Research* 24, 271 (2023), 1–61.

[47] Brit Youngmann, Michael Cafarella, Amir Gilad, and Sudeepa Roy. 2024. Summarized Causal Explanations For Aggregate Views. *Proceedings of the ACM on Management of Data* 2, 1 (2024), 1–27.

[48] Brit Youngmann, Michael Cafarella, Yuval Moskovitch, and Babak Salimi. 2023. On Explaining Confounding Bias. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 1846–1859.

[49] Brit Youngmann, Michael Cafarella, Babak Salimi, and Anna Zeng. 2023. Causal Data Integration. *arXiv preprint arXiv:2305.08741* (2023).

A ON THE EXISTENCE OF CONFOUNDERS

Theorem A.1. *Fixing a causal diagram $\mathcal{G}$, a causal query $\mathcal{Q} = (T, O)$, and a set of variables $\mathbf{Z}$ where $\mathbf{Z} \cap De(T) = \emptyset$. If there exists an open non-causal path for $\mathcal{Q}$ conditioned on $\mathbf{Z}$ in $\mathcal{G}$, there must exist a variable $Z \in \mathbf{V}(\mathcal{G})$ satisfying (1) there is an open causal path from Z to T and all non-causal paths between Z and T are closed conditioned on $\mathbf{Z}$ in the projection of $\mathcal{G}$ onto $\{T, Z\} \cup \mathbf{Z}$; and (2) Z is a confounder for $\mathcal{Q}$ conditioned on $\mathbf{Z}$.*

Proof. We prove by contradiction. If $\mathbf{Z}$ is an admissible adjustment set, then the theorem holds automatically. Suppose there exists an open non-causal path p for $\mathcal{Q}$ conditioned on $\mathbf{Z}$. Then p must be an open back-door path between T and O conditioned on $\mathbf{Z}$ (p begins with $T \leftarrow$).

Lemma A.1. *Let p be an open back-door path between T and O conditioned on $\mathbf{Z}$. Then there exists a variable Z on path p such that:*

(1) The triplet (Z_l, Z, Z_r) in p forms a fork $Z_l \leftarrow Z \rightarrow Z_r$.
(2) The sub-path p_Z from Z to T in p is a directed path with $p_Z \cap \mathbf{Z} = \emptyset$.

Proof. Since p is a back-door path, we proceed by induction on the parent sets $Pa^i(Z) \cap p$ (e.g., $Pa^2(Z) = Pa(Pa(Z))$). For each $i \geq 1$, we must have $(Pa^i(Z) \cap p) \cap \mathbf{Z} = \emptyset$, as otherwise p would be closed. This induction terminates at some finite k where $Pa^k(Z) \cap p = \emptyset$. □

Given Z and p_Z in Lemma A.1, we denote $p_{\backslash Z} = (Z, p \backslash p_Z)$ as the complementary sub-path of p that starts from Z. Both p_Z and $p_{\backslash Z}$ must be open paths conditioned on $\mathbf{Z}$. If Z and T are unconfounded conditioned on $\mathbf{Z}$, then Z satisfies the two conditions; if Z and T are confounded, there must be an open back-door path p'_Z between T and Z conditioned on $\mathbf{Z}$. Then, we concatenate p'_Z with $p_{\backslash Z}$ to form a new path between T and O.

Lemma A.2. *Let p' be the path formed by concatenating p'_Z with $p_{\backslash Z}$. Then p' is an open back-door path conditioned on $\mathbf{Z}$.*

Proof. The result follows from two observations:

(1) Both p'_Z and $p_{\backslash Z}$ are open back-door paths conditioned on $\mathbf{Z}$
(2) Since $p_{\backslash Z}$ begins with $Z \rightarrow$, Z cannot be a collider in p'

Thus, p' preserves the open back-door property. □

Then, we may iteratively apply Lemma A.1 and Lemma A.2 until the Z variable from Lemma A.1 in the newly constructed back-door path satisfies the two conditions. A termination is guaranteed because we always find variables ranking higher in the topological order. As a result, we are guaranteed to arrive at a variable satisfying both conditions. □

B MULTI-DIMENSIONAL SEMIRING HISTOGRAMS FOR ESTIMATING CONDITIONAL MUTUAL INFORMATION

In the main body, we use the fact that we are able to estimate the conditional mutual information $\mathrm{MI}(X; Y|Z)$ via histograms. We prove that the mutual information estimated via histograms converges to the true mutual information:

Theorem B.1. *Given a mixed random vector (X, Y, Z),*

$$\lim_{b \to 0} \lim_{n \to \infty} \mathrm{MI}^h(X; Y|Z) = \mathrm{MI}(X; Y|Z)$$

almost surely, where n refers to the sample size and b refers to the maximum of the semiring histogram volumes for bins in B' (defined in Definition B.4 and Section B.6), $\mathrm{MI}^h(X; Y|Z)$ is the mutual information computed via histograms, and $\mathrm{MI}(X; Y|Z)$ is the true mutual information.

Although we show explicit proofs with respect to the conditional mutual information, the statements below also apply to mutual information. (This is because mutual information is a special case of its conditional mutual information.)

Also, for the proofs below, we make no distinction between differential entropy (applies to continuous random variables and can be negative) and entropy (applies only to discrete random variables and is non-negative). Both definitions can be unified via measure theory. In particular, probability mass functions (pmf) can be translated to probability density functions (pdf) with respect to the counting measure.

Definition B.1 (Differential Entropy). *For continuous random variable X with range $\mathcal{R}$ and pdf $p(x)$, the **differential entropy** is*

$$h(X) = -\int_{x \in \mathcal{R}} p(x) \log p(x) dx = -\int_{x \in \mathcal{R}} p(x) \log p(x) d\mu(x),$$

where μ is the Lebesgue measure on $\mathcal{R}$.

Definition B.2 (Entropy). *For discrete random variable X with range Ω and pmf $p(x)$, the **entropy** is*

$$H(X) = -\sum_{x \in \Omega} p(x) \log p(x) = -\int_{x \in \Omega} p(x) \log p(x) d\mu(x),$$

where μ is the counting measure on Ω.

B.1 Generalized One-Dimensional Entropy

We start with the one-dimensional case of the general problem: entropy estimation for variables that can take both discrete and continuous values. Let $p_X(x)$ represent the probability distribution of a random variable X, and for any subset S, define the probability over S as $p_X(S) = \sum_{x \in S} p_X(x)$.

Now, consider the one-dimensional entropy. For a random variable X, entropy H is expressed as:

$$H(X) = -\int_{x \in \mathbb{R}} p_X(x) \log p_X(x) \, d\mu(x), \tag{1}$$

where $\mu(\cdot)$ is a measure defined over one-dimensional Borel sets and $p_X(x) = \Pr(X = x)$. If $\mu(\cdot)$ is the Lebesgue measure, denoted as $u(\cdot)$, then $H(X)$ corresponds to differential entropy. On the other hand, if $\mu(\cdot)$ is a counting measure, $H(X)$ gives the standard discrete entropy. (Refer to Definitions B.1 and B.2 for more details.)

If X is a mixture of discrete and continuous components, the measure μ is constructed as follows. Partition $\mathbb{R}$ into three disjoint subsets, so that $\mathbb{R} = S' \cup T' \cup U'$. First, U' is the region where X has zero probability, i.e., $p_X(U') = 0$. Second, S' contains the discrete values, meaning that S' is countable and for all $x \in S'$, $p_X(x) > 0$. Third, T' comprises the continuous part, such that $p_X(T') + p_X(S') = 1$, and for any Borel set $A \subseteq T'$ where $u(A) = 0$,

we also have $p_X(A) = 0$. Based on these subsets S', T', and U', the measure μ is defined as:

$$\mu(A) = u(A \cap T') + |A \cap S'|, \tag{2}$$

where $|A \cap S'|$ denotes the cardinality of the intersection.

B.2 Generalizing to Multi-Dimensional Entropy

Next, we extend the entropy definition to a k-dimensional random vector $W = (W_1, \ldots, W_k)$. For each component W_i, we define the subsets S'_i, T'_i, and U'_i along with the measure μ^i as described before.

Definition B.3. *The* product measure *for the entire random vector W is then given by:*

$$\mu = \mu^1 \times \ldots \times \mu^k.$$

The entropy for W is defined as:

$$H(W) = -\int_{w \in \mathbb{R}^k} p_W(w) \log p_W(w)\, d\mu_W(w). \tag{3}$$

To ensure that this entropy definition in Equation 3 is valid, we establish in Lemma B.5 that $p_W(\cdot)$ exists.

B.3 One-Dimensional Histogram Models

Now, we consider how to define histograms over the mixture variables. A histogram model is typically defined based on a set of consecutive intervals called *bins*. However, to deal with discrete-continuous mixture random variables, we define the set of bins, denoted as B, such that each bin is either an interval or a set containing only a single point. That is, $B = \underline{B} \cup \overline{B}$, where $\underline{B}$ and $\overline{B}$ are sets of subsets of $\mathbb{R}$, with $\underline{B}$ consisting of countable consecutive intervals and $\overline{B}$ consisting of countable single point sets. Last, we define the "width" of a bin using the measure μ as defined in Eq. 2, i.e.„ for a bin $B_j \in B$ we have

$$\mu(B_j) = u(B_j \cap \underline{B}) + |B_j \cap \overline{B}|\ . \tag{4}$$

As any $B_j \in \overline{B}$ contains only a single discrete point, $\mu(B_j) = 1$ for all $B_j \in \overline{B}$.

Further, we define a histogram model M as a set of bins equipped with a parameter vector of length K, where $K = |B|$ is the number of bins:

Definition B.4 (Histogram Model M). *A **histogram model** M is a family of probability distributions parameterized by the vector $\theta = (\theta_1, \ldots, \theta_K)$. Each element of θ represents the density of each bin.*

We define $B = \underline{B} \cup \overline{B}$, where $\underline{B}$ and $\overline{B}$ are sets of subsets of $\mathbb{R}$, with $\underline{B}$ consisting of countable consecutive intervals and $\overline{B}$ consisting of countable single point sets. The number of bins is $K = |B|$.

Note that this definition generalizes to purely continuous random variables when $\overline{B} = \emptyset$ and also to discrete random variables if $\underline{B} = \emptyset$. For the latter case, the histogram model degenerates to a multinomial model.

B.4 Maximum Likelihood Estimator

Given a possibly multi-dimensional histogram with K bins, we denote the density function $p_{W,\theta}$ as f^h_θ and its maximum likelihood estimator as $f^h_{\hat{\theta}}$. Observe that for any parameter $\theta_j \in \theta$, the product $\theta_j \mu(B_j)$ follows a multinomial distribution. Thus, given a dataset $D = \{D_i\}_{i=1,\ldots,n}$, with D_i representing a row, the maximum log-likelihood is denoted as and equivalent to

$$l_M(D) = \log f^h_{\hat{\theta}(D)}(D) = \log \prod_{j=1}^{K} \left(\frac{c_j}{n \cdot \mu(B_j)} \right)^{c_j}, \tag{5}$$

where c_j and $\mu(B_j)$ are respectively the number of data points and the bin volumes of bin $j \in \{1 \ldots K\}$. Notice that this maximum likelihood generalizes to the purely discrete case (i.e.„ multinomial distribution) where all $\mu(B_j) = 1$, and to the purely continuous case where μ becomes the Lebesgue measure.

B.5 Conditional Mutual Information Estimator

Combining all previous theoretical discussions, we can now estimate conditional mutual information for three (possibly multivariate) random variables X, Y and Z by

$$\mathsf{MI}^h(X; Y|Z) = H^h(X,Z) + H^h(Y,Z) - H^h(X,Y,Z) - H^h(Z)\ .$$

The corresponding measure-theoretic entropies are estimated from k-dimensional data over (X, Y, Z), where k_X, k_Y and k_Z are the corresponding number of dimensions of X, Y and Z. We estimate the entropies as

$$\begin{aligned} H^h(X,Y,Z) &= -\int_{\mathbb{R}^k} f^h_{\hat{\theta}}(x,y,z) \log(f^h_{\hat{\theta}}(x,y,z))\, d\mu \\ H^h(X,Z) &= -\int_{\mathbb{R}^{k_X+k_Z}} f^h_{\hat{\theta}}(x,z) \log(f^h_{\hat{\theta}}(x,z))\, d\mu \\ H^h(Y,Z) &= -\int_{\mathbb{R}^{k_Y+k_Z}} f^h_{\hat{\theta}}(y,z) \log(f^h_{\hat{\theta}}(y,z))\, d\mu \\ H^h(Z) &= -\int_{\mathbb{R}^{k_Z}} f^h_{\hat{\theta}}(z) \log(f^h_{\hat{\theta}}(z))\, d\mu \end{aligned} \tag{6}$$

in which $f^h_{\hat{\theta}}(x,y,z)$ is the maximum likelihood estimator given the data, while we obtain $f^h_{\hat{\theta}}(x,z)$, $f^h_{\hat{\theta}}(y,z)$, and $f^h_{\hat{\theta}}(z)$ via marginalization from $f^h_{\hat{\theta}}(x,y,z)$. Next, we will prove that I^h is a strongly consistent estimator for conditional mutual information on mixed data.

B.6 Multi-Dimensional Histograms

First, we define the set of multi-dimensional bins. For a mixed k-dimensional random vector $W = (W_1, \ldots, W_k)$, we define the set of *bins* for each W_i as in Sec. B.3, denoted as B^i. Consequently, we can define a set of k-dimensional *bins*, denoted B, by the Cartesian product $B = B^1 \times \ldots \times B^k$.

Since each B^i is countable, B is also countable, and we can hence assume B is indexed by j. Then, we split B in a similar way as in the one-dimensional case, i.e.„ $B = \underline{B} \cup \overline{B}$, where $\overline{B}$ contains *only discrete values*. That is, for any k-dimensional bin $B_j \in \overline{B}$, each dimension of B_j is a set that contains a single one-dimensional point. Note that, however, for any $B_j \in \underline{B}$, each dimension of B_j can either be a one-dimensional interval or a one-dimensional single-point set. Further, we define the volume of a multi-dimensional bin $B_j \in B$ using the product measure $\mu(B_j)$ (see Sec. B.2).

Similar to one-dimensional histograms, a multi-dimensional histogram model M can be described by a probability distribution $p_{W,\theta}$ parametrized by the vector $\theta = (\theta_1, \ldots, \theta_K)$, where K is the number of bins and θ_i is the density for each bin.

B.7 Proofs

Lemma B.5. *Given a mixed k-dimensional random vector $W = (W_1, \ldots, W_k)$ with measure μ_W, $p_W(\cdot)$ always exists.*

Proof. Given a k-dimensional Borel set A, there exist one-dimensional Borel sets $A_1, \ldots, A_k$ such that $A = A_1 \times \ldots \times A_k$. If $\mu(A) = 0$, then there exists at least one $\mu^i, i \in \{1, \ldots, k\}$, such that $\mu^{\mathsf{MI}}(A_i) = 0$. Thus, $p_{W_i}(A_i) = 0 \Rightarrow p_W(\mathbb{R} \times \ldots \times \mathbb{R} \times A_i \times \mathbb{R} \times \ldots \times \mathbb{R}) = 0 \Rightarrow p_W(A) = 0$, as $A = A_1 \times \ldots \times A_k \subseteq \mathbb{R} \times \ldots \times \mathbb{R} \times A_i \times \mathbb{R} \times \ldots \times \mathbb{R}$. □

Last, based on Lemma B.5, we can prove that just like for a purely continuous or discrete random vector, conditional mutual information for a mixed random vector can be written as a sum of entropies.

Lemma B.6. *Given a mixed random vector (X, Y, Z) with joint probability p_{XYZ}, we can write $\mathsf{MI}(X; Y|Z) = H(X, Z) + H(Y, Z) - H(Z) - H(X, Y, Z)$, where each entropy can be defined as in Eq.* (3).

Proof. We first prove the statement for $Z \neq \emptyset$, for which we can write $\mathsf{MI}(X; Y|Z) = \mathsf{MI}(X; \{Y, Z\}) - \mathsf{MI}(X; Z)$ by the chain rule for mutual information. Thus, it suffices to prove that $\mathsf{MI}(X; Z) = H(X) + H(Z) - H(X, Z)$ and $\mathsf{MI}(X; \{Y, Z\}) = H(X) + H(Y, Z) - H(X, Y, Z)$. Next, denote μ as the product measure defined based on (X, Z), where $\mu = \mu^1 \times \ldots \times \mu^{k_{XZ}}$, and k_{XZ} is the number of dimensions of X plus that of Z; then by Lemma B.5, p_{XZ} is absolutely continuous with respect to μ. Then, we show that $p_X p_Z$ is absolutely continous with respect to μ. For some k_{XZ}-dimensional Borel set $A = A_1 \times \ldots \times A_{k_{XZ}}$ satisfying $\mu(A) = 0$ there exists $\mu^i \in \{\mu^1, \ldots, \mu^{k_{XZ}}\}$ such that $\mu^{\mathsf{MI}}(A_i) = 0$. Hence, $p_X p_Z(A) = 0$ because $0 \leq p_X p_Z(A) = p_X p_Z(A_1 \times \ldots \times A_{k_{XZ}}) \leq p_X p_Z(\mathbb{R} \times \ldots \mathbb{R} \times A_i \times \mathbb{R} \ldots \times \mathbb{R}) = p_{\mathsf{MI}}(A_i) = 0$, where p_i is the marginalization of the product $p_X p_Z$ to the ith dimension and $p_{\mathsf{MI}}(A_i) = 0$ is because $\mu^{\mathsf{MI}}(A_i) = 0$ by the definition of μ.

Finally, we have that

$$\mathsf{MI}(X; Z) = \int p_{XZ}(x, z) \log \frac{p_{XZ}(x, z)}{p_X(x) p_Z(z)} \mu_X(x) \mu_Z(z) \tag{7}$$

$$= \int p_{XZ}(x, z) \log \frac{p_{XZ}(x, z)}{p_X(x) p_Z(z)} \mu(x, z) \tag{8}$$

$$= H(X) + H(Z) - H(X, Z) \ . \tag{9}$$

The proof for $\mathsf{MI}(X; \{Y, Z\})$ is equivalent. If $Z = \emptyset$, CMI reduces to $\mathsf{MI}(X; Y)$, for which we can prove the statement in the same manner. □

B.7.1 Proof of Theorem B.1. To prove Theorem B.1 we need several intermediate results. Lemma B.11 shows that a histogram results in a valid discretization as all terms corresponding to volumes in I^h cancel out, and hence I^h can be written as a sum of plug-in estimators of *discrete entropies*. Then, Lemma B.9 shows a classic result that the plug-in estimator of discrete entropies will converge to the true entropy almost surely. Further, we show in Lemma B.10 that as the volumes of semiring histogram bins containing continuous values go to 0, the true entropies of discretized variables (which are discretized by the semiring histogram) converges to the true entropy of original variables.

Definition B.7. *Given a random vector (X, Y, Z) that contains mixture variables, and a semiring histogram grid B, we define the discretized random variable X', Y', Z', with probability mass function*

$$p_{X',Y',Z'}((j_1, j_2, j_3)) = \int_{B_j} p_{XYZ} d\mu \ ,$$

where B_j denotes the jth bin of B.

Definition B.8. *Given discrete random variables X', Y', Z' (possibly multi-dimensional), with support S'_X, S'_Y, S'_Z, and given dataset $D = \{(x_i, y_i, z_i)\}_{i \in \{1,\ldots,n\}}$ with sample size n, the plug-in estimator of discrete entropy H is denoted and defined as*

$$H_n(X', Y', Z') = - \sum_{j \in S'_X \times S'_Y \times S'_Z} \hat{p}(j) \log \hat{p}(j)$$

with probability estimates

$$\hat{p}(j) = \frac{|\{(x_i, y_i, z_i)_{i \in \{1,\ldots,n\}} : (x_i, y_i, z_i) = j\}|}{n} \ ,$$

where $|\cdot|$ represents the cardinality of a set, and j is an element in $S'_X \times S'_Y \times S'_Z$.

Lemma B.9 ([7]). *Given a discrete random vector (X', Y', Z'), $lim_{n \to \infty} H_n(X', Y', Z') = H(X', Y', Z')$ almost surely.*

Lemma B.10. *Given a random vector (X, Y, Z) that contains discrete-continuous mixture random variables, with bins $B = \underline{B} \cup \overline{B}$ and the resulting discretized random vector (X', Y', Z'), where $\overline{B}$ contains discrete data points (of which every dimension has a discrete value) and $\underline{B} = B \setminus \overline{B}$, we have*

$$\lim_{b \to 0} H(X', Y', Z') = H(X, Y, Z) \ ,$$

where $b = \max_{B_j \in \underline{B}}(\mu(B_j))$.

Proof. Firstly, it is well-known that this result holds if (X, Y, Z) is a continuous random vector [11]; then, if (X, Y, Z) contains mixture variables,

$$H(X, Y, Z) = \lim_{b \to 0} \sum_{B_j \in \underline{B}} \frac{p_{X'Y'Z'}}{\mu(B_j)} \log \frac{p_{X'Y'Z'}}{\mu(B_j)} \tag{10}$$

$$+ \sum_{B_j \in \overline{B}} \frac{p_{X'Y'Z'}}{\mu(B_j)} \log \frac{p_{X'Y'Z'}}{\mu(B_j)} \tag{11}$$

$$= \lim_{b \to 0} H(X', Y', Z') \ , \tag{12}$$

which concludes the proof. □

Lemma B.11. *Given a k-dimensional random vector (X, Y, Z) that contains mixture variables with an unknown probability function p_{XYZ}, a dataset $D = \{(x_i, y_i, z_i)\}_{i \in \{1,\ldots,n\}}$ generated by p_{XYZ}, a semiring histogram model M, and corresponding discretized random vector (X', Y', Z'), we have $\mathsf{MI}^h(X, Y|Z)$ is equivalent to*

$$H_n(X', Z') + H_n(Y', Z') - H_n(X', Y', Z') - H_n(Z') \ .$$

That is, the terms corresponding to volumes in I^h cancel out and our semiring histogram model results a valid discretization.

Proof. Denote the grid of semiring histogram model M as B^{XYZ}, which is the Cartesian product of bins defined on X, Y, Z—i.e., $B^{XYZ} = B^X \times B^Y \times B^Z$, and denote the corresponding MLE of histogram density function as $f^h_{\hat{\theta}_{XYZ}}$. Further, define a function μ_X, such that for each x_i in D, $\mu_X(x_i) = \mu(B^X_j)$ if $x_i \in B^X_j$, where B^X_j is

a bin of B^X and μ is defined based on the random variable X. Then, define $\mu_Y, \mu_Z, \mu_{XZ}, \mu_{YZ}, \mu_{XYZ}$ similarly.

By the definition $\mathsf{MI}^h(X, Y|Z)$ is equivalent to

$$H^h(X,Z) + H^h(Y,Z) - H^h(X,Y,Z) - H^h(Z) \ .$$

First consider $H^h(X,Z)$. We write $B^{XZ} = B^X \times B^Z$, with marginal density function $f^h_{\hat{\theta}_{XZ}}$. W.l.o.g. suppose that B_{XZ} consists of K bins, denoted as $B^{XZ}_j, j \in \{1, \ldots, K\}$. Then,

$$\begin{aligned} H^h(X,Z) &= -\int_{\mathbb{R}^{k_X+k_Z}} f^h_{\hat{\theta}_{XZ}} \log f^h_{\hat{\theta}_{XZ}} d\mu \\ &= -\sum_{j=1}^{K} \int_{B^{XZ}_j} f^h_{\hat{\theta}_{XZ}} \log f^h_{\hat{\theta}_{XZ}} d\mu \\ &= -\sum_{j=1}^{K} c_j \log\left(\frac{c_j}{n\mu(B_j)}\right) \\ &= -\sum_{j=1}^{K} c_j \log\left(\frac{c_j}{n}\right) + \sum_{i=1}^{n} \log(\mu_{XZ}(x_i, z_i)) \\ &= H_n(X', Z') + \sum_{i=1}^{n} \log(\mu_{XZ}(x_i, z_i)) \ , \end{aligned} \tag{13}$$

where c_j is the number of data points in B_j and $\mu_{XZ}(x_i, z_i) = \mu_X(x_i)\mu_Z(z_i)$. The remaining entropies can be calculated similarly. Hence, $\mathsf{MI}^h(X, Y|Z) = H_n(X', Z') + H_n(Y', Z') - H_n(X', Y', Z') - H_n(Z')$, as the sum of the volume related terms

$$\sum_{i=1}^{n} \log(\mu_{XZ}(x_i, z_i)) + \sum_{i=1}^{n} \log(\mu_{YZ}(y_i, z_i)) \tag{14}$$

$$-\sum_{i=1}^{n} \log(\mu_{XYZ}(x_i, y_i, z_i)) - \sum_{i=1}^{n} \log(\mu_Z(z_i)) \tag{15}$$

is equivalent to zero. □

To prove Theorem B.1, we link the above results:

$$\begin{aligned} &\lim_{b\to 0} \lim_{n\to\infty} \mathsf{MI}^h(X; Y \mid Z) \\ &= \lim_{b\to 0} \lim_{n\to\infty} (H^h(X,Z) + H^h(Y,Z) - H^h(X,Y,Z) - H^h(Z)) \\ &= \lim_{b\to 0} \lim_{n\to\infty} (H_n(X',Z') + H_n(Y',Z') - \\ &\quad H_n(X',Y',Z') - H_n(Z')) \\ &= \lim_{b\to 0} (H(X',Z') + H(Y',Z') - H(X',Y',Z') - H(Z')) \\ &= \mathsf{MI}(X; Y \mid Z) \ . \end{aligned} \tag{16}$$

C IMPLEMENTATION DISCUSSION

Notably, we conduct BCD between T and every other variable in $\mathbf{A}$. In implementation, we process variables in $\mathbf{A}$ in batches where $\beta / w_{\widetilde{\epsilon_Z}}$ differs between variables in $\mathbf{A}$, resulting in duplicating $R[T]$. To avoid this, we apply another layer of approximation based on $|\lfloor ab \rfloor - a\lfloor b \rfloor| \in [0, |a|]$ such that $\lfloor c_1 \lfloor c_2 \cdot t[\widetilde{T}] \rfloor \rfloor \approx \lfloor \beta \cdot t[\widetilde{T}] / w_{\widetilde{\epsilon_Z}} \rfloor$ while c_2 is constant across all variables in $\mathbf{A}$.