# On Explaining Confounding Bias

Brit Youngmann
*CSAIL MIT*
brity@mit.edu

Michael Cafarella
*CSAIL MIT*
michjc@csail.mit.edu

Yuval Moskovitch
*Ben Gurion University of the Negev*
yuvalmos@bgu.ac.il

Babak Salimi
*University of California, San Diego*
bsalimi@ucsd.edu

***Abstract*—When analyzing large datasets, analysts are often interested in the explanations for unexpected results produced by their queries. In this work, we focus on aggregate SQL queries that expose correlations in the data. A major challenge that hinders the interpretation of such queries is *confounding bias*, which can lead to an unexpected correlation. We generate explanations in terms of a set of *potential confounding variables* that explain the unexpected correlation observed in a query. We propose to mine candidate confounding variables from external sources since, in many real-life scenarios, the explanations are not solely contained in the input data. We present an efficient algorithm that finds a concise subset of attributes (mined from external sources and the input dataset) that explain the unexpected correlation. This algorithm is embodied in a system called MESA. We demonstrate experimentally over multiple real-life datasets and through a user study that our approach generates insightful explanations, outperforming existing methods even when are given with the extracted attributes. We further demonstrate the robustness of our system to missing data and the ability of MESA to handle input datasets containing millions of tuples and an extensive search space of candidate confounding attributes.**

## I. Introduction

When analyzing large datasets, analysts often query their data to extract insights. Oftentimes, there is a need to elaborate upon the queries' answers with additional information to assist analysts in understanding unexpected results, especially for aggregate queries, which are harder to interpret [1], [2]. While aggregate query results expose correlations in the data, the human mind cannot avoid a causal interpretation. Thus, we provide explanations for unexpected correlations observed in aggregate queries using causation terms.

In this work, we focus on SQL queries that are aggregating an *outcome attribute* ($O$) based on some groups of interest indicated by a grouping attribute, referred to as the *exposure* ($T$) [3]. A major challenge that hinders the interpretation of such queries is *confounding bias* [4] that can lead to a spurious association between $T$ and $O$ and hence perplexing conclusions. Confounding bias occurs when analysts try to determine the effect of an exposure on an outcome but unintentionally measures the effect of another factor(s) (i.e., a *confounding variable(s)*) on the outcome. This results in a distortion of the actual association between $T$ and $O$ [3]. We are interested in generating explanations in terms of a set of confounding attributes that explain unexpected correlations observed in query results.

A key observation that guides this work is that in many cases, uncontrolled confounding variables might be found outside the narrow query results that the analyst observes and the database being used. Thus, there is a need to develop automated solutions that can explain unexpected correlations to analysts, which goes beyond just the data accessed by the query. To illustrate, consider the following example.

**Example I.1.** *Ann is an analyst in the WHO organization who aims to understand the coronavirus pandemic for improved policymaking. She examines a dataset containing information describing Covid-19-related facts in multiple cities worldwide. It consists of the number of deaths-/recovered-/active-/new-per-100-cases in each city. Ann evaluates the following query over this dataset:*

```
SELECT Country, avg(Deaths_per_100_cases)
FROM Covid-Data
GROUP BY Country
```

*A visualization of the query results is given in Figure 1. Here, the exposure is* COUNTRY *and the outcome is* DEATHS_PER_100_CASES. *Ann observes a puzzling correlation between the exposure and outcome; namely, she wonders why the choice of the country has such a substantial effect on the death rate. She is interested in finding a set of confounding variables that explain this association. She sees that the attribute* CONFIRMED_CASES *from* COVID-DATA *is correlated with* DEATHS_PER_100_CASES. *However, this attribute alone is not enough to explain the correlation. For example, while Germany had the fifth-most confirmed cases worldwide, it had only a fraction of the death toll in other countries. Ann understands that other factors (that are not in the data) affect this association. She remembers reading in the news that as a country's success (defined by multiple variables, including GDP[1] and HDI[2]) grows, the death rate decreases [5], [6]. However, such properties of countries are not available in her data but could be extracted from external sources.*

We propose to mine candidate confounding attributes from external sources. In general, our framework can extract candidate confounders from any knowledge source (e.g., related tables, data lakes) as long as it can be integrated with the input data. This paper focuses on mining attributes from a Knowledge Graph (KG) for the following reasons. KGs can effectively organize and represent a large amount of data [7]–[9]. KGs have been efficiently utilized in various tasks, such as question-answering and recommendation [10]. Further,

[1]Gross domestic product (GDP) is the monetary value of all goods and services made within a country during a specific period.

[2]The Human Development Index (HDI) is a statistic composite index of life expectancy, health outcome, and per capita income indicators.

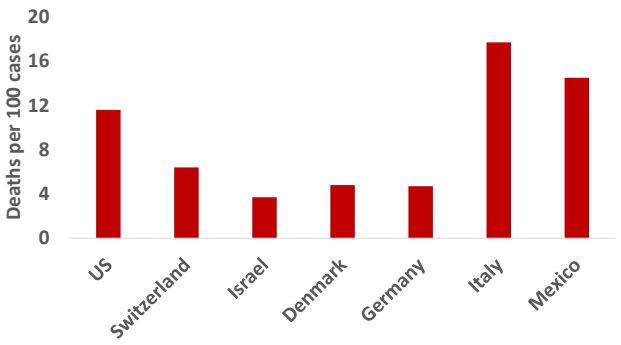


Figure 1: Visualization of the results of the query $Q$.

attribute names in KGs are typically highly informative, allowing analysts to reason about generated explanations. However, the sheer breadth of coverage that makes KGs potentially valuable also creates the need to automate the process of mining relevant confounding variables. There are multiple general-purpose and domain-specific KGs that store data collected from multiple sources. We argue that such data could be utilized for explaining unexpected correlations observed in user queries in a wide range of scenarios.

To this end, we present an efficient algorithm that finds a subset of potential confounding attributes (mined from external sources and from the input dataset) that explain unexpected correlations observed in user queries. We note that our algorithm does not rely on any background knowledge and is focused on identifying correlations. It is up to the analyst to examine the validity of the recommended confounding attributes and conduct a more thorough causal analysis to establish causation. This algorithm is embodied in a system called MESA, which automatically mines candidate attributes from a given knowledge source.

**Example I.2.** *Ann uses* MESA *to search for an explanation for her query.* MESA *mines all available attributes about countries that appear in her data from DBpedia. She learns that besides* CONFIRMED_CASES*, the attributes* HDI*, and* GDP *are potential confounding attributes. She sees that the death rate is similar in countries with a similar number of confirmed cases, HDI, and GDP. She is pleased because she found a plausible real-world explanation for her query results [5], [6].*

Previous work provides explanations for trends and anomalies in query results in terms of predicates on attributes that are shared by one (group of) tuple in the results but not by another (group of) tuple [11]–[15]. However, those methods do not account for correlations among attributes and are thus inapplicable for explaining unexpected correlations. HypDB [1] aims to identify the direct causes of the exposure attribute $T$ and adjust for them in order to eliminate confounding bias. In this sense, it considers the most relevant attributes to $T$ and ignores the outcome attribute $O$ altogether. Further, HypDB relies on very strong assumptions about the underlying causal model that are often impractical, and the algorithm for parent discovery is computationally prohibitive. We share with CajaDE [12] the motivation of considering explanations that are not solely drawn from the input table. CajaDE generates insightful explanations based on contextual information mined from tables related to the table accessed by the query. Their explanations are a set of patterns that are unevenly distributed among $T$, and are *independent of* $O$. Thus, CajaDE may generate explanations that are irrelevant to understanding the correlation between $T$ and $O$.

Our framework supports a rich class of aggregate SQL queries that compare among subgroups, investigating the relationship between the outcome and grouping attributes. To explain the correlation between $T$ and $O$ observed in the results of a query $Q$, we formalize the CORRELATION-EXPLANATION problem that seeks a set of potential confounding attributes, which minimizes the partial correlation between $T$ and $O$ (to measure the correlation between $T$ and $O$, while controlling for the effect of confounding variables). Further, MESA enables analysts to learn the individual responsibility of selected attributes and to automatically identify unexplained data subgroups (correspond to refinements of $Q$) for which the generated explanation might be insufficient.

Given an input database $\mathcal{D}$ and a knowledge source, we extract attributes representing additional properties of entities from $\mathcal{D}$. The attributes are extracted only after the query arrives (as the knowledge source may be a part of the input). Extracted attributes may contain many missing values, especially ones extracted from a KG where data is sparse. Previous work showed that common approaches for handling missing data could cause substantial *selection bias* [16] (which occurs when the obtained data fails to properly represent the population intended to be analyzed) if many values are missing [16]. In contrast to prediction, the quality of explanations is more sensitive to missing data [17]. We, therefore, present a principled way of handling missing values, ensuring the explanations are robust to missing data. We provide sufficient conditions to detect selection bias and an algorithmic approach to handle it properly.

There are potentially hundreds of attributes that could be extracted from external sources. Thus, there is a need to develop an efficient algorithm to search for the optimal attribute set (i.e., explanation) in this extensive search space. Further, the search for the optimal attribute set involves estimating partial correlation for high-dimensional conditioning sets, which is notoriously difficult [18]. To this end, we propose the MCIMR algorithm, a highly efficient algorithm which does not require iterating over all possible attribute sets, and avoids estimating high-dimensional conditioning sets. It selects attributes based on Min-Conditional-mutual-Information (a common measure for partial correlation) and Min-Redundancy criteria. We prove that if the size of the optimal solution $k$ is given, it finds the optimal $k$-size solution. However, in practice, $k$ is unknown. We thus define a heuristic stopping criterion, allowing the algorithm to stop when no further improvement is found.

We conduct an experimental study based on four commonly used datasets that evaluate the quality and efficiency of our algorithm. Our approach is effective whenever the explanation

can be found in a given knowledge source. We show that this was the case in $72.5\%$ of random aggregate queries evaluated on these datasets, using the DBPedia KG [19] for attribute extraction. For quality evaluation, we focus on $14$ representative queries suffering from confounding bias. These queries are inspired by real-life analysis reports, such as Stack Overflow annual reports [20] and academic papers [5]. We ran a user study to evaluate the quality of our explanations compared with seven approaches. We show that the explanations generated by MCIMR are almost as good as those of a computationally infeasible brute-force method and are much better than those of feasible competitors. We also show that previous findings in each domain support our substantive explanations. Our experiments demonstrate the robustness of our solution to missing data and indicate the effectiveness of our algorithm in finding explanations in less than $10$s for queries evaluated on datasets containing more than $5$M tuples.

Our main contributions are summarized as follows:

- We formalize the CORRELATION-EXPLANATION problem that seeks a subset of attributes that explains unexpected correlations observed in SQL queries (Section II).
- We propose to extract unobserved confounding attributes from external sources and focus on KGs. We develop a principled way to avoid selection bias (Section III).
- We devise an efficient algorithm for the CORRELATION-EXPLANATION problem. We embody this algorithm in a system called MESA which enables analysts to automatically identify unexplained data groups (Section IV).
- We qualitatively evaluated the generated explanations over real-life datasets through a user study. We further conducted performance experiments to assess scalability (Section V).

Related work is presented in Section VI and we conclude in Section VII. All proofs are given in the Appendix.

## II. MODEL AND PROBLEM FORMULATION

### A. Data Model

We operate on a standard multi-relational dataset $\mathcal{D}$. To simplify the exposition, we assume $\mathcal{D}$ consists of a single relational table, however, our definitions and results apply to the general case. The table's attributes are denoted by $\mathcal{T}$. We use bold letters for sets of attributes $\boldsymbol{T}\subseteq\mathcal{T}$. We expect the reader is familiar with basic information theory measures, such as entropy and conditional mutual information. Our framework supports a rich class of SQL queries that involve groping, joins and different aggregations to support complex real-world scenarios. The queries we examine compare among subgroups, investigating the relationship between an aggregated attribute $O$ (the outcome) and a grouping attribute $T$ (the exposure). To simplify the exposition, we assume a single grouping attribute. However, our results can be naturally generalized for multiple grouping attributes. We call the condition $C$ (given by the WHERE clause) the context for the query.

We use the following example based on the Stack Overflow (SO) dataset throughout this paper.

**Example II.1.** *SO dataset contains information about people who code around the world, such as their age, income, and country. Consider the following query:*

```
SELECT Country, avg(Salary)
FROM SO
WHERE Continent = Europe
GROUP BY Country
```

*Here, $O$ is* SALARY*, $T$ is* COUNTRY*, the context $C$ is* CONTINENT = EUROPE*, and the aggregation function is average. We aim to explain the difference in the average salary of developers from each country in Europe. While some attributes from SO may partially explain this (e.g.,* GENDER, DEVTYPE*), other important attributes that can cast light on this difference cannot be found in this dataset.*

*Knowledge Extraction.* In general, MESA can extract attributes from any external source, such as related tables, data lakes, or Knowledge Graphs (KGs), as long as it can be integrated with the input dataset. This paper focuses on mining attributes from a KG. KGs can effectively organize a large amount of (domain specific or general) data, and have been successfully utilized in various downstream applications, such as question-answering systems, search engines, and recommendation systems [10]. One of the strengths of KGs is that most of the attributes are already reconciled. Namely, we will not have to match different versions of attributes across different entities. Further, attributes names are typically highly informative, allowing analysts to reason about the generated explanations. Extracting attributes from other sources poses a series of additional challenges, including handling many-to-many relations and uninformative attribute names. We leave these extensions for future research.

To ensure the knowledge source is relevant for a given dataset, MESA allows the analyst decide which source to use for attribute extraction. Given a knowledge source (e.g., domain specific KG [21], [22], publicly available KG [19], [23], [24]), we extract a set of attributes $\mathcal{E}$ representing additional properties of entities from $\mathcal{D}$.

Continuing with our example, $\mathcal{E}$ could be a set of properties of countries extracted from a KG, such as their density, and HDI. We can potentially join $\mathcal{E}$ and $\mathcal{T}$, by linking values from $\mathcal{T}$ with their corresponding entities in the KG that were used for attributes extraction. However, $\mathcal{E}$ may contain many attributes, most of them are irrelevant for explaining the observed correlation.

### B. Problem Formulation

Given a query, the analyst observes an unexpected correlation between the exposure $T$ and the outcome $O$. We assume there is *confounding bias* that causes a spurious association between $T$ and $O$. Confounding bias is a systematic error due to the uneven or unbalanced distribution of a third variable(s), known as the confounding variable(s) in the competing groups. Uncontrolled confounding variables lead to an inaccurate estimate of the true association between $T$ and $O$. Our goal is to discover potential confounding variables.

Let $\mathcal{A}$ denote $\mathcal{E} \cup \mathcal{T} \setminus \{O, T\}$, referred to as the candidate attributes. We search for an attribute set $\boldsymbol{E} \subseteq \mathcal{A}$ that control the correlation between $O$ and $T$, i.e., when conditioning on $\boldsymbol{E}$, the correlation between $O$ and $T$ is diminished. We call such a set the explanation.

**Example II.2.** *It is very likely that countries' economic features (such as GDP, and Gini) affect developers' salaries. To unearth the association between* COUNTRY *and* SALARY*, one must measure the correlation while controlling for such attributes. This will allow analysts to understand which factors affect the differences in developers' salaries. Intuitively, we expect the average developers' salaries to be similar in countries with similar economic characteristics.*

Ideally, we look for a minimal-size set of attributes $\boldsymbol{E} \subseteq \mathcal{A}$ s.t: $(O \perp\!\!\!\perp T | \boldsymbol{E}, C)$. However, in practice, we may not find such perfect explanations (that entirely explains the correlation), hence we search for a minimal-size set of attributes that *minimizes the partial correlation between $T$ and $O$*. Partial correlation measures the strength of a relationship between two variables, while controlling for the effect of other variables. A common measure of partial correlation is multiple linear regression, which is sensitive only to linear relationship. Other partial correlation measures, such as Spearman's coefficient, are more sensitive to nonlinear relationships [25], [26]. Here we use *Conditional Mutual Information* (CMI), a common measure of the mutual dependence between two variables, given the value of a third. We chose CMI because (1) it is a widely used non-parametric measure for partial correlation [27], (2) there is a plethora of techniques for estimating it from data [1], (3) it allows us to develop information-theoretic optimizations. CMI may suffer from underestimation, especially when quantifying dependencies among variables with high associations [28]. However, we avoid such cases since, as we explain in Section IV-B, we discard all attributes that are logically dependent on $T$ or $O$. Note that $(O \perp\!\!\!\perp T | \boldsymbol{E}, C)$ holds iff $I(O; T | \boldsymbol{E}, C) = 0$, where $I(O; T | \boldsymbol{E}, C)$ is the mutual information of $O$ and $T$ while conditioning on $\boldsymbol{E}$. We formalize the CORRELATION-EXPLANATION problem as follows:

**Definition II.1** (**CORRELATION-EXPLANATION**)**.** *Given a set of candidate attributes $\mathcal{A}$ and a query $Q$, find a set of attributes $\boldsymbol{E}^*$ s.t.: $\boldsymbol{E}^* = argmin_{\boldsymbol{E} \subseteq \mathcal{A}} I(O; T | \boldsymbol{E}, C) \cdot |\boldsymbol{E}|$.*

Following previous work [14], [29], [30], besides the explanatory power, we also consider the cardinality of the sets. To combine these two objectives, we multiply the explanatory power by the cardinality of the attribute set. While other aggregation functions could also be used, our approach is invariant to a particular choice of aggregation function.

We assume $\mathcal{A}$ does not contain attributes that have logical dependencies with $T$ or $O$. This reflects a common assumption in causal inference that the underlying distribution is strictly positive. In Section IV-B we explain how we discard such attributes from consideration.

**Example II.3.** *Among other attributes, we extracted from a KG the* GINI *($E_1$),* DENSITY *($E_2$), and* HDI *($E_3$) attributes. An attribute from SO is the developers* GENDER *($E_4$). According to our data, we have $I(O; T | C) = 2.6$. When conditioning on $E_1$, we get: $I(O; T | C, E_1) = 1.3$. Namely, in countries with a similar Gini index, there is less correlation between the country of developers and their salaries. When also considering* DENSITY*, we get: $I(O; T | C, E_1, E_2) = 0.03$. Thus, this set of attributes explains away the correlation in $Q_{so}$. When conditioning on* HDI*, on the other hand, we get: $I(O; T | C, E_3) = 2.5$. Since the HDI of countries in Europe is similar, this attribute does not explain the observed correlation.*

We enable analysts to learn the individual *responsibility* of selected attributes. Given an explanation $\boldsymbol{E}$, we rank the attributes in $\boldsymbol{E}$ in terms of their responsibilities as follows:

**Definition II.2** (Degree of responsibility)**.** *Given a query $Q$ and set of attributes $\boldsymbol{E}$, the degree of responsibility of an attribute $E_i \in \boldsymbol{E}$ is defined as follows:*

$$Resp(E_i) := \frac{I(O; T | \boldsymbol{E} \setminus \{E_i\}, C) - I(O; T | \boldsymbol{E}, C)}{\sum_{E_j \in \boldsymbol{E}} (I(O; T | \boldsymbol{E} \setminus \{E_j\}, C) - I(O; T | \boldsymbol{E}, C))}$$

The responsibility of an attribute $E_i$ is the normalized value of its individual contribution. When all attributes in $\boldsymbol{E}$ contribute to the explanations (i.e., the numerator is positive), the denominator is non-negative. The responsibility of $E_i$ is positive if $E_i$ contributes to the explanation. Thus, a negative responsibility indicates that adding $E_i$ only harms the explanation (it happens since $E_i$ has a negative interaction information with $O$ and $T$). The higher the responsibility of an attribute, the greater is its individual explanatory power.

**Example II.4.** *Recall that $E_1 =$* GINI*, and $E_2 =$* DENSITY*. Let $\boldsymbol{E} = \{E_1, E_2\}$. According to our data we have: $I(O; T | C, E_2) = 1.51$. We get: $Resp(E_1) = 0.54$, and $Resp(E_2) = 0.46$. The attribute* HOBBY *($E_5$) indicates whether a developer is coding as an hobby. It has a negative interaction information with $O$ and $T$. We have $I(O; T | C, E_5) = 2.7 > I(O; T | C)$. Let $\boldsymbol{E} = \{E_1, E_5\}$. We get: $I(O; T | C, \boldsymbol{E}) = 1.5$, $Resp(E_1) = 1.2$, and $Resp(E_5) = -0.2$. Since $E_5$ did not contribute to the explanation, its responsibility is negative.*

The Shapley value [31] is a game-theoretic concept that has recently been shown to be useful in explaining complex data-intensive computations, such as query results and model performance [32]–[37]. Our responsibility scores can be combined with Shapley values to account for interactions between attributes. Specifically, the responsibility score can be used to quantify the contribution of an attribute in a particular coalition (an attribute subset). However, computing Shapley values is generally intractable [34], [38]. While this is not the focus of the current work, extending the responsibility scores with Shapley values is an interesting direction for future research.

*Key Assumption*. We generally believe that attributes with low responsibility are of little interest to analysts and that XOR-like explanations (in which the explanation power of

each individual attribute is low, but their combination makes a good explanation) are hard to understand; thus, they are less likely to be considered good explanations. Our view is motivated by [39]. A similar assumption is often made in feature selection [40], [41], where they assume the optimal feature set does not contain multivariate associations among features, which are individually irrelevant to a target class but become relevant in the presence of others. We further believe true XOR phenomena are likely to be uncommon in real datasets; the practical success of feature selection methods that make this assumption [27] is some evidence for this view. Further, generating XOR explanations would be a substantial additional technical challenge. It would eliminate our ability to prune low-relevance attributes and to define a stopping criterion for our algorithm.

## III. Attributes Extraction

### A. Extracting the Candidate Attributes

MESA extracts attributes representing additional properties of entities from $\mathcal{D}$ from a given knowledge source. In general, we may extract attributes from any given source as long as it can be integrated with the input dataset. For example, we may extract attributes from a data lake, leveraging existing methods to join an input table with other tables [26], [42]–[45]. As mentioned in Section II-A, here we focus on extracting attributes from a given KG.

**Extracting Attributes from a KG**: Given a KG, the first step is to map values that appear in the table $\mathcal{T}$ to their corresponding unique entities in the KG $\mathcal{G}$. This task is often referred to as the Named Entity Disambiguation (NED) problem [46]. We can use any off-the-shelf NED algorithm (e.g., [46], [47]) to match any non-numerical value in $\mathcal{T}$ to an entity in $\mathcal{G}$. Next, given an entity from $\mathcal{T}$, we extract all of its properties from $\mathcal{G}$. We then organize all the extracted properties into a table, setting a null value to all properties whose values were missing. This process is equivalent to building the *universal relation* [48] out of all of the entity specific relations that were derived from $\mathcal{G}$. To extract more attributes and potentially improve the explanations, one may "follow" links in $\mathcal{G}$. Namely, extract also properties of values which are entities in $\mathcal{G}$ as well. This process can be done up to any number of hops in $\mathcal{G}$. All properties are then flattened and stored as a single table.

**Accommodating One-to-Many Relations**: The process described above assumes that each entity is associated with a single value. However, real-world data often contain multiple categorical values (see Example III.1). Because correlation is only defined for sets of paired values, downstream applications typically aggregate the values into a single number [45]. MESA supports any user-defined function (e.g., mean, sum, max, first) to perform the aggregation.

**Example III.1.** *A country's leader is an attribute extracted for each country. We can extract properties of the leaders, such as their age and gender, adding to $\mathcal{E}$ additional properties such as* Leader Age*, and* Leader Gender*. Other properties may point to multiple entities. The* US *entity has the property* Ethnic-Group*, which points to different ethnic groups. Each ethnic group is also an entity, and has the property* Population size*. One may add the property* Avg Population size of Ethnic-Group *to $\mathcal{E}$ by averaging the population sizes.*

### B. Handling Missing Data

Extracted attributes, especially ones from KGs where data is sparse, may contain missing values. Our goal is to develop a principled approach to ensure the generated explanations are robust to missing data. Handling missing data is an enduring problem for many systems [49]. The simplest approach to dealing with missing values is to restrict the analysis to complete cases, i.e., discard cases that have missing values.However, this can induce *selection bias* if the excluded tuples are systematically different from those included. For example, if the HDI values of only countries with a very high HDI are missing, restricting the analysis only to complete cases may lead to misleading explanations. A common solution is to impute missing values. Data imputation is unlikely to cause substantial bias if few data are missing, but bias may increase as the number of missing data increases [16]. The approach that we followed is Inverse Probability Weighting (IPW), a commonly used method to correct selection bias [16]. In IPW, we consider only complete cases, but more weight is given to some complete cases than others. We next explain how to adapt IPW into our setting.

For simplicity of presentation, we assume that $\mathcal{T}$ and $\mathcal{E}$ have been joined into a single table. As we will explain in Section IV, for an attribute $E{\in}\mathcal{E}$ we estimate $I(O;T|E,C)$ and $I(E;E')$ for $E'{\in}\mathcal{E}$. Therefore, we need to recover the probabilities $P(O|C,E), P(O|C,T,E), P(E)$, and $P(E|E')$. But since $E$ may contain missing values, we must ensure that those probabilities are *recoverable*. Given an attribute $E$, let $R_E$ denote a selection attribute that indicates if the values of $E$ for the $i$-th tuple in the results of $Q$ is missing. I.e., $R_E[i]{=}1$ if the value of $E$ for the $i$-th tuple was extracted, and $R_E[i]{=}0$ otherwise. A complete cases analysis means that we examine only cases in which $R_E[i]{=}1$. Let $R_E{=}1$ denote the selection of all tuples in which for them $R_E[i]{=}1$ holds. We say the probability of an event $X$ which involves $E$ (e.g., $P(O|E)$) is recoverable if: $P(X){=}P(X|R_E{=}1)$. We next provide sufficient conditions to ensure recoverability.

We prove that $I(O;T|C,E)$ is recoverable if the complete cases are a representative sample of the original data, and each complete case is a random sample from the population of individuals with the same $E$ and $T$ values.

**Proposition III.1.** *If $(O \perp\!\!\!\perp R_E = 1|E,C)$ and $(O \perp\!\!\!\perp R_E = 1|E,T,C)$, then $I(O;T|C,E)$ is recoverable.*

We prove $I(E;E')$ is recoverable if the completeness of a case is independent of $E$, and remains independent given $E'$.

**Proposition III.2.** *If $(E_i \perp\!\!\!\perp R_{E_i}{=}1, R_{E_j}{=}1)$ and $(E_i \perp\!\!\!\perp R_{E_i}{=}1, R_{E_j}{=}1|E_j)$, then $I(E;E')$ is recoverable.*

In situations other than those described above, the probabilities will generally not be recoverable. Following the IPW approach, we assign weights to complete cases, where the weight of an event $X$ is defined as $P(R_E{=}1)/P(R_E{=}1|X)$. However, since $E$ contains missing values, $P(X)$ is unknown. We thus estimate $P(X)$. Commonly, a logistic regression model is fitted [50], [51]. Data available for this are the values in $\mathcal{D}$. We employ a logistic regression to estimate $P(X)$. Note that while this is similar to data imputation, we use existing values for prediction but only predict the weights of existing values, rather than predicting missing values.

## IV. Algorithms

### A. The MCIMR Algorithm

We present the MCIMR algorithm for the Correlation-Explanation problem. Its key advantages are that it *avoids iterating over all possible attribute sets*, and it *avoids estimating CMI for high-dimensional conditioning attribute sets*, which is computationally difficult [52]. However, it does not necessarily outputs the optimal solution to the Correlation-Explanation problem. Nevertheless, our experimental study using real-life datasets and scenarios demonstrates that this algorithm is highly efficient and useful in practice.

Estimating CMI of high-dimensional conditioning sets requires estimating multivariate probability in high dimensions. It is often hard to get an accurate estimation for multivariate probability because of the following difficulties in the high-dimensional space. First, the number of tuples in the dataset is often insufficient to do it accurately. Second, multivariate density estimation often involves computing the inverse of the the high-dimensional covariance matrix, which is typically an ill-posed problem [52]. These problems are more acute for continuous attributes. However, even for discrete attributes, the practical problems in estimating high-dimensional joint probabilities cannot be fully avoided [52].

Our algorithm, therefore, avoids iterating over all possible attribute sets and *calculates only bivariate probabilities*, which is much more accurate. We do so by incrementally selecting attributes based on Minimal-Conditional-mutual-Information (MCI) and Minimal-Redundancy (MR) criteria.

We begin by assuming that the size of the optimal solution $k$ is known and show that MCIMR yields the optimal $k$-size solution in this case. We then remove this assumption and propose a heuristic criterion to stop the algorithm.

For a fixed $k$, the Correlation-Explanation problem becomes finding a $k$-size attribute set $\boldsymbol{E_k}$ such that:

$$\boldsymbol{E_k} = argmin_{\boldsymbol{E}\subseteq\mathcal{A},|\boldsymbol{E}|=k} I(O;T|C,\boldsymbol{E}) \quad (1)$$

Obviously, when $k$ equals 1, the solution is the attribute $E$ that minimizes $I(O;T|C,E)$. When $k{>}1$, a simple incremental solution is to add one attribute at one time: given the set with $k{-}1$ attributes, $E_{k-1}$, the $k$-th attribute to be added can be determined as the one that contributes to the largest decrease of $I(O;T|C,\boldsymbol{E}_{k-1})$.

Importantly, note that we cannot directly compute Equation 1. Instead, we show that the combination of the Min-Conditional-mutual-Information (MCI) and Min-Redundancy (MR) criteria is equivalent to Equation 1 if one feature is selected at each iteration.

**Algorithm 1:** The MCIMR Algorithm.

**input :** A number $k$, a set of attributes $\mathcal{A}$, the outcome, treatment attributes $O$ and $T$, and the context $C$
**output:** An explanation $\boldsymbol{E}$.

```
MCIMR(k, A, O, T, C):
E ← ∅.
for i ∈ [1, k] do
    E_i ← NextBestAtt (O, T, C, E, A)
    if O ⫫ E_i|E                    // The responsibility test for E_i
     then
        return E
    E ← E ⋃ {E_i}
return E
NextBestAtt (O, T, C, E, A):
E* ←None, v ← ∞
foreach E ∈ A \ E do
    /* Weights are added if selection bias was detected */
    v_1 ← I(O; T|C, E), v_2 ← 0          // Min CI computation
    foreach E' ∈ E do
        /* Weights are added if selection bias was detected */
        v_2 ← v_2 + I(E; E')      // Min redundancy computation
    if v_1 + v_2/|E| < v then
        E* ← E, v ← v_1 + v_2/|E|
return E*
```

The idea behind MCI is to search a $k$-size attribute set $\boldsymbol{E_k}$ that satisfies Equation 2, which approximates Equation 1 with the mean value of all CMI values between the individual attributes in $\boldsymbol{E_k}$ and $O$ and $T$:

$$\boldsymbol{E_k} = argmin_{\boldsymbol{E_k}\subseteq\mathcal{A},|\boldsymbol{E_k}|=k} MCI(O,T,C,\boldsymbol{E_k}) \quad (2)$$

where $MCI(O,T,C,\boldsymbol{E_k}){=}\frac{1}{k}\sum_{E\in\boldsymbol{E_k}} I(O;T|C,E)$.

However, it is likely that attributes selected according to MCI are redundant. Thus, the following minimal redundancy condition is added:

$$\boldsymbol{E_k} = argmin_{\boldsymbol{E_k}\subseteq\mathcal{A},|\boldsymbol{E_k}|=k} MR(\boldsymbol{E_k}) \quad (3)$$

where $MR(\boldsymbol{E_k}) = \frac{1}{k^2}\sum_{E_i,E_j\in\boldsymbol{E_k}} I(E_i;E_j)$.

Our goal is to minimize MCI and MR simultaneously. Namely, we look for a $k$-size attribute set $\boldsymbol{E_k^*}{\subseteq}\mathcal{A}$ such that:

$$\boldsymbol{E_k^*}{=}argmin_{|\boldsymbol{E_k}|=k}[MCI(O,T,C,\boldsymbol{E_k}){+}MR(\boldsymbol{E_k})] \quad (4)$$

In the $k$-th iteration we have the $k{-}1$-size attribute set $\boldsymbol{E_{k-1}}$. The $k$-th attribute to be added is the attribute that minimizes the following condition:

$$E_k{=}argmin_{E\in\mathcal{A}\setminus\boldsymbol{E_{k-1}}}[I(O;T|C,E){+}\frac{1}{k{-}1}\sum_{E_i\in\boldsymbol{E_{k-1}}} I(E;E_i)] \quad (5)$$

We prove that the combination of the MCI and MR criteria is equivalent to Equation 1. Namely, when $k$ is given, the MCIMR algorithm computes the optimal $k$-size solution.

**Theorem IV.1.** *The combination of the MCI and MR criteria is equivalent to Equation 1.*

However, in practice, the size of the optimal solution is unknown. A straightforward approach is to generate $m$

attribute sets of sizes $1, \ldots, m$, where $m$ is $|\mathcal{A}|$, using our algorithm. It will then select the optimal solution by comparing these solutions. However, given two solutions of sizes $k$ and $k'$, we cannot accurately determine whether $I(O;T|C,\boldsymbol{E_k})< I(O;T|C,\boldsymbol{E_{k'}})$ or vice versa, since it requires to estimate joint probabilities for high dimensional attribute sets (which, as mentioned above, cannot be done accurately).

**Stopping Criterion**. Therefore, we define a heuristic stopping criterion for the MCIMR algorithm. Specifically, we propose a responsibility test for the next attribute to be added. As mentioned, we assume that attributes in which their marginal explanatory power is small are of no interest to analysts. Thus, given a set of $k$ attributes $\boldsymbol{E}_k$, this test verifies if the responsibility of a candidate attribute $E_{k+1}$ to be added is (approximately) 0. If so, we stop the algorithm without including this attribute.

To implement this responsibility test, we prove that given a $k$-size attribute set $\boldsymbol{E_k}$, the responsibility of a candidate attribute $E_{k+1}$ is close to 0 if $O \perp\!\!\!\perp E_{k+1}|\boldsymbol{E_k}$.

**Lemma IV.1** (Responsibility test)**.** *If* $O \perp\!\!\!\perp E_{k+1}|\boldsymbol{E_k}$ *then* $Resp(E_{k+1}) \leq 0$.

For this test we use the conditional independence test proposed in [1] (which can only be used to determine conditional independence and not for estimating partial correlation).

The full MCIMR algorithm is depicted in Algorithm 1. This algorithm does not directly optimizes the objective of the CORRELATION-EXPLANATION problem. It instead takes as an input a bound $k$ on the maximal explanation size, which the analyst provides. If it has not stopped earlier (according to the stopping criterion), it will terminate after $k$ iterations. Attributes are iteratively added according to the NEXTBESTATT procedure (line 4). The algorithm then applies the responsibility test to a selected attribute. If the responsibility of this attribute is $\approx 0$, it terminates and returns the solution obtained until this point (lines 5-7). Otherwise, it terminates after $k$ iterations (line 9). Given the attribute set selected up until the $i$-th iteration, the NEXTBESTATT procedure finds the $i$-th attribute to be added. It implements Equation 5, by iterating over all candidate attributes and computing their individual explanatory power (line 14), and their redundancy w.r.t. selected attributes (lines 16-18). For simplicity, we omitted parts dedicated to handling missing data from presentation. In our implementation, before executing lines 14 and 18, we check if weights are needed to be added and adjust the computation accordingly.

We next summarize the complexity of our algorithm.

**Proposition IV.1.** *The time complexity of the incremental MCIMR algorithm is* $O(k|\mathcal{A}|)$.

The size of $\mathcal{A}$ is potentially very large. Thus, in the next section, we propose several optimizations to reduce it.

### B. Pruning Optimizations

We propose several optimizations to reduce the size of $\mathcal{A}$ and thereby reduce execution times. These optimizations are used to prune attributes that are either uninteresting as an explanation or cannot be a part of the optimal solution. We propose two types of optimizations: Across-queries optimizations that could be executed at pre-processing, and query-specific once that could be done only once $O$ and $T$ are known. Full details are given in the appendix.

*Preprocessing pruning*. Attributes discarded at this phase either have a fixed value, a unique value for each tuple, or lots of missing values. Thus, such attributes are uninteresting as an explanation [1], [12]. **Simple Filtering**: We drop all attributes with a constant value (e.g., the attribute TYPE which has the value *Country* to all countries), and attributes in which the percentage of missing values is $>90\%$. **High Entropy**: we discard attributes such as WIKIID, that have high entropy and (almost) a unique value for each tuple (as was done in [1]).

*Online pruning*. **Logical Dependencies**: Our goal is to identify potential confounding variables, affecting both $T$ and $O$ and create a spurious correlation between them. The presence of logical dependencies can hinder this process, as they can obscure the true relationships between attributes. This reflects a common assumption in causal inference that the underlying distribution is strictly positive, meaning that all events have non-zero probability. This assumption breaks down in the presence of logical dependencies. We thus discard all attributes that are functionally dependent on $T$ or $O$ (e.g., COUNTRYCODE $\Rightarrow$ COUNTRY) using a test for functional dependencies suggested in [1]. **Low Relevance**: As mentioned, we assume that the optimal explanation does not contain attributes which are individually unimportant but become important in the context of others. We leverage this assumption to prune attributes in which their individual explanatory power is low.

Another possible optimization is to cluster highly correlated attributes, such as HDI and HDI RANK, as was done in [12]. However, we found this optimization to be not useful because of: (1) It could only be done after the query arrives, and the clustering process took longer than running our algorithm on all attributes. (2) We found that attributes clustered together were not necessarily semantically related.

### C. Identifying Unexplained Subgroups

The MCIMR algorithm finds the explanation for the correlation between $T$ and $O$. While the generated explanation is insightful considering the whole data, it may be insufficient for some parts in the data. We thus propose an algorithm the analyst may use after getting the explanation, to identify unexplained data subgroups. It receives the original query $Q$ and the generated explanation. The output is a set of data groups correspond to context refinements of $Q$, in which a different explanation is required and thus may be of interest. In other words, it finds the top-$k$ largest data groups for which the generated explanation might be insufficient.

**Example IV.1.** *Consider a query compare the average salary of developers among countries. The explanation found by* MESA *is* $\boldsymbol{E}$=$\{$HDI, GINI$\}$. *As mentioned, the HDI of all*

*countries in Europe is similar. Thus, for countries in Europe, it is likely that $\boldsymbol{E}$ is not a satisfactory explanation.*

For simplicity, numerical attributes are assumed to be binned. Data groups are defined by a set of attribute-value assignments and correspond to refinement of the context $C$ of $Q$. Treating the context $C$ as a set of conditions, a refinement $C'$ of $C$ is a set s.t. $C' \subset C$. We aim to find the largest data groups s.t. $\boldsymbol{E}$ can not serve as their explanation. Formally, given an explanation $\boldsymbol{E}$, $I(O;T|C,\boldsymbol{E})$ is referred to as the explanation score for $C$. We are inserted in the top-$k$ data groups (in terms of size), each correspond to a context refinement $C'$ of $C$, s.t. their explanation score is $>\tau$ for some threshold $\tau$ ($\tau$ can be set based on the initial explanation score).

**Example IV.2.** *Continuing with Example IV.1, we refine $Q$ by adding a WHERE clause selecting only countries in Europe ($C' = \{$*CONTINENT = EUROPE*$\}$). Let $Q_{EU}$ denote this query. We get: $I(O;T|C',\boldsymbol{E})$=2.13. As mentioned in Example II.3, the optimal explanation for $Q_{EU}$ is* {GINI, DENSITY}.

A naive algorithm would traverse over all possible contexts refinements $C'$, check if the explanation score is $>\tau$, and will choose the largest data groups for which $\boldsymbol{E}$ is not a satisfactory explanation. We propose an efficient algorithm, exploiting the notion of pattern graph traversal [53]. Intuitively, the set of all context refinements can be represented as a graph where nodes correspond to refinements and there is an edge between $C$ and $C'$ if $C'$ can be obtained from $C$ by adding a single value assignment. This graph can be traversed in a top-down fashion, while generating each node at most once.

Algorithm 2 depicts the search for the largest $k$ data groups that for which $\boldsymbol{E}$ is not a satisfactory explanation. It traverses the refinements graph in a top-down manner, starting for the children of $C$. It uses a max heap $MaxHeap$ to iterate over the refinements by their size. It first initialize the result set $\mathcal{R}$ (line 1) and $MaxHeap$ with the children of $C$ (line 2). Then, while the $\mathcal{R}$ consists of less then $k$ refinements (line 3), the algorithm extracts the largest (by data size) refinement $C'$ (line 4) and computes $I(O;T|C',\boldsymbol{E})$. If it exceeds the threshold $\tau$ (line 5), $C'$ is used to update $\mathcal{R}$ (line 6). The procedure `update` checks whether any ancestor of $C'$ is already in $\mathcal{R}$ (this could happen because the way the algorithm traverses the graph). If not, $C'$ is added to $\mathcal{R}$. If $I(O;T|C',\boldsymbol{E}) \leq \tau$ (line 5), the children of $C'$ are added to the heap (lines 8– 9).

**Proposition IV.2.** *Algorithm 2 yields the top-$k$ largest data groups in which their explanation score is grater than $\tau$.*

In the worst case, there are no such $k$ data groups and hence the algorithm traverses over every possible context refinement of $Q$ However, as we show, in practice this algorithm efficiently identifies the data groups of interest, while exploring only an handful of context refinements.

## V. EXPERIMENTAL STUDY

We present experiments that evaluate the effectiveness and efficiency of our solution. We aim to address the following research questions. $Q1$: What is the quality of our explanations, and how does it compare to that of existing methods? $Q2$: How robust are the explanations to missing data? $Q3$ What is the efficiency of the proposed algorithm and the optimization techniques? $Q4$: How useful are our proposed extensions?

**Algorithm 2:** Top-$k$ unexplained data groups.

**input :** A number $k$, a set of attributes $\mathcal{A}$, the attributes $O$ and $T$, the context $C$, an explanation $\boldsymbol{E}$, and a threshold $\tau$.
**output:** Context refinements $\{C_1,\ldots,C_k\}$ s.t. the corresponding groups are the largest $k$ groups and $I(O;T|C_i,\boldsymbol{E})>\tau$

1 $\mathcal{R} \leftarrow \emptyset$
2 $MaxHeap \leftarrow$ GENCHILDREN$(C)$
3 **while** $|\mathcal{R}| < k$ *or* $MaxHeap.isEmpty()$ **do**
4  $C' \leftarrow MaxHeap.extractMax()$
5  **if** $I(O;T|C',\boldsymbol{E}) > \tau$ **then**
6  UPDATE$(\mathcal{R}, C')$ // If none of the ancestors of $C'$ are in $\mathcal{R}$, insert $C'$ into $\mathcal{R}$.
7  **else**
8  **for** $C'' \in$ GENCHILDREN$(C')$ **do**
9  $MaxHeap.insert(C'')$
10 **return** $\mathcal{R}$

Table I: Examined Datasets.

| Dataset | n | —$\mathcal{E}$— | Columns used for extraction |
|---|---|---|---|
| SO [56] | 47623 | 461 | Country, Continent |
| COVID-19 [57] | 188 | 463 | Country, WHO-Region |
| Flights [58] | 5819079 | 704 | Airline, Origin/Destination city/state |
| Forbes [59] | 1647 | 708 | Name |

Our code and datasets are available at [54]. We used DBPedia KG [19] for attribute extraction, and the Pyitlib library [55] for information-theoretic computations. The experiments were executed on a PC with a 4.8GHz CPU, and 16GB memory.

*Datasets*: We examine four commonly used datasets: **(1) SO**: Stack Overflow's annual developer survey is a survey of people who code around the world. It has more than 47K records containing information about developers' such as their age, income, and country. **(2) Covid-19**: This dataset includes information such as the number of confirmed, death and new cases in 2020 across the globe. **(3) Flights**: This dataset contains transportation statistics of over 5.8M domestic flights operated by large air carriers in the USA. **(4) Forbes**: This dataset contains annual earning information of 1.6K celebrities between $2005-2015$ It contains the celebrities' annual pay, and category (e.g., Actors, Producers).

The attributes used for property extraction and the number of extracted attributes in each dataset are given in Table I.

*Baseline Algorithms*: We compare MESA against the following baselines: **(1) Brute-Force**: The optimal solution according to Def. II.1. This algorithm implements an exhaustive search over all attribute subsets. To make it feasible, we run it after employing our pruning optimizations. **(2) Top-K**: This baseline ranks the attributes according to their individual explanatory power (equivalent to Max-Relevance only). **(3) MRMR [52]** This feature selection algorithm selects attributes based on Max-Relevance (measured by the mutual information with $O$) and Min-Redundancy criteria. We also tested a version of MRMR that includes $T$ in the selected attribute set (but does not include it as part of the explanation) to account for the redundancy w.r.t. $T$. **(4) HypDB** [1]: This system employs an algorithm for confounding variable detection based

on causal analysis. It identifies an attribute set that has uneven or unbalanced distribution w.r.t $T$ (ignoring $O$). **(5) MESA$^-$**: To examine the effect of pruning, we examine the explanations generated by MESA without the pruning optimizations.

We also examined the explanations generated using linear regression and CajaDE [12], a system that generates query results explanations based on augmented provenance information. However, since in all cases, those baselines generated explanations obtaining the lowest scores, we omit their results from presentation. More details are provided in the Appendix.

Unless mentioned otherwise, we set the maximal explanation size, $k$, to 5 and extracted attributes for 1-hop in the KG. For a fair comparison, we run all baselines (except for MESA$^-$) after employing our pruning optimizations.

### A. Quality Evaluation (Q1)

We validate our intuition that attributes extracted from KGs can explain correlations in common scenarios. To this end, we randomly generated 40 queries (10 from each dataset) as follows. We set $T$ to be an attribute used for attribute extraction. We set $O$ to be an attribute that could be predicted from the data (e.g., DEPARTURE/ARRIVAL DELAY in Flights, NEW/DEATH CASES in Covid-19). We then added a WHERE clause by randomly picking an attribute-value assignment, ensuring selected subsets contain more than 10% of the dataset. Full details are given in the Appendix. We say our approach was useful if (1) the CMI between $T$ and $O$ (while conditioning on the generated explanation) is lower than the original correlation and (2) the explanation contains at least one extracted attribute. We report this was the case in 72.5% percent of the queries.

Next, we aim to assess the quality of the generated explanations to validate our problem definition. We present a user study consisting of explanations produced by each algorithm. Since a standard benchmark for results explanation does not exist, we consider 14 representative queries suffering from confounding bias, as shown in Table II. Our queries are inspired by real-life sources, such SO annual reports [20], media websites (e.g., Vanity Fair [60], USA Today [61] for Forbes and Flights), and academic papers [5], [6]. Similar experiments were conducted in [1], [2], [12]. To compare the generated explanations with real-world explanations, we show that our explanations are supported by previous findings. These explanations were obtained manually and serve as our "ground truth" to be compared with the generated explanations. A similar approach was taken in [1].

We recruited 150 subjects on Amazon MTurk. This sample size enables us to observe a 95% confidence level with a 10% margin of error. Subjects were asked to rank each explanation of each method (shown together with its corresponding query) on a scale of $1-5$ (a higher score is better).

HypDB's time complexity is exponential in the size of $\mathcal{A}$ [1]. We run it over all attributes in $\mathcal{A}$ (after pruning) and report that it never terminates within 10 hours. Thus, we have no choice but to limit the number of attributes for HypDB to allow it to generate explanations in a reasonable time. For HypDB, besides pruning, we omitted attributes uniformly at random, ensuring that $|\mathcal{A}|\leq 50$. We only report the results of Brute-Force for the small Covid-19 and Forbes datasets, as it was infeasible to compute them for the larger datasets. We do not randomly drop attributes for computational efficiency here because Brute-Force is intended to be an optimal solution for our problem definition against which our algorithm is judged. The explanations generated by different methods are given in Table II, and the average explanation scores given by the subjects are depicted in Table III.

We summarize our main finding as follows:

- The subjects found the explanations generated by Brute-Force, MESA$^-$, and MESA to be the most convincing. This supports our mathematical definition (Def 2.1) of what constitutes a good explanation.
- MESA explanations are supported by previous in-domain findings, which serve as "ground-truth" explanations.
- Our pruning has little effect on explanation quality.
- Even when given with extracted attributes, our approach outperforms existing solutions in terms of either quality (e.g., MRMR, CajaDE) or scalability (HypDB).

First, subjects found the explanations generated by Brute-Force, MESA$^-$, and MESA to be the most convincing. The pairwise differences between the average scores of these methods are not statistically significant. Previous in-domain findings also support these explanations. For example, in SO $Q_1$, it was shown in [20] that there is a correlation between developers' salaries and countries' economies. For Flights $Q_1$, it was stated in [61] that weather is one of the top reasons for flights delay. For Covid-19 $Q_1$, it was shown that there is a correlation between countries' economies and Covid-19 death rate [5], [6]. More details can be found in the Appendix. In all cases where the results of Brute-Force and MESA are different, it happens because MESA drops attributes with insignificant responsibility (according to the responsibility test). For example, in Forbes $Q_1$, MESA dropped AGE. The low difference between the results of MESA$^-$ and MESA indicates that pruning has little effect on explanation quality. Namely, MESA *is able to execute efficiently without compromising on explanation quality.*

The explanations of all methods consist of extracted attributes. This validates our assumptions that KGs can serve as valuable sources for results explanations. The next best competitor is HypDB (the average score is worse than that of MESA. This difference is statistically significant, $p<.05$). This is not surprising as HypDB finds confounding attributes. However, its main disadvantage is its ability to scale. The explanations generated by Top-K and MRMR were considered to be less convincing (their average scores are statistically significant from all other methods, $p<.05$). For Top-K, this is substantially because it ignores redundancy among attributes. For example, in Flights $Q_1$, it chose the attributes YEAR LOW F and YEAR AVERAGE F, which are highly correlated. For MRMR, it substantially because it ignores $T$, as it seeks for attributes that are only correlated with $O$.

Table II: User study: The best and second best explanations are marked in red and blue, resp.

| Dataset | | Query | Brute-Force | MESA- | MESA | Top-K | MRMR | HypDB |
|---|---|---|---|---|---|---|---|---|
| SO | $Q_1$ | Average salary per country | - | **HDI Rank, Gini** | **HDI, Gini** | HDI, Established Date | Population Census, Gini | GDP |
| | $Q_2$ | Average salary per continent | - | **GDP Rank, Density** | **GDP,Density** | GDP,Area rank | GDP, Gini | GDP |
| | $Q_3$ | Average salary per country in Europe | - | **Population Census, Gini Rank** | **Population Census, Gini** | Population Census, Population Estimate | Population Census, HDI | Gini, Area Rank |
| Flights | $Q_1$ | Average delay per origin city | - | **Precipitation Days, Year UV, Airline** | Population urban, Year Low F, Airline | Year Low F, Year Avg F, December Low F | Year Low F, Arrival Delay, Year UV | **Year Low F, May Precipitation Inch, Airline** |
| | $Q_2$ | Average delay per origin state | - | **Density, Year Snow, Airline** | **Population estimation, Year Low F, Airline** | Population estimation, Population Urban, Population Rank | Record Low F, Arrival Delay, Year Low F | Record Low F, Population estimation, Day |
| | $Q_3$ | Average delay per origin cities in CA | - | **Density, Population Metropolitan, Security Delay** | Density, Population Total,Security Delay | Population Metropolitan,Security Delay | Population Size, Density, Arrival Delay | **Density, Population Ranking, Cancelled** |
| | $Q_4$ | Average delay per origin state and airline | - | **Population Total, Fleet size** | **Population Ranking, Fleet size** | Density, Arrival Delay | Year Low F, Density | Revenue, Dec Record Low F |
| | $Q_5$ | Average delay per airline | - | **Equity, Fleet Size** | **Equity, Fleet Size** | Equity, Net Income | Equity, Arrival Delay | **Num of Employees, Revenue** |
| Covid-19 | $Q_1$ | Deaths per country | **HDI, GDP, Confirmed cases** | **HDI, GDP Rank, Confirmed cases** | **HDI, GDP, Confirmed cases** | GDP Rank, GDP Nominal, HDI | Confirmed cases, Recovered cases, New cases | Density, Time Zone, Confirmed cases |
| | $Q_2$ | Deaths per country in Europe | **Gini, Population Census, Confirmed cases** | **Gini Rank, Density, Confirmed cases** | **Gini, Population Census, Confirmed cases** | Gini Rank, Gini, GDP | Confirmed cases, Recovered cases, New cases | Currency, GDP, New cases |
| | $Q_3$ | Average deaths per WHO-Region | **Density, Confirmed Cases** | **Density,Confirmed Cases** | **Density,Confirmed Cases** | **Density,Confirmed Cases** | Confirmed cases, Recovered case | **Area Km,Confirmed Cases** |
| Forbes | $Q_1$ | Salary of Actors | **Net Worth, Gender, Age** | **Net Worth, ActiveSince, Gender** | Net Worth, Gender | Net worth, Awards | Net Worth, Honors | Gender, Honors |
| | $Q_2$ | Salary of Directors/Producers | **Net Worth, Awards** | **Years Active, Net Worth** | **Net Worth, Awards** | Net Worth, Age | **Net Worth, Awards** | Years Active |
| | $Q_3$ | Salary of Athletes | **Cups, Draft Pick, Active Years** | National Cups, Draft Pick | Cups, Draft Pick | Total Cups, National Cups | Active Years, Net Worth | **Cups, Active Years** |

Table III: Avg. explanation scores according to the subjects.

| Baseline | Average Score | Average Variance |
|---|---|---|
| Brute-Force | 3.8 | 0.8 |
| MESA- | 3.7 | 1.1 |
| MESA | 3.5 | 0.9 |
| HypDB | 2.8 | 1.1 |
| MRMR | 2.2 | 0.5 |
| Top-K | 2.1 | 0.8 |

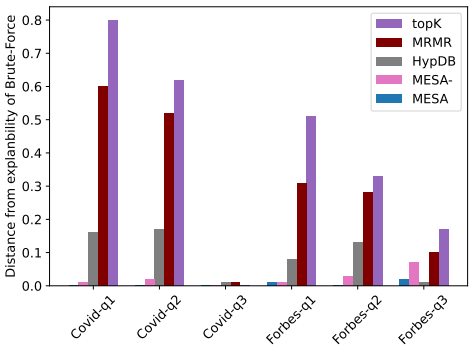



Figure 2: Distance from explainability scores of Brute-Force.

*Explainability scores*. Let $\boldsymbol{E}$ denote the explanation found by an algorithm. We call $I(O;T|\boldsymbol{E})$ the explainability score. Explainability score equal to 0 means that $\boldsymbol{E}$ perfectly explains the correlation between $O$ and $T$. The explainability scores of Brute-Force serve as the gold standard (as by definition, it aims to minimize this score). In some cases, the explanations generated by all algorithms, including Brute-Force, cannot fully explain the correlations. E.g., in Flights $Q_2$, the explainability score of Brute-Force is 0.25. This means that other factors affecting flight delays may not exist in the KG (e.g., labor problems). The results are depicted in Figure 2. The y-axis is the distance between the explainability scores of each method and Brute-Force. The lower the distance, the better the explanation. Observe that the explainability scores of MESA are almost as good as the ones of Brute-Force and are much better than those of the competitors.

Additional experiments are given in the Appendix. We found that: (i) the choice of how to combine cardinality and explanatory power in our problem definition has little effect on quality; (ii) the quality of the used entity linker and regression models to compute weights were largely satisfactory; (iii) attributes extracted through more than one hop in the KG were found to be irrelevant.

### B. *Robustness to Missing Data (**Q2**)*

Statistics regarding the percentage of missing values and the percentage of extracted attributes suffering from selection bias are given in the Appendix. We report that, on average, selection bias was detected in 19% of extracted attributes.

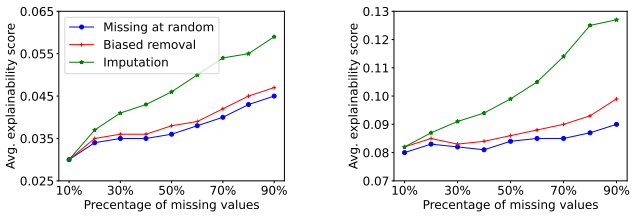



Figure 3: Explainability as a function of missing data.

We examine the robustness of our explanations to missing data by varying the percentage of missing values from the top 10 most relevant attributes. We examine two ways to omit values: missing-at-random and biased removal, where the top-$x$ highest values were omitted (when varying $x$). We examine the effect on our generated explanations' average explainability score. Explainability should not be affected if an explanation is robust to missing data. We also examine the effect on the explainability scores while imputing missing values (using mean imputation [62]). The results for the SO and Covid datasets are depicted in Figure 3. As expected, data imputation has a huge negative effect on explainability. Our approach is much less sensitive to missing data: Even with 50% missing values (at random or not), the explainability scores have hardly changed. When the percentage of missing values is above 50%, a lot of the information is lost, and thus it is harder to estimate partial correlation correctly.

### C. Efficiency Evaluation (Q3)

To examine the contribution of our optimizations, we report the running times of the following baselines: **No Pruning** —the MCIMR algorithm without pruning; **Offline Pruning** —MCIMR with only offline pruning. We study the effect of multiple parameters on running times. For each dataset, we report the average execution time of the queries presented in Section V-A. In all cases, the execution time of MCIMR was less than 10 seconds. We omit the results obtained on the (smallest) Covid-19 dataset from presentation, as the results demonstrated similar trends to those of Forbes.

*Candidate Attributes*. In this experiment, we discarded attributes from $\mathcal{A}$ uniformly at random. The results are depicted in Figure 4. In all datasets, we exhibit a (near) linear growth in running times as a function of the size of $\mathcal{A}$. The execution times of No-Pruning are significantly higher than those of Offline Pruning and MCIMR, *indicating the usefulness of offline pruning*. The difference in times across datasets is due to their size. Estimating CMI on large datasets takes longer than on small datasets. In Forbes, Offline Pruning is faster than MCIMR, implying that in small datasets online pruning is not necessary, as it takes longer than running MCIMR.

*Data Size*. We vary the number of tuples in the datasets by removing tuples uniformly at random. The results are depicted in Figure 5. In SO and Flights, observe that the dataset size has little effect on running times. This is because of the fact that when randomly omitting tuples, the number of considered groups in the queries is almost unchanged. On the other hand, since in Forbes, each group contained only a few records, we exhibit a (near) linear growth in running times.

Table IV: Top-5 unexplained groups for SO Q1.

| Rank | Size | Data group |
|---|---|---|
| 1 | 18342 | CONTINENT = EUROPE |
| 2 | 17899 | CONTINENT = ASIA |
| 3 | 15466 | CONTINENT = NORTH AMERICA |
| 4 | 14788 | CURRENCY = EURO |
| 5 | 12754 | CONTINENT = AFRICA |

*Explanation size*. We vary the bound on the explanation size. Recall that given a bound $k$, MCIMR returns an explanation of size $\leq k$. It may return an explanation of size $l<k$ if the responsibility of the $l+1$-th attribute is $\approx 0$. We report that in all cases, the size of the explanations was no bigger than 3. Thus, $k$ has almost no effect on running times, as the algorithms terminate after 4 iterations.

### D. Extensions (Q4)

We demonstrate the effectiveness of the Top-K unexplained groups algorithm by focusing on SO $Q_1$, setting $\tau > 0.2$. The top-5 largest unexplained data groups are given in Table IV. Observe that economy-related attributes (e.g., GDP, HDI) of selected data groups are internally consistent (e.g., the HDI of countries in Europe is similar). Thus, it makes sense that the explanation for SO $Q_1$ ({HDI, GINI}) will not be a satisfactory explanation for these data groups. Indeed, the explanation for the top-1 unexplained group (SO $Q_3$) is different from the one found for all countries. We ran this algorithm over all other queries. The average execution time is $4.4s$. This demonstrates the ability of our algorithm to efficiently identify data subgroups that are likely to be of interest to users.

## VI. RELATED WORK

*Results Explanations*. Methods explaining why data is missing or mistakenly included in query results have been studied in [63]–[66]. Explanations for unexpected query results have been presented in [67], [68]. Those works are orthogonal to our work, as we aim to explain unexpected correlations. Another line of work provides explanations on how a query result was derived by analyzing its provenance and pointing out tuples that significantly affect the results [69]–[71]. Those methods are designed to generate tuple-level explanations and not attribute-level explanations that are required for unearthing correlations. Another type of explanation for query results is a set of patterns that are shared by one (group of) tuple but not by another (group of) tuple [11]–[15]. However, those works do not account for correlations among attributes.

We share with [12] the motivation for considering explanations that are not solely drawn from the input table. [12] presented CajaDE, a system that generates query results explanations based on information from tables related to the table accessed by the query. However, related tables often do not exist. Moreover, their explanations are *independent of the outcome*. Thus, even if CajaDE is given with the extracted attributes, it may generate explanations that are irrelevant to the correlation between the exposure and outcome.

*Causal Discovery*. While methods for identifying confounding variables through causal models using, for example,

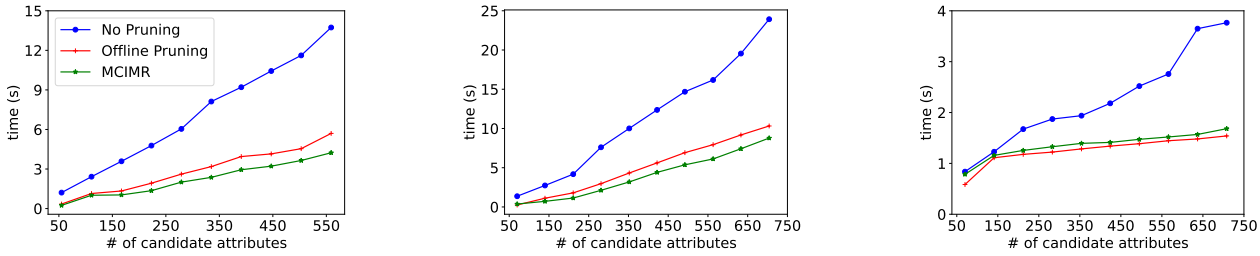



Figure 4: Running times as a function of the number of candidate attributes.

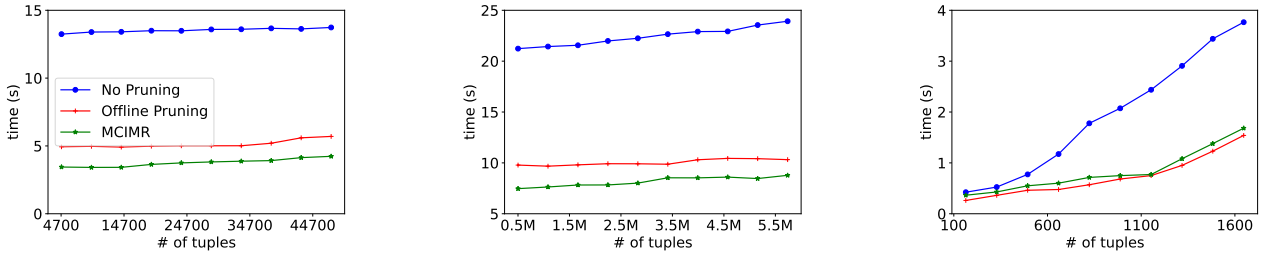



Figure 5: Running times as a function of the number of rows in the dataset.

backdoor and front-door criteria are well understood, they all rely on the availability of causal models from background knowledge [3]. However, in practice, causal models are often not available. Specifically in our framework, in which we dynamically integrate data with external sources and augment it with potentially hundreds of attributes, obtaining causal models is impractical. An alternative would be to use existing methods for automatic discovery of causal DAGs. However, it is in principle impossible to fully discover causal models [72]–[74]. Furthermore, these existing DAG discovery methods are generally intractable and do not scale in our setting. Our approach can facilitate causal discovery by providing the analyst with a set of potential confounding variables that may explain the observed correlation, even in situations where there is not enough information available to establish causality.

HypDB [1] aims to identify the direct causes of the exposure $T$ and adjust for them in order to eliminate confounding bias. In this sense, it seeks to identify the most relevant attributes to $T$ and ignores the outcome $O$ altogether. However, this process has several limitations: (1) the parents of $T$ can only be discovered from data under very strong assumptions about the underlying causal model that are often impractical, e.g., the parents should not be directly influenced by each other. It is generally accepted that the process of discovering confounding variables should rely on background knowledge and cannot be fully automated. (2) it only works if all parents of $T$ are observed in the data; (3) the proposed algorithm for parent discovery is computationally prohibitive. In contrast, our work does not claim to discover the causal relationships but rather aims to discover potential confounding attributes that can explain the observed correlation using an algorithm that simultaneously consider both $T$ and $O$ and is tractable. This could facilitate the process of identifying confounding variables for a more thorough causal analysis.

*Dataset Discovery*. Given an input dataset, dataset discovery methods find related tables that can be integrated via join or union operations. Existing methods estimate how joinable or unionable two datasets are [43], [44], [75], [76]. Other works focused on automating the data augmentation task to discover relevant features for ML models [77]. While these works focus on finding datasets that are joinable or unionable, we aim to find unobserved attributes that explain unexpected correlations. Recent work proposed solutions to discover datasets that can be joined with an input dataset and contain a column that is correlated with a target column [26], [45]. Such techniques can be integrated into our system for extracting candidate attributes from tabular data. We focus on finding attributes that minimize the partial correlation between two columns rather than finding columns that are correlated with a target column. Thus, future work will extend these techniques to support our goal.

*Feature Selection*. The CORRELATION-EXPLANATION problem is related to the well-studied Feature Selection (FS) problem [18], [27], [78], which aims to eliminate redundant or irrelevant variables from input data in order to reduce computational cost, improve understanding of the data, and increase prediction accuracy [27]. In this sense, FS algorithms can be seen as methods that identify the most relevant attributes to the outcome $O$. However, the CORRELATION-EXPLANATION problem is conceptually different, as it seeks to discover a minimal set of attributes that can explain the observed correlation between $O$ and $T$, and therefore must consider both attributes at the same time. The closest to our problem is a line of work using information-theoretic methods for FS [18], such as the MRMR algorithm [52], which selects features based on Max-Relevance and Min-Redundancy criteria. However, the main difference is that in MCIMR we consider the relevance

for the association of $T$ and $O$, whereas MRMR considers the relevance to the target attribute $O$ only. We thus define the min-conditional-mutual-information (CMI) criterion to account for the contribution of attributes to explaining the *relationship* between $T$ and $O$. Another key difference is the stopping condition: while in MRMR the size $k$ of the selected feature set is determined using the underlying learning model, in MCIMR we set $k$ using responsibility scores.

## VII. Conclusion and Limitations

This paper presented the Correlation-Explanation problem, whose goal is to identify uncontrolled confounding attributes that explain unexpected correlations observed in query results. When interpreting the explanations generated by our system, it is important to consider the following limitations: First, the quality of the generated explanation may be affected by factors such as the quality of extracted data (e.g., incorrect values) and the quality of black-box components (e.g., the entity linker, the regression model used for computing weights). Second, the generated explanations may not be complete, meaning that other unobserved confounding attributes were not extracted. Finally, since we only measure correlations, the generated explanations consist of potential confounders and may include attributes that are not actually confounding attributes. Therefore, it is up to the analysts to interpret the causal relationships among the selected attributes, the exposure, and the outcome.

## References


[1] B. Salimi, J. Gehrke, and D. Suciu, "Bias in olap queries: Detection, explanation, and removal," in *Proceedings of the 2018 International Conference on Management of Data*, 2018, pp. 1021–1035.

[2] Y. Lin, B. Youngmann, Y. Moskovitch, H. Jagadish, and T. Milo, "On detecting cherry-picked generalizations," *Proceedings of the VLDB Endowment*, vol. 15, no. 1, pp. 59–71, 2021.

[3] J. Pearl, *Causality*. Cambridge university press, 2009.

[4] M. A. Pourhoseingholi, A. R. Baghestani, and M. Vahedi, "How to control confounding effects by statistical analysis," *Gastroenterology and hepatology from bed to bench*, vol. 5, no. 2, p. 79, 2012.

[5] A. K. Upadhyay and S. Shukla, "Correlation study to identify the factors affecting covid-19 case fatality rates in india," *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, vol. 15, no. 3, pp. 993–999, 2021.

[6] A. Kaklauskas, V. Milevicius, and L. Kaklauskiene, "Effects of country success on covid-19 cumulative cases and excess deaths in 169 countries," *Ecological indicators*, p. 108703, 2022.

[7] F. Akrami, M. S. Saeef, Q. Zhang, W. Hu, and C. Li, "Realistic re-evaluation of knowledge graph completion methods: An experimental study," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 1995–2010.

[8] W. Zheng, J. X. Yu, L. Zou, and H. Cheng, "Question answering over knowledge graphs: question understanding via template decomposition," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1373–1386, 2018.

[9] C. De Sa, A. Ratner, C. Ré, J. Shin, F. Wang, S. Wu, and C. Zhang, "Deepdive: Declarative knowledge base construction," *ACM SIGMOD Record*, vol. 45, no. 1, pp. 60–67, 2016.

[10] X. Chen, S. Jia, and Y. Xiang, "A review: Knowledge reasoning over knowledge graph," *Expert Systems with Applications*, vol. 141, p. 112948, 2020.

[11] K. El Gebaly, P. Agrawal, L. Golab, F. Korn, and D. Srivastava, "Interpretable and informative explanations of outcomes," *Proceedings of the VLDB Endowment*, vol. 8, no. 1, pp. 61–72, 2014.

[12] C. Li, Z. Miao, Q. Zeng, B. Glavic, and S. Roy, "Putting things into context: Rich explanations for query answers using join graphs," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1051–1063.

[13] S. Roy, L. Orr, and D. Suciu, "Explaining query answers with explanation-ready databases," *Proceedings of the VLDB Endowment*, vol. 9, no. 4, pp. 348–359, 2015.

[14] S. Roy and D. Suciu, "A formal approach to finding explanations for database queries," in *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, 2014, pp. 1579–1590.

[15] E. Wu and S. Madden, "Scorpion: Explaining away outliers in aggregate queries," 2013.

[16] S. R. Seaman and I. R. White, "Review of inverse probability weighting for dealing with missing data," *Statistical methods in medical research*, vol. 22, no. 3, pp. 278–295, 2013.

[17] K. Mohan, F. Thoemmes, and J. Pearl, "Estimation with incomplete data: The linear case," in *Proceedings of the International Joint Conferences on Artificial Intelligence Organization*, 2018.

[18] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–45, 2017.

[19] "Dbpedia," https://www.dbpedia.org/, 2022.

[20] "2021 stackoverflow developer survey," 2021, https://insights.stackoverflow.com/survey/2021.

[21] A. Santos, A. R. Colaço, A. B. Nielsen, L. Niu, M. Strauss, P. E. Geyer, F. Coscia, N. J. W. Albrechtsen, F. Mundt, L. J. Jensen *et al.*, "A knowledge graph to interpret clinical proteomics data," *Nature Biotechnology*, pp. 1–11, 2022.

[22] S. K. Mohamed, V. Nováček, and A. Nounu, "Discovering protein drug targets using knowledge graph embeddings," *Bioinformatics*, vol. 36, no. 2, pp. 603–610, 2020.

[23] "Wikidata," https://www.wikidata.org/wiki/Wikidata:Main_Page, 2022.

[24] T. Rebele, F. Suchanek, J. Hoffart, J. Biega, E. Kuzey, and G. Weikum, "Yago: A multilingual knowledge base from wikipedia, wordnet, and geonames," in *International semantic web conference*. Springer, 2016, pp. 177–185.

[25] F. E. Croxton and D. J. Cowden, "Applied general statistics." 1939.

[26] M. Esmailoghli, J.-A. Quiané-Ruiz, and Z. Abedjan, "Cocoa: Correlation coefficient-aware data augmentation." in *EDBT*, 2021, pp. 331–336.

[27] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[28] J. Zhao, Y. Zhou, X. Zhang, and L. Chen, "Part mutual information for quantifying direct associations in networks," *Proceedings of the National Academy of Sciences*, vol. 113, no. 18, pp. 5130–5135, 2016.

[29] R. Pradhan, J. Zhu, B. Glavic, and B. Salimi, "Interpretable data-based explanations for fairness debugging," *arXiv preprint arXiv:2112.09745*, 2021.

[30] B. Lockhart, J. Peng, W. Wu, J. Wang, and E. Wu, "Explaining inference queries with bayesian optimization," *Proc. VLDB Endow.*, vol. 14, no. 11, pp. 2576–2585, 2021.

[31] L. S. Shapley, "A value for n-person games," *Classics in game theory*, vol. 69, 1997.

[32] S. Davidson, D. Deutch, N. Frost, B. Kimelfeld, O. Koren, and M. Monet, "Shapgraph: An holistic view of explanations through provenance graphs and shapley values," in *Proceedings of the 2022 International Conference on Management of Data*, 2022, pp. 2373–2376.

[33] D. Deutch, N. Frost, B. Kimelfeld, and M. Monet, "Computing the shapley value of facts in query answering," in *Proceedings of the 2022 International Conference on Management of Data*, 2022, pp. 1570–1583.

[34] E. Livshits and B. Kimelfeld, "The shapley value of inconsistency measures for functional dependencies," *arXiv preprint arXiv:2009.13819*, 2020.

[35] D. Deutch, N. Frost, A. Gilad, and O. Sheffer, "Explanations for data repair through shapley values," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 362–371.

[36] S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 56–67, 2020.

[37] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[38] A. Lahiri, K. Alipour, E. Adeli, and B. Salimi, "Combining counterfactuals with shapley values to explain image models," *arXiv preprint arXiv:2206.07087*, 2022.

[39] T. Lombrozo, "Simplicity and probability in causal explanation," *Cognitive psychology*, vol. 55, no. 3, pp. 232–257, 2007.

[40] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov, "Algorithms for large scale markov blanket discovery." in *FLAIRS conference*, vol. 2. St. Augustine, FL, 2003, pp. 376–380.

[41] L. E. Brown and I. Tsamardinos, "Markov blanket-based variable selection in feature space," 2008.

[42] Y. Zhang and Z. Ives, "Finding related tables in data lakes for interactive data science," in *SIGMOD*, pp. 1951–1966.

[43] F. Nargesian, E. Zhu, K. Q. Pu, and R. J. Miller, "Table union search on open data," *Proceedings of the VLDB Endowment*, vol. 11, no. 7, pp. 813–825, 2018.

[44] E. Zhu, D. Deng, F. Nargesian, and R. J. Miller, "Josie: Overlap set similarity search for finding joinable tables in data lakes," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 847–864.

[45] A. Santos, A. Bessa, F. Chirigati, C. Musco, and J. Freire, "Correlation sketches for approximate join-correlation queries," in *Proceedings of the 2021 International Conference on Management of Data*, 2021, pp. 1531–1544.

[46] A. Parravicini, R. Patra, D. B. Bartolini, and M. D. Santambrogio, "Fast and accurate entity linking via graph embedding," in *Proceedings of the 2nd Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, 2019, pp. 1–9.

[47] G. Zhu and C. A. Iglesias, "Exploiting semantic similarity for named entity disambiguation in knowledge graphs," *Expert Systems with Applications*, vol. 101, pp. 8–24, 2018.

[48] R. Fagin, A. O. Mendelzon, and J. D. Ullman, "A simplied universal relation assumption and its properties," *ACM Transactions on Database Systems (TODS)*, vol. 7, no. 3, pp. 343–360, 1982.

[49] B. Efron, "Missing data, imputation, and the bootstrap," *Journal of the American Statistical Association*, vol. 89, no. 426, pp. 463–475, 1994.

[50] J. D. Kang and J. L. Schafer, "Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data," *Statistical science*, vol. 22, no. 4, pp. 523–539, 2007.

[51] D. Hinkley, "Transformation diagnostics for linear models," *Biometrika*, vol. 72, no. 3, pp. 487–496, 1985.

[52] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[53] A. Asudeh, Z. Jin, and H. V. Jagadish, "Assessing and remedying coverage for a given dataset," in *35th IEEE International Conference on Data Engineering, ICDE 2019, Macao, China, April 8-11, 2019*. IEEE, 2019, pp. 554–565. [Online]. Available: https://doi.org/10.1109/ICDE.2019.00056

[54] "Technical report," https://github.com/ResultsExplanations/ExplanationsFromKG, 2022.

[55] "Pyitlib library," https://pypi.org/project/pyitlib/, 2022.

[56] "Stack overfow developer survey," https://insights.stackoverflow.com/survey, 2021.

[57] "Covid-19 dataset," https://www.kaggle.com/imdevskp/corona-virus-report?select=usa_county_wise.csv, 2020.

[58] "Flights delay dataset," https://www.kaggle.com/usdot/flight-delays?select=flights.csv, 2020.

[59] "Forbes dataset," https://www.kaggle.com/datasets/slayomer/forbes-celebrity-100-since-2005, 2021.

[60] "The vanity fair," 2018, https://www.vanityfair.com/hollywood/2018/02/hollywood-movie-salaries-wage-gap-equality.

[61] "The usa today," 2019, https://www.usatoday.com/story/travel/airline-news/2022/06/19/why-us-flights-canceled-delayed-sunday/7677552001/.

[62] Z. Zhang, "Missing data imputation: focusing on single imputation," *Annals of translational medicine*, vol. 4, no. 1, 2016.

[63] N. Bidoit, M. Herschel, and K. Tzompanaki, "Query-based why-not provenance with nedexplain," in *Extending database technology (EDBT)*, 2014.

[64] A. Chapman and H. Jagadish, "Why not?" in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, 2009, pp. 523–534.

[65] S. Lee, B. Ludäscher, and B. Glavic, "Approximate summaries for why and why-not provenance (extended version)," *arXiv preprint arXiv:2002.00084*, 2020.

[66] B. ten Cate, C. Civili, E. Sherkhonov, and W.-C. Tan, "High-level why-not explanations using ontologies," in *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, 2015, pp. 31–43.

[67] A. Bessa, J. Freire, T. Dasu, and D. Srivastava, "Effective discovery of meaningful outlier relationships," *ACM Transactions on Data Science*, vol. 1, no. 2, pp. 1–33, 2020.

[68] Z. Miao, Q. Zeng, B. Glavic, and S. Roy, "Going beyond provenance: Explaining query answers with pattern-based counterbalances," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 485–502.

[69] T. Milo, Y. Moskovitch, and B. Youngmann, "Contribution maximization in probabilistic datalog," in *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 2020, pp. 817–828.

[70] A. Meliou, W. Gatterbauer, K. F. Moore, and D. Suciu, "The complexity of causality and responsibility for query answers and non-answers," *arXiv preprint arXiv:1009.2021*, 2010.

[71] ——, "Why so? or why no? functional causality for explaining query answers," *arXiv preprint arXiv:0912.5340*, 2009.

[72] P. Spirtes *et al.*, *Causation, prediction, and search*. MIT press, 2000.

[73] S. Shimizu, P. O. Hoyer, A. Hyvärinen, A. Kerminen, and M. Jordan, "A linear non-gaussian acyclic model for causal discovery." *Journal of Machine Learning Research*, vol. 7, no. 10, 2006.

[74] D. Chickering, "Optimal structure identification with greedy search," *JMLR*, vol. 3, no. Nov, pp. 507–554, 2002.

[75] Y. Yang, Y. Zhang, W. Zhang, and Z. Huang, "Gb-kmv: An augmented kmv sketch for approximate containment similarity search," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 458–469.

[76] E. Zhu, F. Nargesian, K. Q. Pu, and R. J. Miller, "Lsh ensemble: Internet-scale domain search," *arXiv preprint arXiv:1603.07410*, 2016.

[77] N. Chepurko, R. Marcus, E. Zgraggen, R. C. Fernandez, T. Kraska, and D. Karger, "Arda: automatic relational data augmentation for machine learning," *arXiv preprint arXiv:2003.09758*, 2020.

[78] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[79] "Daxx," 2022, https://www.daxx.com/blog/development-trends/it-salaries-software-developer-trends.

[80] "Tech gig," 2022, https://content.techgig.com/career-advice/what-is-the-average-salary-of-software-engineers-in-different-countries/articleshow/91121900.cms.

[81] "Bts gov," 2022, https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations.

[82] R. Pascoal and H. Rocha, "Population density impact on covid-19 mortality rate: A multifractal analysis using french data," *Physica A: Statistical Mechanics and its Applications*, vol. 593, p. 126979, 2022.

[83] "The guardian," 2019, https://www.theguardian.com/world/2019/sep/15/hollywoods-gender-pay-gap-revealed-male-stars-earn-1m-more-per-film-than-women.

[84] "Go banking rates," 2022, https://www.gobankingrates.com/money/jobs/how-much-do-actors-make/.

[85] "Climb the ladder," 2022, https://climbtheladder.com/producer-salary/.


## APPENDIX

### A. Missing Proofs

In this part, we provide missing proofs.

*Proof of Proposition III.1.* For any two random variables, if $X \perp\!\!\!\perp Y$ we have: $H(Y|X) = H(Y)$. This can be generalized to conditional independence as well. We get:

$$I(O;T|E,R_E=1,C) =$$
$$H(O|E,R_E=1,C) - H(O|T,E,R_E=1,C) =$$
$$H(O|E,C) - H(O|T,E,C) = I(O;T|E,C)$$

□

*Proof of Proposition III.2.* We have:

$$I(E_i; E_j | R_{E_i} = 1, R_{E_j} = 1) = H(E_i | R_{E_i} = 1, R_{E_j} = 1) - H(E_i | E_j, R_{E_i} = 1, R_{E_j} = 1) = H(E_i) - H(E_i | E_j) = I(E_i; E_j)$$

□

In what follows, to ease the exposition, we assume that there is no WHERE clause in the query, i.e., $C = \emptyset$. Our results also hold for cases where $C$ is not empty.

*Proof of Theorem IV.1.* Recall that by definition of the algorithm, we assume that $\boldsymbol{E_{k-1}}$, i.e., the set of $k-1$ attributes, has already been obtained, and thus $\boldsymbol{E_{k-1}}, O$, and $T$ are fixed when selecting the $k$-th attribute. The goal is to select the optimal $k$-th attribute to be added, $E_k$, from $\mathcal{D} \setminus \boldsymbol{E_{k-1}}$.

By the definition of conditional mutual information, we have:

$$I(O; T | \boldsymbol{E_{k-1}}, E_k) = I(O; T | \boldsymbol{E_k}) = H(O; \boldsymbol{E_k}) + H(T; \boldsymbol{E_k}) - H(O; T; \boldsymbol{E_k}) - H(\boldsymbol{E_k})$$

We use the following definition of [52] for the attributes $E_1, \ldots, E_k$: $J(\boldsymbol{E_k}) = J(E_1, \ldots, E_k)$ where:

$$J(\boldsymbol{E_k}) = \sum \ldots \sum Pr(E_1, \ldots, E_k) \frac{Pr(E_1, \ldots, E_k)}{Pr(E_1) \cdot \ldots \cdot Pr(E_k)}$$

Similarly, we have:

$$J(O, T, \boldsymbol{E_k}) = \sum \ldots \sum Pr(E_1, \ldots, E_k, O, T) \frac{Pr(E_1, \ldots E_k, O, T)}{Pr(E_1) \ldots Pr(E_k) Pr(O) Pr(T)}$$

$$J(X, \boldsymbol{E_k}) = \sum \ldots \sum Pr(E_1, \ldots E_k, X) \frac{Pr(E_1, \ldots E_k, O)}{Pr(E_1) \cdot \ldots Pr(E_k) Pr(X)}$$

We can derive:

$$H(O; \boldsymbol{E_k}) + H(T; \boldsymbol{E_k}) - H(O; T; \boldsymbol{E_k}) - H(\boldsymbol{E_k}) =$$

$$H(O) + \sum_{i=1}^{k} H(E_i) - J(O, \boldsymbol{E_k}) + H(T) + \sum_{i=1}^{k} H(E_i) - J(T, \boldsymbol{E_k}) - H(O) - H(T) - \sum_{i=1}^{k} H(E_i) + J(O, T, \boldsymbol{E_k}) - \sum_{i=1}^{k} H(E_i) + J(\boldsymbol{E_k}) =$$

$$J(O, T, \boldsymbol{E_k}) + J(\boldsymbol{E_k}) - J(O, \boldsymbol{E_k}) - J(T, \boldsymbol{E_k})$$

Thus we consider the following expression:

$$J(O, T, \boldsymbol{E_k}) + J(\boldsymbol{E_k}) - J(O, \boldsymbol{E_k}) - J(T, \boldsymbol{E_k}) \quad (6)$$

We argue that (6) is minimized when the $k$-th attribute minimizes the Min-CIM and Min-Redundancy criteria.

As stated in [52], the maximum of $J(O, \boldsymbol{E_k})$ is attained when all variables are maximally dependent. When $O, \boldsymbol{E_{k-1}}$ are fixed, this indicates that the attribute $E_k$ should have the maximal dependency to $O$. In this case, we get that $J(O, T, \boldsymbol{E_k}) = J(T, \boldsymbol{E_k})$. Note that when the dependency of $O$ or $T$ in $E_k$ increases, the conditional mutual information $I(O; T | \boldsymbol{E_k})$ decreases. This is the Min-CIM criterion.

Moreover, as noted in [52], the minimum of $J(\boldsymbol{E_k})$ is attained when the attributes $E_1, \ldots, E_k$ are independent of each other. As all the attributes $E_1, \ldots, E_{k-1}$ are fixed at this point, this pair-wise independence condition means that the mutual information between the attribute $E_k$ and any other attribute $E_i$ is minimized. This is the Min-Redundancy criterion.

Thus, we get that the overall expression in (6) is minimized (i.e., $J(O, \boldsymbol{E_k})$ is maximized, $J(O, \boldsymbol{E_k}, T) = J(T, \boldsymbol{E_k})$, and $J(\boldsymbol{E_k})$ is minimized) when we are minimizing the Min-CIM and Min-Redundancy criteria.

□

*Proof of Lemma IV.1.* First, since $(O \perp\!\!\!\perp E_{k+1} | \boldsymbol{E_k})$ we have: $I(O, E_{k+1} | \boldsymbol{E_k}) = 0$. We get:

$$I(O; T | \boldsymbol{E_k}) - I(O; T | \boldsymbol{E_k}, E_{k+1}) = H(O | \boldsymbol{E_k}) - H(O | T, \boldsymbol{E_k}) - H(O | \boldsymbol{E_k}, E_{k+1}) + H(O | T, \boldsymbol{E_k}, E_{k+1})$$

Since $H(O | \boldsymbol{E_k}) - H(O | E_{k+1}, \boldsymbol{E_k}) = H(O | \boldsymbol{E_k}) - H(O | \boldsymbol{E_k}) = 0$, we get:

$$I(O; T | \boldsymbol{E_k}) - I(O; T | \boldsymbol{E_k}, E_{k+1}) = H(O | T, \boldsymbol{E_k}, E_{k+1}) - H(O | T, \boldsymbol{E_k}) \leq 0$$

For the last inequality we used the fact that for every three ransom variables $X, Y, X$: $H(X|Y) \leq H(X|Y, Z)$, since adding more conditions can only reduce the uncertainty of $X$.

We get that the numerator of the responsibility score of $E_{k_1}$ is $\leq 0$, and thus $Resp(E_{k+1}) \leq 0$ □

**Proposition A.1.** *The time complexity of the incremental MCIMR algorithm is $O(k|\mathcal{A}|)$.*

*Proof.* At each iteration, the MCIMR algorithm selects a new attribute to be added based on the condition defined in Equation (5). In the worst case, it examines all attributes in $\mathcal{A}$. Since it stops after at most $k$ iterations, we get that the time complexity is $O(k|\mathcal{A}|)$. □

We next prove that logical dependencies can lead to a misleading conclusion that we found a confounding attribute.

**Lemma A.1.** *If for an attribute $E$ we have: $FD : E \Rightarrow T$ then we get $I(O; T | E, C) = 0$.*

*Proof.* If for an attribute $E$ we have: $FD : E \Rightarrow T$ then we have $H(T|E) \approx 0$. We get: $I(O; T | E, C) = H(O | E, C) - H(O | T, E, C)$. But since $T$ and $E$ are dependent, we get: $H(O | T, E, C) \approx H(O | E, C)$ and thus $I(O; T | E, C) = 0$. □

The lemma also holds for the case that the attribute $E$ logically depends on the outcome $O$.

**Relevance Test**: Given a candidate attribute $E$, if $(O \perp\!\!\!\perp E|C)$ and $(O \perp\!\!\!\perp E|C,T)$ we get that $H(O|E,C)=H(O|C)$ and $H(O|T,E,C)=H(O|T,C)$. Thus:

$$I(O;T|E,C)=H(O|E,C)-H(O|T,E,C)= \\ H(O|C)-H(O|T,C)=I(O;T|C)$$

That means that the individual explanation power of $E$ is low, and thus it can be dropped as we assume $E$ cannot be a part of the optimal explanation.

### B. Experiments

*Omitted Baselines:* : We also consider the explanations generated by the following baselines:

**LR**: This baseline employs the OLS method to estimate the coefficients of a linear regression describing the relationship between the outcome and the candidate attributes. The explanations are defined as the top-$k$ attributes with the highest coefficients (s.t. the $p$ value is $<.05$). Note that Pearson's $r$ is the standardized slope of LR and thus can be viewed as part of our competing baselines. *This baseline demonstrates that considering explanations that have only a linear correlation with the outcome is too restrictive*.

**CajaDE [12]**: a system that generates query results explanations based on augmented provenance information. The explanations are defined to be the top-$k$ patterns with the highest F-scores, as defined in [12]. Since CajaDE only provides explanations for the difference between two (groups of) tuples in a query answer, here we picked two tuples from the query results (one from the bottom 10% percent for the value of the outcome, and the second from the top 10% with the highest outcome value) and generated explanations for the difference between them.

Both these baselines generated explanations that were considered to be not convincing by the subjects. For LR, in many cases, it failed to generate explanations, as there were no attributes with low enough p-values. Even when it succeeded, the subjects found them to be not convincing. The reason is that LR focuses on finding linear correlations. The reason for that CajaDE explanations are a set of patterns that are unevenly distributed among groups in the query results, which are independent of the outcome variable. Thus, it cannot generate explanations that explain the correlation between $T$ and $O$. *This demonstrates that CajaDE is orthogonal to our approach.*

*Explanation quality:* Next, we provide references supporting the explanations generated by MESA. These in-domain findings serve as "domain-expert" explanations.

**SO Q1**: It was shown in [20] that there is a correlation between the developers salary and countries' economies. In [79], it was also shown that the countries with the highest salary for developers are countries with a relatively high HDI (e.g., the US, Switzerland, Denmark).

**SO Q2 + Q3**: It was mentioned in [80] that countries that have a scarcity of software graduates tend to offer higher salaries than countries like India which produce hundred of thousands developers every year. This suggests that besides the economy of a country (resp., continent), the population size is also a factor that affects the average salary of developers.

**Flights Q1**: It was stated in [61] that weather is one of the top reasons for flights delay in the US.

**Flights Q2-Q4**: It was mentioned in https://www.bts.gov/topics/airlines-and-airports/understanding-reporting-causes-flight-delays-and-cancellations that besides weather conditions, main causes for flights delay in the US are heavy traffic volume, and air traffic control. Those two factors are highly correlated with population size. In bigger and more dense area, the air traffic increases.

**Flights Q5**: It was mentioned in [81] that a main cause of the delay of flights in the US is the airline's control (e.g., maintenance or crew problems).

**Covid Q1**: It was shown that there is a correlation between countries' economies and Covid-19 death rate [5], [6].

**Covid Q2-Q3**: It was stated in [82] that population density impact on COVID-19 mortality rate.

**Forbes Q1**: It was shown in [83] that there is a gender pay gap for actors in Hollywood. Thus, it make sense that gender is a factor affecting the average salary of actors. It was also stated in [84] that actors get paid according to their experience, which is reflected in their net worth.

**Forbes Q2**: It was mentioned in [85] that what affects directors and producers salary is their level of experience (which is reflected in the awards and net worth attributes).

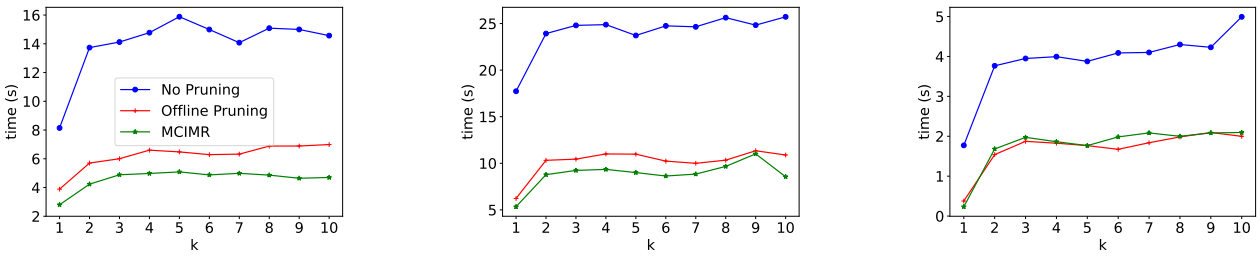
**Forbes Q3**: It was stated in [85] that very often professional athletes salaries are performance-based. The performance quality is reflected in the Cups and Draft Pick attributes (for tennis, basketball and football athletes, which are the majority of athletes in the Forbes dataset).

We next present additional experiments.

*Combining cardinality and explanatory power*. In the definition of the CORRELATION-EXPLANATION problem, we use multiplication to combine cardinality and explanatory power. We used a weighted average instead and examined the optimal results in this case. Since our algorithm is invariant to this definition, we obtained the same results. As for Brute Force, we report that in 5 out of 6 queries, we obtained the same explanations. This indicates that the choice of aggregation function has little effect on quality.

*Impact of pruning*. We next examine how useful were our pruning techniques. **Offline Pruning**. We found that our two offline pruning optimizations to be highly useful: On average, we dropped 41%, 59%, 45%, and 73% of the extracted attributes, in the SO, Flights, Covid-19, and Forbes dataset, resp. **Online Pruning**. At query time, we filter the extracted attributes using the logical dependency and the low relevance techniques. Not surprisingly, as most irrelevant attributes were already dropped in the offline phase, we dropped many fewer attributes at this phase. On average, we dropped 14%, 6%, 11% and 3% of the remaining attributes, in SO, Flights, Covid-19, and Forbes, resp.

*Entity linker*. Many of the missing values were caused by an unsuccessful matching of values from the table to their




SO Flights Forbes

Figure 6: Running times as a function of the bound on the explanation size.

entities in the KG. For example, in SO, for some developers, their origin country is `Russian Federation`. However, the corresponding entity in DBpedia is called `Russia`. We thus failed to extract the properties of this country. In other cases, the values that appear in the tables were ambiguous, and thus we failed to match them to DBpedia entities. For example, in Forbes, one of the athletes is called `Ronaldo`. SpaCy entity linker could not decide whether to link this value to the entity `Ronaldo Luís Nazário de Lima` (Brazilian footballer) or to `Cristiano Ronaldo` (Portuguese footballer).

*Computing Weights*. Recall that we use logistic regression models to compute weights to overcome selection bias. We report that the average accuracy of the models was $91.6\%$, indicating that these models' performances were largely satisfactory.

*Multi-Hops*. We examine the effect of extracting attributes following more than one hop in the KG. We report that in the vast majority of cases, MESA's explanations were unaffected. Namely, almost all attributes extracted from 2 or more hops were found to be irrelevant (and were pruned). In some cases, we found at most one more attribute that was included in the explanations. For example, in Forbes $Q_1$, an attribute representing the average budget of the films played by actors (attribute extracted from 2-hops) was included in the explanation. In all cases, no attributes from 3 or more hops was considered to be relevant. Further, since the number of candidate attributes was increased (in 145%, on average), running times were increased (by up to 15 seconds). This indicates that most of the relevant information can be found in the first hop. Future research will predict which paths in the KG may lead to relevant attributes.

*Missing Values*. On average, the percentage of missing values in extracted attributes is 37%, 42%, 45%, and 73% in Covid-19, SO, Flights and Forbes, resp. The high prevalence of missing values in Forbes is because DBpedia uses different attributes to describe a person. For example, for athletes, facts regarding their height and weight are typically present. However, these properties can rarely be found in actors. In Covid-19, SO, Flights, and Forbes, the percentage of attributes with selection bias is 13%, 14%, 24%, and 29%, resp. *This verifies that selection bias exists in attributes extracted from KGs and thus should be appropriately handled.*

Last, Figure 6 depicts running times as a function of the bound on the explanation size. Observe that, as mentioned in Section V, the bound on the explanation size has little effect on running times.