Causal DAG Summarization [Technical Report]

Anonymous Author(s)

ABSTRACT

Causal inference aids researchers in discovering cause-and-effect relationships, leading to scientific insights. Accurate causal estimation requires identifying confounding variables to avoid false discoveries. Pearl's causal model uses causal DAGs to identify confounding variables, but incorrect DAGs can lead to unreliable causal conclusions. However, for high dimensional data, the causal DAGs are often too complex beyond human verifiability. Graph summarization is a logical step, but current methods are not designed to preserve critical causal information. This paper addresses these challenges by proposing a causal graph summarization objective that balances graph simplification for better understanding with retaining essential causal information for reliable inference. Specifically, we formulate the causal DAG summarization problem, adding size and semantic constraints to improve the summary DAG's intelligibility, while aiming to preserve, to the greatest extent, the most crucial causal information encoded in the input causal DAG. We developed an efficient greedy algorithm and extended Pearl's framework to allow reliable causal inference directly over summary causal DAGs. Experimenting with five real-life datasets, we compared our algorithm to three existing solutions, showing its effectiveness in handling high-dimensional data and its ability to generate summary DAGs that ensure reliable causal inference.

1 INTRODUCTION

Causal inference sits at the heart of informed decision-making across various disciplines, offering analysts tools to unravel complex cause-effect relationships that govern our world. From policies shaping economies [90] to social structures influencing human behavior [25], from groundbreaking medical interventions [35] to intricacies of environmental changes [73], causal inference provides a critical lens through which insights are derived. Beyond its foundational applications in economics, sociology, and medicine, causal inference is becoming increasingly critical in ML. It underpins efforts to ensure algorithmic fairness [73], reduce biases [102], improve AI interpretability [24], and handle distribution shifts [46, 86]. Causal inference has also become a major theme in recent data management research [70, 73, 98]. It has been shown to be useful for various data management tasks, including query result explanation [44, 72, 96], and hypothetical reasoning [23].

Causal inference critically hinges on background knowledge and assumptions that data alone cannot verify [61]. *Causal DAGs* provide a fundamental way to encode this knowledge, automating the causal inference process [61]. They provide an explicit representation of assumed causal relationships, enabling a systematic approach to reasoning about the effects of interventions and facilitating inference through methodologies such as Pearl's *do*-calculus [61]. This representation is crucial for identifying confounders, which are instrumental in ML for enhancing model interpretability, mitigating bias, and ensuring fairness. Causal DAGs can distill complex causal relationships into understandable formats and decompose causal effects into finer-grained impacts such as direct and indirect effects of interventions [61]. This decomposition is crucial for applications like algorithmic fairness [54, 73], where understanding nuanced pathways of influence can inform more equitable decision-making processes.

Example 1. Consider the example causal DAG for a Flights Delay dataset depicted in Fig. 1, which outlines potential causal relationships among various factors. Notably, the `Airport` influences `Dep. Delay`, albeit indirectly, via `Airport Traffic`. This illustration clarifies that `Temp.` does not directly affect `Dep. City`; rather, it is `Dep. City` that has a causal effect on `Temp`. To accurately estimate the causal effect of `Airport` on `Dep. Delay`, it is essential to account for all relevant confounders. An insufficient set of confounders may result in incorrect conclusions based on spurious correlations. In this scenario, `Dep. City` emerges as a critical confounder that needs to be adjusted for, as dictated by Pearl's backdoor criterion [61]. This example highlights the necessity of comprehending the causal structure and making appropriate adjustments for confounders to obtain reliable causal estimates. □

Access to high-quality causal models, which accurately encode background knowledge, is crucial for facilitating the diverse applications mentioned earlier. However, such models are often not available and must be constructed, typically involving *domain knowledge* [11, 91, 98] or utilizing *causal discovery* methods [15, 28, 79, 93, 103]. Despite the value of expert input, it is error-prone [57], and gathering insights from multiple experts can be time-consuming. Causal discovery methods, while useful, are limited to identifying a class of models that statistically coincide with the observed data, not a singular, definitive model [28]. This challenge is intensified by the complex and often intractable nature of causal discovery, where exact solutions are unfeasible, and approximations are necessary. Such approximations can introduce inaccuracies, frequently resulting in poor performance on real-world data [16, 32, 82].

Moreover, practical applications necessitate that datasets be thoroughly curated with all relevant confounding factors to ensure robust causal inference, a goal typically achieved through data integration [21, 74, 97, 98]. When dealing with high-dimensional data, initial models from causal discovery methods require significant human intervention for verification [11, 16, 32, 82]. Verifying and refining these models becomes increasingly challenging as data dimensionality grows [68, 87], emphasizing the essential role of analysts in adding further information. The complexity and volume of data magnify these challenges, underscoring an urgent need for novel methods to navigate this complexity and facilitate the integration of expert knowledge into the causal modeling process [58].

This paper tackles the aforementioned challenges by introducing a novel graph summarization technique tailored for causal inference. This technique aims to simplify complex causal DAGs into more manageable forms without compromising critical causal information essential for accurate analysis. By doing so, we facilitate more efficient integration of expert knowledge and streamline the refinement processes of causal models. This approach not only enhances the interpretability of causal graphs but also addresses the

significant challenge of handling high-dimensional data. In providing a method for summarizing causal DAGs, we offer new avenues for practitioners to conduct causal inference with greater ease and reliability, opening up possibilities for applications in fields where understanding causality is crucial.

Graph summarization has been extensively studied, with state-of-the-art methods designed to efficiently generate concise representations aimed at minimizing reconstruction errors [40, 95], facilitating accurate query answering [45, 80], or enhancing visualizations [33, 37]. However, existing methods prove inadequate for summarizing causal graphs, a task that demands the preservation of causal information crucial for reliable inference. This paper confronts these challenges by introducing a causal DAG summarization objective, which balances simplifying the graph for enhanced comprehensibility and retaining essential causal information. Our approach is tailored for causal inference, acknowledging the unique semantics of causal graphs and distinguishing them from other graph types like transportation or social networks. We argue that while general-purpose methods are adept at managing massive graphs, they fall short in preserving causal integrity, often resulting in summaries with self-edges, cycles, or overlooking semantic similarities among variables—making them unsuitable for causal inference. Conversely, our method ensures acyclicity, semantic comprehensibility, and direct applicability to causal inference without requiring the reconstruction of the original graph. We illustrate that with an example.

Example 2. Consider the summary graph generated by SSumM [40] for the Flights Delay causal DAG, as depicted in Fig. 2(b). SSumM is a top-performing general-purpose graph summarization method known for effectively balancing conciseness (measured by the number of bits needed to store the graph), and accuracy (measured by reconstruction error). The graph exhibits cycles and self-loops, characteristics that are incompatible with foundational principles of causal inference[1]. This shows that SSumM is not suitable for summarizing causal DAGs. Consequently, it becomes impossible to identify confounders set for accurately estimating the causal effect of `Airport` on `Dep. Delay`, given the bidirectional edge between `Airport` and `{City, Traffic}`, and the self-loop of `{City, Traffic}`. An in-depth comparison with an additional graph summarization method, k-Snap [88], is given in Section 7.4. We demonstrate that while k-Snap can be adapted to generate summary DAGs compatible with causal inference principles, it results in summary DAGs that do not optimally preserve critical causal information.

In contrast, our approach generates a summary causal DAG that faithfully preserves the causal information from the original DAG, ensuring the reliability of causal inference over the summary DAG. We also ensure that only semantically related variables are grouped, to ensure coherent and meaningful clusters. The 7-node summary DAG that maximally retains causal information is shown in Fig. 2(a). This summary graph facilitates the derivation of a reliable estimate for the causal effect of `Airport` on `Dep. Delay`. Although the identified confounders set now includes the variable `Dep. State` alongside `Dep. City` (unlike in the original DAG where only `Dep. City` was included), the estimate remains unbiased. □

[1]Similar issues were also observed in the summary graph generated by [95].

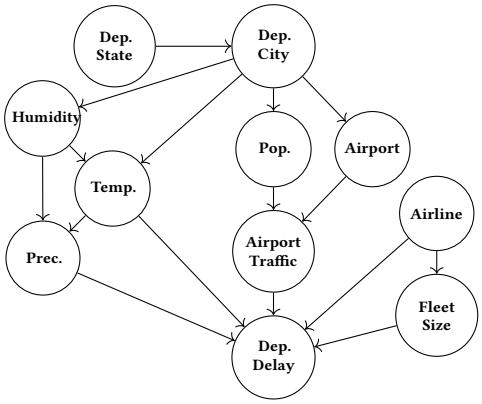
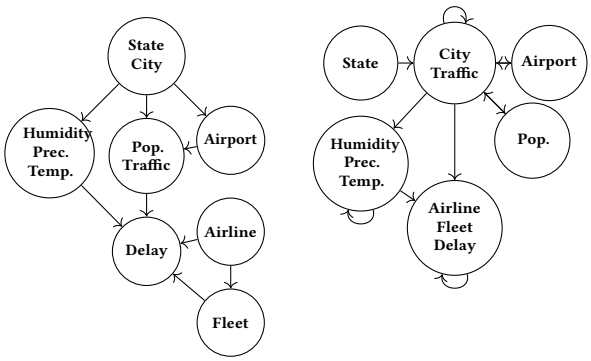


Figure 1: Example causal DAG



(a) The optimal summary DAG (b) Summary Graph By SSumM

Figure 2: 7-nodes summary graphs for the DAG in Fig. 1.

Our main contributions are summarized as follows.

Causal DAG Summarization. We introduce the problem of summarizing causal DAGs as the problem of distilling an input causal DAG into an interpretable and compact form that maintains the utility of the model for reliable causal inference (Section 3). This process necessitates preserving the causal dependency between nodes. Importantly, causal DAGs encode information through missing edges between variables, indicating no direct causal relationship and thus translating to Conditional Independence (CI) constraints. These constraints are fundamental for various applications, such as efficient inference [19, 36], algorithmic fairness [73], and missing data [52]. They are also essential for the interpretability of the causal DAG as they lead to sparser graphs [61]. Furthermore, we incorporate size and semantic constraints to enhance the intelligibility of summary graphs. The problem of causal DAG summarization is formalized as finding a summary DAG that maintains causal dependencies and CI statements while meeting these constraints. We prove that this problem is NP-hard, highlighting the computational challenge in generating summaries that strike a balance between simplification and preserving essential causal information.

Summary Causal DAGs and s-Separation. We introduce the concept of *summary causal DAGs* as a class of graphical models derived by grouping nodes within the original DAG via *node contractions*[63]. Node contraction, while inherently leading to information

loss, enables summary DAGs to compactly encapsulate various potential causal DAGs from which the summary DAG could have originated. Moving beyond the conventional concept of d-separation, we introduce s-separation and develop a sound and complete algorithm for identifying all CIs encoded by a summary DAG. This is crucial for identifying optimal summary DAGs and for utilizing them for causal inference. At the heart of this development is the link between node contractions and the addition of edges to the input causal DAG. We show that contracting nodes results in a reduction of causal information in a causal DAG, akin to adding edges. This underpins our strategy to decode from a given summary DAG, the causal information it carries, thereby extending techniques for conventional causal DAGs to summary DAGs (Section 4).

The CaGreS Algorithm. We devise an efficient, bottom-up greedy algorithm for the causal DAG summarization problem, called CaGreS. A key feature of CaGreS is its methodical approach to choosing which node pair to merge, a critical decision affecting the summary DAG quality. This process is informed by the connection between node contraction and the addition of edges to the input causal DAG, prioritizing node pairs that add the fewest edges upon merging. Additionally, CaGreS incorporates several optimizations, including caching mechanisms, making it a practical tool for generating summary causal DAGs (Section 5).

Causal Inference over Summary Causal DAGs. We illustrate that summary causal DAGs not only offer an interpretable summary but also can be directly utilized for causal inference. In this direction, we establish that Pearl's *do-calculus* framework [61], which provides a set of sound and complete rules for reasoning about the effects of interventions using causal DAGs, remains sound and complete for summary causal DAGs. By examining the implications of node contractions and the consequent addition of edges to a causal DAG, we offer clear insights into how these modifications affect the soundness and completeness of do-calculus rules when applied within the framework of summary causal DAGs (Section 6).

Experimental Evaluation. We conduct an extensive experimental study over synthetic and five real-life datasets, demonstrating the effectiveness of CaGreS compared to three existing solutions and two variations of CaGreSas additional comparison points. Our results demonstrate the robust quality of CaGreS, even compared to the optimal solution for the causal DAG summarization problem. The results further show the efficiency of CaGreS in handling high-dimensional datasets and its ability to generate summary causal DAGs that ensure reliable causal inference (Section 7).

Relate work is discussed in Section 8 and we conclude in Section 9. All proofs are provided in the Appendix.

2 BACKGROUND

We consider a single-relation database over a schema $\mathbb{A}$. The schema is a vector of attribute names, i.e., $\mathbb{A}=(A_1,\ldots,A_s)$, where each A_i can be categorical or continuous. We use upper case letters to denote a variable from $\mathbb{A}$ and bold symbols to represent a set of variables.

The broad goal of causal inference is to estimate the effect of an *exposure variable* $T\in\mathbb{A}$ (e.g., `Airport`) on an *outcome variable* $O\in\mathbb{A}$ (e.g., `Dep. Delay`). In this work, we use Pearl's model for conducting such inference using observational data [61].

To get an unbiased estimate for the causal effect of the exposure T on the outcome O, one must mitigate the effect of *confounding variables*, i.e., attributes that can affect the exposure assignment and outcome [61]. For instance, when estimating how an airport can affect the departure delay, one would avoid a source of *confounding bias* by considering the departure city and weather conditions. Pearl's causal model provides ways to account for these confounding variables to get an unbiased causal estimate from observational data using *causal DAGs* [61]. Causal DAGs provide a simple way of graphically representing causal relationships within a set of variables. A causal DAG $\mathcal{G}$ for the variables in $\mathbb{A}$ is a specific type of a Bayesian network and is formally defined as follows.

Causal DAG. A Bayesian network is a DAG $\mathcal{G}$ in which nodes represent random variables and edges express direct dependence between the variables. Each node X_i is associated with the conditional distribution $\mathbb{P}(X_i|\pi(X_i))$, where $\pi(X_i)$ is the set of parents of X_i in $\mathcal{G}$. The joint distribution over all variables $\mathbb{P}(X_1,\ldots,X_n)$, is given by the product of all conditional distributions. That is,

$$\mathbb{P}(X_1,\ldots,X_n)=\prod_{i=1}^{n}\mathbb{P}(X_i|\pi(X_i)) \tag{1}$$

A causal DAG is a Bayesian network where edges signify direct causal influence rather than statistical dependence. We say that U is a potential cause of V if there is a directed path from U to V. Fig. 1 shows an example causal DAG for the flight delay dataset.

Let $\mathcal{G}$ be a causal DAG. A *directed path* $t=(V_1,\ldots,V_n)$ is a sequence of nodes such that there is an edge $(V_i\to V_{i+1})\in\mathsf{E}(\mathcal{G})$ for every $i\in\{1,\ldots,n-1\}$. We say that V is a *descendant* of U, and U an *ancestor* of V if there is a directed path from U to V. We denote by $\mathsf{ch}_{\mathcal{G}}(V)$ the child-nodes of V in $\mathcal{G}$, by $\mathtt{Dsc}_{\mathcal{G}}(V)$, the descendants of V (we assume that $V\in\mathtt{Dsc}_{\mathcal{G}}(V)$), and by $\mathtt{NDsc}_{\mathcal{G}}(V)$ the nodes of $\mathcal{G}$ that are not descendants of V. For a set of nodes $\mathbf{S}\subseteq\mathsf{V}(\mathcal{G})$, we let $\mathtt{Dsc}_{\mathcal{G}}(\mathbf{S})\stackrel{\text{def}}{=}\bigcup_{U\in\mathbf{S}}\mathtt{Dsc}_{\mathcal{G}}(U)$, and by $\mathtt{NDsc}_{\mathcal{G}}(\mathbf{S})\stackrel{\text{def}}{=}\bigcap_{U\in\mathbf{S}}\mathtt{NDsc}_{\mathcal{G}}(U)$.

A *trail* $t=(V_1,\ldots,V_n)$ is a sequence of nodes such that there is an edge between V_i and V_{i+1} for every $i\in\{1,\ldots,n-1\}$. That is, $(V_i\to V_{i+1})\in\mathsf{E}(\mathcal{G})$ or $(V_i\leftarrow V_{i+1})\in\mathsf{E}(\mathcal{G})$ for every $i\in\{1,\ldots,n-1\}$. A node V_i is said to be *head-to-head* with respect to t if $(V_{i-1}\to V_i)\in\mathsf{E}(\mathcal{G})$ and $(V_i\leftarrow V_{i+1})\in\mathsf{E}(\mathcal{G})$. A trail $t=(V_1,\ldots,V_n)$ is *active* given $\mathbf{Z}\subseteq\mathcal{V}$ if (1) every V_i that is a head-to-head node with respect to t either belongs to $\mathbf{Z}$ or has a descendant in $\mathbf{Z}$, and (2) every V_i that is not a head-to-head node w.r.t. t does not belong to $\mathbf{Z}$. If a trail t is not active given $\mathbf{Z}$, then it is *blocked* given $\mathbf{Z}$.

***d*-Separation & Conditional Independence.** Let $\mathcal{G}$ be a causal DAG and $\mathbf{X},\mathbf{Y},\mathbf{Z}\subseteq\mathsf{V}(\mathcal{G})$ be pairwise disjoint. We say that $\mathbf{Z}$ *d-separates* $\mathbf{X}$ from $\mathbf{Y}$ if every trail between $X\in\mathbf{X}$ and $Y\in\mathbf{Y}$ is blocked given $\mathbf{Z}$. We denote by $\mathbf{X}\perp\!\!\!\perp_d\mathbf{Y}\mid\mathbf{Z}$ that $\mathbf{Z}$ d-separates $\mathbf{X}$ from $\mathbf{Y}$.

Causal DAGs encode a set of Conditional Independence statements (CIs) that can be read off the graph using d-separation [61]. These statements describe the absence of an active trail between two sets of variables when conditioning on other variables. In our example (Fig. 1), some examples of CIs are: (`Pop. density` $\perp\!\!\!\perp_d$ `Airport` | `Dep. city`), and (`Prec.` $\perp\!\!\!\perp_d$ `Airport` | `Humidity`, `Temp.`).

ATE& do-Calculus. The do-operator, a fundamental concept in causal inference, is used to denote interventions on variables in a causal model. It represents the intervention on a variable to observe the resulting change in an outcome variable while holding the

external factors constant. In computing the *Average Treatment Effect* (ATE) [61], a popular measure of causal estimate, the *do*-operator is applied to represent the treatment assignment for treatment and control groups. The ATE quantifies the average causal effect of a treatment T on an outcome variable O in a population:

$$ATE(T, O) = \mathbb{E}[O \mid do(T = 1)] - \mathbb{E}[O \mid do(T = 0)] \quad (2)$$

To compute the causal effect of a treatment T on an outcome O, identifying and adjusting for confounders is crucial to revealing the true causal relationship. The backdoor criterion, introduced by Pearl [61], serves as a sufficient condition for this purpose by specifying a set $\mathbf{Z}$ that, when conditioned upon, blocks all backdoor paths between T and O. This criterion offers a practical way to adjust for confounders within the causal DAG framework. However, it is part of the larger *do*-calculus system, a more generic axiomatic framework designed for reasoning about interventions and their effects within causal models. The *do*-calculus comprises three rules that facilitate the substitution of probability expressions containing the *do*-operator with standard conditional probabilities [61]. This system underpins a systematic methodology for deriving causal relationships from observational data. Given its soundness and completeness, this framework offers a broad toolkit for causal inference beyond the specific application of the backdoor criterion. Since these concepts are not directly employed in this paper, we do not delve into a detailed review of them here.

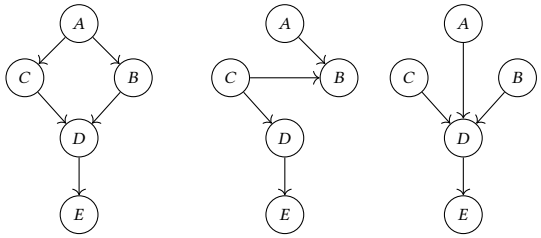
3 PROBLEM FORMULATION

Our aim is to distill an input causal DAG into a more interpretable summary DAG by grouping nodes while maintaining its utility for reliable causal inference. To achieve this: (1) *Comprehension through Constraints:* We introduce size and semantic constraints to improve the summary DAG's intelligibility, ensuring the core complexity of the original DAG is preserved in a simplified form. (2) *Integrity for Inference:* We meticulously preserve the most crucial causal information, which guarantees the ability to conduct reliable causal inference from the summary DAG. The summary causal DAG is expected to meet the following criteria:

- **Size**: It should be concise to reduce the cognitive load on the analyst while providing a clear view of the causal relationships [9].
- **Semantics**: It should merge only variables that share semantic relationships, reflecting coherent and meaningful clusters.
- **Causal Dependence and Directionality**: It must maintain the causal dependencies present in the original causal DAG, including the directionality of those dependencies. If variable A has a directed causal path to variable B in the original DAG, then this directional relationship should be faithfully preserved in the summary DAG.
- **Causal Independence**: It should also preserve causal independence represented in the original DAG. If variables A and B are not connected by a direct path, or equivalently, if they are conditionally independent, this lack of dependence should be reflected, to the greatest extent possible, in the summary DAG. Moreover, the summary DAG should remain faithful by not introducing any spurious CIs that are not implied by the original causal DAG.

However, since graph summarization inevitably entails some loss of information, our objective must be to preserve the utility of the summary causal DAG for causal inference. We note that in a causal DAG, information encoded by missing edges, or causal independence, is more critical than the information suggested by existing edges, which indicate *potential* causal dependence. This implies that adding edges to a causal DAG, provided acyclicity and directionality are maintained, does not necessarily compromise validity, whereas removing edges can undermine it [61]. Therefore, we rigorously enforce conditions on causal dependence to maintain directionality. Nevertheless, we relax the conditions on causal independence and assert that the summary DAG should preserve a subset of the independence assumptions encoded in the original DAG. We show that, with these considerations, the summary causal DAG remains a viable tool for causal inference (Section 6).

We first formalize the concept of a summary causal DAG, then rigorously formalize the problem of causal DAG summarization.



(a) $\mathcal{G}_1$ **(b)** $\mathcal{G}_2$ **(c)** $\mathcal{G}_3$

Figure 3: Three causal DAGs over the same set of nodes.

Table 1: The recursive bases of the summary DAGs in Figure 4

Graph	Recursive Basis
$\mathcal{G}_1$	$(C \perp\!\!\!\perp B\|A), (D \perp\!\!\!\perp A\|BC), (E \perp\!\!\!\perp ABC\|D)$
$\mathcal{H}_1$	$(D \perp\!\!\!\perp A\|BC), (E \perp\!\!\!\perp ABC\|D)$
$\mathcal{H}_2$	$(E \perp\!\!\!\perp AC\|BD)$
$\mathcal{H}_3$	$(E \perp\!\!\!\perp ABC\|D)$
$\mathcal{H}_4$	$(DE \perp\!\!\!\perp A\|BC)$

3.1 Summary Causal DAGs

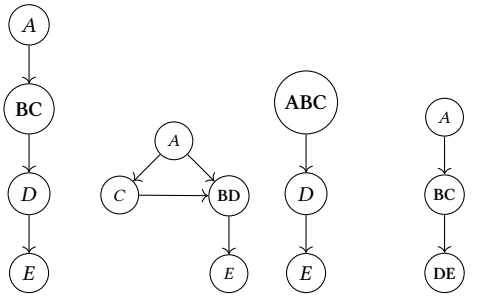
A *summary graph* is obtained by grouping nodes of the original graph based on a given partition of the nodes. The resulting graph retains the essential connectivity and structural information of the original graph but with a reduced number of nodes. We obtain a summary graph by applying *node contraction* operations [63].

Given a graph $\mathcal{G}$, the contraction of a pair of nodes $X, Y \in \mathsf{V}(\mathcal{G})$ is the operation that produces a graph $\mathcal{H}$ in which the two nodes X and Y are replaced with a single node $\mathbf{C}=\{X, Y\} \in \mathsf{V}(\mathcal{H})$, where $\mathbf{C}$ is now neighbors with nodes that X and Y were originally adjacent to (edge directionality is preserved). In node contraction, it does not matter if X and Y are connected by an edge; if they are, the edge is removed upon contraction.

Definition 1 (Summary-DAG)**.** A summary DAG of a DAG $\mathcal{G}$ is a pair $(\mathcal{H}, f)$, where $\mathcal{H}$ is a DAG with nodes $\mathsf{V}(\mathcal{H})$, edges $\mathsf{E}(\mathcal{H})$, and $f : \mathsf{V}(\mathcal{G}) \to \mathsf{V}(\mathcal{H})$ is a function that partitions the nodes $\mathsf{V}(\mathcal{G})$ among the nodes $\mathsf{V}(\mathcal{H})$, such that: If $(U, V) \in \mathsf{E}(\mathcal{G})$, then $f(U) = f(V)$ or $(f(U), f(V)) \in \mathsf{E}(\mathcal{H})$. We define the inverse $f^{-1} : \mathsf{V}(\mathcal{H}) \to 2^{\mathsf{V}(\mathcal{G})}$ as follows: $f^{-1}(X) \stackrel{\text{def}}{=} \{V \in \mathsf{V}(\mathcal{G}) : f(V) = X\}$

To simplify the notations, we omit f whenever possible.

Example 3. Consider Fig. 3(a) which depicts a DAG $\mathcal{G}_1$. After contracting B and C, the resulting summary DAG $\mathcal{H}_1$ is displayed



(a) $\mathcal{H}_1$ **(b)** $\mathcal{H}_2$ **(c)** $\mathcal{H}_3$ **(d)** $\mathcal{H}_4$

Figure 4: Summary causal DAGs for $\mathcal{G}_1$.

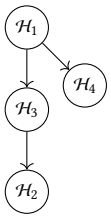


Figure 5: Partial order among the summary DAGs in Fig. 4.

in Fig. 4(a). In $\mathcal{H}_1$, the nodes B and C have been contracted into the node **BC**. Namely, $f(B) = f(C) = \mathbf{BC}$, and $f^{-1}(\mathbf{BC}) = \{B, C\}$. □

Let $\mathcal{G}$ be a causal DAG and let $\mathcal{G}'$ be a causal DAG where $\mathsf{V}(\mathcal{G})=\mathsf{V}(\mathcal{G}')$. We say that $\mathcal{G}'$ is a *supergraph* of $\mathcal{G}$ if $\mathsf{E}(\mathcal{G})\subseteq\mathsf{E}(\mathcal{G}')$. In this case, we also say that $\mathcal{G}$ is *compatible* with $\mathcal{G}'$.

Definition 2 (Compatibility)**.** Let $(\mathcal{H}, f)$ be a summary DAG. A DAG $\mathcal{G}$ is *compatible* with $\mathcal{H}$ if $\mathcal{H}$ is a summary-DAG for $\mathcal{G}$. We use $\{\mathcal{G}_i\}_{\mathcal{H}}$ to denote the set of all causal DAGs compatible with $\mathcal{H}$.

Example 4. Consider again Fig. 3. Both $\mathcal{G}_1$ and $\mathcal{G}_2$ are compatible with the summary DAG $\mathcal{H}_1$ shown in Fig. 4(a) (achieved by contracting B and C). However, $\mathcal{G}_3$ is not compatible with $\mathcal{H}_1$, since the edge between D and A is not preserved. □

We are interested in finding acyclic summary DAGs. Thus, we prove a simple lemma that characterizes node contractions that maintain acyclicity.

Lemma 3.1. *Let $\mathcal{G}$ be a DAG, and let $X, Y\in\mathsf{V}(\mathcal{G})$. Let $\mathcal{H}_{XY}$ denote the summary graph that results from $\mathcal{G}$ by contracting X and Y. Then $\mathcal{H}_{XY}$ contains a directed cycle if and only if $\mathcal{G}$ contains a directed path P from X to Y (or Y to X), where $|P|\geq 2$.*

A *summary causal DAG* is a specific type of summary graph obtained through node contraction operations over a given causal DAG $\mathcal{G}$ and ensures acyclicity. Namely, only contractions involving nodes without a directed path of length ≥ 2 are permitted.

The Recursive Basis. The *Recursive Basis* (RB) [27] for a causal DAG comprises a set of at most n CIs, signifying that each node is conditionally independent of its preceding nodes given its parents. This succinct set of CIs holds significance, as it can be used for constructing the causal DAG through a recursive process, and all other CIs encoded in the causal DAG can be deduced from it.

Formally, given a causal DAG $\mathcal{G}$, let $\langle X_1, \ldots, X_n\rangle$ denote a complete topological order over $\mathsf{V}(\mathcal{G})$. Equation 1 implicitly encodes a set of n CIs, called the RB for $\mathcal{G}$, defined as follows:

$$\Sigma_{\text{RB}}(\mathcal{G}) \stackrel{\text{def}}{=} \{(X_i \perp\!\!\!\perp X_1 \ldots X_{i-1} \backslash \pi(X_i) \mid \pi(X_i)) : i \in [n]\} \quad (3)$$

Let $\tau \stackrel{\text{def}}{=} (\mathbf{A} \perp\!\!\!\perp_d \mathbf{B}|\mathbf{C})$ be a CI where $\tau \notin \Sigma_{\text{RB}}(\mathcal{G})$. The implication problem associated with causal DAGs is determining whether τ holds in every probability distribution in which all the CIs in Σ_{RB} hold. In notation, $\Sigma_{\text{RB}}(\mathcal{G}) \implies \tau$. Let $\mathcal{G}$ be a DAG generated by the RB Σ_{RB}. That is, the nodes of $\mathcal{G}$ are the random variables $\{X_1, \ldots, X_n\}$, and its edges are $\mathsf{E}(\mathcal{G})=\{X_i \to X_j | X_i \in \pi(X_j)\}$. Given a CI $\tau=(\mathbf{A} \perp\!\!\!\perp_d \mathbf{B}|\mathbf{C})$, the d-separation algorithm efficiently determines if $\mathbf{C}$ d-separates $\mathbf{A}$ from $\mathbf{B}$ in $\mathcal{G}$. It has been shown that $\Sigma_{\text{RB}} \implies \tau$ iff $\mathbf{C}$ d-separates $\mathbf{A}$ from $\mathbf{B}$ in the DAG $\mathcal{G}$ generated by Σ_{RB} [27]. In other words, both the semi-graphoid axioms (given in Appendix A) and the d-separation criterion are sound and complete for inferring CIs from the RB.

Example 5. Consider the causal DAG $\mathcal{G}_1$ in Fig. 3(a). In the nodes' topological order, A precedes B and C, which in turn, precedes D. The last node is E. The RB of $\mathcal{G}_1$ is given in Table 1 □

The RB of a summary causal DAG is defined in a manner akin to the RB of a causal DAG, as denoted by Eq. (3). The only difference is that in a summary causal DAG, a node may represent a subset of nodes of the original DAG. To illustrate, Table 1 shows the RBs of summary causal DAGs depicted in Figure 4.

3.2 The causal DAG summarization Problem

As mentioned, we aim to reduce the size of an input causal DAG by partitioning its nodes into semantically related sets, while retaining maximal causal information. We covered the three criteria of our problem before proceeding with formalizing it.

Size Constraint A size constraint on a summary graph is a key motivating constraint for graph summarization work and may be imposed on the number of edges, nodes, storage space, minimum description length, etc. [43]. In this work, we focus on a node-based size constraint, as limited-size graphs are generally more accessible for inspection [9, 31]. Moreover, it is relatively straightforward for analysts to set and adjust a limit on the number of nodes [88].

Semantic Constraint We aim to ensure that only semantically related variables are merged, supporting comprehension and semantic coherency in the summary DAG. Thus, we seek to quantify the semantic similarity among variable subsets considered for contraction. For example, merging `Precipitation` and `Temperature` is sensible, but merging `Humidity` and `Fleet size` would be challenging to interpret in a summary DAG. To achieve this, we draw upon previous work on semantic similarity [29], measured using embedding techniques [50] or large language models [5].

A semantic constraint imposes a minimum threshold on the *inter-cluster semantic similarity* within a subset of nodes. We can accommodate various definitions of inter-cluster semantic similarity, such as between the farthest nodes or the average similarity within a subset. We assume a semantic similarity measure $sim(\cdot, \cdot)$ that assigns a value in $[0, 1]$ to a variable pair. A score of 0 indicates that two variables have no semantic relationship, and a score of 1 signifies that they are semantically equivalent. Let $InterSim(\boldsymbol{C})$ denote the inter-cluster similarity of a set of nodes $\boldsymbol{C}$. Given a summary

DAG $\mathcal{H}$ and a threshold τ, we say that $\mathcal{H}$ satisfies the semantic constraint if for every $\mathbf{C}\in\mathbf{V}(\mathcal{H})$, $InterSim(\mathbf{C})\geq\tau$.

Causal Dependence and Independence As mentioned, to preserve causal dependence and directionality, if two variables have a directed path between them in the original DAG, then this relationship should be faithfully preserved in the summary DAG. Indeed, this follows from the definition of a summary DAG (Def. 1).

Given two summary DAGs derived from the same causal DAG $\mathcal{G}$, both adhering to the constraints, we prefer the one that preserves, to a larger degree, the set of CIs represented in $\mathcal{G}$. To this end, we devise a measure to compare summary DAGs based on their RBs. When comparing two summary DAGs $\mathcal{H}_1$ and $\mathcal{H}_2$, we assert that $\mathcal{H}_1$ is *superior* to $\mathcal{H}_2$ if the RB of $\mathcal{H}_2$ is implied by the RB of $\mathcal{H}_1$. Namely, all the CIs encoded by $\mathcal{H}_2$ can also be deduced from $\mathcal{H}_1$. We are searching for a maximal summary causal DAG, namely, that its RB is not implied by any other valid summary DAG.

Let $\Omega\stackrel{\text{def}}{=}\{X_1,\ldots,X_n\}$ be a set of jointly distributed random variables with distribution $\mathbb{P}$ (i.e., nodes of the original DAG). Formally,

Definition 3 (I-Map)**.** A DAG $\mathcal{G}$ is an *I-Map* for $\mathbb{P}$ if for every disjoint sets $\mathbf{X},\mathbf{Y}$, and $\mathbf{Z}$ it holds that $(\mathbf{X}\perp\!\!\!\perp_d\mathbf{Y}\mid\mathbf{Z})_{\mathcal{G}}$ only if $(\mathbf{X}\perp\!\!\!\perp_{\mathbb{P}}\mathbf{Y}\mid\mathbf{Z})$.

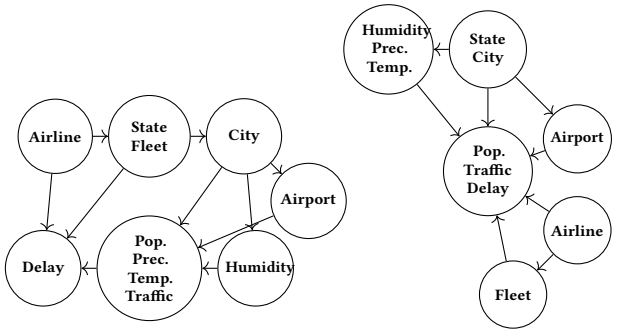
We denote by $\mathcal{G}(\mathbb{P})$ the set of DAGs that are an I-Map for $\mathbb{P}$. Let $\mathcal{G}_1$ and $\mathcal{G}_2$ be two DAGs that are I-Maps for $\mathbb{P}$. We say that $\mathcal{G}_2$ is superior to $\mathcal{G}_1$, in notation $\mathcal{G}_2\succ\mathcal{G}_1$, if for every $\sigma\in\Sigma_{\text{RB}}(\mathcal{G}_1)$, it holds that $\Sigma_{\text{RB}}(\mathcal{G}_2)\implies\sigma$. Note that the relation $\succ$ does not necessarily form a complete order. We say that $\mathcal{G}$ is *maximal for* $\mathbb{P}$ if $\mathcal{G}$ is an I-Map for $\mathbb{P}$, and there does not exist any $\mathcal{G}'\in\mathcal{G}(\mathbb{P})$ such that $\mathcal{G}'\succ\mathcal{G}$. Our goal is to find a summary DAG that is maximal for $\mathbb{P}$.

Example 6. Consider the causal DAG $\mathcal{G}_1$ in Fig. 3(a). Fig. 4(a) presents a 4-size summary DAG $\mathcal{H}_1$ for $\mathcal{G}_1$. The RBs of both DAGs are shown in Table 1. Clearly, $\Sigma_{\text{RB}}(\mathcal{H}_1)\subset\Sigma_{\text{RB}}(\mathcal{G}_1)$, and hence $\mathcal{H}_1$ is an I-Map for $\mathbb{P}$. Fig. 4(b) presents $\mathcal{H}_2$, another 4-size summary DAG for $\mathcal{G}_1$, where $\Sigma_{\text{RB}}(\mathcal{H}_2)=\{(E\perp\!\!\!\perp AC|BD)\}$. From the semi-graphoid axioms, it holds that $(E\perp\!\!\!\perp ABC|D)\implies(E\perp\!\!\!\perp AC|BD)$. Thus, $\mathcal{H}_1\succ\mathcal{H}_2$. Hence, $\mathcal{H}_1$ is a superior summary DAG. Similarly, Figures 4(c) and 4(d) illustrate $\mathcal{H}_3$ and $\mathcal{H}_4$, 3-size summary DAGs for $\mathcal{G}_1$. Their RBs are given in Table 1. The partial order among all summary DAGs is presented in Fig. 5. Despite $\mathcal{H}_3$ having only three nodes, it surpasses $\mathcal{H}_2$. However, $\mathcal{H}_3$ and $\mathcal{H}_4$ are incomparable, i.e., neither $\Sigma_{\text{RB}}(\mathcal{H}_3)\implies\Sigma_{\text{RB}}(\mathcal{H}_4)$ nor $\Sigma_{\text{RB}}(\mathcal{H}_4)\implies\Sigma_{\text{RB}}(\mathcal{H}_3)$. □

We define the causal DAG summarization problem as follows:

Problem 1 (causal DAG summarization)**.** Given a causal DAG $\mathcal{G}$ defined over a joint distribution $\mathbb{P}$, a bound k, a semantic measure $sim(\cdot,\cdot)$, and a threshold τ, find a summary causal DAG $\mathcal{H}$ s.t.:
Size Constraint: the number of nodes in $\mathcal{H}$ is $\leq k$.
Semantic Constraint: the inter-cluster semantic similarity (measure by $sim(\cdot,\cdot)$) of each node $\mathbf{V}\in\mathbf{V}(\mathcal{H})$, $InterSim(\mathbf{V})$, is $\geq\tau$.
Maximality Constraint: $\mathcal{H}\in\mathcal{G}(\mathbb{P})$ and is maximal for $\mathbb{P}$.

Example 7. Consider the causal DAG in Fig. 1. We set k=7 and τ=0.5 (for brevity, the semantic similarity matrix is given in Appendix B). Fig. 2(a) depicts the optimal summary causal DAG. Another example of a summary DAG that satisfies the constraints is shown in Fig. 6(b). The summary DAG in Fig. 2 (a) is preferable concerning



(a) Maximal, no semantic **(b) Not maximal**

Figure 6: Summary causal DAGs for the DAG in Fig. 1.

the preservation of causal information. This is because this DAG conducts the same node contractions, with the additional contraction of {Pop, Traffic} with Delay. Merging Traffic and Delay results in the loss of information that there was no direct causal link between, e.g., either Fleet or Humidity to Traffic In the original DAG. The optimal summary DAG, without regard for the semantic constraint, is given in Fig. 6(a). In this summary DAG, the semantic constraint is violated since Fleet cannot be merged with State due to having a similarity score <0.5. □

We show that the causal DAG summarization problem is *NP*-hard via a reduction from the k-Min-Cut problem [30].

Theorem 3.2. *causal DAG summarization is NP-hard.*

4 NODE-CONTRACTION AS EDGE ADDITION

Next, we establish the connection between node contractions and the addition of edges to the input causal DAG. This connection will be used to read off, from a given summary causal DAG, all the CIs it encodes. It also serves as a pivotal factor in guiding our algorithm for selecting promising node pairs to merge. Additionally, in Section 6, we will leverage this connection to demonstrate how causal inference can be directly conducted over summary DAGs.

4.1 The Canonical Causal DAG

Given a summary causal DAG $\mathcal{H}$, we define its corresponding canonical causal DAG, denoted as $\mathcal{G}_{\mathcal{H}}$. In this causal DAG, cluster nodes are decomposed into distinct nodes connected by edges. Each node within a cluster in $\mathcal{H}$ is linked by an edge to all the parents and children of the cluster node in $\mathcal{G}_{\mathcal{H}}$. We show that the RB of canonical causal DAG is *equivalent* to that of $\mathcal{H}$. We begin by defining the notion of equivalence for sets of CIs.

Definition 4 (CI Sets Equivalence)**.** Let $\mathbf{S}$ and $\mathbf{T}$ denote two sets of CIs over the variable-set $\{X_1,\ldots,X_n\}$. We say that $\mathbf{S}\implies\mathbf{T}$ if $\mathbf{S}\implies\sigma$ for every CI $\sigma\in\mathbf{T}$. We say that $\mathbf{S}$ and $\mathbf{T}$ are *equivalent*, in notation $\mathbf{S}\equiv\mathbf{T}$, if $\mathbf{S}\implies\mathbf{T}$ and $\mathbf{T}\implies\mathbf{S}$.

Next, we formally define the notion of the *canonical causal DAG* for a given summary DAG.

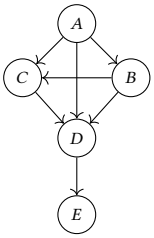


Figure 7: The canonical causal DAG of $\mathcal{H}_3$ (Fig. 4(c))

Definition 5 (Canonical Causal DAG)**.** Let $(\mathcal{H}, f)$ be a summary DAG for a causal DAG $\mathcal{G}$. Let $\langle X_1, \ldots, X_n \rangle$ denote a complete topological order over $\mathsf{V}(\mathcal{G})$. We define the canonical causal DAG associated with $(\mathcal{H}, f)$, denoted $\mathcal{G}_{\mathcal{H}}$ as follows: $\mathsf{V}(\mathcal{G}_{\mathcal{H}}) = \mathsf{V}(\mathcal{G})$, and

$$(X_i, X_j) \in \mathsf{E}(\mathcal{G}_{\mathcal{H}}) \text{ if and only if } \quad (X_i, X_j) \in \mathsf{E}(\mathcal{G})$$
$$\text{or} \quad (f(X_i), f(X_j)) \in \mathsf{E}(\mathcal{H})$$
$$\text{or} \quad f(X_i) = f(X_j) \text{ and } i < j$$

We observe that, by definition, $\mathcal{G}_{\mathcal{H}}$ is compatible with the summary DAG $(\mathcal{H}, f)$.

Example 8. Consider the summary DAG $\mathcal{H}_3$ in Figure 4(c). Its canonical causal DAG $\mathcal{G}_{\mathcal{H}_3}$ is depicted in Figure 7. Assume that in the topological order A proceeds B which in turn proceeds C. All nodes within the cluster node **ABC** are connected by edges in $\mathcal{G}_{\mathcal{H}_3}$, according to the topological order. Since the cluster node **ABC** is the parent of D in $\mathcal{H}_3$, in $\mathcal{G}_{\mathcal{H}_3}$ all A, B and C are parents of D. □

We show that the RB of the canonical causal DAG $\mathcal{G}_{\mathcal{H}}$ is equivalent to that of the summary DAG $\mathcal{H}$ obtained by node contractions to a causal DAG $\mathcal{G}$. In other words, node contractions can be conceptualized as the addition of edges to the input causal DAG, since $\mathsf{V}(\mathcal{G}_{\mathcal{H}})=\mathsf{V}(\mathcal{G})$ and $\mathsf{E}(\mathcal{G})\subseteq\mathsf{E}(\mathcal{G}_{\mathcal{H}})$ (i.e., $\mathcal{G}_{\mathcal{H}}$ is a supergraph of $\mathcal{G}$).

Theorem 4.1. *Let $\mathcal{H}$ be a summary causal DAG, and $\mathcal{G}_{\mathcal{H}}$ is its corresponding canonical causal DAG. We have: $\Sigma_{RB}(\mathcal{H}) \equiv \Sigma_{RB}(\mathcal{G}_{\mathcal{H}})$.*

Continuing with Example 8, the RB of $\mathcal{G}_{\mathcal{H}3}$ is $(E \perp\!\!\!\perp ABC|D)$, which is identical to that of $\mathcal{H}_3$ (see Table 1).

4.2 *s*-Separation

We introduce the notion of *s-separation*, an extension of *d*-separation, tailored to identify CIs encoded by a summary DAG. Intuitively, a summary DAG represents a collection of causal DAGs that are compatible with it, meaning that it could have been obtained from any of those DAGs (similar to *possible worlds* in probabilistic database [18]). Each of these DAGs encodes a different set of CIs. The set of CIs encoded by a summary DAG is defined as the intersection of CIs that hold in all compatible DAGs. In this way, we can ensure we restrict ourselves only to CIs that are certainly present in a particular context and can be reliably used for inference.

The validity of a CI statement, as derived from summary DAG $\mathcal{H}$, is given by the following definition:

Definition 6 (Validity of a CI in a summary DAG)**.** A CI statement is deemed *valid* in a summary causal DAG $\mathcal{H}$ if and only if it is implied by all causal DAGs within $\{\mathcal{G}_i\}_{\mathcal{H}}$.

s-separation captures all certain CIs that hold across all DAGs in $\{\mathcal{G}_i\}_{\mathcal{H}}$ from which $\mathcal{H}$ could have been derived. We propose the following criterion for *s*-separation to encapsulate this notion of validity.

Definition 7 (*s*-separation)**.** Given a summary DAG $(\mathcal{H}, f)$ and disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathsf{V}(\mathcal{H})$, we say that $\mathbf{X}$ and $\mathbf{Y}$ are *s*-separated in $\mathcal{H}$ by $\mathbf{Z}$, denoted by $(\mathbf{X} \perp\!\!\!\perp_S \mathbf{Y} \mid \mathbf{Z})_{\mathcal{H}}$, iff $f^{-1}(\mathbf{X})$ and $f^{-1}(\mathbf{Y})$ are *d*-separated by $f^{-1}(\mathbf{Z})$ in every causal DAG within $\{\mathcal{G}_i\}_{\mathcal{H}}$.

We say that $\mathbf{X}$ and $\mathbf{Y}$ are *s*-connected in $(\mathcal{H}, f)$ by $\mathbf{Z}$, if there exists a causal DAG $\mathcal{G} \in \{\mathcal{G}_i\}_{\mathcal{H}}$, such that $f^{-1}(\mathbf{X})$ and $f^{-1}(\mathbf{Y})$ are *d*-connected in $\mathcal{G}$ by $f^{-1}(\mathbf{Z})$.

4.2.1 *s*-separation Algorithm.

Given a summary causal DAG $\mathcal{H}$, we aim to derive the set of CIs it encodes. A naive approach would be to employ *d*-separation algorithms [61]. However, $\mathcal{H}$ can potentially encompass more CIs than those discerned through *d*-separation alone, as demonstrated in the following example.

Example 9. Referring back to Fig. 3, $(B \perp\!\!\!\perp_d E \mid D)$ and $(C \perp\!\!\!\perp_d E \mid B, D)$, and thus also $(B, C \perp\!\!\!\perp_d E \mid D)$, all hold in $\mathcal{G}_1$ and $\mathcal{G}_2$. Likewise, $(\mathbf{BC} \perp\!\!\!\perp_d E \mid D)$ holds in $\mathcal{H}_1$ (Fig. 4(a)) where $f^{-1}(\mathbf{BC})=\{B, C\}$. However, since $\mathcal{H}_1$ does not contain B or C as separate nodes, we cannot establish $(B \perp\!\!\!\perp_d E \mid D)$ or $(C \perp\!\!\!\perp_d E \mid B, D)$ from $\mathcal{H}_1$. □

To address this, a simple solution is to find the set of CIs shared across all causal DAGs compatible with $\mathcal{H}$. However, this approach is costly, as it necessitates examining all compatible causal DAGs. We, therefore, present a simple algorithm for *s*-separation that leverages the connection between a summary DAG and its canonical causal DAG. This algorithm operates as follows: Given a summary DAG $\mathcal{H}$, establish a topological order for its nodes.[2] Using this order, construct the canonical causal DAG $\mathcal{G}_{\mathcal{H}}$. Next, apply *d*-separation over $\mathcal{G}_{\mathcal{H}}$ and return the resulting CI set. We demonstrate that this algorithm is sound and complete.

Theorem 4.2 (Soundness and Completeness of *s*-separation). *In a summary DAG $(\mathcal{H}, f)$, let $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathsf{V}(\mathcal{H})$ be disjoint sets of nodes. If $\mathbf{X}$ and $\mathbf{Y}$ are d-separated by $\mathbf{Z}$ in $\mathcal{H}$, then in any causal DAG $\mathcal{G} \in \{\mathcal{G}_i\}_{\mathcal{H}}$, $f^{-1}(\mathbf{X})$ and $f^{-1}(\mathbf{Y})$ are d-separated by $f^{-1}(\mathbf{Z})$. That is:*

$$(\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y}|\mathbf{Z})_{\mathcal{H}} \Longrightarrow (f^{-1}(\mathbf{X}) \perp\!\!\!\perp_d f^{-1}(\mathbf{Y})|f^{-1}(\mathbf{Z}))_{\mathcal{G}} \Longrightarrow (\mathbf{X} \perp\!\!\!\perp_s \mathbf{Y}|\mathbf{Z})_{\mathcal{H}}$$

If $\mathbf{X}$ and $\mathbf{Y}$ are d-connected by $\mathbf{Z}$ in $\mathcal{H}$, then there exists a DAG $\mathcal{G} \in \{\mathcal{G}_i\}_{\mathcal{H}}$, s.t $f^{-1}(\mathbf{X})$ and $f^{-1}(\mathbf{Y})$ are d-connected by $f^{-1}(\mathbf{Z})$ in $\mathcal{G}$.

5 THE CAGRES ALGORITHM

We introduce an algorithm, named CaGreS for the causal DAG summarization problem. Although lacking theoretical guarantees, CaGreS effectively meets the size and semantic constraints and efficiently produces high-quality summary causal DAGs in practice. A brute force approach explores all summary DAGs with up to k nodes; for each candidate, it materializes its RB and selects one with a maximal RB. It finds the optimal summary causal DAG but runs in exponential time due to the exponential number of graphs to explore. CaGreS overcomes this by avoiding iterating over every possible summary DAG and merging nodes based on an estimation of the merging effect on the canonical causal DAG.

[2]The order of nodes within a cluster is considered arbitrary, or it may be determined based on the topological order of the input causal DAG if such information is preserved.

Algorithm 1: The CaGreS Algorithm

input : A causal DAG $\mathcal{G}$ and a number k.
output: A summary causal DAG $\mathcal{H}$ with k nodes.

1 $\mathcal{H} \leftarrow \mathcal{G}$
2 /* Merge node-pairs in which their cost is ≤ 1 */
3 $\mathcal{H} \leftarrow$ LowCostMerges($\mathcal{H}$)
4 **while** *size*($\mathcal{H}$.*nodes*) $> k$ **do**
5 $min\_cost \leftarrow \infty$
6 $(\boldsymbol{X}, \boldsymbol{Y}) \leftarrow Null$
7 **for** $(\boldsymbol{U}, \boldsymbol{V}) \in \mathcal{H}$.*nodes* **do**
8 **if** IsValidPair $(\boldsymbol{U}, \boldsymbol{V}, \mathcal{H})$ **then**
9 $\text{cost}_{UV} \leftarrow$ GetCost($\boldsymbol{U}, \boldsymbol{V}, \mathcal{H}$)
10 **if** $cost_{UV} < min\_cost$ **then**
11 min_cost $\leftarrow \text{cost}_{UV}$
12 $(\boldsymbol{X}, \boldsymbol{Y}) \leftarrow (\boldsymbol{U}, \boldsymbol{V})$
13 **if** $cost_{UV} == min\_cost$ **then**
14 Randomly decide if to replace $\boldsymbol{X}$ and $\boldsymbol{Y}$ with $\boldsymbol{U}$ and $\boldsymbol{V}$
15 $\mathcal{H}$.Merge($\boldsymbol{X}, \boldsymbol{Y}$)
16 **return** $\mathcal{H}$

Overview The CaGreS algorithm follows a previous line of work (e.g., [88]), using a bottom-up greedy approach to identify promising node pairs for contraction. Its main contribution lies in how it estimates merge costs: It counts the number of edges to be added in the canonical causal DAG for each node pair (a proxy for the RB's effect, as discussed in Section 4). In each iteration, the algorithm contracts the node pair resulting in the minimal number of edges (i.e., minimal cost). We also introduce optimizations for runtime efficiency, such as fast low-cost merges and caching mechanisms.

The CaGreS algorithm is given in Algorithm 1. Given a bound k and an input causal DAG, this algorithm iteratively seeks for the next-best pair of nodes to be merged, until the size constraint is met (lines 4-15). The next-best pair of nodes to merge is the node pair whose contraction has the lowest cost (lines 10-12). The algorithm randomly breaks ties (lines 13-14). The GetCost procedure is shown in Algorithm 2. The cost of merging two (clusters of) nodes $\mathbf{U}$ and $\mathbf{V}$ is equal to the number of edges to be added in the corresponding canonical causal DAG: (1) edges to be added between the nodes within the combined cluster $\mathbf{U} \bigcup \mathbf{V}$ (lines 3-4), (2) new parents for the nodes in $\mathbf{U}$ or $\mathbf{V}$ post-merge (lines 6-11), and (3) new children for the nodes in $\mathbf{U}$ or $\mathbf{V}$ after the merge (lines 13-18).

We next propose two optimizations to improve runtime.

Low Cost Merges As a pre-processing step, we contract node pairs with low costs (line 4). This involves merging nodes that share identical children and parents, with a cost of at most 1 (requiring, in the worst case, only the addition of an edge between them in the canonical causal DAG). Additionally, we merge nodes linked along a non-branching path of nodes, each having at most one parent and one child, incurring a cost of 1.

Caching Mechanisms We employ two caching mechanisms. The first is dedicated to storing node pairs deemed invalid for contraction, while the second is utilized for storing cost scores.

We initialize the invalid node pairs cache during the low-cost merge phase. An invalid pair is a pair of nodes with semantic

Algorithm 2: The GetCost Procedure

input : A summary causal DAG $\mathcal{H}$ and a pair of nodes $\boldsymbol{U}$ and $\boldsymbol{V}$.
output: The cost of contracting $\boldsymbol{U}$ and $\boldsymbol{V}$.

1 cost $\leftarrow 0$
2 /* New edges among the nodes in the cluster */
3 **if** $\mathcal{H}$.HasEdge($\boldsymbol{U}, \boldsymbol{V}$) == *False* **then**
4 cost $\leftarrow$ cost $+ size(\boldsymbol{U}) \cdot size(\boldsymbol{V})$
5 /* New parents */
6 $\text{parents}_U \leftarrow \mathcal{H}$.predecessors($\boldsymbol{U}$)
7 parents_U.RemoveSharedParents($\boldsymbol{V}$)
8 cost $\leftarrow$ cost + size(parents_U) $\cdot$ size($\boldsymbol{V}$)
9 $\text{parents}_V \leftarrow \mathcal{H}$.predecessors($\boldsymbol{V}$)
10 parents_V.RemoveSharedParents($\boldsymbol{U}$)
11 cost $\leftarrow$ cost + size(parents_V) $\cdot$ size($\boldsymbol{U}$)
12 /* New children */
13 $\text{children}_U \leftarrow \mathcal{H}$.successors($\boldsymbol{U}$)
14 children_U.RemoveSharedChildren($\boldsymbol{V}$)
15 cost $\leftarrow$ cost + size(children_U) $\cdot$ size($\boldsymbol{V}$)
16 $\text{children}_V \leftarrow \mathcal{H}$.successors($\boldsymbol{V}$)
17 children_V.RemoveSharedChildren($\boldsymbol{U}$)
18 cost $\leftarrow$ cost + size(children_V) $\cdot$ size($\boldsymbol{U}$)
19 **return** cost

similarity exceeding the threshold or connected by a directed path of length greater than 2 (according to Lemma 3.1). Throughout the execution of CaGreS, whenever we encounter an invalid pair, we add it to the cache. Before computing the cost in each iteration, we verify that the node pair is valid for contraction.

The cost of a node pair $\mathbf{U}, \mathbf{V}$ remains unchanged after merging another node pair $\mathbf{X}, \mathbf{Y}$ if neither $\mathbf{U}$ nor $\mathbf{V}$ are neighbors of $\mathbf{X}$ or $\mathbf{Y}$. Following the merge of $\mathbf{X}$ and $\mathbf{Y}$, we update the cost cache by removing the cost scores of all node pairs involving one of their neighbors. When calculating the cost for a node pair, we check if the score is in the cache. If not, we compute and add it, ensuring the cache reflects node pair mergers' impact on neighboring pairs.

Time Complexity A single cost computation in the input DAG $\mathcal{G}$ with n=$|\mathbf{V}(\mathcal{G})|$ takes $O(n)$ due to the maximum number of neighbors a node can have. The low-cost merge phase operates in $O(n^3)$ by iterating over every node pair in $\mathcal{G}$ and considering their neighbors. The algorithm undergoes $n-k$ iterations, evaluating all node pairs in the current summary DAG with no more than n neighbors. Thus, the overall time complexity is $O((n-k)\cdot n^3)$.

6 DO-CALCULUS IN SUMMARY CAUSAL DAGS

Next, we show that the rules of *do*-calculus are sound and complete in summary causal DAGs. This is vital to ensure that the summary causal DAGs are effective formats that support causal inference by enabling direct causal inference on the summary DAGs. Our proof relies on the connection between node contraction and the addition of edges, namely, on the equivalence between the RB of a summary DAG and its canonical causal DAG (Theorem 4.1). This result is not surprising because the canonical causal DAG is a supergraph of the input causal DAG. Pearl already observed in [61] that: *"The addition of arcs to a causal diagram can impede, but never assist, the identification of causal effects in nonparametric models. This is because such addition reduces the set of d-separation conditions*

Table 2: Datasets

Dataset	# Nodes (Variables)	# Edges	# Tuples
Flights	11	15	1M
Adult	13	48	32.5K
German	21	43	1000
Accidents	41	368	2.8M
URLs	60	310	1.7M

carried by the diagram; hence, if causal effect derivation fails in the original diagram, it is bound to fail in the augmented diagram".

Given a causal DAG $\mathcal{G}$, for a set of nodes $\mathbf{X} \subseteq \mathsf{V}(\mathcal{G})$, let $\mathcal{G}_{\overline{\mathbf{X}}}$ denote the graph that results from $\mathcal{G}$ by removing all incoming edges to nodes in $\mathbf{X}$, by $\mathcal{G}_{\underline{\mathbf{X}}}$ the graph that results from $\mathcal{G}$ by removing all outgoing edges from the nodes in $\mathbf{X}$. For a set of nodes $\mathbf{X} \subseteq \mathsf{V}(G)\backslash\mathbf{Z}$, we denote by $\mathcal{G}_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}$ the graph that results from $\mathcal{G}$ by removing all incoming edges into $\mathbf{X}$ and all outgoing edges from $\mathbf{Z}$.

Theorem 6.1 (Soundness of Do-Calculus in summary causal DAGs). *Let $\mathcal{G}$ be a causal DAG encoding an interventional distribution $P(\cdot \mid do(\cdot))$, compatible with the summary causal DAG $(\mathcal{H}, f)$. For any disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W} \subseteq \mathsf{V}(\mathcal{H})$, the following rules hold:*

$$\mathbf{R}_1: \quad (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} | \mathbf{X}, \mathbf{W})_{\mathcal{H}_{\overline{\mathbf{X}}}} \implies P(Y \mid do(\boldsymbol{X}), \boldsymbol{Z}, \boldsymbol{W}) = P(Y \mid do(\boldsymbol{X}), \boldsymbol{W})$$

$$\mathbf{R}_2: \quad (Y \perp\!\!\!\perp \boldsymbol{Z} | \boldsymbol{X}, \boldsymbol{W})_{\mathcal{H}_{\overline{\boldsymbol{X}}\underline{\boldsymbol{Z}}}} \implies P(Y \mid do(\boldsymbol{X}), do(\boldsymbol{Z}), \boldsymbol{W}) = P(Y \mid do(\boldsymbol{X}), \boldsymbol{Z}, \boldsymbol{W})$$

$$\mathbf{R}_3: \quad (\perp\!\!\!\perp \boldsymbol{Z} | \boldsymbol{X}, \boldsymbol{W})_{\mathcal{H}_{\overline{\boldsymbol{X}\boldsymbol{Z}(\boldsymbol{W})}}} \implies P(Y \mid do(\boldsymbol{X}), do(\boldsymbol{Z}), \boldsymbol{W}) = P(Y \mid do(\boldsymbol{X}), \boldsymbol{W})$$

where, $\boldsymbol{U} \stackrel{def}{=} (U)$ for every $U \in \mathsf{V}(\mathcal{H})$, and $Z(W)$ is the set of nodes in Z that are not ancestors of any node in W.

Theorem 6.2 (Completeness of Do-Calculus in summary causal DAGs). *Let $(\mathcal{H}, f)$ be a summary causal DAG for $\mathcal{G}$, and let $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{W}, \boldsymbol{Z} \subseteq \mathsf{V}(\mathcal{H})$ be disjoint sets of variables. If $\boldsymbol{Y}$ is d-connected to $\boldsymbol{Z}$ in $\mathcal{H}_{\overline{\boldsymbol{X}}}$ w.r.t. $\boldsymbol{X} \cup \boldsymbol{W}$, then there exists a causal DAG $\mathcal{G}'$ compatible with $\mathcal{H}$, such that $f(\boldsymbol{Y})$ is d-connected to $f(\boldsymbol{Z})$ in $\mathcal{G}'_{\overline{f(\boldsymbol{X})}}$ w.r.t. $f(\boldsymbol{X} \cup \boldsymbol{W})$.*

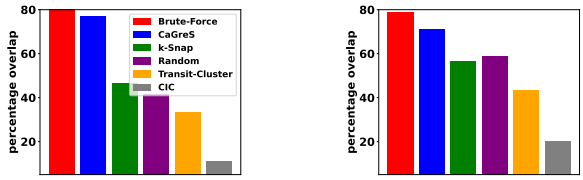
7 EXPERIMENTAL EVALUATION

We empirically demonstrate the following claims: (**C1**) Our summary DAGs support reliable causal analysis. (**C2**) Our objective evaluation method effectively determines superior summary causal DAGs. (**C3**) CaGreS outperforms other methods in causal DAG summarization and achieves efficient performance.

7.1 Experimental Setting

All algorithms are implemented in Python 3.7. The experiments were executed on a PC with a 4.8GHz CPU, and 16GB memory. Our code and datasets are available at [3].

Examined datasets We examine five public datasets, as shown in Table 2. We used a FastText model [51] to generate semantic similarity scores and build the input causal DAG using the approaches outlined in [98]. **Flights** [2]: a dataset describing domestic flight statistics in the US. This dataset was enriched with attributes describing the weather, population, and properties of the airline carriers. **Adult** [1]: a dataset comprises demographic information of individuals including their education, occupation, and income. **German** [7]: a dataset contains details of bank account holders, including demographic and financial information. **Accidents** [53]: This dataset provides information on various factors that are pertinent to the severity of car accidents, including weather conditions and the presence of traffic signs. **URLs** [4]: a dataset containing descriptions of malicious and non-malicious URLs. It encompasses properties such as URL length, the number of digits, and the occurrence of sensitive words.



(a) Flights (b) German

Figure 8: Average percentage overlap with ground truth.

We created **synthetic data** using the DoWhy package [77], enabling manipulation of node count, edge count, and data size while retaining knowledge of the causal DAG structure.

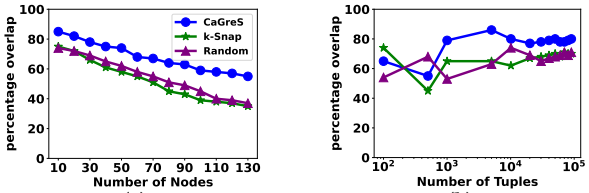
Baseline Methods We examine the following baseline methods:
Brute-Force: The optimal solution according to Def. 1. This algorithm implements an exhaustive search over all possible summary DAGs that satisfy the constraints.
k-Snap [88]: A general-purpose graph summarization algorithm that employs bottom-up node contractions (akin to CaGreS). The primary distinction lies in the objective function: while CaGreS aims to maximize retained causal information, k-Snap focuses on ensuring homogeneity among nodes within a cluster. We have enhanced k-Snap to address acyclicity and the semantic constraint.
Transit-Cluster In [89], the authors proposed Transit Clusters as a specific type of summary causal DAG that maintains identifiability properties under certain conditions. They introduced an algorithm to identify all transit clusters for a graph. For a fair comparison, we assess our solution against the transit cluster that meets the constraints and has the maximal RB.
CIC [56] The authors of [56] proposed a Clustering Information Criterion (CIC), based on information-theoretic measures that represent various complex interactions among variables in a causal DAG. Based on this criterion, they developed a greedy-based approach to learn clustered causal DAGs directly from the data.
Random: As a sanity check, we assess an algorithm that generates a random summary causal DAG that adheres to the constraints.

Metrics of evaluation As mentioned, in some cases, summary DAGs are incomparable, meaning that their RBs are not strictly implied by one another. To nevertheless compare their quality, we quantify the number of additional edges in their corresponding canonical causal DAG — edges that are absent in the original DAG. A smaller number of such edges implies a more sparse summary DAG that encodes more CIs. Additionally, with a smaller number of edges, the adjustment sets used to compute causal effects will likely be smaller and closer to the ones computed over the original causal DAG. As we show, these two metrics are highly correlated.

As a default configuration, we set the semantic threshold τ=0.33 and the size constraint k to $\frac{n}{2}$, where n is the number of nodes in the input causal DAG. The runtime cutoff was set at 1 hour. Causal effect computation was performed using the DoWhy library [77].

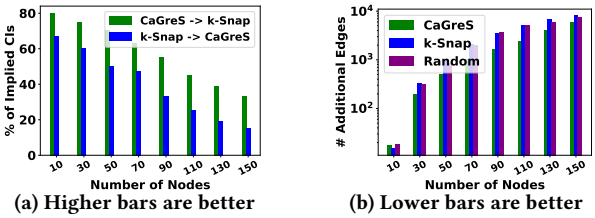
7.2 Usability Evaluation (C1)

We assess the utility of the summary causal DAGs for causal inference. To this end, we compare the causal effects estimated within

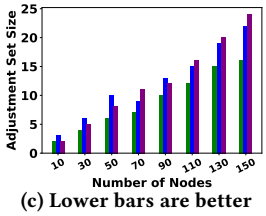


(a) (b)

Figure 9: Average percentage overlap vs. data properties.



(a) Higher bars are better (b) Lower bars are better



(c) Lower bars are better

Figure 10: Quality metrics vs. the number of nodes.

the original DAG with those computed within the summary causal DAGs. Each causal effect estimation yields an interval (of 95% confidence). We compare the intervals derived from the input DAG (the ground truth) with those obtained by the baselines. Given that the adjustment sets in the summary DAGs may differ from those in the original DAG, we anticipate getting different intervals.

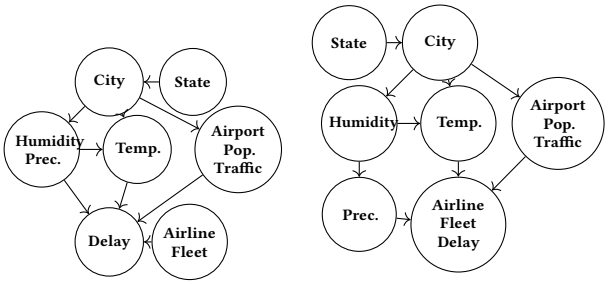
ATE Computation over Summary DAGs: In these experiments, we calculate the ATE (see in Section 2) between pairs of nodes connected by a causal path in the original DAG $\mathcal{G}$. If the treatment or outcome is part of a cluster node in the summary DAG $\mathcal{H}$, we proceed as follows: For a node pair U, V, we estimate the causal effect $ATE(U, V)$ over the corresponding canonical causal DAG $\mathcal{G}_{\mathcal{H}}$ (this is valid as demonstrated in Section 6). To reduce the adjustment set's size, we arrange for U to precede all nodes within its cluster node in the $\mathcal{G}_{\mathcal{H}}$. An alternative approach might involve constraining the causal effect across the summary DAG by considering all subsets in U's cluster in $\mathcal{H}$, a direction we leave for future work.

Average Percentage Overlap: We report the average percentage of overlap of the causal interval across all examined node pairs. A higher percentage overlap indicates greater robustness in causal inference. The results for Flights and German are shown in Fig. 8. CaGreS's average percentage overlap is close to that of Brute-Force, suggesting a high degree of similarity between the two summary DAGs. CaGreS surpasses all other competitors. *This underscores the superior suitability of CaGreS for causal inference compared to the baselines.*

In what comes next, we use synthetic data, allowing us to manage the number of nodes in the input DAG and database tuples. We omit from presentation the Brute-Force, Transit-Cluster, and CIC baselines as they exceeded our time limit cutoff.

Table 3: Pair-wise percentage of the RB's CIs implied.

	Brute-Force	CaGreS	k-Snap	Random	TC	CIC
Brute-Force	-	83.3%	50%	50%	16.6%	16.6%
CaGreS	50%	-	60%	16.6%	0%	16%
k-Snap	0%	16.6%	-	50%	16.6%	0%
Random	16.6%	0%	50%	-	0%	16.6%
TC	0%	0%	16.6%	16.6%	-	50%
CIC	0%	0%	0%	16.6%	0%	-



(a) CaGreS (b) k-Snap

Figure 11: Summary causal DAGs for the Flights dataset.

# of attributes: We examine how the number of nodes in the original causal DAG (i.e., # of attributes) affects the performance. With a larger number of nodes, the task of finding the optimal summary DAG becomes harder. Here, the number of data tuples is fixed at 10,000. The results are depicted in Figure 9 (a). Indeed, for all baselines with more data attributes, their alignment with the original causal DAG diminishes. Nevertheless, CaGreS consistently outperforms the competing methods.

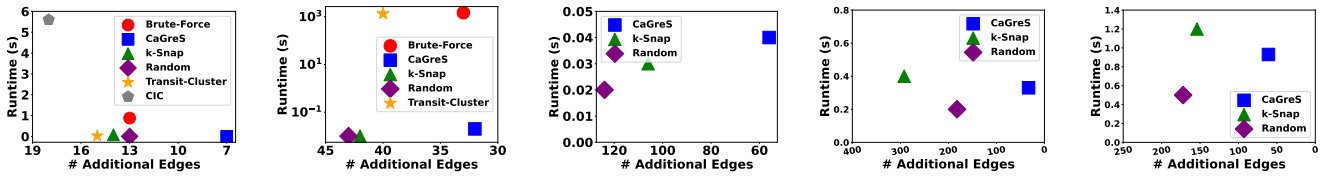
# of tuples: We examine the effect of data size (# of tuples) on performance. Given that causal effects are statistical measures that are sensitive to the sample size, we anticipate that as the number of tuples grows, the effects evaluated on the summary DAG will converge towards their counterparts evaluated on the original DAG. Here, the number of nodes in the input causal DAG is fixed at 30. Our findings are shown in Fig. 9 (b). For all baselines, when the data size is small, the results are noisy, and when it increases, the results tend to stabilize. *Again, CaGreS outperforms its competitors.*

7.3 Quality Evaluation (C2)

As mentioned, in many cases, there are many summary DAGs with maximal RBs. We, therefore, we investigate several evaluation metrics to determine which summary causal DAG is superior: (1) The percentage of CIs in the RB of one summary causal DAG that is implied by the RB of another summary DAG.(2) The number of additional edges in the canonical causal DAG. (3) The size of the adjustment sets in the evaluation of causal estimations. With smaller adjustment sets, the estimation will likely be more accurate. As we will show, these metrics are highly correlated.

We generated a series of random causal DAGs, each with a progressively larger number of nodes (five random DAGs for each node count), while keeping all other parameters fixed (the graph density is fixed to 0.3, k is set to $\frac{n}{2}$ and the semantic threshold τ is set to = 0). We omit from presentation the Brute-Force, Transit-Cluster, and CIC baselines as they exceeded our time limit cutoff.

The results are depicted in Fig. 10. Fig. 10(a) depicts the percentage of the CIs in the RB of k-Snap that are implied by that of CaGreS and vice versa. Similar trends were observed for Random,



(a) Flights (b) Adult (c) German (d) Accidents (e) URLs

Figure 12: Number of additional edges vs. runtime. The optimal solution should be located in the lower right region.

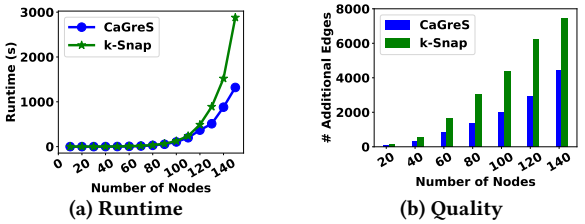
and thus are omitted. A higher percentage of k-Snap's CIs are implied by CaGreS compared to the percentage of CaGreS's CIs that are implied by k-Snap. Hence, while no RB entirely implies the other RB, we can still conclude that the summary DAG of CaGreS is superior to that of k-Snap. Fig. 10(b) depicts the number of additional edges in the . CaGreS consistently yields summary DAGs with fewer additional edges. This is not surprising, as the objective of CaGreS is to minimize the number of such edges. Fig. 10(c) shows the average size of the adjustment sets in the computation of causal estimations. Once again, CaGreS outperforms the competitors, consistently yielding smaller adjustment sets. As shown above, this results in more accurate causal estimations, as fewer redundant variables are considered. *Since these three metrics are closely interrelated, we deduce that it is appropriate to use the count of additional edges for comparing the quality of summary DAGs.*

7.4 Effectiveness Evaluation (C3)

We assess CaGreS based on quality and runtime performance.

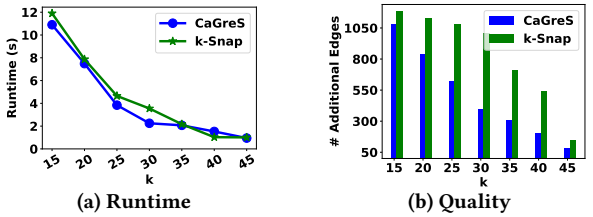
Case Study: Flights We present the pairwise percentage of the CIs in the RB implied by all baseline pairs. The results are shown in Table 3. The summary DAGs obtained by CaGreSand k-Snap are given in Fig. 11 (The optimal summary DAG by Brute-Force is given in Fig. 2(a)). Brute-Force yields the most effective summary DAG, as it implies the highest percentage of CIs of any other baseline. While 60% of the CIs of k-Snap are implied by the RB of CaGreS, only 16% of the CIs of CaGreS are implied by the RB of k-Snap. This superiority of CaGreS over k-Snap is further supported by a lower number of additional edges (7 for CaGreS, 13 for Brute-Force, and 14 for k-Snap). Intuitively, this stems from k-Snap's decision to form two 3-size clusters, connected by an edge. In the resulting canonical causal DAG, every pair of nodes within and between the clusters is connected by an edge.

Next, for each dataset, we report the runtime and the number of additional edges in the canonical causal DAG as a quality metric. The results are depicted in Fig. 12. Only CaGreS, k-Snap, and Random can effectively handle input causal DAGs with more than 20 nodes within a responsive runtime. While Random and k-Snap exhibit runtimes comparable to that of CaGreS, the latter consistently produces summary DAGs with fewer additional edges. As shown, this implies more accurate causal estimations. As expected, Brute-Force outperforms CaGreS in terms of quality but is impractical for interactive interaction. CIC exhibits relatively low performance, primarily due to the inclusion of a causal discovery component. Similarly, Transit-Cluster faces limitations in handling input causal DAGs with more than 20 nodes, as the algorithm materializes all transit clusters, and then selects the maximal one.



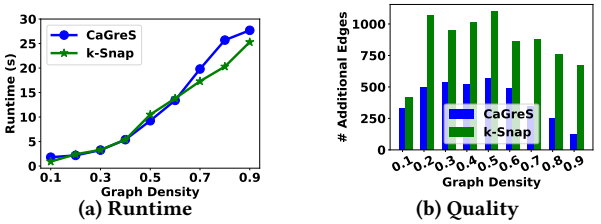
(a) Runtime (b) Quality

Figure 13: Number of nodes vs. running times and quality.



(a) Runtime (b) Quality

Figure 14: Summary size k vs. running times and quality.



(a) Runtime (b) Quality

Figure 15: Graph density vs. running times and quality.

7.4.1 Sensitivity Analysis. We analyze the influence of different parameters on performance. In these experiments, our focus shifts to synthetic data, which enables us to manipulate data-related factors.
Input DAG size We vary the number of nodes in the input DAG. To this end, we generated a series of random DAGs, progressively increasing the number of nodes (5 DAGs per node count) while keeping all other parameters constant. The results are shown in Fig. 13. As expected, k-Snap and CaGreS exhibit a polynomial increase in runtime (Fig. 13(a)). This aligns with our time complexity analysis (Section 5). The improvement relative to k-Snap is attributed to our caching mechanisms, particularly beneficial for large causal DAGs. Even when the causal DAG comprises over 100 nodes (i.e., over 100 variables in the data), CaGreS summarizes the DAG in less than 2 minutes. *This underscores CaGreS's scalability to high-dimensional data.* Observe that CaGreS consistently generates summary DAGs with fewer additional edges (Fig. 13(b)), indicating better quality.
Summary size We vary the size constraint k. Here, the node count is set to 50, the graph density is held constant at 0.3, and $\tau = 0$. The results are depicted in Fig.14. The runtime of both CaGreS and k-Snap demonstrate a linear increase with k (Fig. 14(a)). This is because larger k values necessitate more merges. As expected,

CaGreS manages to generate summary DAGs corresponding to canonical causal DAGs with fewer edges (Fig. 14(b)).

Graph density We investigate the influence of graph density on performance. We observe a nearly linear increase in runtime for both CaGreS and k-Snap as graph density rises (Figure 15(a)). This is because both algorithms examine the neighboring nodes of each node pair, and with increased density, there are more neighboring nodes. In terms of quality, as density increases, the number of additional edges also rises for both algorithms. However, at high densities (above 0.7), where the graph already includes many edges, there are fewer edges left to be added. Consequently, the number of additional edges for decreases (Figure 15(b)).

Semantic threshold We examine the effect of the semantic threshold τ on performance. The results are omitted from presentation due to space limitations. We report that increasing τ leads to a decrease in runtime for both CaGreS and k-Snap. This reduction is attributed to fewer valid solutions that satisfy the semantic constraint. Here again, CaGreS outperforms k-Snap, regardless of τ.

Data size We report that the data size, i.e., number of tuples, has no effect on the performance of CaGreS and k-Snap. This is because both algorithms only examine the input causal DAGs.

8 RELATED WORK

Summary Causal DAGs. The abstraction of causal models has been studied in literature [75]. Previous work [8, 59] investigated the problem of determining under what assumptions a DAG over sets of variables can represent the same CIs between those individual variables. The authors of [12, 13, 71] explored the problem of determining the causes of a target behavior (a macro-variable) from micro-variables (e.g., image pixels). Other works include chain graphs and ancestral causal graphs [39, 99], which were developed to represent sets of causal diagrams equivalent under specific properties. The authors of [66] presented a method to compress causal graphs by removing nodes to remove redundant information. In contrast, our work studies the problem of causal DAG summarization, wherein certain causal information is inevitably lost, yet the resultant summary DAG maintains reliable causal inference.

The authors of [6] expanded the *do-calculus* framework [61] to clustered causal graphs, a related but distinct concept. Our contribution lies in presenting a more streamlined proof of this principle, relying on the connection between node contraction and edge addition. Another difference is that we also propose an algorithm to generate a summary DAG from a given causal DAG. As discussed in our experiments, two approaches have addressed this problem. Transit-Cluster [89], focuses on clustering mediator variables but overlooks semantic relationships and lacks control over the summary DAG size. Another approach, the CIC algorithm [56], directly learns a summary DAG from data but does not consider semantic information or prioritize preserving independence information. In contrast, CaGreS is designed to conserve independence information, ensuring that crucial causal information is retained, thus supporting reliable causal inference over summary DAGs.

General-Purpose Summary Graphs Graph summarization aims to condense an input graph into a more concise representation. This condensed form not only reduces the graph's size but also facilitates efficient query answering [22, 45, 67, 80], enables enhanced data visualization and pattern discovery [17, 20, 33, 37, 41, 78], and supports extraction of influence dynamics [48]. Various graph summarization techniques have been explored, including grouping nodes based on similarity measures [40, 42, 49, 55, 67, 80, 81, 83, 88, 95, 101], reducing the number bits required to represent graphs [10, 47, 55, 69, 76], and removing unimportant nodes and edges [45, 84]. We argue that existing techniques are ill-suited for the causal DAG summarization problem. Graph summarization objectives differ across applications, often prioritizing minimizing the reconstruction error [40, 95], facilitating accurate query answering [45, 80], or enhancing visualizations [33, 37]. Consequently, existing methods inadequately cater to the objective of preserving causal information, often yielding graphs unsuitable for causal inference, as shown in the Introduction and experimental study.

Data Summarization. Data summarization involves distilling meaningful insights from large datasets into a compact, understandable format, which is crucial for uncovering trends and patterns. Related efforts have concentrated on summarizing tabular datasets [34, 38, 92], aiding analysts in extracting insights that might otherwise remain obscured. While our focus is different, its impact is akin, potentially assisting analysts in grasping causal relationships within high-dimensional datasets.

Causal Discovery. Causal discovery is a well-studied problem [28, 98, 100], whose goal is to infer causal relationships among variables. While background knowledge is crucial [62], causal DAGs can be inferred from data under certain assumptions [15, 28]. Existing methods include constraint-based [85] and score-based algorithms [15, 79, 93, 103]. Pashami et al. [60] proposed a clustering-based causal discovery method, using a cluster-based conflict resolution mechanism to determine the relationship among variables. Our aim is to summarize causal DAGs representing relationships in high-dimensional data. Consequently, our work serves as a complementary endeavor to existing research in causal discovery.

9 LIMITATIONS & CONCLUSIONS

A mixed graph, incorporating both directed and undirected edges, is a typical output of causal discovery algorithms [14, 64, 85]. It is often utilized when a complete causal DAG cannot be obtained. For simplicity in exposition, we concentrated on regular causal DAGs throughout this paper. Nevertheless, our results and algorithms apply to mixed graphs as well, as detailed in the Appendix.

When interpreting the summary causal DAG, it is crucial to acknowledge the following limitations: Firstly, if the input causal DAG's quality is low (e.g., missing nodes/edges), the summary DAG quality may suffer likewise. Secondly, user-defined parameters such as size constraint, semantic similarity measure, and the semantic threshold significantly influence the generated summary DAG. These parameters might need tuning by the analyst to achieve a desirable summary DAG. Future research will focus on principled approaches to suggest optimal values for these parameters.

This paper opens up promising future research directions. This includes the development of compact representations of node sets tailored specifically for causal inference, addressing additional size constraints, refining algorithms with theoretical guarantees, and improving the efficiency of the CaGreS algorithm.

REFERENCES

[1] 2016. Adult Census Income Dataset. https://www.kaggle.com/datasets/uciml/adult-census-income. Accessed: 2024-04-04.

[2] 2020. Kaggle Datasets: Flights Delay. https://www.kaggle.com/usdot/flight-delays.

[3] 2024. Code Repository and Technical Report. https://anonymous.4open.science/r/CausalDAGSummarization-D29E/. Accessed: 2024-04-07.

[4] 2024. Kaggle Datasets: malicious url detection. https://www.kaggle.com/datasets/pilarpieiro/tabular-dataset-ready-for-malicious-url-detection.

[5] 2024. OpenAI ChatGPT (3.5) [Large language model]. https://openai.com/blog/chatgpt. Accessed: 2024-04-04.

[6] Tara V Anand, Adele H Ribeiro, Jin Tian, and Elias Bareinboim. 2023. Causal Effect Identification in Cluster DAGs. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[7] Arthur Asuncion and David Newman. 2007. UCI machine learning repository.

[8] Sander Beckers and Joseph Y Halpern. 2019. Abstracting causal models. In *Proceedings of the aaai conference on artificial intelligence*, Vol. 33. 2678–2685.

[9] Sourav S Bhowmick and Byron Choi. 2022. Data-driven visual query interfaces for graphs: Past, present, and (near) future. In *Proceedings of the 2022 International Conference on Management of Data*. 2441–2447.

[10] Paolo Boldi and Sebastiano Vigna. 2004. The webgraph framework I: compression techniques. In *Proceedings of the 13th international conference on World Wide Web*. 595–602.

[11] Alessandro Castelnovo, Riccardo Crupi, Fabio Mercorio, Mario Mezzanzanica, Daniele Potertì, and Daniele Regoli. 2024. Marrying LLMs with Domain Expert Validation for Causal Graph Generation. (2024).

[12] Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. 2016. Multi-level cause-effect systems. In *Artificial intelligence and statistics*. PMLR, 361–369.

[13] Krzysztof Chalupka, Pietro Perona, and Frederick Eberhardt. 2015. Visual causal feature learning. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 181–190.

[14] Wenyu Chen, Mathias Drton, and Ali Shojaie. 2023. Causal Structural Learning via Local Graphs. *SIAM Journal on Mathematics of Data Science* 5, 2 (2023), 280–305.

[15] D.M Chickering. 2002. Optimal structure identification with greedy search. *JMLR* 3, Nov (2002), 507–554.

[16] Anthony C Constantinou, Yang Liu, Kiattikun Chobtham, Zhigao Guo, and Neville K Kitson. 2021. Large-scale empirical validation of Bayesian Network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning* 131 (2021), 151–188.

[17] Diane J Cook and Lawrence B Holder. 1993. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research* 1 (1993), 231–255.

[18] Nilesh Dalvi and Dan Suciu. 2007. Efficient query evaluation on probabilistic databases. *The VLDB Journal* 16 (2007), 523–544.

[19] Xavier De Luna, Ingeborg Waernbaum, and Thomas S Richardson. 2011. Covariate selection for the nonparametric estimation of an average treatment effect. *Biometrika* 98, 4 (2011), 861–875.

[20] Cody Dunne and Ben Shneiderman. 2013. Motif simplification: improving network visualization readability with fan, connector, and clique glyphs. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3247–3256.

[21] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. 2021. COCOA: COrrelation COefficient-Aware Data Augmentation.. In *EDBT*. 331–336.

[22] Wenfei Fan, Jianzhong Li, Xin Wang, and Yinghui Wu. 2012. Query preserving graph compression. In *Proceedings of the 2012 ACM SIGMOD international conference on management of data*. 157–168.

[23] Sainyam Galhotra, Amir Gilad, Sudeepa Roy, and Babak Salimi. 2022. Hyper: Hypothetical reasoning with what-if and how-to queries using a probabilistic causal approach. In *Proceedings of the 2022 International Conference on Management of Data*. 1598–1611.

[24] Sainyam Galhotra, Romila Pradhan, and Babak Salimi. 2021. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *Proceedings of the 2021 International Conference on Management of Data*. 577–590.

[25] Markus Gangl. 2010. Causal inference in sociological research. *Annual review of sociology* 36 (2010), 21–47.

[26] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer. 1976. Some Simplified NP-Complete Graph Problems. *Theor. Comput. Sci.* 1, 3 (1976), 237–267.

[27] Dan Geiger, Thomas Verma, and Judea Pearl. 1990. Identifying independence in bayesian networks. *Networks* 20, 5 (1990), 507–534.

[28] Clark Glymour, Kun Zhang, and Peter Spirtes. 2019. Review of causal discovery methods based on graphical models. *Frontiers in genetics* 10 (2019), 524.

[29] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. 2015. Semantic Similarity from Natural Language and Ontology Analysis. *ArXiv* abs/1704.05295 (2015).

[30] Juris Hartmanis. 1982. Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson). *Siam Review* 24, 1 (1982), 90.

[31] Weidong Huang, Peter Eades, and Seok-Hee Hong. 2009. Measuring effectiveness of graph visualizations: A cognitive load perspective. *Information Visualization* 8, 3 (2009), 139–152.

[32] Johannes Huegle, Christopher Hagedorn, Lukas Boehme, Mats Poerschke, Jonas Umland, and Rainer Schlosser. 2021. MANM-CS: Data generation for benchmarking causal structure learning from mixed discrete-continuous and nonlinear data. *WHY-21 at NeurIPS* 2021 (2021).

[33] Lisa Jin and Danai Koutra. 2017. Ecoviz: Comparative vizualization of time-evolving network summaries. In *ACM SIGKDD 2017 Workshop on Interactive Data Exploration and Analytics*.

[34] Alexandra Kim, Laks VS Lakshmanan, and Divesh Srivastava. 2020. Summarizing hierarchical multidimensional data. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 877–888.

[35] Samantha Kleinberg and George Hripcsak. 2011. A review of causal inference for biomedical informatics. *Journal of biomedical informatics* 44, 6 (2011), 1102–1112.

[36] Daphne Koller and Nir Friedman. 2009. *Probabilistic graphical models: principles and techniques*. MIT press.

[37] Danai Koutra, U Kang, Jilles Vreeken, and Christos Faloutsos. 2014. Vog: Summarizing and understanding large graphs. In *Proceedings of the 2014 SIAM international conference on data mining*. SIAM, 91–99.

[38] Laks VS Lakshmanan, Jian Pei, and Jiawei Han. 2002. Quotient cube: How to summarize the semantics of a data cube. In *VLDB'02: Proceedings of the 28th International Conference on Very Large Databases*. Elsevier, 778–789.

[39] Steffen L Lauritzen and Thomas S Richardson. 2002. Chain graph models and their causal interpretations. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 64, 3 (2002), 321–348.

[40] Kyuhan Lee, Hyeonsoo Jo, Jihoon Ko, Sungsu Lim, and Kijung Shin. 2020. Ssumm: Sparse summarization of massive graphs. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 144–154.

[41] Chenhui Li, George Baciu, and Yunzhe Wang. 2015. Modulgraph: modularity-based visualization of massive graphs. In *SIGGRAPH Asia 2015 Visualization in High Performance Computing*. 1–4.

[42] Xingjie Liu, Yuanyuan Tian, Qi He, Wang-Chien Lee, and John McPherson. 2014. Distributed graph summarization. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. 799–808.

[43] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. 2018. Graph summarization methods and applications: A survey. *ACM computing surveys (CSUR)* 51, 3 (2018), 1–34.

[44] Pingchuan Ma, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. 2023. Xinsight: explainable data analysis through the lens of causality. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–27.

[45] Antonio Maccioni and Daniel J Abadi. 2016. Scalable pattern matching over compressed graphs via dedensification. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1755–1764.

[46] Sara Magliacane, Thijs Van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. 2018. Domain adaptation by using causal inference to predict invariant conditional distributions. *Advances in neural information processing systems* 31 (2018).

[47] Sebastian Maneth and Fabian Peternek. 2016. Compressing graphs by grammars. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*. IEEE, 109–120.

[48] Yasir Mehmood, Nicola Barbieri, Francesco Bonchi, and Antti Ukkonen. 2013. Csi: Community-level social influence analysis. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part II 13*. Springer, 48–63.

[49] Arpit Merchant, Michael Mathioudakis, and Yanhao Wang. 2023. Graph Summarization via Node Grouping: A Spectral Algorithm. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 742–750.

[50] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[51] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in Pre-Training Distributed Word Representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

[52] Karthika Mohan. 2022. Causal Graphs for Missing Data: A Gentle Introduction. In *Probabilistic and Causal Inference: The Works of Judea Pearl*. 655–666.

[53] Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. 2019. Accident risk prediction based on heterogeneous sparse data: New dataset and insights. In *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*. 33–42.

[54] Razieh Nabi and Ilya Shpitser. 2018. Fair inference on outcomes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[55] Saket Navlakha, Rajeev Rastogi, and Nisheeth Shrivastava. 2008. Graph summarization with bounded error. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 419–432.

[56] Xueyan Niu, Xiaoyun Li, and Ping Li. 2022. Learning Cluster Causal Diagrams: An Information-Theoretic Approach. (2022).

[57] Chris J Oates, Jessica Kasza, Julie A Simpson, and Andrew B Forbes. 2017. Repair of partly misspecified causal diagrams. *Epidemiology* 28, 4 (2017), 548–552.

[58] RT O'donnell, Ann E Nicholson, B Han, Kevin B Korb, MJ Alam, and LR Hope. 2006. Incorporating expert elicited structural information in the CaMML causal discovery program. In *Proceedings of the 19th Australian Joint Conference on Artificial Intelligence: Advances in Artificial Intelligence*. 1–16.

[59] Pekka Parviainen and Samuel Kaski. 2016. Bayesian networks for variable groups. In *Conference on Probabilistic Graphical Models*. PMLR, 380–391.

[60] Sepideh Pashami, Anders Holst, Juhee Bae, and Sławomir Nowaczyk. 2018. Causal discovery using clusters from observational data. In *FAIM'18 Workshop on CausalML, Stockholm, Sweden, July 15, 2018*.

[61] Judea Pearl. 2000. *Causality : models, reasoning, and inference*. Cambridge University Press.

[62] J. Pearl and D. Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

[63] Sriram Pemmaraju, Steven Skiena, et al. 2003. *Computational discrete mathematics: Combinatorics and graph theory with mathematica®*. Cambridge university press.

[64] Jose M Peña. 2016. Learning acyclic directed mixed graphs from observations and interventions. In *Conference on Probabilistic Graphical Models*. PMLR, 392–402.

[65] Emilija Perkovic. 2020. Identifying causal effects in maximally oriented partially directed acyclic graphs. In *Conference on Uncertainty in Artificial Intelligence*. PMLR, 530–539.

[66] Cristina Puente, José Angel Olivas, E Garrido, and R Seisdedos. 2013. Compressing the representation of a causal graph. In *2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS)*. IEEE, 122–127.

[67] Sriram Raghavan and Hector Garcia-Molina. 2003. Representing web graphs. In *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*. IEEE, 405–416.

[68] Wullianallur Raghupathi and Viju Raghupathi. 2014. Big data analytics in healthcare: promise and potential. *Health information science and systems* 2 (2014), 1–10.

[69] Ryan A Rossi and Rong Zhou. 2018. Graphzip: a clique-based sparse graph compression method. *Journal of Big Data* 5, 1 (2018), 1–14.

[70] Sudeepa Roy. 2022. Toward interpretable and actionable data analysis with explanations and causality. *Proc. VLDB Endow.* 15, 12 (2022), 3812–3820.

[71] Paul K Rubenstein, Sebastian Weichwald, Stephan Bongers, Joris M Mooij, Dominik Janzing, Moritz Grosse-Wentrup, and Bernhard Schölkopf. 2017. Causal consistency of structural equation models. *arXiv preprint arXiv:1707.00819* (2017).

[72] Babak Salimi, Johannes Gehrke, and Dan Suciu. 2018. Bias in olap queries: Detection, explanation, and removal. In *SIGMOD*. 1021–1035.

[73] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional fairness: Causal database repair for algorithmic fairness. In *Proceedings of the 2019 International Conference on Management of Data*. 793–810.

[74] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. 2021. Correlation sketches for approximate join-correlation queries. In *Proceedings of the 2021 International Conference on Management of Data*. 1531–1544.

[75] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. 2021. Toward causal representation learning. *Proc. IEEE* 109, 5 (2021), 612–634.

[76] Neil Shah, Danai Koutra, Lisa Jin, Tianmin Zou, Brian Gallagher, and Christos Faloutsos. 2017. On Summarizing Large-Scale Dynamic Graphs. *IEEE Data Eng. Bull.* 40, 3 (2017), 75–88.

[77] Amit Sharma and Emre Kiciman. 2020. DoWhy: An End-to-End Library for Causal Inference. *arXiv preprint arXiv:2011.04216* (2020).

[78] Zeqian Shen, Kwan-Liu Ma, and Tina Eliassi-Rad. 2006. Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE transactions on visualization and computer graphics* 12, 6 (2006), 1427–1439.

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

[80] Kijung Shin, Amol Ghoting, Myunghwan Kim, and Hema Raghavan. 2019. Sweg: Lossless and lossy summarization of web-scale graphs. In *The World Wide Web Conference*. 1679–1690.

[81] Maryam Shoaran, Alex Thomo, and Jens H Weber-Jahnke. 2013. Zero-knowledge private graph summarization. In *2013 IEEE International Conference on Big Data*. IEEE, 597–605.

[82] Karamjit Singh, Garima Gupta, Vartika Tewari, and Gautam Shroff. 2018. Comparative benchmarking of causal discovery algorithms. In *Proceedings of the ACM India Joint International Conference on Data Science and Management of Data*. Association for Computing Machinery, 46–56.

[83] Qi Song, Yinghui Wu, Peng Lin, Luna Xin Dong, and Hui Sun. 2018. Mining summaries for knowledge graph search. *IEEE Transactions on Knowledge and Data Engineering* 30, 10 (2018), 1887–1900.

[84] Daniel A Spielman and Nikhil Srivastava. 2008. Graph sparsification by effective resistances. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. 563–568.

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

[86] Xinwei Sun, Botong Wu, Xiangyu Zheng, Chang Liu, Wei Chen, Tao Qin, and Tie-Yan Liu. 2021. Recovering latent causal factor for generalization to distributional shifts. *Advances in Neural Information Processing Systems* 34 (2021), 16846–16859.

[87] Etsuji Suzuki, Tomohiro Shinozaki, and Eiji Yamamoto. 2020. Causal diagrams: pitfalls and tips. *Journal of epidemiology* 30, 4 (2020), 153–162.

[88] Yuanyuan Tian, Richard A Hankins, and Jignesh M Patel. 2008. Efficient aggregation for graph summarization. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. 567–580.

[89] Santtu Tikka, Jouni Helske, and Juha Karvanen. 2021. Clustering and Structural Robustness in Causal Diagrams. *arXiv preprint arXiv:2111.04513* (2021).

[90] Hal R Varian. 2016. Causal inference in economics and marketing. *Proceedings of the National Academy of Sciences* 113, 27 (2016), 7310–7315.

[91] Aniket Vashishtha, Abbavaram Gowtham Reddy, Abhinav Kumar, Saketh Bachu, Vineeth N Balasubramanian, and Amit Sharma. 2023. Causal inference using llm-guided discovery. *arXiv preprint arXiv:2310.15117* (2023).

[92] Yuhao Wen, Xiaodan Zhu, Sudeepa Roy, and Jun Yang. 2018. Interactive summarization and exploration of top aggregate query answers. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, Vol. 11. NIH Public Access, 2196.

[93] Marco A Wiering et al. 2002. Evolving causal neural networks. In *Benelearn'02: Proceedings of the Twelfth Belgian-Dutch Conference on Machine Learning*. 103–108.

[94] Raymond W. Yeung. 2008. *Information Theory and Network Coding* (1 ed.). Springer Publishing Company, Incorporated.

[95] Quinton Yong, Mahdi Hajiabadi, Venkatesh Srinivasan, and Alex Thomo. 2021. Efficient graph summarization using weighted lsh at billion-scale. In *Proceedings of the 2021 International Conference on Management of Data*. 2357–2365.

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

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

[98] Brit Youngmann, Michael Cafarella, Babak Salimi, and Zeng Anna. 2023. Causal Data Integration. *Proceedings of the VLDB Endowment* 16, 1- (2023), 2665–2659.

[99] Jiji Zhang. 2008. On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. *Artificial Intelligence* 172, 16-17 (2008), 1873–1896.

[100] Boxiang Zhao, Shuliang Wang, Lianhua Chi, Qi Li, Xiaojia Liu, and Jing Geng. 2023. Causal Discovery via Causal Star Graphs. *ACM Transactions on Knowledge Discovery from Data* 17, 7 (2023), 1–24.

[101] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2009. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment* 2, 1 (2009), 718–729.

[102] Jiongli Zhu, Sainyam Galhotra, Nazanin Sabri, and Babak Salimi. 2023. Consistent Range Approximation for Fair Predictive Modeling. *Proceedings of the VLDB Endowment* 16, 11 (2023), 2925–2938.

[103] Shengyu Zhu, Ignavier Ng, and Zhitang Chen. 2020. Causal Discovery with Reinforcement Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A SEMIGRAPHOID-AXIOMS

The semi-graphoid axioms are the following:

(1) Triviality: $I(A;\emptyset|C)$=0.
(2) Symmetry: $I(A;B|C)$=0 $\Longrightarrow$ $I(A;C|B)$=0.
(3) Decomposition: $I(A;BD|C)$=0 $\Longrightarrow$ $I(A;B|D)$=0.
(4) Contraction: $I(A;B|C)$=0, $I(A;D|BC)$=0 $\Longrightarrow$ $I(A;BD|C)$=0.
(5) Weak Union: $I(A;BD|C)$=0 $\Longrightarrow$ $I(A;B|CD)$=0, $I(A;D|BC)$=0.

The semi-graphoid axioms can be summarized using the following identity, which follows from the *chain-rule* for mutual information [94].

$$I(A;BD|C)=0 \text{ if and only if } I(A;B|C)=0 \text{ and } I(A;D|BC)=0 \quad (4)$$

B SEMANTIC SIMILARITY

The pair-wise semantic similarity score for the variables of the Flights dataset is depicted in Table 4.

C PROOFS

Next, we provide the missing proofs.

C.1 Proofs for Section 3

Proof of Lemma 3.1. Let P be a directed path from A to B, such that $|P| \geq 2$. Let X be A's successor in P, and Y be B's predecessor in P. By our assumption that $|P| \geq 2$, $X \notin \{A, B\}$, but Y may be the same as X. Now, consider the graph H. By definition, H contains a node AB, with an incoming edge from Y, and an outgoing edge to X. If $X = Y$, we immediately get the cycle $AB \to X \to AB$. Otherwise, we consider the subpath P' (of P) from X to Y ($X \underset{P'}{\rightsquigarrow} Y$) in G. This results in the following cycle in H: $Y \to AB \to X \underset{P'}{\rightsquigarrow} Y$.

Now, suppose that H contains the cycle C. C must contain the node AB. Otherwise, the cycle is included in G, which leads to a contradiction that G is a DAG. Let Y and X be the incoming and outgoing vertices, respectively, to AB in C. Then, there is a directed path P from X to Y in H that avoids AB. That is, every vertex and edge on the path P belongs to G as well. Hence, P is a directed path from X to Y in G. Since Y is incoming to AB, then Y is incoming to either A or B in G. Assume, wlog, that $Y \to A \in E$. Since X is an outgoing vertex from AB, then it is outgoing from from either A or B (or both). If X is outgoing from A, then we get the following cycle in G: $Y \to A \to X \underset{P}{\rightsquigarrow} Y$. Since G is a DAG, this brings us to a contradiction. Therefore, X must be outgoing from B and not A. But this gives us the following directed path from B to A:

$$A \leftarrow Y \underset{P}{\rightsquigarrow} X \leftarrow B.$$

This completes the proof. □

Next, we show that the causal DAG summarization problem is *NP*-hard via a reduction from the k-Min-Cut problem [30]. Let G be an undirected graph with weighted edges (i.e., $w : \mathsf{V}(G) \to \mathbb{R}_{\geq 0}$). The k-Min-Cut problem consists of partitioning $\mathsf{V}(G)$ into k disjoint clusters so as to minimize the sum of weights of the edges joining vertices in different clusters. It is well-known that k-Min-Cut is NP-hard even if $k = 2$ [26]. In other words, deciding whether there exists a k-clustering of $\mathsf{V}(G)$ to clusters $\{V_1, \ldots, V_k\}$, where the sum of weights of edges between vertices in distinct clusters is at most a given threshold γ is NP-hard.

Proof of Theorem 3.2. In this proof we make the assumption that $InterSim(\mathbf{S}) = \sum_{u,v \in \mathbf{S}} InterSim(\{u, v\})$ where $\mathbf{S} \subseteq \mathsf{V}(G)$. Given a DAG G, a similarity threshold τ, and its summary graph $\mathcal{H}$, it is easy to verify that $\sum_{\mathbf{S} \in \mathsf{V}(\mathcal{H})} InterSim(\mathbf{S}) \geq \tau$. Therefore, the causal DAG summarization problem is in NP.

We prove hardness by reduction from k-Min-Cut. Let G be an undirected weighted graph, and let $\gamma > 0$ be the threshold for the k-Min-Cut problem. For every pair of vertices $u, v \in \mathsf{V}(G)$, define $InterSim(\{u, v\}) \stackrel{\text{def}}{=} w(u, v)$; define $\tau \stackrel{\text{def}}{=} \sum_{u,v \in \mathsf{V}(G)} w(u, v) - \gamma$. Let G' be the DAG that results from G by orienting its edges such that no directed cycles are generated (this is simple to do by fixing a complete order over the vertices, and orienting the edges accordingly). It is easy to see that there is a k-clustering $\{V_1, \ldots, V_k\}$ of G where the sum of weights of edges between vertices in distinct clusters is at least γ if and only if $\sum_{i=1}^{k} InterSim(V_i) \geq \tau$ in G'. □

C.2 Proofs for Section 4

Next, we prove some simple lemmas that will be useful later on. We denote by $\mathcal{H}_{UV}$ the summary DAG where nodes U and V are contracted.

Lemma C.1. *The following holds:*

$$\pi_{\mathcal{H}_{UV}}(\mathbf{X_{UV}}) = \pi_{\mathcal{G}_{UV}}(U) \qquad \pi_{\mathcal{G}_{UV}}(V) = \pi_{\mathcal{G}_{UV}}(U) \cup \{U\} \quad (5)$$

$$\mathsf{ch}_{\mathcal{H}_{UV}}(\mathbf{X_{UV}}) = \mathsf{ch}_{\mathcal{G}_{UV}}(V) \qquad \mathsf{ch}_{\mathcal{G}_{UV}}(U) = \mathsf{ch}_{\mathcal{G}_{UV}}(V) \cup \{V\} \quad (6)$$

$$\mathsf{NDsc}_{\mathcal{H}_{UV}}(\mathbf{X_{UV}}) = \mathsf{NDsc}_{\mathcal{G}_{UV}}(U) \qquad \mathsf{NDsc}_{\mathcal{G}_{UV}}(V) = \mathsf{NDsc}_{\mathcal{G}_{UV}}(U) \cup \{U\} \quad (7)$$

Lemma C.2. *Let $T \in \mathcal{V}$ such that $T \notin \{U, V\} \cup \mathsf{ch}_{\mathcal{G}}(U) \cup \mathsf{ch}_{\mathcal{G}}(V)$. Then it holds that:*

$$\pi_{\mathcal{G}_{UV}}(T) = \pi_{\mathcal{H}_{UV}}(T) \text{ and} \quad (8)$$

$$\mathsf{NDsc}_{\mathcal{G}_{UV}}(T) \backslash \{UV\} = \mathsf{NDsc}_{\mathcal{H}}(T) \backslash \{\mathbf{X_{UV}}\} \text{ and} \quad (9)$$

$$\{U, V\} \subseteq \mathsf{NDsc}_{\mathcal{G}_{UV}}(T) \text{ if and only if } \mathbf{X_{UV}} \in \mathsf{NDsc}_{\mathcal{H}}(T) \quad (10)$$

Now, let $T \in \mathcal{V}$ such that $T \notin \{U, V\} \cup \pi_{\mathcal{G}}(U) \cup \pi_{\mathcal{G}}(V)$. Then:

$$\mathsf{ch}_{\mathcal{G}_{UV}}(T) = \mathsf{ch}_{\mathcal{H}_{UV}}(T) \text{ and} \quad (11)$$

$$\mathsf{Dsc}_{\mathcal{G}_{UV}}(T) \backslash \{U, V\} = \mathsf{Dsc}_{\mathcal{H}_{UV}}(T) \backslash \{\mathbf{X_{UV}}\} \quad (12)$$

$$\{U, V\} \subseteq \mathsf{Dsc}_{\mathcal{G}_{UV}}(T) \text{ if and only if } \mathbf{X_{UV}} \in \mathsf{Dsc}_{\mathcal{H}_{UV}}(T) \quad (13)$$

Proof of Lemma C.1. By the definition of G', it holds that $\pi_{G'}(u) = \pi_G(u) \cup \pi_G(v)$. Since $(u, v) \in \mathsf{E}(G')$, then $\pi_{G'}(v) = \pi_{G'}(u) \cup \{u\}$. Similarly, $\mathsf{ch}_{G'}(v) = \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$, and since $(u, v) \in \mathsf{E}(G')$, then $\mathsf{ch}_{G'}(u) = \mathsf{ch}_{G'}(v) \cup \{v\}$. By definition of edge contraction, it holds that $\pi_H(X_{uv}) = \pi_G(u) \cup \pi_G(v) = \pi_{G'}(u)$, proving (5). Also, by definition of edge contraction, it holds that $\mathsf{ch}_H(X_{uv}) = \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v) = \mathsf{ch}_{G'}(v)$, proving (6).

We now prove (7). Let $t \in \mathsf{NDsc}_H(X_{uv})$. If $t \notin \mathsf{NDsc}_{G'}(u)$, then $t \in \mathsf{Dsc}_{G'}(u) \backslash \{v\}$. This means that there is a directed path P from u to t in G'. Let s be the first vertex on this path (after u). Since $s \in \mathsf{ch}_{G'}(u) \backslash \{v\}$, then by the definition of G', $s \in \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. By the definition of edge contraction, $s \in \mathsf{ch}_H(X_{uv})$. Since $s \notin uv \cup \pi_G(u) \cup \pi_G(v)$, then every directed path starting at s in G remains a directed path in H. But this means that there is a directed path from X_{uv} to t (via s); contradicting the assumption that $t \in \mathsf{NDsc}_H(X_{uv})$. Now, let $t \in \mathsf{NDsc}_{G'}(u)$. If $t \notin \mathsf{NDsc}_H(X_{uv})$, then $t \in \mathsf{Dsc}_H(X_{uv}) \backslash \{X_{uv}\}$. This means that there is a directed path P from X_{uv} to t in H. Let s be the first vertex on this path (after X_{uv}). Since $s \in \mathsf{ch}_H(X_{uv})$, then by the definition of H, $s \in \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. But then, by the definition of G', it holds that $s \in \mathsf{ch}_{G'}(u)$. Since no edges are removed by the transition from G to G', there is a directed path from u to (via s) in G'; contradicting the assumption that $t \in \mathsf{NDsc}_{G'}(u)$. □

Table 4: Semantic similarity scores for the variables in the Flights Delay dataset.

	Dep. State	Dep. City	Humidity	Airport	Density	Temp.	Traffic	Airline	Prec.	Fleet size	Dep. Delay
Dep. State	1	0.9	0.4	0.5	0.7	0.6	0.5	0.7	0.5	0.3	0.2
Dep. City		1	0.5	0.5	0.6	0.7	0.6	0.7	0.4	0.5	0.3
Humidity			1	0.3	0.5	0.9	0.4	0.5	0.8	0.2	0.1
Airport				1	0.5	0.5	0.8	0.8	0.2	0.8	0.8
Density					1	0.4	0.8	0.6	0.3	0.4	0.5
Temp.						1	0.3	0.2	0.9	0.3	0.5
Traffic							1	0.8	0.5	0.4	0.8
Airline								1	0.4	0.8	0.9
Prec.									1	0.2	0.4
Fleet size										1	0.6
Dep. Delay											1

Proof of Lemma C.2. By the definition of contraction, the only vertices in G whose parent-set can potentially change following the contraction of u and v belong to the set $uv \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. By the definition of $\mathsf{E}(G')$, the only vertices in G whose parent-set can potentially change belong to the set $uv \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. Therefore, if $t \notin uv \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$, then $\pi_{G'}(t) = \pi_H(t) = \pi_G(t)$. This proves (8).

By the definition of contraction, the only vertices in G whose child-set can potentially change following the contraction of u and v belong to the set $uv \cup \pi_G(u) \cup \pi_G(v)$. By the definition of $\mathsf{E}(G')$, the only vertices in G whose child-set can potentially change belong to the set $uv \cup \pi_G(u) \cup \pi_G(v)$. Therefore, if $t \notin uv \cup \pi_G(u) \cup \pi_G(v)$, then $\mathsf{ch}_{G'}(t) = \mathsf{ch}_H(t) = \mathsf{ch}_G(t)$. This proves (11).

We now prove (9); Let $s \in \mathsf{NDsc}_H(t) \backslash \{X_{uv}\}$. If $s \notin \mathsf{NDsc}_{G'}(t)$, then $s \in \mathsf{Dsc}_{G'}(t)$. That is, there is a directed path P from t to s in G'. Let us assume wlog that P is the shortest directed path from t to s in G'. By this assumption, exactly one of the following holds: (1) $u, v \notin \mathsf{V}(P)$ (2) $u \in \mathsf{V}(P)$, $v \notin nodes(P)$ (3) $v \in \mathsf{V}(P)$, $u \notin nodes(P)$, or (4) $(u, v) \in \mathsf{E}(P)$. In the first case, every edge of P is also an edge of $\mathsf{E}(G)$, that does not enter or exit $\{u, v\}$. Therefore, P is a directed path in H, a contradiction. In case (2), since $\pi_{G'}(u) = \pi_H(X_{uv})$ (see (5)), and $\mathsf{ch}_{G'}(u) = \mathsf{ch}_H(X_{uv}) \backslash \{v\}$ (see (6)), then the path with nodes $X_{uv} \cup (\mathsf{V}(P) \setminus \{u\})$, is a directed path in H from s to t; a contradiction. In case (3), since $\mathsf{ch}_{G'}(v) = \mathsf{ch}_H(X_{uv})$ (see (6)), and $\pi_{G'}(v) = \pi_H(X_{uv}) \backslash \{u\}$ (see (5)), then the path with nodes $X_{uv} \cup (\mathsf{V}(P) \setminus \{v\})$, is a directed path in H from s to t; a contradiction. Finally, if $(u, v) \in \mathsf{E}(P)$, then since $\pi_{G'}(u) = \pi_H(X_{uv})$ and $\mathsf{ch}_{G'}(v) = \mathsf{ch}_H(X_{uv})$, then the path with nodes $X_{uv} \cup (\mathsf{V}(P) \backslash uv)$, is a directed path from s to t in H; a contradiction. For the other direction, let $s \in \mathsf{NDsc}_{G'}(t) \backslash uv$. If $s \notin \mathsf{NDsc}_H(t)$, then there is a directed path P from t to s in H. If $X_{uv} \notin \mathsf{V}(P)$, then $\mathsf{E}(P) \subseteq \mathsf{E}(G) \subseteq \mathsf{E}(G')$, and hence P is a directed path from t to s in G'. Otherwise, if $X_{uv} \in \mathsf{V}(P)$, then since $\pi_H(X_{uv}) = \pi_{G'}(u)$, $\mathsf{ch}_H(X_{uv}) = \mathsf{ch}_{G'}(v)$, and $(u, v) \in \mathsf{E}(G')$, then replacing X_{uv}, with the edge (u, v) results in a directed t, s-path in G'; a contradiction. □

Proof of Theorem 4.1. We first prove that $\Sigma_{\mathrm{RB}}(G') \implies \Sigma_{\mathrm{RB}}(H)$. We divide to cases. Let $(X_i; B_i | \pi_H(X_i)) \in \Sigma_{\mathrm{RB}}(H)$, where $X_{uv} \notin B_i \cup \pi_H(X_i) \cup \{X_i\}$. In particular, $X_i \in \mathsf{V}(G)$, and $X_i \notin \{u, v\} \cup \mathsf{ch}_H(X_{uv}) = \{u, v\} \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. By (9), we have that $\pi_{G'}(X_i) = \pi_H(X_i)$, and that $\mathsf{NDsc}_{G'}(X_i) \backslash uv = \mathsf{NDsc}_H(X_i) \backslash X_{uv}$. Therefore, we have that $(X_i; \mathsf{NDsc}_H(X_i) \backslash \{X_{uv}\} | \pi_{G'}(X_i))_{G'}$. Since $B_i \subseteq \mathsf{NDsc}_H(X_i) \backslash \{X_{uv}\}$, then, by decomposition, we have that $\Sigma_{\mathrm{RB}}(G') \implies (X_i; B_i | \pi_H(X_i))$.

Now, let $(X_i; X_{uv} B_i | \pi_H(X_i)) \in \Sigma_{\mathrm{RB}}(H)$. In this case as well $X_i \in \mathsf{V}(G)$, and $X_i \notin \{u, v\} \cup \mathsf{ch}_H(X_{uv}) = \{u, v\} \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. By (9), we have that $\pi_{G'}(X_i) = \pi_H(X_i)$, and that $\mathsf{NDsc}_{G'}(X_i) \backslash uv = \mathsf{NDsc}_H(X_i) \backslash X_{uv}$. By , we have that $X_{uv} \in \mathsf{NDsc}_H(X_i)$ iff $uv \subseteq \mathsf{NDsc}_{G'}(X_i)$. Therefore, $B_i X_{uv} \subseteq \mathsf{NDsc}_H(X_i)$ iff $B_i uv \subseteq \mathsf{NDsc}_{G'}(X_i)$. This means that $\Sigma_{\mathrm{RB}}(G') \implies (X_i; \mathsf{NDsc}_{G'}(X_i) | \pi_{G'}(X_i))$. By decomposition, we have that $\Sigma_{\mathrm{RB}}(G') \implies (X_i; B_i uv | \pi_H(X_i))$ as required.

Now, suppose that $X_{uv} \in \pi_H(X_i)$, or that $X_i \in \mathsf{ch}_H(X_{uv})$. Since $X_i \in \mathsf{V}(G) \backslash \{u, v\}$, then by (6), we have that $X_i \in \mathsf{ch}_{G'}(v) \backslash \{v\}$. Therefore, $\pi_{G'}(X_i) = \pi_H(X_i) \backslash \{X_{uv}\} \cup \{u, v\}$. By (9), we have that:

$$\begin{aligned}
\mathsf{NDsc}_{G'}(X_i) \backslash \pi_{G'}(X_i) &= \mathsf{NDsc}_{G'}(X_i) \backslash (\pi_H(X_i) \backslash \{X_{uv}\} \cup uv) \\
&= (\mathsf{NDsc}_{G'}(X_i) \backslash uv) \backslash (\pi_H(X_i) \backslash \{X_{uv}\}) \\
&\underbrace{=}_{(9)} (\mathsf{NDsc}_H(X_i) \backslash \{X_{uv}\}) \backslash (\pi_H(X_i) \backslash \{X_{uv}\}) \\
&= \mathsf{NDsc}_H(X_i) \backslash \pi_H(X_i)
\end{aligned}$$

Therefore, $\Sigma_{\mathrm{RB}}(G') \implies (X_i; \mathsf{NDsc}_H(X_i) \backslash \pi_H(X_i) \mid \pi_H(X_i) \backslash \{X_{uv}\} \cup \{u, v\})$. Finally, we consider the case where $X_i = X_{uv}$. By construction of G', and by (7), it holds that:

$$\Sigma_{\mathrm{RB}}(G') \implies (u; \mathsf{NDsc}_{G'}(u) \backslash \pi_{G'}(u) \mid \pi_{G'}(u)) \tag{14}$$

$$\Sigma_{\mathrm{RB}}(G') \implies (v; \mathsf{NDsc}_{G'}(u) \backslash \pi_{G'}(u) \mid \pi_{G'}(u) \cup \{u\}) \tag{15}$$

By applying the contraction axiom on (14) and (15), we get that

$$\Sigma_{\mathrm{RB}}(G') \implies (uv; \mathsf{NDsc}_{G'}(u) \backslash \pi_{G'}(u) \mid \pi_{G'}(u)).$$

Using the fact that hat $\pi_H(X_{uv}) = \pi_{G'}(u)$ (see (5)), and that $\mathsf{NDsc}_H(X_{uv}) = \mathsf{NDsc}_{G'}(u)$ (see (7)), we get that

$$\Sigma_{\mathrm{RB}}(G') \implies (uv; \mathsf{NDsc}_H(X_{uv}) \backslash \pi_H(X_{uv}) \mid \pi_H(X_{uv})).$$

Since $B_i \subseteq \mathsf{NDsc}_H(X_{uv}) \backslash (\pi_H(X_{uv}) \cup \{X_{uv}\})$, this proves the claim.

Now, for the other direction. Let $(X_i; B_i \mid \pi_{G'}(X_i)) \in \Sigma_{\mathrm{RB}}(G')$. If $u, v \notin X_i \cup B_i \cup \pi_{G'}(X_i))$, then $X_i \notin \{u, v\} \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. By (9), it holds that $\pi_H(X_i) = \pi_{G'}(X_i)$, and that $\mathsf{NDsc}_{G'}(X_i) \backslash uv = \mathsf{NDsc}_H(X_i) \backslash X_{uv}$. Since $B_i \subseteq \mathsf{NDsc}_{G'}(X_i) \backslash uv = \mathsf{NDsc}_H(X_i) \backslash X_{uv}$, then $\Sigma_{\mathrm{RB}}(H) \implies (X_i; B_i \mid \pi_{G'}(X_i))$.

If $uv \subseteq B_i$, then $u, v \notin \pi_{G'}(X_i)$, then $X_i \notin uv \cup \mathsf{ch}_G(u) \cup \mathsf{ch}_G(v)$. By (9), we have that $\pi_H(X_i) = \pi_{G'}(X_i)$, and that $\mathsf{NDsc}_H(X_i) \backslash X_{uv} = \mathsf{NDsc}_{G'}(X_i) \backslash uv$. Therefore, $B_i \backslash uv \subseteq \mathsf{NDsc}_H(X_i)$, and by (10), if $uv \subseteq B_i \subseteq \mathsf{NDsc}_{G'}(X_i)$, then $X_{uv} \in \mathsf{NDsc}_H(X_i)$. Therefore, $\Sigma_{\mathrm{RB}}(H) \implies$

$(X_i; B_i \backslash uv \cup X_{uv} \mid \pi_{G'}(X_i))$, and since $X_{uv} = uv$, then $\Sigma_{\text{RB}}(H) \implies (X_i; B_i \mid \pi_{G'}(X_i))$.

Since $(u, v) \in \mathtt{E}(G')$, we are left with two other cases. First, that $(u; B_u \mid \pi_{G'}(u))$, and second $(v; B_v \mid \pi_{G'}(v))$. By d-separation in H, the following holds:

$$\begin{aligned}\Sigma_{\text{RB}}(H) &\implies (X_{uv}; \mathtt{NDsc}_H(X_{uv}) \backslash \pi_H(X_{uv}) \mid \pi_H(X_{uv})) \\ &\underbrace{\implies (X_{uv}; \mathtt{NDsc}_{G'}(u) \backslash \pi_{G'}(u) \mid \pi_{G'}(u))}_{(5),\ (7)} \\ &\implies (uv; \mathtt{NDsc}_{G'}(u) \backslash \pi_{G'}(u) \mid \pi_{G'}(u)) \qquad (16)\end{aligned}$$

By (5), it holds that $\pi_{G'}(v) = \pi_{G'}(u) \cup \{u\}$. By (7), it holds that

$$\begin{aligned}B_v \subseteq \mathtt{NDsc}_{G'}(v) \backslash \pi_{G'}(v) &= (\mathtt{NDsc}_{G'}(u) \cup \{u\}) \backslash (\pi_{G'}(u) \cup \{u\}) \\ &= \mathtt{NDsc}_{G'}(u) \backslash \pi_{G'}(u)\end{aligned}$$

Therefore, $B_v \cup B_u \subseteq \mathtt{NDsc}_{G'}(u) \backslash \pi_{G'}(u)$. In other words, by (16), we have that;

$$\Sigma_{\text{RB}}(H) \implies (uv; B_u \cup B_v \mid \pi_{G'}(u)) \text{ if and only if}$$
$$\Sigma_{\text{RB}}(H) \implies (u; B_u \cup B_v \mid \pi_{G'}(u)), (v; B_u \cup B_v \mid \pi_{G'}(u) \cup \{u\})$$

Since $\pi_{G'}(v) = \pi_{G'}(u) \cup \{u\}$, then overall, we have that $\Sigma_{\text{RB}}(H) \implies (u; B_u \mid \pi_{G'}(u))$, and $\Sigma_{\text{RB}}(H) \implies (v; B_v \mid \pi_{G'}(v))$. This completes the proof. □

To prove Theorem 4.2, we first show the following lemma that establishes the connection between d-separation on the canonical causal DAG and the original causal DAG.

Lemma C.3. *Let $\mathcal{G}$ and $\mathcal{G}'$ be causal DAGs defined over the same set of nodes, i.e., $\mathtt{V}(\mathcal{G}) = \mathtt{V}(\mathcal{G}')$, where $\mathcal{G}'$ is a supergraph of $\mathcal{G}$ ($\mathtt{E}(\mathcal{G}') \supseteq \mathtt{E}(\mathcal{G})$). Then, for any three disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \subseteq \mathtt{V}(\mathcal{G})$, it holds that: $(\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y} | \mathbf{Z})_{\mathcal{G}'} \implies (\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y} | \mathbf{Z})_{\mathcal{G}}$*

Proof of Lemma C.3. Suppose that X and Y are d-separated by Z in G' (i.e., $(X; Y | Z)_{G'}$). If X and Y are d-connected by Z in G, then let P denote the unblocked path between X and Y, relative to Z. Since $\mathtt{E}(G') \supseteq \mathtt{E}(G)$, then clearly P is a path in G' as well. Consider any triple (x, w, y) on this path. If this triple has one of the forms

$$\{x \to w \to y, x \leftarrow w \leftarrow y, x \leftrightarrow w \to y, x \leftarrow w \leftrightarrow y\},$$

then since P is unblocked in G, relative to Z, then $w \notin Z$. Since $w \notin Z$, then the subpath (x, w, y) is also unblocked in G'. If the triple has one of the forms:

$$\{x \leftrightarrow w \leftarrow y, x \to w \leftrightarrow y, x \to w \leftarrow y\},$$

then since P is unblocked in G, relative to Z, then $\mathtt{Dsc}_G(w) \cap Z \neq \emptyset$. Since $\mathtt{E}(G') \supseteq \mathtt{E}(G)$, then $\mathtt{Dsc}_G(w) \subseteq \mathtt{Dsc}_{G'}(w)$. Therefore, $\mathtt{Dsc}_{G'}(w) \cap Z \neq \emptyset$. Consequently, we have, again, that the subpath (x, w, y) is unblocked in G'. Overall, we get that every triple (x, w, y) on the path P is unblocked in G', relative to Z, and hence X and Y are d-connected in G', a contradiction.

By definition, G' is compatible with G'. Therefore, if X and Y are d-connected by Z in G', then by the completeness of d-separation, there exists a probability distribution that factorizes according to G' in which the CI $(X; Y | Z)$ does not hold. This proves completeness. □

Proof of Theorem 4.2. Let $\mathcal{G}_{\mathcal{H}}$ denote the canonical causal DAG corresponding to $\mathcal{H}$. By Theorem 4.1, $\Sigma_{\text{RB}}(\mathcal{H}) \equiv \Sigma_{\text{RB}}(\mathcal{G}_{\mathcal{H}})$. Therefore, $(\mathbf{X} \perp\!\!\!\perp_d \mathbf{Y} | \mathbf{Z})_{\mathcal{H}} \iff (f^{-1}(\mathbf{X}) \perp\!\!\!\perp_d f^{-1}(\mathbf{Y}) | f^{-1}(\mathbf{Z}))_{\mathcal{G}_{\mathcal{H}}}$ Since $\mathtt{E}(\mathcal{G}) \subseteq \mathtt{E}(\mathcal{G}_{\mathcal{H}})$, the claim immediately follows from Lemma C.3. □

C.3 Proofs for Section 6

We next show a smile lemma that will be useful for proving the soundness and completeness of do-calculus in summary graphs.

Lemma C.4. *Let G be ADMG, and let G' be an ADMG where $\mathtt{V}(G') = \mathtt{V}(G)$, and $\mathtt{E}(G') \supseteq \mathtt{E}(G)$. Let $A, B, C \subseteq \mathtt{V}(G)$ be disjoint sets of variables, and let $X, Z \subseteq \mathtt{V}(G)$. Then:*

$$(A; B | C)_{G'_{\overline{X}\underline{Z}}} \implies (A; B | C)_{G_{\overline{X}\underline{Z}}} \qquad (17)$$

Corollary C.4.1. *Let G be ADMG, and let (H, f) be a summary-DAG for G. Let $A, B, C \subseteq \mathtt{V}(H)$ be disjoint sets of nodes, and let $X, Z \subseteq \mathtt{V}(H)$. Then:*

$$(A; B | C)_{H_{\overline{X}\underline{Z}}} \implies (\mathbf{A}; \mathbf{B} | \mathbf{C})_{G_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}} \qquad (18)$$

where for $U \subseteq \mathtt{V}(H)$, we denote $\mathbf{U} \stackrel{def}{=} f(U)$.

Theorem C.5 (Soundness of Do-Calculus in supergraphs). *Let $\mathcal{G}$ be a causal DAG encoding an interventional distribution $P(\cdot \mid do(\cdot))$. Let $\mathcal{G}'$ be a causal DAG where $\mathtt{V}(\mathcal{G}) = \mathtt{V}(\mathcal{G}')$ and $\mathtt{E}(\mathcal{G}) \subseteq \mathtt{E}(\mathcal{G}')$. For any disjoint subsets $\mathbf{X}, \mathbf{Y}, \mathbf{Z}, \mathbf{W} \subseteq \mathtt{V}(\mathcal{G})$, the following three rules hold:*

$$\mathbf{R}_1: \quad (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} | \mathbf{X}, \mathbf{W})_{\mathcal{G}'_{\overline{\mathbf{X}}}} \implies P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{Z}, \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{W})$$
$$\mathbf{R}_2: \quad (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} | \mathbf{X}, \mathbf{W})_{\mathcal{G}'_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}} \implies P(\mathbf{Y} \mid do(\mathbf{X}), do(\mathbf{Z}), \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{Z}, \mathbf{W})$$
$$\mathbf{R}_3: \quad (\mathbf{Y} \perp\!\!\!\perp \mathbf{Z} | \mathbf{X}, \mathbf{W})_{\mathcal{G}'_{\overline{\mathbf{X}\mathbf{Z}(\mathbf{W})}}} \implies P(\mathbf{Y} \mid do(\mathbf{X}), do(\mathbf{Z}), \mathbf{W}) = P(\mathbf{Y} \mid do(\mathbf{X}), \mathbf{W})$$

where $\mathbf{Z}(\mathbf{W})$ is the set of nodes in $\mathbf{Z}$ that are not ancestors of any node in $\mathbf{W}$. That is, $\mathbf{Z}(\mathbf{W}) = \mathbf{Z} \backslash \mathtt{Ancs}_{\mathcal{G}'}(\mathbf{W})$ where $\mathtt{Ancs}_{\mathcal{G}'}(\mathbf{W}) \stackrel{def}{=} \bigcup_{W \in \mathbf{W}} \mathtt{Ancs}_{\mathcal{G}'}(W)$.

Theorem C.6 (Soundness of Do-Calculus in supergraphs). *Let G be a causal BN (CBN) encoding an interventional distributions $P(\cdot \mid do(\cdot))$. Let G' be an ADMG where $\mathtt{E}(G) \subseteq \mathtt{E}(G')$. For any disjoint subsets $X, Y, Z, W \subseteq \mathtt{V}(G)$, the following three rules hold:*

$$\mathbf{R}_1: \quad (Y; Z | X, W)_{G'_{\overline{X}}} \implies P(Y \mid do(X), Z, W) = P(Y \mid do(X), W)$$
$$\mathbf{R}_2: \quad (Y; Z | X, W)_{G'_{\overline{X}\underline{Z}}} \implies P(Y \mid do(X), do(Z), W) = P(Y \mid do(X), Z, W)$$
$$\mathbf{R}_3: \quad (Y; Z | X, W)_{G'_{\overline{XZ(W)}}} \implies P(Y \mid do(X), do(Z), W) = P(Y \mid do(X), W)$$

where $Z(W)$ is the set of vertices in Z that are not ancestors of any vertex in W. That is, $Z(W) = Z \backslash \mathtt{Ancs}_{G'}(W)$ where $\mathtt{Ancs}_{G'}(W) \stackrel{def}{=} \bigcup_{w \in W} \mathtt{Ancs}_{G'}(w)$.

Proof of Lemman C.4. We show that $\mathtt{E}(G_{\overline{X}\underline{Z}}) \subseteq \mathtt{E}(G'_{\overline{X}\underline{Z}})$, and the claim then follows from Theorem **??**. Let $(u, v) \in \mathtt{E}(G_{\overline{X}\underline{Z}}) \subseteq \mathtt{E}(G) \subseteq \mathtt{E}(G')$. By definition, $u \notin Z$ and $v \notin X$. But this means that $(u, v) \in \mathtt{E}(G'_{\overline{X}\underline{Z}})$, which completes the proof. □

Proof of Corollary C.4.1. Let $G_{H_{\overline{X}\underline{Z}}}$ denote the grounded DAG corresponding to $H_{\overline{X}\underline{Z}}$. By Theorem 4.1, $\Sigma_{\text{RB}}(H_{\overline{X}\underline{Z}}) \equiv \Sigma_{\text{RB}}(G_{H_{\overline{X}\underline{Z}}})$,

and hence $(A; B|C)_{H_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}}$ if and only if $(\mathbf{A}; \mathbf{B}|\mathbf{C})_{G_{H_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}}}$. Since $\mathtt{E}(G_H) \supseteq \mathtt{E}(G)$, then by Lemma C.4, it holds that if $(\mathbf{A}; \mathbf{B}|\mathbf{C})_{G_{H_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}}}$, then $(\mathbf{A}; \mathbf{B}|\mathbf{C})_{G_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}}$. Overall, we have that:

$$(A; B|C)_{H_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}} \Leftrightarrow (\mathbf{A}; \mathbf{B}|\mathbf{C})_{G_{H_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}}} \implies (\mathbf{A}; \mathbf{B}|\mathbf{C})_{G_{\overline{\mathbf{X}}\underline{\mathbf{Z}}}} \tag{19}$$

which proves the claim. □

Proof of Theorem C.6. If $(Y; Z|X, W)_{G'_{\overline{X}}}$, then by Lemma C.4, it holds that $(Y; Z|X, W)_{G_{\overline{X}}}$. By the soundness of do-calculus for causal BNs, we get that $P(\boldsymbol{Y} \mid do(\boldsymbol{X}), \boldsymbol{Z}, \boldsymbol{W}) = P(\boldsymbol{Y} \mid do(\boldsymbol{X}), \boldsymbol{W})$. If $(Y; Z|X, W)_{G'_{\overline{X}\underline{Z}}}$, then by Lemma C.4, it holds that $(Y; Z|X, W)_{G_{\overline{X}\underline{Z}}}$. By the soundness of do-calculus for causal BNs, we get that $P(Y \mid do(X), do(Z), W) = P(Y \mid do(X), Z, W)$. Finally, if $(Y; Z|X, W)_{G'_{\overline{XZ(W)}}}$, then by Lemma C.4, it holds that $(Y; Z|X, W)_{G_{\overline{XZ(W)}}}$. By the soundness of do-calculus for causal BNs, we get that $P(Y \mid do(X), do(Z), W) = P(Y \mid do(X), W)$. □

Proof of Theorem 6.1. If $(Y; Z|X, W)_{H_{\overline{X}}}$, then by Corollary C.4.1, it holds that $(\boldsymbol{Y}; \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{W})_{G_{\overline{X}}}$. By the soundness of do-calculus for causal BNs, we get that $P(\boldsymbol{Y} \mid do(\boldsymbol{X}), \boldsymbol{Z}, \boldsymbol{W}) = P(\boldsymbol{Y} \mid do(\boldsymbol{X}), \boldsymbol{W})$. If $(Y; Z|X, W)_{H_{\overline{X}\underline{Z}}}$, then by Corollary C.4.1, it holds that $(\boldsymbol{Y}; \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{W})_{G_{\overline{X}\underline{Z}}}$. By the soundness of do-calculus for causal BNs, we get that $P(\boldsymbol{Y} \mid do(\boldsymbol{X}), do(\boldsymbol{Z}), w) = P(\boldsymbol{Y} \mid do(\boldsymbol{X}), \boldsymbol{Z}, \boldsymbol{W})$. Finally, if $(Y; Z|X, W)_{H_{\overline{XZ(W)}}}$, then by Corollary C.4.1, it holds that $(\boldsymbol{Y}; \boldsymbol{Z}|\boldsymbol{X}, \boldsymbol{W})_{G_{\overline{XZ(W)}}}$. By the soundness of do-calculus for causal BNs, we get that $P(\boldsymbol{Y} \mid do(\boldsymbol{X}), do(\boldsymbol{Z}), \boldsymbol{W}) = P(\boldsymbol{Y} \mid do(\boldsymbol{X}), \boldsymbol{W})$. □

Proof of Theorem 6.2. Consider G_H, the grounded-DAG of (H, f), that is, by definition, compatible with H. If Y is d-connected to Z in $H_{\overline{X}}$ with respect to $X \cup W$, then by Definition 5, it holds that y is d-connected to z in $G_{H_{\overline{f(X)}}}$ with respect to $f(X \cup W)$, for every $y \in f(Y)$ and $z \in f(Z)$. Therefore, $f(Y)$ is d-connected to $f(Z)$ in $G_{H_{\overline{f(X)}}}$ with respect to $f(X \cup W)$. □

D HANDLING MIXED GRAPHS

While one dominant form of graph input for causal inference is a causal DAG, other graph representations are also used when a full causal DAG is not retrievable, say, by a causal discovery algorithm (e.g., [64]). Many of these graph representations are referred to as *mixed graphs* due to their inclusion of undirected, bidirected, and other types of edges [15, 65].

One commonly used of mix graph is an acyclic-directed mixed graph (ADMG), which consists of a DAG with bidirected edges. As mentioned in the introduction, all of our results apply to scenarios where the input graph is an ADMG. Subsequently, we present an extension to the CaGreS algorithm to accommodate an ADMG.

In this scenario, we modify the cost function (Algorithm 2) as follows: When we remove a bidirected edge between nodes U and V (i.e., $U \leftrightarrow V$) by merging U and V into a single node, the cost incurred is doubled compared to removing a "standard" directed edge. For instance, in line 4 of Algorithm 2, if $\mathbf{U}$ and $\mathbf{V}$ were linked by a bidirected edge, the line would be updated to:

$$cost \leftarrow cost + 2 \cdot size(\mathbf{U}) \cdot size(\mathbf{V})$$

This adjustment is necessary because losing a bidirected edge should carry a higher cost than losing a regular directed edge, given that more information is lost.