CausaLens: A System for Summarizing Causal DAGs

Noam Chen
noamchen@campus.technion.ac.il
Technion

Anna Zeng
annazeng@mit.edu
CSAIL, MIT

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

Batya Kenig
batyak@technion.ac.il
Technion

Markos Markakis
markakis@mit.edu
CSAIL, MIT

Oren Mishali
omishali@cs.technion.ac.il
Technion

Brit Youngmann
brity@technion.ac.il
Technion

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

ABSTRACT

Causal inference aids researchers in discovering causal relationships, leading to scientific insights. Pearl's causal model uses causal DAGs to estimate causal effects, so DAG correctness is essential for reliable causal conclusions. However, for high dimensional data, the causal DAGs are often complex beyond human verifiability. Graph summarization is a logical next step, but current methods for general-purpose graph summarization are inadequate for causal DAG summarization, as they are not designed to preserve causal information. In this demonstration, we present a system called CausaLens that summarizes a given causal DAG, and balances graph simplification for better understanding and retention of essential causal information for reliable inference directly on the summary DAG. We illustrate that causal inference on the summary DAG is more robust to misspecification in the initial causal DAG compared to performing inference directly on the initial causal DAG, thereby enhancing the robustness of causal inference. We will demonstrate the utility of CausaLens for generating useful summary causal DAGs by interacting with the SIGMOD'25 participants, who will act as data analysts aiming to perform causal analysis on high dimensional datasets. A companion video for this submission is available at [1].

1 INTRODUCTION

Causal inference is central to informed decision-making in economics, sociology, and medicine. It has become increasingly relevant in machine learning, where it supports multiple tasks, including algorithmic fairness [15] and explainable AI [5]. Causal inference has also become a major theme in data management research, enhancing tasks like query results explanation [14, 21], data discovery [4], and data cleaning [15].

The core idea behind causal inference is that computing the effect of a variable T (the *treatment* variable) on another variable O (the *outcome* variable) should be aware of the causal structure of the system of which they are a part. This structure is often represented as a Causal Directed Acyclic Graph (DAG) [12], where each directed edge represents possible causal influence. Graphical criteria, such as the backdoor criterion [12], can then be applied to a causal DAG to find a sufficient set of *confounding variables* to adjust for. This ensures sound causal inference, even if the underlying data is purely *observational* (i.e., passively collected).

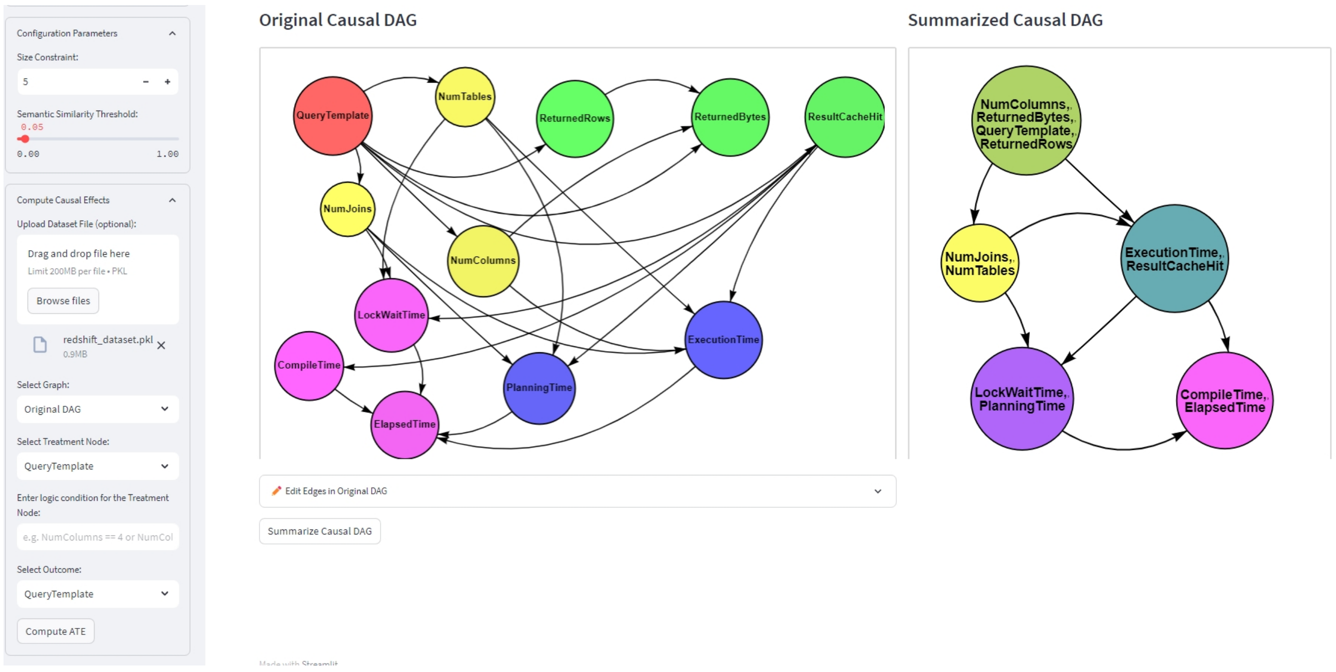
The reliability of causal inference thus depends on high-quality causal DAGs, which are often unavailable. Such DAGs are often constructed using domain knowledge, a process that is error-prone and time-consuming. Alternatively, multiple causal discovery methods have been proposed [7], which attempt to recover a causal DAG from available data alone. While useful, often they do not perform well on real-world data and require significant human effort for verification [18]. This challenge is exacerbated in high-dimensional data, underscoring the need for efficient methods to simplify and validate causal DAGs. We illustrate this need via an example:

Example 1. *Consider a dataset about query performance on a cloud-based data warehouse service*[1] *that includes performance metrics and query features, such as the number of tables and columns referenced. This dataset can help answer key causal questions for performance optimization, such as understanding how caching impacts latency, or analyzing the effect of join complexity on query planner performance. Fig. 1 (left) shows a small possible causal DAG for this task, constructed manually. For example, the edge from* `NumColumns}` *to* `ExecTime` *suggests that the number of columns referenced in a query may affect its execution time. In this example DAG,* `QueryTemplate` *is a confounder that must be adjusted for in order to answer the causal queries above in an unbiased way: it affects both performance metrics (e.g.,* `ElapsedTime`, `PlanTime`*) and the mechanisms being analyzed (e.g.,* `ResultCacheHit`, `NumJoins`*). If the DAG was mis-specified and some of the relevant edges from* `QueryTemplate` *were missing, we would reach inaccurate conclusions. This underscores the importance of domain expert verification for each edge, a task that can be overwhelming even in this causal DAG with just 12 nodes, as it requires examining 66 potential edges.*

Causal DAGs might be generated by an automated method or by a different team. In either case, the DAG here is very complicated, so graph summarization is a logical next step, as it reduces the number of nodes and edges, making it easier for users to verify. Graph summarization has been extensively studied in the data management community [8, 9, 17, 20], with techniques such as node grouping based on similarity [19], bit reduction [13], and removing unimportant nodes [9]. While these general-purpose methods excel at handling large graphs, they are unsuitable for summarizing causal DAGs, as they focus on objectives like minimizing reconstruction errors or enhancing visualizations rather than preserving causal information crucial for reliable inference.

In this work, we demonstrate CausaLens, a system that simplifies an input causal DAG into a *summary causal DAG*, making the verification process easier. In our full paper, recently accepted for publication[2] [1], we show that the generated summary DAG

[1]Amazon Redshift Serverless: https://aws.amazon.com/redshift/redshift-serverless/
[2]Since not published yet, we include the peer-reviewed paper in the git repository



Configuration Parameters

Size Constraint: 5

Semantic Similarity Threshold: 0.05

Compute Causal Effects

Upload Dataset File (optional):

redshift_dataset.pkl 0.9MB

Select Graph: Original DAG

Select Treatment Node: QueryTemplate

Enter logic condition for the Treatment Node: e.g. NumColumns == 4 or NumCol

Select Outcome: QueryTemplate

Figure 1: CausaLens UI showing an example causal DAGs for analyzing the Redshift dataset on query performance. Left: Manual causal DAG with color-coded semantic relationships. Right: Generated summary causal DAG based on user-defined size and similarity constraints.

can be used for sound causal inference and is more resilient to errors compared to the original DAG. CausaLens thereby enhances *interpretability*, *verifiability*, and *robustness* in causal inference. In CausaLens, the user provides an input causal DAG or generates one using a causal discovery algorithm via the UI. They can then set a bound on the number of nodes in the summary DAG. To ensure semantic coherence, CausaLens estimates the semantic similarity between variables (visualized through color codes in Fig. 1), allowing the user to set a threshold for the maximum semantic distance between nodes within the same cluster in the summary DAG. In the demonstration, we will show how the summary DAG can be used for reliable causal inference and how it is more robust to errors in the input DAG. This demo complements our full paper by showcasing the usability of CausaLens, demonstrating its end-to-end implementation, and illustrating practical scenarios for using it.

2 TECHNICAL BACKGROUND

We provide an overview of our theoretical foundations of the development of CausaLens. Full details can be found in [1].

2.1 Background on Causal Inference

We consider a single-relation database over a schema $\mathbb{A}$. We use upper case letters for a variable from $\mathbb{A}$ and bold symbols for variable sets. The broad goal of causal inference is to estimate the effect of a *treatment variable* $T \in \mathbb{A}$ (`NumJoins`) on an *outcome variable* $O \in \mathbb{A}$ (`ElapsedTime`). The gold standard approach is performing a *randomized controlled experiment*, where the population is randomly divided into a *treated* group (denoted $\mathsf{do}(T = 1)$ for a binary treatment) and a *control* group ($\mathsf{do}(T = 0)$). Randomization mitigates the effect of *confounding factors*, i.e., variables that would have affected both T and O, were the treatments not explicitly assigned.

One popular measure of a causal estimate is the *Average Treatment Effect* (ATE). In a randomized experiment, the ATE is the difference in the average outcomes of the treated and control groups [12]:

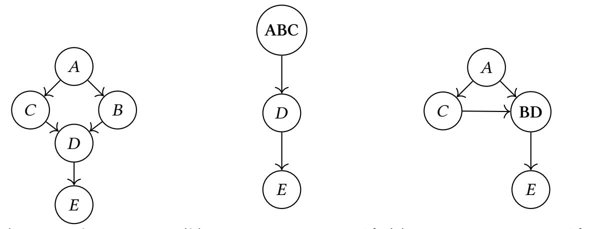
$$ATE(T, O) = \mathbb{E}[O \mid \mathsf{do}(T = 1)] - \mathbb{E}[O \mid \mathsf{do}(T = 0)] \qquad (1)$$

However, experiments where treatments are assigned at random cannot be done in many practical scenarios due to, e.g., economic or ethical issues. In such cases, we can still account for the confounding attributes to get an unbiased causal estimate *from observational data*, by using Pearl's model [12]. To achieve this, we must replace the *do*-operator in Equation 1 with appropriate conditional probabilities.

Finding the right conditional probabilities to substitute can be achieved using the rules of the *do*-calculus framework [12], which requires access to a causal DAG. A causal DAG is a specific type of a Bayesian network, where nodes represent random variables and edges signify *potential* causal influence. Starting from such a graph, Pearl developed the *backdoor criterion* [12], which allows us to graphically identify a sufficient set of confounding variables.

2.1.1 Causal Information. A causal DAGs encodes causal information as Conditional Independence (CI) relationships among the variables. These relationships can be read off the DAG using a graphical criterion called *d-separation* [12], which determines whether two sets of nodes $\mathbf{X}$ and $\mathbf{Y}$ are conditionally independent, given a third set $\mathbf{Z}$. The *Recursive Basis* (RB) [6] for a causal DAG consists of up to n CIs, where n is the number of nodes, indicating each node's independence from its non-descendant nodes, given its parents. This set of CIs holds significance, as it can be used to derive all other CIs encoded in the DAG using the semi-graphoid axioms.

Example 2. *Consider the causal DAG shown in Fig. 2(a). An example CI it encodes is $B \perp\!\!\!\perp_d C \mid A$. Its RB is given in Table 1.*



(a) Causal DAG $\mathcal{G}$ (b) Summary DAG $\mathcal{H}_1$ (c) Summary DAG $\mathcal{H}_2$

Figure 2: A causal DAG, and two summary DAGs.

Table 1: The recursive bases of the DAGs in Figure 2

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

2.1.2 CIs & Missing Edges. In causal DAGs, the information encoded by *missing* edges implies the set of CIs the DAG represents. Namely, removing edges can undermine the causal model as it implies additional CIs that do not necessarily hold. On the other hand, existing edges only indicate *potential* causal influence. This implies that adding edges to a causal DAG, provided acyclicity is maintained, does not necessarily compromise validity [12].

2.2 The Causal DAG Summarization Problem

CausaLens takes an input causal DAG and simplifies it for better interpretability while preserving its usefulness for causal inference. We begin by defining summary causal DAGs.

2.2.1 Summary Causal DAG. A *summary causal DAG* is created through *node contraction*, where only acyclicity-preserving contractions are allowed. Contracting nodes $U, V \in \mathsf{V}(\mathcal{G})$ produces a graph $\mathcal{H}$, replacing U and V with a single node $\mathsf{C} = \{U, V\}$. C retains the neighbors of U and V with preserved edge directionality, while any edge between U and V is removed.

Example 3. *Consider the example causal DAG $\mathcal{G}$ shown in Figure 2(a). After contracting the nodes* B *and* D*, the resulting summary DAG $\mathcal{H}_2$ is shown in Figure 2(c).*

A causal DAG $\mathcal{G}$ is said to be *compatible* with a summary DAG $\mathcal{H}$, if, there exists a partition of the nodes $\mathsf{V}(\mathcal{G})$ among the nodes $\mathsf{V}(\mathcal{H})$, such that: If $(U, V) \in \mathsf{E}(\mathcal{G})$, then either U and V are in the same cluster node in $\mathsf{V}(\mathcal{H})$ or there is an edge from the cluster node of U to the cluster node of V in $\mathsf{E}(\mathcal{H})$.

The RB of a summary causal DAG is defined in a similar manner. The only difference is that in a summary causal DAG, a node may represent a subset of nodes of the original DAG. Table 1 presents the recursive bases of the example causal DAG and two possible summary DAGs for it, as illustrated in Figure 2.

2.2.2 Problem Formulation. Our objectives are (1) to maintain causal information to the greatest extent possible, and (2) to simplify the input causal DAG.

<u>Causal Information Preservation</u>: Given two summary DAGs derived from the same DAG $\mathcal{G}$, we prefer the one that preserves a larger number of the CIs encoded in $\mathcal{G}$. We compare summary DAGs based on their RBs. Given 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 $\mathcal{H}_1$.

Example 4. *Consider again the causal DAG $\mathcal{G}$ shown in Figure 2, together with two possible summary DAGs $\mathcal{H}_1$ and $\mathcal{H}_2$. Despite $\mathcal{H}_1$ having only three nodes, it surpasses $\mathcal{H}_2$. From $\mathcal{H}_1$, for example, we can infer (using the semi-graphoid axioms) that $(E \perp\!\!\!\perp_d A \mid D)$, while this CI cannot be deduced from $\mathcal{H}_2$.*

Additionally, a summary DAG should not introduce any spurious CIs that the original causal DAG does not imply. We refer to this property as an I-Map. Our goal is to find a summary DAG that is an I-Map for the input causal DAG and its RB is maximal.

<u>Size Constraint</u> The summary DAG should be concise to reduce cognitive load for analysts [3]. We therefore allow users to impose a size constraint on the number of nodes in the summary DAG, enabling them to inspect it more easily.

Given a causal DAG $\mathcal{G}$ and a bound on the number of nodes $k>1$, CausaLens finds a summary causal DAG $\mathcal{H}$ s.t. (i) the number of nodes in $\mathcal{H}$ is $\leq k$; (ii) $\mathcal{G}$ is compatible with $\mathcal{H}$, is an I-Map for $\mathcal{G}$ and its RB is maximal.

In [1], 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. In a nutshell, we show that contracting nodes is akin to adding edges to the input causal DAG. Based on this connection, we can think of node contraction as the operation of adding edges to the input causal DAG. As mentioned, adding edges to a causal DAG does not undermine the causal model, but it reduces the amount of CIs it encodes. Our goal is then to find the summary DAG that results in the minimal number of added edges.

2.3 Causal DAG Summarization Algorithm

In our full paper, we show that the causal DAG summarization problem is NP-hard. We, therefore, employ a heuristic, scalable algorithm that meets the size constraints. CausaLens's underlying algorithm follows a previous line of work (e.g., [19]), where a bottom-up greedy approach is used to identify promising node pairs for contraction. Its main contribution lies in *how* it estimates merge costs, to preserve the causal interpretation of the graph: It counts the number of edges to be added to the original causal DAG for each node pair (a proxy for the effect of summarization on the RB, per Section 2.2.2). In each iteration, the algorithm contracts the node pair resulting in the minimal number of additional edges. We also introduce low-cost merges as a pre-processing step, and caching mechanisms to store invalid node-pairs and previously calculated cost estimations.

Semantic Constraint. The user can ensure that only semantically related variables (i.e., nodes) are merged, thereby maintaining semantic coherence in the summary DAG. To achieve this, CausaLens evaluates pairwise semantic similarity scores between all node pairs using the GPT-4 model [11]. The scores are displayed to the user by color-coding the nodes in the input causal DAG. For instance, in Fig. 1, we observe that the pairwise semantic similarity scores of `NumTable`, `NumJoins`, and `NumColumns` are relatively high, indicated by their similar yellow shades. Users can specify a semantic similarity threshold. A pair of nodes is eligible for contraction only if its semantic similarity score exceeds the specified threshold. The color coding of the cluster nodes in the summary causal DAG is consistent with the colors used in the input DAG. For example, in Fig. 1, if two nodes with a yellow shade are merged into a single cluster node, the cluster node will also be colored yellow.

3 SYSTEM & DEMONSTRATION

We implemented CausaLens in Python. To generate semantic similarity scores, we used GPT-4 [11]. We used the DoWhy [16] package to compute causal effects and Streamlit[3] for the UI.

System overview: The UI of CausaLens is shown in Fig. 1. The analyst begins by providing a dataset with a corresponding causal DAG. If a causal DAG is not provided, CausaLens can leverage one of the causal discovery algorithms in DoWhy [16] to infer one. Next, the analyst specifies a size constraint and optionally a semantic similarity threshold (see the left-hand side of Figure 1). The system then constructs a summary causal DAG that satisfies the size and semantic constraints. If no summary DAG can be found (e.g., when k is low, and the semantic threshold is high), CausaLens notifies the user, allowing them to adjust the parameters. The analysts can then use this summary causal DAG for causal analysis (in CausaLens it can be directly used for ATE computations).

In this demonstration, we will show that the obtained summary causal DAGs are valuable for reliable causal inference. We will also highlight how they effectively help users identify and address errors in their input causal DAGs, as error detection is significantly easier in summary DAGs compared to the original causal DAGs.

Demo scenarios: We demonstrate the operation of CausaLens over six datasets, including the Redshift dataset (12 nodes and 23 edges in the causal DAG), the German Credit dataset [2] (21 nodes, 43 edges), and a Car Accidents Data [10] (41 nodes, 368 edges). More dataset information is available in our main paper [1]. The participants will act as data analysts, aiming to explore and summarize their input causal DAG, before using it for causal analysis.

Introducing CausaLens: In this part of the demonstration, attendees will explore pre-loaded causal DAGs, such as the one in Figure 1 for Redshift. We will invite the attendees to select a dataset and load its corresponding causal DAG. We will then explain the available configuration knobs in CausaLens, setting them to initial values and obtaining a summary causal DAG. Attendees can then examine how the configuration knobs affect the summary DAG, as well as estimate causal effects over the summary and original DAGs to demonstrate the summary DAG's utility for causal inference. Specifically, we will compare ATEs estimated from the original DAG and the summary DAGs, each with a 95% confidence interval. While confounding variable sets may differ between the summary and original DAGs, resulting in some variation, we will show significant overlap in the intervals.

Demonstrating Robustness: In this part, the participants will be invited to change the input causal DAGs using the system UI. They can either modify the pre-loaded input DAG by adding or removing edges, or use a causal discovery algorithm to derive a new input DAG. This scenario simulates a real-life task where an analyst tries to verify the correctness of a causal DAG. Participants will then inspect the generated summary DAG to see the robustness of the summary DAG against errors in the input causal DAG. In particular, we will demonstrate that the ATEs calculated from summary causal DAGs are robust to errors in the input DAGs. This highlights their practical importance in real-world scenarios where a perfect input causal DAG may not be available.

Additionally, we will present some flawed input causal DAGs and ask participants to identify the errors (e.g., missing or redundant edges). Participants will be invited to inspect the input causal DAG directly, or examine the corresponding summary DAG. They will quickly realize that reviewing a summary DAG is a simpler task, as it requires inspecting fewer edges and nodes, thereby facilitating the verification process of the input causal DAG.

Looking under the hood: Interested participants will be invited to examine how various parameters affect quality and performance. For example, we will run an exhaustive approach that considers all summary causal DAGs, and show that while the summary DAG generated by CausaLens is similar to that of the exhaustive approach, CausaLens is at least one order of magnitude faster than this baseline, and the difference in runtime between the algorithms increases as the size of the causal DAG and the parameter k increase. Further, we will show users the GPT prompt used to generate the semantic similarity scores and allow them to select from several other prompts to get different semantic similarity values.

[3]https://streamlit.io/

REFERENCES

[1] Code repository and technical report. https://github.com/TechnionTDK/causalens, 2024. Accessed: 2025-01-30.
[2] A. Asuncion and D. Newman. Uci machine learning repository, 2007.
[3] S. S. Bhowmick and B. Choi. Data-driven visual query interfaces for graphs: Past, present, and (near) future. In *SIGMOD*, pages 2441–2447, 2022.
[4] S. Galhotra, Y. Gong, and R. C. Fernandez. Metam: Goal-oriented data discovery. In *ICDE*, pages 2780–2793. IEEE, 2023.
[5] S. Galhotra, R. Pradhan, and B. Salimi. Explaining black-box algorithms using probabilistic contrastive counterfactuals. In *SIGMOD*, pages 577–590, 2021.
[6] D. Geiger, T. Verma, and J. Pearl. Identifying independence in bayesian networks. *Networks*, 20(5):507–534, 1990.
[7] C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.
[8] K. Lee, H. Jo, J. Ko, S. Lim, and K. Shin. Ssumm: Sparse summarization of massive graphs. In *SIGKDD*, pages 144–154, 2020.
[9] A. Maccioni and D. J. Abadi. Scalable pattern matching over compressed graphs via dedensification. In *SIGKDD*, pages 1755–1764, 2016.
[10] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath. Accident risk prediction based on heterogeneous sparse data: New dataset and insights. In *SIGSPATIAL*, pages 33–42, 2019.
[11] OpenAI. Gpt-4 technical report, 2023.
[12] J. Pearl. *Causality : models, reasoning, and inference.* Cambridge University Press, 2000.
[13] R. A. Rossi and R. Zhou. Graphzip: a clique-based sparse graph compression method. *Journal of Big Data*, 5(1):1–14, 2018.
[14] B. Salimi, J. Gehrke, and D. Suciu. Bias in olap queries: Detection, explanation, and removal. In *SIGMOD*, pages 1021–1035, 2018.
[15] B. Salimi, L. Rodriguez, B. Howe, and D. Suciu. Interventional fairness: Causal database repair for algorithmic fairness. In *SIGMOD*, pages 793–810, 2019.
[16] A. Sharma and E. Kiciman. Dowhy: An end-to-end library for causal inference. *arXiv preprint arXiv:2011.04216*, 2020.
[17] K. Shin, A. Ghoting, M. Kim, and H. Raghavan. Sweg: Lossless and lossy summarization of web-scale graphs. In *WWW*, pages 1679–1690, 2019.
[18] K. Singh, G. Gupta, V. Tewari, and G. Shroff. Comparative benchmarking of causal discovery algorithms. In *CODS-COMAD*, pages 46–56. Association for Computing Machinery, 2018.
[19] Y. Tian, R. A. Hankins, and J. M. Patel. Efficient aggregation for graph summarization. In *SIGMOD*, pages 567–580, 2008.
[20] Q. Yong, M. Hajiabadi, V. Srinivasan, and A. Thomo. Efficient graph summarization using weighted lsh at billion-scale. In *SIGMOD*, pages 2357–2365, 2021.
[21] B. Youngmann, M. Cafarella, A. Gilad, and S. Roy. Summarized causal explanations for aggregate views. *PACMMOD*, 2(1):1–27, 2024.