ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240863

Quantifying a Causal Effect from a CPDAG with Targeted Exogenous Causal Knowledge

Mahdi HADJ ALI[a,b], **Yann LE BIANNIC**[b] **and Pierre-Henri WUILLEMIN**[a,*]

[a]LIP6 (UMR 7606 Sorbonne Université – CNRS), 4 pl. Jussieu 75005 Paris, France.
[b]SAP France, 35 rue d'Alsace, 92300 Levallois-Perret, France.
ORCID (Mahdi HADJ ALI): https://orcid.org/0009-0000-3674-1853, ORCID (Pierre-Henri WUILLEMIN): https://orcid.org/0000-0003-3691-4886

Abstract. Machine Learning models are getting more accurate, yet their complexity and opacity are also increasing. Explainable AI improved their general interpretability by quantifying the contributions of input features to predictions. Despite these advancements, practitioners still seek to gain causal insights into the underlying data-generating mechanisms. To this end, a possible solution is to rely on classical probabilistic causal analysis, which offers tools to quantify causal effects. However, causal analysis assumes a sufficient knowledge of causal structure, which is often unreachable from data alone. Indeed, causal discovery algorithms produce, at most, partial causal structures, namely Completed Partially Directed Acyclic Graphs (CPDAG). The conventional approach involves fully orienting the structure with exogenous causal knowledge through expert interaction or real-world experiments. In this paper, we focus on quantifying a specific total causal effect. Within this context, we emphasize that a partial structure can be sufficient to answer the query, and can be reached by different sequences of additional causal knowledge. Whether coming from an expert or an experiment, each addition has a cost difficult to assess a priori. The contribution of this paper is twofold: given a CPDAG and a specific query, we identify a set of irrelevant edges, and we introduce an algorithm for ranking the remaining informative edges, providing a guide to iteratively obtain a partial structure sufficient for resolving the query. Simulations show that these two contributions significantly reduce the number of requests for exogenous causal information, corroborating the feasibility of a causal impact quantification with very limited exogenous information.

Introduction

In recent years, the multiplication of sophisticated machine learning (ML) models has driven remarkable advancements in artificial intelligence (AI) across diverse domains. Despite their notable success in achieving state-of-the-art performance across numerous tasks, these models often function as opaque "black boxes", leaving users in the dark regarding the complexity of their process. The lack of interpretability and transparency impedes their adoption in critical applications and raises ethical and fairness concerns [2, 5].

Explainable Artificial Intelligence (XAI) has emerged as a key field addressing these challenges by developing methods to clarify the decision-making mechanisms of complex AI systems. Despite significant progress made by current XAI techniques in quantifying how observed feature values influence predictions [22], estimating the actual impact of an intervention on the underlying data-generating mechanism remains a challenge.

When the underlying causal model is fully known, employing the causal framework and specialized tools like do-calculus [15] provides a solution to common causal questions, such as identifying causes and effects, predicting intervention effects, and addressing counterfactual questions. When the complete causal structure is known but not its parameters, various works such as causal Shapley values or Shapley Flow [11, 23] focus on quantifying causal effects (direct and/or indirect) from a given predictive model. Especially, to quantify a specific causal effect [8] introduced a query-driven methodology involving the do-calculus and estimates from a tailored predictive model. A common assumption in those works is the knowledge of the complete causal structure.

Yet obtaining such a structure is challenging. Many causal discovery algorithms, such as PC, FCI, RFCI [3, 12, 20, 21] yield only a partially directed causal graph defined as a Completed Partially Directed Acyclic Graph (CPDAG).

Our contribution is twofold. Firstly, we identify, among the undirected edges of CPDAG, a subset that prevents the application of the Generalized Backdoor criterion [13]. Secondly, we propose an interactive algorithm that iteratively sorts pending undirected edges and propagates exogenous information.

More precisely, we propose extending the query-driven methodology presented in [8] to CPDAGs, leveraging previous results from (i) the Generalized Backdoor criterion [13], which proposes conditions for the extension of quantification of causal effects in such graphs, and (ii) [14], which defines the Meek rules that propagate orientations. Indeed, combining Meek rules with parsimonious enrichment of the learned causal structure (CPDAG) by the addition of exogenous causal information (the orientation of an edge) helps reach a CPDAG where [13] is applicable and ensures that the causal model remains consistent.

The first section presents a guiding thread example for our contributions. The second section offers essential background concepts. The third section gives some intuition about the possible issues that may arise and their solutions. The fourth section demonstrates how we can narrow the questions proposed to the expert. Finally, the last section shows experimental results that assess the relevance of our

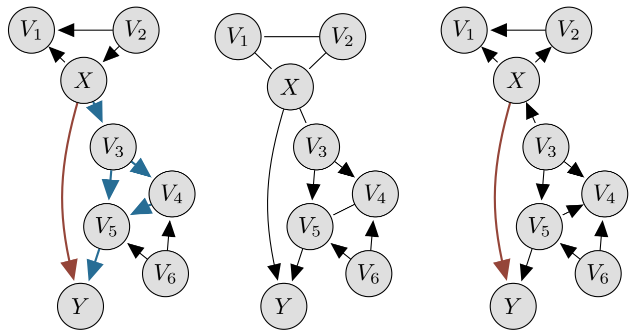
* Corresponding Author. Email: pierre-henri.wuillemin@lip6.fr

approach.

1 Motivating example

Suppose a complete causal model with structure as Fig.1a, we can then use this model to estimate the effect of X on Y thanks to do-calculus [15, 16, 17].

However, if we only have access to a database generated by this causal model, learning the complete structure is complex. Although several algorithms are available for causal structural learning [7], precisely determining the causal DAG proves inaccessible in numerous scenarios. Indeed, from data alone, some patterns can not be distinguished by the conditional independence tests used to recover the structure. Thus, these algorithms lead to a partially oriented graph, which is representative of the Markov Equivalence class [1, 14]. A Markov equivalence class contains all DAGs sharing identical conditional independences as the original causal DAG, and can be represented as a distinct graph structure known as an Essential Graph (EG) or a Completed Partially Directed Acyclic Graph (CPDAG). For instance, when attempting to learn the DAGs in Fig.1a and Fig.1c, we end up with the same CPDAG of Fig.1b.



(a) A DAG where X has a direct and indirect effect on Y.

(b) CPDAG $\mathcal{C}$.

(c) A DAG where X has only a direct effect on Y.

Figure 1: Two DAGs that result in the same CPDAG when learned. The direct effects are in red, and the indirect effects are in blue.

Even if the learning produces identical CPDAG, the effect we aim to calculate may be different. For instance, in Fig.1a, X influences Y both directly and indirectly through the red and blue paths. Whereas in Fig.1c, X has solely a direct effect on Y through the red path.

Our main objective is to answer the causal query, i.e., to quantify a causal effect, even in situations involving a partially oriented CPDAGs. To discern between these cases, additional information is required to adequately address the request. Further, we assume we can obtain exogenous causal knowledge such as expert knowledge to obtain missing information within the CPDAG.

In this paper, we aim to reduce the expert knowledge needed to answer our causal query. We assume fairly realistically that acquiring expert information is quite difficult. This paper then focuses on defining and estimating the relevance of each external causal information with respect to the causal query, providing a ranking that helps the expert to choose the next causal knowledge to assess.

2 Background and Notations

The notations and definitions are derived from [13].

Graphs A graph denoted by $\mathcal{G}$ is composed of a node set $\mathbf{V} = \{V_1, \ldots, V_n\}$. The nodes represent random variables, and the links indicate dependencies and causal relationships. Two nodes can be connected by either an arc, denoting a directed relationship, or an edge, representing an undirected relationship The relations are represented as " $\rightarrow$ " for arcs and " $-$ " for edges. We use the term link to refer to a relationship, regardless of whether it is directed or undirected. This discussion focuses solely on graphs without directed cycles, categorizing them into two types: directed acyclic graphs (DAGs), which only contain arcs, and partially directed acyclic graphs (PDAGs), which allow for both relation types.

A partially directed graph G is a chain graph if it contains no partially directed cycle. A graph G is chordal if every cycle of length $k \geq 4$ contains a chord, i.e., a link that is not part of the cycle but connects two nodes of the cycle.

Let G' be the undirected graph we get by removing all arcs from a chain graph G. The set of all nodes that are connected to X in G' is called the chain component of X: $\mathcal{T}_{G'}(X)$. The completed partially directed acyclic graphs (CPDAGs) are a subset of chordal chain graphs [1].

Paths Two nodes are adjacent if connected by a link. A path is a sequence of distinct, adjacent nodes, represented as $p = \langle V_i, V_{i+1}, \ldots, V_{i+l} \rangle$. The direction of the arc between V_i and V_{i+1} determines whether the path is considered to go "out of" V_i or "into" V_{i+1}. If, for every $k \in \{1, \ldots, l\}$, the arc $V_{i+k-1} \rightarrow V_{i+k}$ exists, the path is classified as a directed path from V_i to V_{i+l}. If, for each $k \in \{1, \ldots, l\}$, the link between V_{i+k-1} and V_{i+k} does not point into V_{i+k-1}, it is considered a partially directed path from V_i to V_{i+l}. If it contains only edges, it is classified as an undirected path.

Node relations If there is an arc $V_j \rightarrow V_i$, we define V_i as the child of V_j and V_j as a parent of V_i. The corresponding sets of parents and children are denoted as $\text{pa}(V_i, \mathcal{G})$ and $\text{ch}(V_i, \mathcal{G})$, respectively. If there is an edge $V_j - V_i$, we define V_i and V_j as neighbors. If a (partially) directed path exists from V_i to V_j, V_i is regarded as a (possible) ancestor of V_j, and conversely, V_j is a (possible) descendant of V_i.

The sets of neighbors, ancestors, descendants, possible ancestors, and possible descendants for a node V_i in $\mathcal{G}$ are represented as $\text{Nei}(V_i, \mathcal{G})$, $\text{An}(V_i, \mathcal{G})$, $\text{Desc}(V_i, \mathcal{G})$, $\text{possAn}(V_i, \mathcal{G})$, and $\text{possDesc}(V_i, \mathcal{G})$, respectively (we sometimes omit specifying the graph in these notations when the context is clear). If V_j is a possible descendant of V_i, the path from V_i to V_j is undirected or partially directed and is called a possible descendancy path.

3 Quantifying causal impact in a CPDAG

This section explores three key points. First, we discuss causal effects in entirely directed graphs and highlight how these computations intersect with XAI frameworks to provide interpretable insights into complex causal structures. Next, we present established methods for quantifying causal effects in partially directed graphs and show their continued relevance to XAI. Finally, we explore strategies for scenarios where these methods do not apply directly. Henceforth, we will consider a CPDAG $\mathcal{C}$, with a node set $\mathbf{V} = \{V_1, \ldots, X, \cdots V_n, Y\}$.

3.1 Interpretability with Complete Causal Knowledge

Starting with a DAG as a complete causal structure, causal effects can be identified and estimated from observational data using do-calculus tools. For instance, the Backdoor criterion [17] is a sufficient condition for effective adjustment, stating that a set of variables is suitable if it meets some graphical criteria within the DAG. If identifiable, with the classic Backdoor criterion, the causal effect of X on Y is given by the following adjustment:

$$P(Y|do(X=x)) = \sum_{z} P(Y|X=x, Z=z)P(Z=z) \quad (1)$$

With $Z \subset \mathbf{V}$, a set that satisfies the Backdoor criterion relative to (X, Y).

Renaming $P(Y|X=x, Z=z)$ as $f(x, Z)$, Equation 1 above can be seen as an expectation of f over Z.

$$P(Y|do(X=x)) = \mathbb{E}_Z[f(x, Z)] \quad (2)$$

Thus, to compute this quantity one can rely on a Monte-Carlo integration over the distribution of the set Z and get Equation 3:

$$P(Y|do(X = x)) \simeq \frac{1}{D} \sum_{\substack{i=1 \\ Z^i \sim Z}}^{D} f(X = x, Z^i) \quad (3)$$

In the context of ML, Zhao and Hastie established the analogy between the backdoor adjustment and the partial dependence plot (PDP) [24]. Indeed, one can estimate f by building a predictive model $\hat{f}$, and the data gives a collection of $Z^i \sim Z$. Thus we can approximate Equation 3 by:

$$P(Y \mid do(X = x)) \simeq \frac{1}{D} \sum_{i=1}^{D} \hat{f}(X = x, Z^i) \quad (4)$$

Given a predictive model $\hat{f}(\mathbf{X})$, a partial dependence plot (PDP) [6] grants visualization and analysis of the dependence of the predictions on an input feature of interest X (let $\bar{X}$ be its complement). The PDP can be computed as shown in Equation 5.

$$PDP_X(x) = E_{\bar{X}}[\hat{f}(x, \bar{X})] = \int \hat{f}(x, \bar{x}) d\bar{x} \quad (5)$$

$$\simeq \frac{1}{D} \sum_{i=1}^{D} \hat{f}(x, \bar{X}^i) \quad (6)$$

The Monte-Carlo integration in Equation 6 and Equation 4, are identical when the set Z is identified with $\bar{X}$.

This implies that a sufficient condition to obtain a causal interpretation from the PDP is to have trained the predictive model on a set of variables that satisfies the backdoor criterion.

In this regard, [8] proposes to use predictive models to quantify causal effects with XAI techniques.

One significant advantage of predictive models is that they are not limited by graphical model structure constraints. For instance, Bayesian networks sometimes limit the number of parents because the size of the conditional probability tables grows exponentially in the number of parent nodes. Some also only deal with discrete variables, thus necessitating a discretization of the data. In contrast, predictive models can handle numerous inputs with greater flexibility and manage both continuous and discrete variables.

3.2 Interpretability with Partial Causal Knowledge

The previous results assume perfect knowledge of the entire causal graph. However, as previously mentioned, the outcome of causal discovery methods does not always yield a fully directed graph.

From this point forward, we limit ourselves to the CPDAG. Note that this is a weaker assumption than knowing the complete structure but still quite strong, as it assumes the perfection of all independency tests employed to recover the structure from the data.

Maathuis and Colombo [13] and Perković [18, 19], expand the identification and estimation of causal effects in CPDAG.

Maathuis and Colombo extended the concept of backdoor adjustment to CPDAG, and other types of graphs. Given our sole focus on CPDAG, our analysis will exclusively consider results about this specific graph type, particularly the findings in Corollary 4.2 of [13] (referred here as Theorem 1).

Theorem 1 (Generalized backdoor criterion for CPDAG)**.** *Let X and Y be two distinct nodes in a CPDAG $\mathcal{C}$. We note $\mathcal{C}_{\underline{X}}$ be the graph obtained from $\mathcal{C}$ by removing all arcs out of X in $\mathcal{C}$. Then there exists a generalized back-door set relative to (X, Y) and $\mathcal{C}$ if and only if:*

1. *$Y \notin pa(X, \mathcal{C})$*
2. *and $Y \notin possDe(X, \mathcal{C}_{\underline{X}})$*

Moreover, if such a generalized back-door set exists, then $pa(X, \mathcal{C})$ is such a set.

If a set satisfies the Generalized Backdoor criterion, the effect is computed using the classic backdoor adjustment formula (Eq.1), therefore the parallel with Eq.4 still holds. Thus, we can compute the causal effect within a CPDAG by employing an adapted PDP.

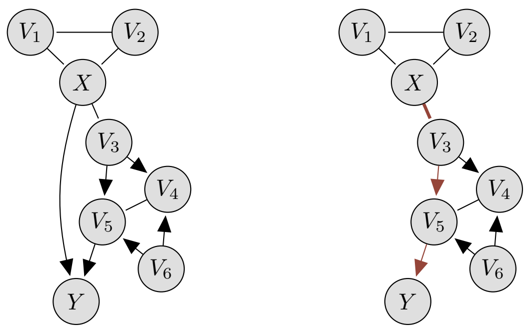
3.3 Analyzing conditions in Generalized Backdoor

The Generalized Backdoor criterion may not be satisfied for two potential reasons:

(i) if $Y \in pa(X, \mathcal{C})$, given that arrows represent the flow of causality, the causal effect of X can not go backward to Y; we can conclude that X has no effect on Y.

(ii) If Y is a possible descendant of X in the graph $\mathcal{C}_{\underline{X}}$, some additional causal knowledge (orientation of edges) is required to apply Theorem.1.

This subsection aims to provide insights into potential solutions for addressing the second item.



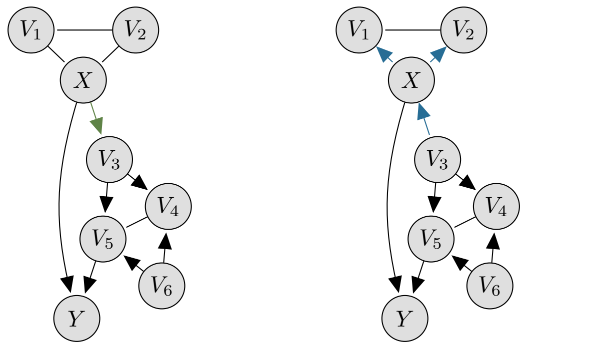
(a) CPDAG $\mathcal{C}$

(b) $\mathcal{C}_{\underline{X}}$ with a possible descendancy path in red.

Figure 2: A CPDAG $\mathcal{C}$ where condition (2) of Theorem 1 is not satisfied.

Consider the example in Fig.2, Y is a possible descendant of X through the path $\langle X, V_3, V_5, Y \rangle$ causing the Generalized Backdoor to be not applicable. However, this issue can be resolved by clarifying whether or not Y is a descendant of X in $\mathcal{C}_{\underline{X}}$.

For instance, if the edge $X - V_3$ is oriented as $X \to V_3$ (Fig.3a), the arc is no longer in $\mathcal{C}_{\underline{X}}$ and thus Y ceases to be a possible descendant in this graph. Therefore, Theorem 1 is applicable and results in $\{\emptyset\}$ as the adjustment set. Conversely, if orienting $V_3 \to X$ (Fig.3b), Y is not a possible descendent anymore, and thus, the Generalized Backdoor can be used and, for instance, results in $\{V_3\}$ as the adjustment set. Therefore to resolve the ambiguity regarding whether Y is a descendant of X in $\mathcal{C}_{\underline{X}}$, some edges have to be oriented.



(a) $X \to V_3$: Backdoor set $\{\emptyset\}$ (b) $X \leftarrow V_3$: Backdoor set $\{V_3\}$

Figure 3: Identifying the direction of causality between $V_3 - X$ enables the Generalized Backdoor.

Nevertheless, not all orientations carry the same level of information, for instance:

- Orienting $V_1 \to X$ forces the orientation $X \to V_3$ due to Meek rule 1 (Fig.4). Similarly, orienting $V_2 \to X$ forces $X \to V_3$.
- Conversely, orienting $X \to V_1$ (or $X \to V_2$) does not trigger any Meek rules. Therefore, the orientation for $V_3 - X$ remains unknown. Resolving this ambiguity requires supplementary information to determine the orientation.

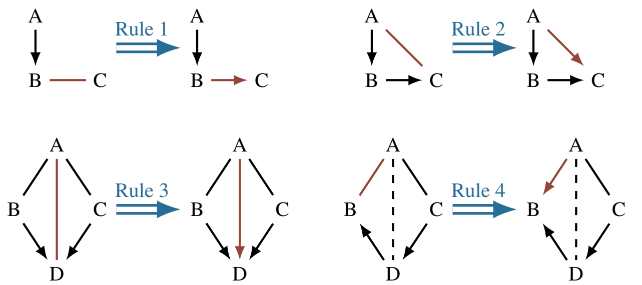


Figure 4: Meek rules for orientation in PDAGs [14]. A dashed line represents a link (it can be an arc or an edge).

Thus, if the Generalised Backdoor cannot be applied in an initial CPDAG, we can incrementally add orientations and then propagate them until Theorem 1 is applicable for our causal query. As a reminder, the initial CPDAG already contains the maximum amount of information that can be retrieved from data. Therefore, it is necessary to seek out information from other sources, such as expert knowledge or experiments. Previous studies suggest leveraging experimental interventional data from randomized controlled trials [4]. In many real-world applications, experimental interventions are time-consuming and/or expensive. In this paper, we propose substituting external experience with expert knowledge. Yet, the expert may not always know how to orient the required edges. Despite this, we can propose a set of edges that could lead to the necessary orientations for problem resolution. Doing so opens up multiple possibilities for resolving the issue.

4 Eliciting Sufficient Background Knowledge

This section provides the key contributions of the article. As a refresher, the challenge lies in determining the orientation of the edges that induce incompatibility with condition (2) of the Generalized Backdoor, i.e., the edges that provoke possible descendancy. Gaining knowledge about the orientation of these blocking edges would lead to a structure where Theorem 1 becomes applicable. To this end, we introduce several theoretical findings aimed at identifying and excluding edges that do not contribute to solving our problem. Subsequently, we propose a methodology for ranking edges to reduce the amount of additional exogenous causal knowledge, i.e., requested orientations from an expert.

Firstly, we report an established proposition that states that any CPDAG can be divided into components that can be oriented independently.

Proposition 1. *(From [10]) Every essential graph is a chain graph with chordal chain components. Moreover, orientations in one chain component do not affect orientations in other components.*

For instance, in the chain graph of Fig.5c, edges are either in the component $\{V_1, V_2, X, V_3\}$ or the component $\{V_4, V_5\}$. Orienting edges in one component does not affect the edges of the other.

4.1 Informative Edges Given a Specific Query

The subsection is dedicated to exploring the impact of X on Y in a CPDAG $\mathcal{C}$ such as $Y \in \text{possDesc}(X, \mathcal{C}_{\underline{X}})$ where $\mathcal{C}_{\underline{X}}$ is the graph obtained by removing all directed edges out of X in $\mathcal{C}$.

To assess the possible descendancy from X to Y, we need to orient some edges to transform all the possible directed paths from X to Y into:

- Either a path starting with an arrow out of (or into) X:
$$X \to \cdots Y \text{ or } X \leftarrow \cdots Y$$
- Or into a path in the form:
$$X - \cdots V_i \leftarrow V_{i+1} \cdots - Y$$

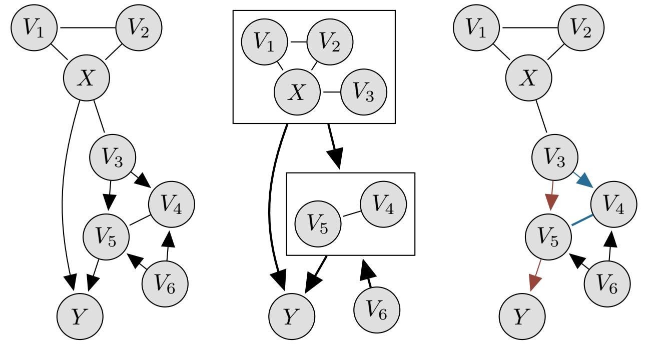
Thus, we are interested in the edges that, once oriented, can resolve the question of possible descendancy from X to Y, directly or by propagation. To this end, we will use two new concepts.

Definition 1. *(Informative edges) An edge is informative if its orientation can contribute to the resolution of possible descendancy from X to Y.*

Definition 2. *(Bypass path) Let $p = \langle U_1, \cdots, U_l \rangle$ be a partially directed path. If there exists a directed path p' from U_1 to U_l, then p' bypasses p (and p is bypassed by p').*

In Fig.5c, the red path $\langle V_3, V_5, Y \rangle$ bypasses the blue path $\langle V_3, V_4, V_5, Y \rangle$.

In the following development, we will demonstrate that informative edges are only in the component of X ($\mathcal{T}_{\mathcal{C}}(X)$). Initially, we prove that all edges within the intermediate components between X and Y are bypassed via an entirely directed path. Then, we show that every edge in a bypassed component is non-informative. Lastly, we conclude that the only component containing informative edges is $\mathcal{T}_{\mathcal{C}}(X)$.



(a) Initial CPDAG $\mathcal{C}$ (b) Components of $\mathcal{C}$ (c) Bypassed paths in $\mathcal{C}_{\underline{X}}$

Figure 5: Example of components and bypass paths

Lemma 2. *(Bypassed paths are irrelevant.)*

Let $Y \in possDesc(X, \mathcal{C})$, and p be a partially directed path on a possible descendancy path from X to Y.

If a directed path p' bypasses p, then orienting an edge in p can not resolve the question of possible descendancy from X to Y.

Proof. Consider paths p partially directed and p' totally directed, such that p and p' are on possible descendancy paths. Let U_1 (respectively, U_l) be the start (respectively, end) node of p.

Since p' bypasses p, a directed path exists from U_1 to U_l. Hence, any edges oriented on path p will not revoke the descendancy of U_l from U_1. Consequently, orienting any edges on p will not resolve the possible descendancy from X to X. □

In Fig.5c, the partially directed path $\langle V_3, V_4, V_5, Y\rangle$ is on a possible descendancy path from X to Y. The directed path $\langle V_3, V_5, Y\rangle$ bypasses the path $\langle V_3, V_4, V_5, Y\rangle$ so the edge $V_4 - V_5$ is irrelevant, i.e. it is not necessary to seek for the causality between these two nodes to assess the effect of X on Y.

Lemma 3. *(Parent of a chain component)*

Let $\mathcal{C}$ be a CPDAG with a node set $\mathbf{V} = \{V_1, \cdots, V_p\}$, and $\boldsymbol{\mathcal{T}}_\mathcal{C}$ be the set of components of the graph. For $V_i \in \mathbf{V}$, $\mathcal{T}_\mathcal{C}(V_i)$, is the component containing node V_i.

Let $V^{in} \in \mathbf{V}$ such that $V^{in} \notin \mathcal{T}_\mathcal{C}(V_i)$.

If an arc $V^{in} \to V_i$ exists, then for all $V \in \mathcal{T}_\mathcal{C}(V_i)$, an arc $V^{in} \to V$ exists.

We denote such a node V^{in} as a parent of the chain component $\mathcal{T}_\mathcal{C}(V_i)$.

Proof. Let $\mathcal{C}$ be a CPDAG. Thus, in $\mathcal{C}$, orientations have already been propagated using the Meek rules.

Let $V^{in}, V_i \in \mathbf{V}$ such that $V^{in} \notin \mathcal{T}_\mathcal{C}(V_i)$ and there exists a directed arc $V^{in} \to V_i$. Nei(V_i) is the set of nodes that are connected to V_i with an edge.

- We start by establishing that for all $N \in$ Nei(V_i), there exists a directed arc $V^{in} \to N$.
  - Suppose there is no link between V^{in} and N, then $V^{in} \to V_i - N$ would form an unshielded triplet, thus by Meek Rule 1 (Fig.4), we would have $V^{in} \to V_i \to N$, and hence N would not be in Nei(V_i).
  - Now suppose the arc $V^{in} - N$ is undirected, implying there exists an undirected path between V_i and V^{in}, hence $V^{in} \in \mathcal{T}_\mathcal{C}(V_i)$, contradicting the hypothesis of Lem.3.
  - Suppose now $V^{in} \leftarrow N$, by Meek Rule 2 (Fig.4), we get $V_i \to N$, and hence N would not be in Nei(V_i).

  Thus for all $N \in$ Nei(V_i), there exists an arc $V^{in} \to N$. We now know that for all neighbors N of V_i, a directed arc $V^{in} \to N$ exists.
- Now that we have an arc $V^{in} \to N$, N is in the same conditions as V_i, so we can apply the same reasoning to all neighbors of V_i and their neighbors, and so on, until we obtain an arc $V^{in} \to V$, $\forall V \in \mathcal{T}_\mathcal{C}(V_i)$. □

In Fig.5c, since there is an arc $V_3 \to V_4$, then there is an arc $V_3 \to V_5$ since V_4 and V_5 belong to the same chain component.

Lemma 4. *(Bypassed Chain Component)*

Let $\mathcal{T}_\mathcal{C}(V_i)$ be a chain component with a parent V^{in}, i.e. with an incoming arc $V^{in} \to V_i$, then any partially directed path from V^{in} to a node $V \in \mathcal{T}_\mathcal{C}(V_i)$ is bypassed by the directed path $V^{in} \to V$.

Proof. According to Lem.3, if an incoming arc enters component $\mathcal{T}_\mathcal{C}(V_i)$ from node V^{in}, this implies that source node V^{in} is connected to all nodes within the component, creating a bypass for each node in the component. □

In Fig.5a, $V_3 = V^{in}$ is a parent of the component$\{V_4, V_5\}$,the path $\langle V_3, V_5\rangle$ bypasses the path $\langle V_3, V_4, V_5\rangle$.

Theorem 5. *(Non-Informative Chain Component)*

Let $\mathcal{T}_\mathcal{C}(V_i)$ be a chain component with a parent V^{in}, and with the incoming arc $V^{in} \to V_i$ that belongs to a possible descendancy path p from X to Y, then any edge in $\mathcal{T}_\mathcal{C}(V_i)$ is non informative to assess the effect of X on Y.

Proof. Let $V^{in} \to V_i$ in a possible descendancy path p from X to Y.

If $Y \in \mathcal{T}_\mathcal{C}(V_i)$ then V_{in} is a parent of Y by Lem.3. If $Y \notin \mathcal{T}_\mathcal{C}(V_i)$, then the component has an outgoing arc in p: $V \to V^{out}$, and the directed path $V^{in} \to V \to V^{out}$ bypasses all other paths from V^{in} to V^{out} that go through the component.

In both cases, by Lem.2 and Lem.4, the edges in $\mathcal{T}_\mathcal{C}(V_i)$ are bypassed on p.

Furthermore, for all possible descendancy path p' that goes through this component, p' contains its own V'^{in} parent of the component (otherwise $\mathcal{T}_\mathcal{C}(V_i) = \mathcal{T}_\mathcal{C}(X)$ and then V^{in} could not exist). With the same argument as above, we see that the edges in $\mathcal{T}_\mathcal{C}(V_i)$ are bypassed on p'.

Finally, the edges of $\mathcal{T}_\mathcal{C}(V_i)$ are bypassed on every possible descendancy path that intersects the component. Then by Lem. 2 those edges are non informative to assess the effect of X on Y. □

For example, in Fig.5c, we consider the component $\mathcal{T}_\mathcal{C}(V_4) = \{V_4, V_5\}$ and the source node $V^{in} = V_3$. The edges within the component $\mathcal{T}_\mathcal{C}(V_4)$ are irrelevant regardless of their orientations because the path $\langle V^{in} = V_3, V_5, Y = V^{out}\rangle$ (red path in Fig.5c) will always create a descendancy path.

Theorem 6. *(Localization of informative edges)*

To assess the effect of X to Y, the informative edges are only in the chain component of X.

Proof. Let $\mathcal{C}$ be a CPDAG where $Y \in$ possDesc$(X, \mathcal{C}_{\underline{X}})$, and $\mathcal{T}$ be a chain component of $\mathcal{C}$.

- If $\mathcal{T}$ does not contain any node that belongs to a possible descendancy path from X to Y. Then, by Proposition 1, the orientations in this chain component do not affect the other components and so are non-informative.

- If $\mathcal{T}$ intersects a possible descendancy path p and does not contain X. Then there exists in p an incoming arc that enters $\mathcal{T}$ (otherwise $\mathcal{T} = \mathcal{T}_{\mathcal{C}}(X)$. Let us call this arc $V^{in} \rightarrow V_1$. $\mathcal{T} = \mathcal{T}_{\mathcal{C}}(V_1)$ and V^{in} is a parent of $\mathcal{T}$. By Th.5, the edges of this chain component are then non-informative.

Thus, the only informative edges belong to the chain component that contains X. □

In Fig.5a, we only have to consider edges in $\mathcal{T}_{\mathcal{C}}(X) = \{V_1, V_2, V_3, X\}$.

We have established that we can restrict exogenous information to edges belonging to $\mathcal{T}_{\mathcal{C}}(X)$. It is worth noting that some of these edges may become non-informative after adding exogenous information, as discussed in the next section.

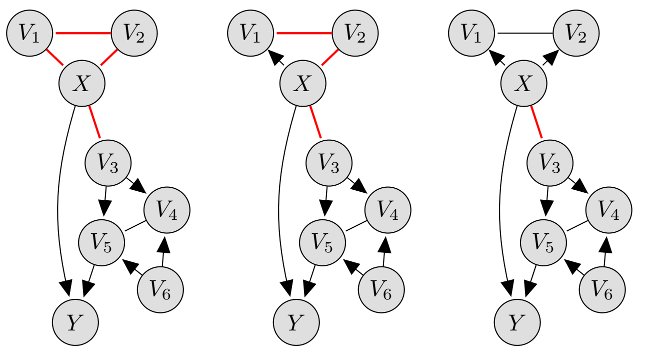
4.2 *Interactive Algorithm for Limiting Expert Input*

Considering a CPDAG $\mathcal{C}$ where $Y \in \mathrm{possDesc}(X, \mathcal{C}_{\underline{X}})$, we can not yet use the Generalized Backdoor in order to adjust for the impact of X on Y. To address this issue, we propose to seek knowledge from an expert. This search takes the form of questions regarding the orientation (the causality) of specific edges. Due to the difficulty of this task, it is necessary to reduce the number of questions asked. Theorem 6 allows us to only focus on orientations in the chain component of X.

Moreover, a causal orientation of an edge can propagate to multiple edges thanks to the Meek rules. We thus propose an iterative approach consisting of asking a causal question, propagating the answer to a new structure, and continuing until this structure is sufficient to assess the impact of X on Y.

Not all edges carry an equal amount of information and some can even become non-informative during this process.

For instance in Fig.6b, the edge $V_1 - V_2$ becomes irrelevant when it gets completely separated from X by new arcs.



(a) Original CPDAG (b) Adding $X \rightarrow V_1$ (c) Adding $X \rightarrow V_2$

Figure 6: A sequence of additional orientations. Edges in red are still relevant after each step.

Lemma 7. *(Separated edges) An edge that does not belong to a possible descendancy path from X to Y, and that is not connected to a possible descendancy path by an undirected path, is not informative.*

Proof. Meek rules propagate an orientation either to an adjacent edge, or through an intermediate edge. Thus, when an edge is not part of a possible descendancy path p, and has no undirected path to p, its orientation cannot propagate to p, and can only propagate to similarly separated edges. □

A classical approach to an interactive algorithm could be to propose a way to decide the best question to ask at each iteration. However, our aim is to leave some initiative to the expert while forewarning about the most complex scenarios. Therefore, we propose presenting the edges (the causal questions) in ascending order based on the maximum number of questions a scenario may induce.

This approach offers another advantage: if the expert encounters difficulties answering questions along the most direct route, she can switch to more accessible questions (i.e. questions with more causal sense from a human point of view) while staying on paths that require few questions.

To this end, the first step is to construct the tree of scenarios as graphs successively enriched by orientations provided by the expert. A detailed illustration of this construction is presented in Fig.7.

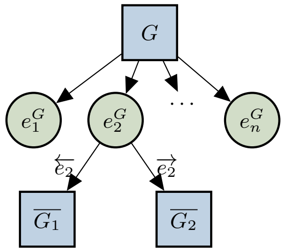


Figure 7: Construction of Trees. e_k^G represents an informative edge in G. $\overleftarrow{e_2}$ and $\overrightarrow{e_2}$ are the orientations used respectively in the left child of G: $\overline{G_1}$, and the right child of G: $\overline{G_2}$.

Starting with a CPDAG $\mathcal{C}$, we initially assess whether the conditions of the Generalized Backdoor are met. If it is the case, we halt the process. If it isn't, we select an undirected edge $a - b$ (e_k^G in Fig.7) from the set of candidate informative edges, defined as follows.

In a graph G obtained by adding orientations to $\mathcal{C}$, candidate informative edges are elements of $\mathcal{T}_{\mathcal{C}}(X)$ that have not yet been oriented and remain informative (Lem.7); the set of candidate edges is denoted as $\mathcal{K}_G(X, Y)$ (at the start of the algorithm $\mathcal{K}_G(X, Y) = \mathcal{T}_{\mathcal{C}}(X)$).

Then, we create two new graphs, incorporating the orientations $a \leftarrow b$ or $a \rightarrow b$ ($\overleftarrow{e_k}$ and $\overrightarrow{e_k}$ in Fig.7). Following this, we apply the Meek rules to both graphs, resulting in two newly partially oriented graphs $\overline{G_1}$ and $\overline{G_2}$. These graphs are designated as the "left child" and the "right child" of G.

Then, we recursively apply the process for the two new graphs, until we reach a graph that satisfies the Generalized Backdoor criterion.

We analyze the depth of the tree that led us to this point, i.e., the number of questions asked to the expert. For any starting graph, we can rank candidates edges based on the worst-case scenario that they can induce. The Algorithm 1 proposes an implementation of the construction of trees for such a process.

5 Experiments

To evaluate our approach, we propose a comparative analysis against a baseline where the expert is solicited to determine the remaining orientations of the CPDAG to apply the conventional causal calculation, which requires a DAG. The key objective is to quantify the reduction in the number of expert interactions achieved through our method.

The simulation methodology follows these steps: The parameters for graph generation range from 10 to 100 for the number of nodes

Algorithm 1 Recursive Tree Construction

function CONSTRUCTTREE(graph G, X, Y)
  if $Y \notin$ possDesc$(X, G_{\underline{X}})$ **then**
    return tree(G, None), 0
  else
    dictTrees $\leftarrow \{\}$
    for $a - b \in \mathcal{K}_G(X, Y)$ **do**
      $G1 \leftarrow$ ApplyMeekRules$(G + a \leftarrow b)$
      $t_1, h_1 \leftarrow$ CONSTRUCTTREE($G1$, X, Y)
      $G2 \leftarrow$ ApplyMeekRules$(G + a \rightarrow b)$
      $t_2, h_2 \leftarrow$ CONSTRUCTTREE($G2$, X, Y)
      tree $\leftarrow$ *new* tree(G, $(a - b)$, t_1, t_2)
      h $\leftarrow \max(h_1, h_2) + 1$
      dictTrees $\leftarrow$ (tree,h)
    end for
    $(t_{min}, h_{min}) \leftarrow \min_h\{(t, h) \in$ dictTrees$\}$
    return $(t_{min}, h_{t_{min}})$
  end if
end function

and from 1.1 to 1.8 for the arc ratio. For each combination of these parameters, we generate 50 DAGs and extract their CPDAGs (the code is available on GitHub[9]).

Within each graph, a variable of interest (X) and a target (Y) are randomly selected. The identification of the causal effect of X on Y with the Generalized Backdoor criterion is then assessed. If the Generalized Backdoor criterion holds true, the graph is ignored. If the Generalized Backdoor criterion is not satisfied, the total number of edges in the CPDAG and the number of edges in the component are counted; we then apply our algorithm to the edges in the component.

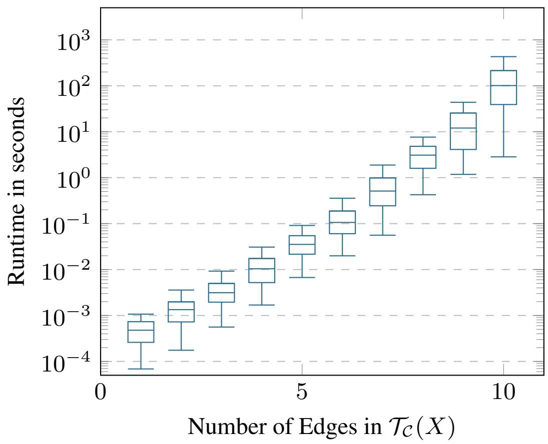


Figure 8: Algorithm Runtime in seconds versus the number of edges in the component of interest. (Note the log scale for the y-axis)

The algorithm 1 is clearly exponential since it involves the exhaustive construction of scenario trees, as shown in Fig.8.

In Fig.9, the blue curve represents the average total number of edges versus the size of the CPDAG, while the red curve illustrates the average number of edges within $\mathcal{T}_{\mathcal{C}}(X)$. This red plot can be seen as the maximum number of questions to be posed in the worst-case scenario, assuming each orientation is independent. The plot in green is the average number of questions asked to guide the expert in assessing the effect of X on Y (i.e., the average depth of the scenario tree).

By application of Theorem 6, orienting the entire component of X, $\mathcal{T}_{\mathcal{C}}(X)$, is sufficient to resolve possible descendancies and apply the

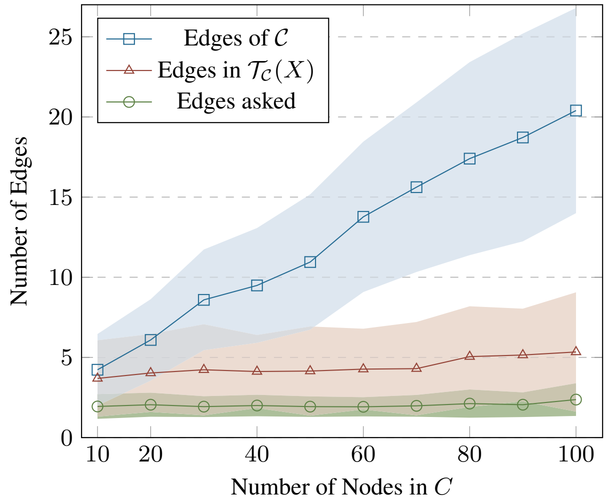


Figure 9: Maximum edges to orient to assess a causal effect. Complete orientation of the CPDAG (blue) and focus on $\mathcal{T}(X)$ (red).

Generalized Backdoor criterion. Thus, the maximum number of interactions with experts is bounded by the number of edges in $\mathcal{T}_{\mathcal{C}}(X)$. Experimentally, we observe that this number slowly increases with the total size of the graph, up to 5 informative edges on average for graphs containing 100 nodes.

In addition to this initial enhancement, we can further minimize the number of questions by prioritizing those directly addressing the issue. As the green graph suggests, with a suitable sequence of orientations, it is possible to answer the query with 2 or 3 inputs from the expert.

Finally, despite having an exponential algorithm, we never apply it to scenarios larger than 5 or 6 possible questions, even for graphs up to 100 nodes, with an acceptable runtime.

Conclusion

To extract causal insights from observational data, it is crucial to consider the causal relationships between variables. However, acquiring causal knowledge presents significant challenges. This paper focuses on leveraging automatically learnable aspects of the causal structure, particularly the CPDAG.

Obtaining a complete Directed Acyclic Graph (DAG) from a CPDAG and then utilizing the tools of the do-calculus requires substantial exogenous causal information. Various studies have enabled the utilization of these tools in partially oriented graphs.

Given a query about a causal effect, our first contribution, presented in Theorem 6, permits limiting the call for exogenous information to a specific subset of the graph. Secondly, we propose an algorithm to iteratively collect a sequence of expert-provided orientations that lead to the resolution of the query.

Our experiments indicate that our proposal significantly decreases the number of questions submitted to the expert; on average, the interaction is drastically limited to 2 or 3 questions, even for large graphs (up to 100 nodes).

This article establishes the validity of our approach, but there is still room for improvement. For instance, generalizing the criterion to do-calculus in a partial causal graph [18] or introducing dynamic programming to speed up the exploration, are promising avenues of research. Finally, our methodology, which assumes knowledge of a flawless CPDAG, could be adapted to handle uncertainty about learned (and therefore approximated) CPDAGs.

Acknowledgements

This work is conducted as part of a CIFRE thesis (no2020/1640) supported by SAP France and ANRT.

References

[1] S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 1997. ISSN 00905364.

[2] N. Burkart and M. F. Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70: 245–317, Jan. 2021.

[3] D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

[4] G. F. Cooper and C. Yoo. Causal discovery from a mixture of experimental and observational data. In *Proceedings of the 15th UAI Conference*, UAI'99, page 116–125, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[5] A. Deeks. The judicial demand for explainable artificial intelligence. *Columbia Law Review*, 119(7):1829–1850, 2019.

[6] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001.

[7] C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019.

[8] M. Hadj Ali, Y. Le Biannic, and P.-H. Wuillemin. Interpreting predictive models through causality: A query-driven methodology. In *The International FLAIRS Conference Proceedings*, volume 36, 2023.

[9] M. HADJALI, Y. LE BIANNIC, and P.-H. WUILLEMIN. Targeted exogenous causal knowledge. https://github.com/mahdihadjali/TargetedExogenousCausalKnowledge, 2024. [Accessed 22-08-2024].

[10] A. Hauser and P. Bühlmann. Two optimal strategies for active learning of causal models from interventional data. *Int. J. Approx. Reasoning*, 55(4):926–939, jun 2014.

[11] T. Heskes, E. Sijben, I. G. Bucur, and T. Claassen. Causal Shapley values: Exploiting causal knowledge to explain individual predictions of complex models, 2020.

[12] V. Louis, N. Sella, S. Affeldt, P. P. Singh, and H. Isambert. Learning causal networks with latent variables from multivariate information in genomic data. *Public Library of Science Computational Biology*, 2017.

[13] M. H. Maathuis and D. Colombo. A generalized back-door criterion. *The Annals of Statistics*, 43(3):1060–1088, 2015.

[14] C. Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the 11th UAI Conference*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[15] J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

[16] J. Pearl. Direct and indirect effects. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[17] J. Pearl and T. Verma. A theory of inferred causation. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 441–452, 1991.

[18] E. Perković. Identifying causal effects in maximally oriented partially directed acyclic graphs. In J. Peters and D. Sontag, editors, *Proceedings of the 36th UAI*, volume 124 of *Proceedings of Machine Learning Research*, pages 530–539. PMLR, 03–06 Aug 2020.

[19] E. Perković, J. Textor, M. Kalisch, and M. H. Maathuis. A complete generalized adjustment criterion. In *Uncertainty in Artificial Intelligence-Proceedings of the 31st Conference (2015)*, pages 682–691. AUAI Press, 2015.

[20] P. Spirtes, C. Meek, and T. Richardson. Causal inference in the presence of latent variables and selection bias. In *Proceedings of the 11th UAI Conference*, UAI'95, page 499–506, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[21] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT press, 2000.

[22] E. Tjoa and C. Guan. A survey on explainable artificial intelligence (xai): Toward medical xai. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813, 2021. doi: 10.1109/TNNLS.2020.3027314.

[23] J. Wang, J. Wiens, and S. Lundberg. Shapley flow: A graph-based approach to interpreting model predictions. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 721–729. PMLR, 13–15 Apr 2021.

[24] Q. Zhao and T. Hastie. Causal interpretations of black-box models. *Journal of business and economic statistics*, 2019, 2019.