Integrating Background Knowledge for Scalable Causal Discovery

Mátyás Schubert
University of Amsterdam

Theofanis Aslanidis*
University of Amsterdam

Tom Claassen
Radboud University Nijmegen

Sara Magliacane
Saarland University
University of Amsterdam

Abstract

Expert background knowledge is often available in practical applications of causal discovery. Such constraints on the true causal graph can help causal discovery in terms of identifiability of causal effects and accuracy of the learned structure, but also in reducing the space of candidate causal graphs. As causal discovery can become computationally expensive for large number of variables, it is crucial to utilize background knowledge effectively *during* the causal discovery process. However, most current methods only use background knowledge in a postprocessing step after causal discovery to refine the learned graph. In this work, we develop a framework for utilizing background knowledge during the causal discovery process, focusing especially on scalable causal discovery methods that recover only a subset of the whole graph. We implement our framework for multiple algorithms and empirically show that utilizing background knowledge can both reduce computational requirements and increase the quality of the learned structures.

1 Introduction

Causal graphs are a popular tool to represent the causal data generating process underlying observations [1]. Causal discovery aims to identify causal relations between causal variables by identifying (parts of) the causal graph from data [2]. These causal relations can then be used for causal effect estimation [3], to deepen our understanding of natural systems [4] and for policy-making [5]. However, causal discovery can become computationally demanding when applied to high-dimensional settings, which can limit its applicability in practical problems.

One way to address this issue is to adapt the objective of causal discovery and focus on the downstream task for which the learned causal relations will be used. For example, if the goal is to estimate the causal effect of a treatment on an outcome, then a large portion of the causal graph may be irrelevant. Motivated by this, several *scalable* causal discovery methods focus on learning only a set of relevant causal relations for a given causal inference task [6, 7, 8]. A popular approach to achieve this is local causal discovery, which identifies only the local structure of a target variable [9, 10, 11]. While these methods improve the computational efficiency and hence applicability of causal discovery methods to hundreds and thousands of variables, they rarely allow for the integration of expert knowledge.

In many real-world settings, experts may have background knowledge on the presence or lack of causal relations between variables, e.g., in the form of direct [12] or ancestral relations [13], or tiers arising in temporal studies [14, 15]. Background knowledge can also be extracted from experiments [16, 17], or even large language models [18, 19]. Such background knowledge can improve the

*Work performed while at University of Amsterdam.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

accuracy and identifiability of learned causal relations [20, 21, 22]. However, with rare exceptions, e.g., tiered background knowledge [14, 15], this knowledge is usually only applied a posteriori *after* causal discovery, and is not leveraged to reduce the search space and improve the scalability of causal discovery [12, 23]. While there exist implementations that integrate limited background knowledge during causal discovery for a specific global method, PC [24], we show that the current algorithms in `pcalg` [25], `causal-learn` [26] and `tetrad` [27] are unsound. Another exception are logic-based methods, which allow for rich background knowledge in the form of logic rules and its seamless integration in the search process, but they can only scale up to tens of variables [28, 29, 30].

In this work we develop sound methods to utilize background knowledge *during* causal discovery, to improve not only the accuracy and identifiability of the learned causal relations, but also the scalability of several current methods. We focus on background knowledge about direct causal relations, i.e., the existence and orientation of specific edges in the true causal graph, and show how it can be used to avoid unnecessary computations. Although integrating background knowledge sounds superficially intuitive, we show that there are some interesting counterintuitive cases that can lead to unsound conclusions if we integrate it naively. We implement our insights in global methods PC [24] and SNAP [8], making them more scalable. Then, we also integrate them into local methods MB-by-MB [9], LDECC [10] and LOAD [11], further increasing their computational efficiency. We prove that our extensions are sound in terms of propagating the background knowledge, and that all, except LDECC, are also complete. Finally, we show the benefits of integrating background knowledge into the causal discovery process in a thorough empirical study on simulated and realistic data, evaluating both scalability (time and number of CI tests) and accuracy of the estimated causal effects.

2 Background

We consider graphs $G = (\mathbf{V}, \mathbf{E})$ with nodes $\mathbf{V}$ and edges $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$. Edges can be undirected as $X - Y$, or directed as $X \to Y$. Undirected graphs contain only undirected edges, directed graphs contain only directed edges, and mixed graphs may contain both. We say that a node is adjacent to another if they are connected by an edge and denote the set of nodes adjacent to X in graph G as $Adj_G(X)$. In case of an undirected edge $X - Y$, we call the nodes siblings and denote the set of siblings of X as $Sib_G(X)$. If an edge is directed as $X \to Y$ then we say that X is a parent of Y and Y is a child of X, and denote the corresponding sets as $Pa_G(Y)$ and $Ch_G(X)$ respectively. The Markov blanket $MB_G(X)$ of X consists of its parents, children and the parents of its children.

A path between two nodes X and Y is a sequence of distinct adjacent nodes with one end in X and the other in Y. If all edges on a path between X and Y are directed towards Y then it is a directed path from X to Y. If there is a directed path from X to Y in a graph G then we say that X is an ancestor of Y and Y is a descendant of X, and denote the set of ancestors and descendants as $An_G(Y)$ and $De_G(X)$ respectively. We consider all nodes to be their own ancestors and descendants. We define adjacencies, siblings, parents, children, ancestors, and descendants on sets of nodes by taking the union, e.g., the nodes adjacent to a set of nodes $\mathbf{S}$ are $Adj_G(\mathbf{S}) = \cup_{X \in \mathbf{S}} Adj_G(X)$.

A directed path from a node to itself is a cycle. A directed acyclic graph (DAG) has only directed edges and no cycles. A causal DAG D can be used to represent the data generating process of a distribution p over some causal variables $\mathbf{V}$, where if a variable X has a direct causal effect on another variable Y then $X \to Y$ in D. We focus on observational distributions p and assume that p is Markov and faithful to D [24], thus every conditional independence (CI) $X \perp\!\!\!\perp Y | \mathbf{S}$ in p is equivalent to a d-separation $X \perp Y | \mathbf{S}$ in D. Furthermore, we assume no latent confounders or selection bias.

Under these assumptions, constraint-based causal discovery methods utilize CI tests to identify the set of causal graphs, including the true graph D, that imply the same set of conditional independences observed in p. This set of graphs is called the Markov equivalence class (MEC) of D can be represented by a unique mixed graph called the *completed partially directed acyclic graph (CPDAG)* [31]. The CPDAG and all DAGs in the MEC share the same adjacencies, and an edge is oriented in the CPDAG if and only if it is oriented the same way in all DAGs of the MEC. Otherwise, if there exist DAGs in the MEC with differing orientations of an edge, then it is undirected in the CPDAG.

Due to these differing orientations, some causal relations may hold in some DAGs in the MEC and not in others. Thus, we define *possible* causal relations on CPDAGs. If X is a parent of Y in at least one DAG in the CPDAG G, then we say that X is a possible parent of Y and denote it as $X \in \text{PossPa}_G(Y)$. We define possible children, ancestors and descendants analogously. In a CPDAG

G, X is a possible ancestor of Y, i.e., $X \in \text{PossAn}_G(Y)$, iff there exists a *possibly directed path* from X to Y, i.e., a path without edges directed towards X.

If there is expert background knowledge (BK) $\mathcal{B}$ available about the true graph D that is not implied by the observed distribution p, then the MEC of D can be further restricted. In this paper, we focus on background knowledge $\mathcal{B}$ that provides information about direct causal relations in the causal DAG D. We consider three types of BK: If $X - Y \in \mathcal{B}$, then X and Y are *adjacent* in D. If $X \to Y \in \mathcal{B}$, then X and Y are adjacent in D and the edge between them is *oriented* from X to Y. If $X \not- Y \in \mathcal{B}$, then there is a *gap* between X and Y in D, i.e., they are non-adjacent. We use $(X, Y) \in \mathcal{B}$ to denote that there is any BK about the pair, and $X \ast\!\!\to Y \in \mathcal{B}$ to denote that $X - Y, X \to Y$ or $X \leftarrow Y$ is in $\mathcal{B}$. For all our theoretical results, we assume that $\mathcal{B}$ is *consistent* with the true causal graph D i.e., it never contradicts it, but we also provide an ablation for imperfect knowledge in App. J.

Perković et al. [12] show how orientation BK can be used to obtain a *maximally oriented partially directed acyclic graph (MPDAG)*, that represents exactly the set of DAGs that adhere to both the observed distribution and the BK $\mathcal{B}$. A CPDAG can be restricted into an MPDAG by applying all orientation BK on it, and exhaustively orienting all edges using Meek's rules [31]. There might still remain undirected edges in the MPDAG. They also show that possible ancestral relations in MPDAGs appear differently than in CPDAGs and define *b-possible ancestors* in Def. 2.1.

Definition 2.1 (Def. 3.1 and 3.2 in [12])**.** *X is a b-possible ancestor of Y in MPDAG G, denoted as $X \in b\text{-}PossAn_G(Y)$, iff there is a b-possibly causal path from X to Y, i.e., a path $p = \langle X = V_0, \dots, V_k = Y \rangle$ such that exists no edge $V_i \leftarrow V_j$ for $0 \leq i < j \leq k$ in G.*

3 Integrating BK into Causal Discovery

Most previous work has explored on how to apply BK *after* causal discovery to constrain the equivalence class of the true DAG [12] or its local structures [23]. We instead aim to use background knowledge also *during* the discovery process to reduce the computational requirements required to identify it, by avoiding unnecessary CI tests. Our core approach is based on three general principles.

First, if $X \ast\!\!-\!\!\ast Y \in \mathcal{B}$, then we know that X and Y are adjacent in the true DAG D and so they cannot be separated by any CI test. Thus, we can skip all CI tests for these pairs of variables. This is already implemented in several causal discovery libraries, such as `pcalg` [25] and `causal-learn` [26]. Second, we can use $\mathcal{B}$ to limit the search space of candidate separating sets. As shown by Lemma 3.3.9 in [24], if X and Y are not adjacent, then they can be d-separated by the parents of X or Y in the underlying causal graph. Hence, when trying to separate X from Y, we can prune the candidate separating sets by considering only nodes for which $\mathcal{B}$ does not contain information that would disqualify them as possible parents of X. A similar strategy is implemented in `tetrad` [27].

The third principle considers integrating BK on gaps $X \not- Y \in \mathcal{B}$. Similarly to adjacencies, gaps inform us about non-adjacency without performing any CI tests. Intuition suggests that if the existence of an edge is already determined by a gap in the BK, then it is unnecessary to test for it. Thus, several popular implementations of PC [24] simply remove all edges corresponding to known gaps at the beginning of the discovery process [25, 27, 26, 32]. However, this intuition is wrong, as naively removing edges purely due to BK without verifying it with a CI tests can lead to incorrect outputs even with consistent BK and oracle CI tests, as we show in App. H. In particular, a gap known in advance by BK still needs a corresponding separating set to be recorded for correct orientations. We address this issue by delaying, instead of fully skipping, CI tests for known gaps. This delay generally allows us to find separating sets from a smaller set of candidate sets due to adjacencies by the end of discovery process. Our strategy of handling gaps is not only sound, but it can also automatically deal with cases when the BK and CI test results contradict each other, which could potentially lead to a missing separating set and hence propagate the error to other parts of the graph.

We first show these three principles in the seminal PC algorithm [24]. Then, we build on our algorithms to integrate BK into other methods including local methods focusing on scalability.

3.1 PC-BK

The PC algorithm [24] is a global causal discovery algorithm that aims to identify the full CPDAG of the true causal DAG D. The PC algorithm initializes a fully connected and undirected skeleton and refines it in three phases. The skeleton phase iterates over $i \in 0 \dots |\mathbf{V}| - 2$ orders,

Algorithm 1 SkelStep-BK

Require: skeleton $\hat{G}$, order i and BK $\mathcal{B}$
1: **for** $X \in \hat{G}, Y \in Adj_{\hat{G}}(X)$ s.t. $(X, Y) \notin \mathcal{B}$ **do**
2: **for** $\mathbf{S} \subseteq \text{PossPa}_{\hat{G}}(X, \mathcal{B}) \setminus \{Y\}$ s.t. $|\mathbf{S}| = i$ **do**
3: **if** $X \perp Y | \mathbf{S}$ **then**
4: Delete $X - Y$ from $\hat{G}$
5: $sepset(\{X, Y\}) \leftarrow \mathbf{S}$
6: **break**
Ensure: Skeleton $\hat{G}$, separating sets $sepset$

Algorithm 2 MissingSepsets-BK

Require: Skeleton $\hat{G}$, order i and BK $\mathcal{B}$
1: **for** $(X \neq Y) \in \mathcal{B}$ **do**
2: **for** $\mathbf{S} \subseteq \text{PossPa}_{\hat{G}}(X, \mathcal{B}) \setminus \{Y\}$ s.t. $|\mathbf{S}| = i$ **do**
3: **if** $X \perp Y | \mathbf{S}$ **then**
4: Delete edge $X - Y$ from $\hat{G}$
5: $sepset(\{X, Y\}) \leftarrow \mathbf{S}$
6: **break**
Ensure: Skeleton $\hat{G}$, separating sets $sepset$

Algorithm 3 PC-BK

Require: Variables $\mathbf{V}$ and BK $\mathcal{B}$
1: $\hat{G} \leftarrow$ fully connected undirected graph over $\mathbf{V}$
2: $sepset \leftarrow \emptyset$
3: **for** $i \in 0 \dots |\mathbf{V}| - 2$ **do**
4: $\hat{G}, sepset \leftarrow$ `SkelStep-BK`$(\hat{G}, i, \mathcal{B})$
5: **for** $i \in 0 \dots |\mathbf{V}| - 2$ **do**
6: $\hat{G}, sepset \leftarrow$ `MissingSepsets-BK`$(\hat{G}, i, \mathcal{B})$
7: $\hat{G} \leftarrow$ `VstructPC`$(\hat{G}, sepset)$ (Alg. 14)
8: $\hat{G} \leftarrow$ Apply all orientations in $\mathcal{B}$
9: $\hat{G} \leftarrow$ `Meek`$(\hat{G})$ (Alg. 12)
Ensure: MPDAG $\hat{G}$

where each step at order i aims to separate variables using separating sets of size i. Then, based on the final skeleton and collected separating sets, PC orients v-structures as shown in Alg. 14. Finally, it orients as many edges as possible using Meek's orientations rules in Alg. 12.

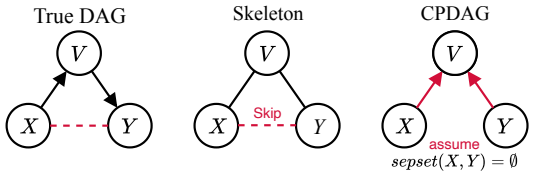


Figure 1: Skipping a CI test between X and Y due to a known gap $\mathcal{B} = X \neq Y$ and assuming that the separating set between X and Y is empty, leads to erroneously orienting an unshielded triple $X - V - Y$ as a v-structure in the CPDAG.

While there exist PC implementations that integrate limited background knowledge during causal discovery, which we discuss in detail in App. H, the current algorithms in `pcalg` [25], `causal-learn` [26] and `tetrad` [27] are unsound, while `pgmpy` [32] crashes due to conflicting information. In particular, using gap BK before causal discovery can lead to skipping CI tests, which can potentially result in incorrect orientations. This can result in assuming the wrong separating sets, as done in `pcalg` [25] or `causal-learn` [26], and therefore orienting an incorrect v-structure as we show in Fig. 1. In this specific example, `tetrad` [27] would correctly not orient this v-structure, because if a separating set is not defined (e.g. due to skipping a CI test), it is not used in orienting v-structures. On the other hand, if the true DAG is a v-structure $X \to V \leftarrow Y$, `tetrad` would also not orient this unshielded triple, leading to a wrong CPDAG. We provide additional details, code examples to reproduce incorrect outputs, and empirical comparisons against these methods in App. H. Our empirical results show that a higher rate of BK only increases the graph estimation accuracy of our approach, while it actually decreases it for the algorithms provided by these libraries on most settings, including oracle d-separation CI tests.

Besides the unsoundness issues, current PC implementations also do not support all types of direct background knowledge we consider, i.e., gaps, adjacencies and orientations. To address these issues, we propose PC-BK, described in Alg. 3. Unlike other implementations of PC, we do not remove edges that are known gaps from the skeleton immediately at the beginning of the algorithm. Instead, PC-BK performs the skeleton discovery in two passes at lines 4 and 6. The first pass, as implemented in Alg. 1, performs the same amount of CI tests as if the gaps were removed from the skeleton in the beginning. It skips all pairs of variables with available BK, including known gaps, at line 1. For every other pair, we limit the separating sets at line 2 to contain only variables that are still possible parents according to the current skeleton $\hat{G}$ and the given BK $\mathcal{B}$ by defining the function $\text{PossPa}_{\hat{G}}(X, \mathcal{B}) = Adj_{\hat{G}}(X) \setminus \{V \in \mathbf{V} | X \to V \in \mathcal{B} \vee X \neq V \in \mathcal{B}\}$.

Algorithm 4 SNAP(k)-BK

Require: variables $\mathbf{V}$, targets $\mathbf{T} \subseteq \mathbf{V}$, max order k, BK $\mathcal{B}$
1: $\hat{U} \leftarrow$ fully connected undirected graph over $\hat{\mathbf{V}} = \mathbf{V}$
2: $seps \leftarrow \emptyset$
3: **for** $i \in 0 \dots k$ **do**
4: $\quad \hat{U} \leftarrow$ induced subgraph of $\hat{U}$ over all $\hat{\mathbf{V}}$
5: $\quad \hat{U}, seps \leftarrow$ `SkelStep-BK`$(\hat{U}, i, \mathcal{B})$ (Alg. 1)
6: $\quad \hat{U}, seps \leftarrow$ `MissingSepsets-BK`$(\hat{U}, i, \mathcal{B})$ (Alg. 2)
7: $\quad$ **if** $i = 0$ **then**
8: $\quad\quad \hat{G} \leftarrow$ `VstructPC`$(\hat{U}, sepset)$ (Alg. 14)
9: $\quad$ **else**
10: $\quad\quad \hat{G}, seps \leftarrow$ `VstructRFCI`$(\hat{U}, seps)$ (Alg. 15)
11: $\quad\quad \hat{U} \leftarrow$ skeleton of $\hat{G}$
12: $\quad \hat{G} \leftarrow$ Apply all orientations in $\mathcal{B}$
13: $\quad \hat{\mathbf{V}} \leftarrow V \in \hat{\mathbf{V}}$ with poss. dir. paths to $T \in \mathbf{T}$ in $\hat{G}$
14: $\hat{G} \leftarrow$ induced sub-graph of $\hat{G}$ over $\hat{\mathbf{V}}$
Ensure: Remaining variables $\hat{\mathbf{V}}$, graph $\hat{G}$

Algorithm 5 SNAP(∞)-BK

Require: variables $\mathbf{V}$, targets $\mathbf{T} \subseteq \mathbf{V}$, BK $\mathcal{B}$
1: $\hat{\mathbf{V}}, \hat{G} \leftarrow$ `SNAP(`k`)-BK`$(\mathbf{V}, \mathbf{T}, |\mathbf{V} - 2|, \mathcal{B})$ (Alg. 4)
2: $\hat{G} \leftarrow$ `Meek`$(\hat{G})$ (Alg. 12)
3: $\hat{G} \leftarrow$ induced subgraph of $\hat{G}$ over all $V \in \mathbf{V}$ with a b-possibly directed path to any $T \in \mathbf{T}$ in $\hat{G}$
Ensure: MPDAG $\hat{G}$

Algorithm 6 MB-by-MB-BK

Require: Variables $\mathbf{V}$, target $T \in \mathbf{V}$, BK $\mathcal{B}$
1: $sepset, MB, L,$ DoneList $\leftarrow \emptyset, \emptyset, \emptyset, \emptyset$
2: WaitList $= [T]$
3: $\hat{G} = (\mathbf{V}, \emptyset)$
4: **repeat**
5: $\quad X \leftarrow$ take a node from the head of WaitList
6: $\quad MB(X) \leftarrow$ GrowShrink-BK($\mathbf{V}, X, \mathcal{B}$) (Alg. 10)
7: $\quad$ WaitList $\leftarrow$ Add $[MB(X) \setminus$ DoneList $\setminus$ WaitList$]$
8: $\quad$ DoneList $\leftarrow$ DoneList $\cup X$
9: $\quad$ **if** $MB^+(X) \subseteq MB^+(X')$ for some $X' \in$ DoneList **then**
10: $\quad\quad$ Set L_X to the subgraph of $L_{X'}$ over $MB^+(X)$
11: $\quad$ **else if** $MB(X) \subseteq$ DoneList **then**
12: $\quad\quad$ Set L_X to the subgraph of G over $MB^+(X)$
13: $\quad$ **else**
14: $\quad\quad L_X \leftarrow$ PC-BK($MB(X)^+, \mathcal{B}$) (with $PossPa^*_{\hat{G}}(\cdot, \mathcal{B})$)
15: $\quad G \leftarrow$ Add edges and v-structures incl. X in L_X
16: $\quad \hat{G} \leftarrow$ Apply all orientations in $\mathcal{B}$
17: $\quad \hat{G} \leftarrow$ `OrientUndirectedEdges`$(\hat{G})$ (Alg. 13)
18: $\quad$ WaitList $\leftarrow$ Remove all nodes whose paths to T in G are blocked by directed edges
19: **until** WaitList is empty
Ensure: $Pa_{\hat{G}}(T)$, $Ch_{\hat{G}}(T)$, $Sib_{\hat{G}}(T)$

As discussed earlier, simply removing edges corresponding to known gaps can lead to unsoundness, if the corresponding separating sets are undefined. Thus, we perform a second pass in the skeleton search specifically to find separating sets corresponding to all known gaps in $\mathcal{B}$ in Alg. 2. Delaying this search after the first pass allows us to find separating sets for gaps from a smaller set of candidate sets due to a sparser skeleton and hence fewer possible parents. Once the missing separating sets are identified, the v-structures can be oriented as usual. If we cannot find any separating set for $X \neq Y \in \mathcal{B}$, possibly due to noisy finite sample data, we keep the edge between X and Y in the skeleton. This ensures that even if $\mathcal{B}$ is inconsistent with the data, our algorithm does not fail. Finally, orientations in $\mathcal{B}$ can be utilized as a post-processing step via orientation rules (Alg. 12) as in [31, 12] at line 9. We show that Alg. 3 is sound and complete in Thm. 3.1 and prove it in App. C.1.

Theorem 3.1. *Given oracle CI tests and consistent BK, PC-BK outputs the true MPDAG.*

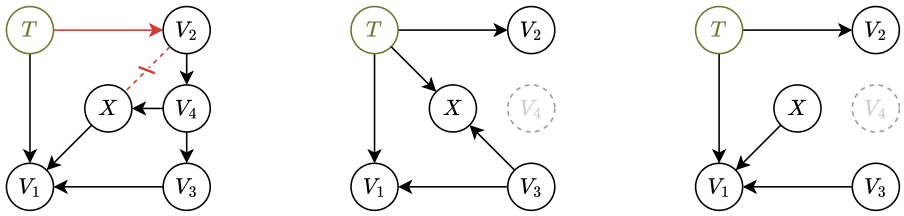
We show how the same principles and subroutines of PC-BK can be integrated into other causal discovery methods, with algorithm-specific adaptations as needed.

3.2 SNAP-BK

Schubert et al. [8] show that for a set of targets $\mathbf{T}$, only their possible ancestors are required to identify their causal relations and efficient adjustment sets. They propose the SNAP(k) algorithm to prune all non-ancestors identified over k iterations, and extend it to SNAP(∞), a stand-alone causal discovery algorithm that is sound and complete over the possible ancestors of targets.

We implement SNAP(k)-BK and SNAP(∞)-BK in Alg. 4 and Alg. 5 respectively. SNAP(k)-BK employs a similar skeleton search as PC-BK, but iteratively orients v-structures and prunes the identified non-ancestors of the targets at every step. Hence, the two subroutines Alg. 1 and Alg. 2 can be easily integrated into SNAP(k)-BK at every order at lines 5 and 6. However, limiting candidate separating sets and orienting v-structures at early steps can lead to SNAP(k) incorrectly pruning ancestors of the targets, as shown in Ex. C.2. We solve this by orienting v-structures using the RFCI orientation rules [33] (Alg. 15) already at order $i = 1$ instead of at $i = 2$ as in the original paper. To use BK even further, we apply all orientations in $\mathcal{B}$ at line 12 on the graph $\hat{G}$ at every iteration.

After this, the original SNAP(k) algorithm prunes variables at each iteration that are not part of the possibly ancestral set containing the targets. Analogously, SNAP(k)-BK prunes variables that are not in the *b-possibly ancestral set* $\mathbf{V}^* \subseteq \mathbf{V}$, which is defined as any set closed under b-possible ancestry,



(a) True DAG/MPDAG. (b) Naive MB-by-MB-BK. (c) Our MB-by-MB-BK.

Figure 2: On the DAG/MPDAG shown in Fig. 2a with $\mathcal{B} = \{T \to V_2, V_2 \neq X\}$ denoted in red, a naively implemented MB-by-MB-BK incorrectly identifies X as a child of target T (Fig. 2b). Our method recovers the correct relations (Fig. 2c). Details are explained in Ex. C.3.

i.e., such that b-PossAn$_G(\mathbf{V}^*) \subseteq \mathbf{V}^*$. In Lem. C.1 we show that if a node V is in a b-possibly ancestral set, then all other nodes with a possibly directed path to V are also in the set. Hence we keep the pruning step at line 13 the same as in the original paper.

We provide two theoretical results, showing that our extensions integrate BK correctly and efficiently into the two SNAP methods. All proofs for SNAP are in App. C.2. We first show that the remaining nodes at every iteration of SNAP(k) form a b-possibly ancestral set that contains $\mathbf{T}$:

Theorem 3.2. *Given oracle CI tests and consistent BK, at each step $i = 0, \dots, k$ of SNAP(k)-BK for the remaining variables $\hat{\mathbf{V}}$ that are output by SNAP, it holds that $\mathbf{T} \subseteq \hat{\mathbf{V}}$ and b-PossAn$_G(\hat{\mathbf{V}}) \subseteq \hat{\mathbf{V}}$.*

Consequently, SNAP(∞)-BK is sound and complete over the b-possible ancestors of the targets.

Corollary 3.1. *Given true MPDAG G, oracle CI tests and consistent BK, let $\hat{G}$ be the output graph of SNAP(∞)-BK for targets $\mathbf{T}$. Then, SNAP(∞)-BK is sound and complete over the b-possible ancestors $\mathbf{T}$, i.e. $\hat{G} = G|_{\text{b-PossAn}(\mathbf{T})}$, and $\hat{G}$ is an MPDAG.*

We now turn to local methods that identify and orient edges around target variables without discovering the whole CPDAG. To find the variables adjacent to a target T, many local methods begin by finding its Markov blanket $MB(T)$ using existing algorithms as a sub-routine. We show how to utilize BK for Markov blanket discovery in App. A, where we provide two extensions for GrowShrink [34] and Total Conditioning [35] that we prove are sound and complete.

As local methods aim to orient edges as soon as possible by exploiting independence patterns, we find that they generally do benefit from searching for separating sets corresponding to known gaps without delay. Thus, when applying Alg. 1 in local methods, we swap the condition $(X, Y) \notin \mathcal{B}$ at line 1, which includes known gaps, with $(X \ast\!\!-\!\!\ast Y) \notin \mathcal{B}$, which only considers known adjacencies.

3.3 MB-by-MB-BK

MB-by-MB [9] is a local causal discovery method that iteratively discovers the Markov blankets $MB(X)$ and local structures around variables X, starting from the target $X = T$, and combines these local structures to identify the parents, children and siblings of T without discovering the global structure. MB-by-MB can directly use one of *our extended versions of MB discovery* described in App. A, e.g., Grow-Shrink-BK in Alg. 10 or Total Conditioning-BK in Alg. 11. However, naively applying PC-BK instead of the original PC over the learned Markov blankets can lead to erroneous results on the local structures, as shown in Fig. 2. In this example, restricting the candidate separating sets only to possible parents prevents us from using V_2 to separate target T and node X even though it is the only available separator in $MB(T)$. In general, this occurs when discovering the local structure of $MB(X)$, but none of the possible parents of a node $V \neq X$ are in $MB(X)$. Thus, we run PC-BK to discover the local structure at line 14 with an adjusted definition of possible parents that only uses BK around X, but ignores it for every other $X \neq V \in MB(X)$, as

$$PossPa^*_{\hat{G}}(V, \mathcal{B}) = \begin{cases} \text{PossPa}_{\hat{G}}(V, \mathcal{B}) & \text{if } V = X \\ Adj_{\hat{G}}(V) & \text{otherwise} \end{cases}. \tag{1}$$

Algorithm 7 LDECC-BK

Require: Variables $\mathbf{V}$, target $T \in \mathbf{V}$, BK $\mathcal{B}$
1: $MB \leftarrow$ `Grow-Shrink-BK`$(\mathbf{V}, T, \mathcal{B})$ (Alg. 10)
2: $Ne, sepset \leftarrow$ `LocalPC-BK`$(T, MB, \mathcal{B})$ (Alg. 16)
3: $Ch \leftarrow$ `UCChildren`$(MB, Ne, sepset)$ (Fig. 4 of [10])
4: $Ch \leftarrow Ch \cup \{V \in \mathbf{V} | T \to V \in \mathcal{B}\}$
5: $Pa \leftarrow \{V \in \mathbf{V} | V \to T \in \mathcal{B}\}$
6: $U \leftarrow$ fully connected undirected graph over $\mathbf{V}$
7: **for** $A \perp\!\!\!\perp B | \mathbf{S} \in$ `SkelStep-BK`$(U, 0.., \mathcal{B})$, $A, B \neq T$ **do**
8: Identical to lines 7 to 22 of Fig. 5 of [10])
9: $Sib \leftarrow Ne \setminus (Pa \cup Ch)$
Ensure: Pa, Ch, Sib

Algorithm 8 LOAD-BK

Require: Variables $\mathbf{V}$, targets $X, Y \in \mathbf{V}$, BK $\mathcal{B}$
1: **if** $X \to Y \in \mathcal{B}$ **then**
2: $T, O \leftarrow X, Y$
3: **else if** $Y \to X \in \mathcal{B}$ **then**
4: $T, O \leftarrow Y, X$
5: **else**
6: Step 1 of LOAD (Alg. 3. in [11])
7: Steps 2 and 3 of LOAD (Alg. 3. in [11])
8: Steps 4 of LOAD with BK on adjacencies and orientations over nodes not projected out (Alg. 3. in [11])
Ensure: Causal relation, identifiability, adjustment sets

Similarly to SNAP(k)-BK, MB-by-MB-BK can also apply orientations in $\mathcal{B}$ at every iteration to terminate quicker. We implement MB-by-MB-BK in Alg. 6 and show that it is sound and complete in Thm. 3.3 with proof in App. C.3.

Theorem 3.3. *Given oracle CI tests and consistent BK, MB-by-MB-BK is sound and complete in identifying the parents, children and siblings of the target.*

3.4 LDECC-BK

LDECC [10] is another local causal discovery algorithm with the same objective as MB-by-MB. We implement LDECC-BK in Alg. 7. It begins by finding the Markov blanket of the target T and then identifies its neighbors which are not already known, as shown in Alg. 16. Then, it identifies some of the children and parents of the target using the Markov blanket and the provided BK.

After this, the algorithm iterates through the same CI tests that Alg. 1 would perform, but tries to orient edges early using *eager collider checks*. This means that the same way as earlier BK algorithms, LDECC-BK can skip any CI test for pairs known to be adjacent by $\mathcal{B}$, and restrict candidate separating sets to subsets of possible parents. The rest of the algorithm is identical to lines 7 to 22 of the original algorithm, where LDECC identifies the parents, children and siblings of the target. However, by utilizing orientation BK during discovery to identify parents at line 5, LDECC-BK may identify additional children of the target through non-collider checks at line 19 of the original algorithm, that simple post-processing would not find.

LDECC-BK relies purely on CI relations to orient edges instead of propagating orientation rules. Thus, as shown in Ex. C.4, it is unable to utilize orientation BK that is outside of the Markov blanket of the target and not implied by the data. This can lead to parents or children getting misclassified as siblings. We found no way to propagate orientations without significantly changing the algorithm. Thus, we only show the soundness of LDECC-BK in Cor. 3.2, which follows from the soundness and completeness of Grow-Shrink-BK in Thm. A.1 and Thm. 5 of [10].

Corollary 3.2. *Given oracle CI tests and consistent BK, LDECC-BK is sound and complete in identifying adjacencies and sound in identifying parents and children of the target.*

3.5 LOAD-BK

LOAD [11] uses local causal discovery algorithms, such as MB-by-MB, to first discover the causal relations of a pair of targets, and then the optimal adjustment set in a modified forbidden projection, to estimate the causal effect between them. Thus, LOAD-BK can utilize our framework indirectly through MB-by-MB-BK. However, BK on gaps might not hold anymore under the modified forbidden projection. Thus, we utilize BK only on adjacencies and orientations in the last step of LOAD-BK. Furthermore, given a target pair X and Y, if $X \to Y$ or $Y \to X$ is in $\mathcal{B}$, then LOAD-BK can skip its first step that aims to determine the causal relation between the targets, as shown in Alg. 8. We show that LOAD-BK is sound and complete Thm. 3.4 and provide proof in App. C.5, for which we extend the modified forbidden projection of Schubert et al. [11] to MPDAGs.

Theorem 3.4. *Given oracle CI tests and consistent BK, LOAD-BK is sound and complete in finding optimal adjustment set for a pair of target variables.*

4 Related Work

We focus on BK about direct causal relations, i.e., whether an edge exists and how it is oriented in the true causal DAG. Meek [31] has done pioneering work on orientation BK, and developed the orientation rules also utilized in this work, shown in Alg. 12, where the fourth rule is specifically designed for extra orientations arising from BK. This theory was further developed by Perković et al. [12] who show how to maximally orient edges and obtain a MPDAG from an available CPDAG.

Fang et al. [13] consider background knowledge on the direction of causal relations such as orientations, and also ancestral and non-ancestral relations, but assume an already given CPDAG. Zheng et al. [23] explore how to use BK on orientations, as well as (non-) ancestral relations to refine the local structures discovered by the MB-by-MB algorithm, which may allow MB-by-MB to terminate earlier, but they do not explicitly avoid unnecessary CI tests like Alg. 6. We discuss how ancestral BK can be utilized by our approach in App. D.1.

A special type of non-ancestral BK is given by ordered tiers of variables arising in, e.g., longitudinal data, which can be represented by tiered MPDAGs [14]. Tiered BK is quite different from the pairwise BK that we consider, as it provides knowledge on (non-)ancestral relations, rather than direct ones. Andrews et al. [15] integrate tiered BK during causal discovery with latent variables by processing tiers one-by-one, discovering only the edges within each tier and into it from earlier tiers. Bang et al. [36] develop the tiered PC (tPC) algorithm to discover tiered MPDAGs and exploit tiered BK during the discovery process by excluding conditioning sets containing future-tier variables, which stems from the same insight by which we exclude all variables that are not possible parents. Tiered BK can be also utilized by causal discovery algorithms allowing for latent variables and multiple datasets [15, 37] as well as in score-based causal discovery [38].

In the causally insufficient case, Wang et al. [39] consider local BK, which fully orients all edges adjacent to any node, and introduce additional orientation rules. Venkateswaran and Perković [40] extend this to general orientation BK, and provide a generalization for the orientation rules of Meek [31] for latent variables. Wang et al. [41] uses BK to efficiently enumerate all graphs in the equivalence class. These works utilize BK after causal discovery instead of during the process.

Logic-based causal discovery methods, e.g., [42, 28, 29, 16, 30], can seamlessly integrate rich background knowledge in the form of logical rules or constraints on causal edges or paths in the causal discovery process, including in challenging settings like latent confounding, selection bias, cycles or data from multiple contexts. In particular, Hyttinen et al. [28], Magliacane et al. [16], Mooij et al. [30] formulate causal discovery as a constraint optimization problem, where weights can be added to each constraint, which allows to model noisy or imperfect knowledge. Since these methods are based on constraints, they can leverage BK during the discovery process to reduce the search space. On the other hand, these methods can only scale up to tens of variables, as opposed to the methods we consider, which can scale up to hundreds or even a few thousand variables.

Gonzales et al. [43] utilize imperfect experts that provide a corresponding confidence probability for each orientation BK. In particular, their algorithm employs a similar strategy to tPC, where descendants of a variable according to the BK are excluded from separating sets. Chen et al. [44] also consider possibly imperfect orientation BK and develop methods to detect and self-correct when they lead to structural inconsistencies during causal discovery.

There are several proposals on how to integrate large language models (LLMs) into causal discovery as imperfect experts [18]. Long et al. [45] and Takayama et al. [46] use LLMs to refine the learned graph after the discovery process. Kampani et al. [47] and Wang et al. [48] use LLMs for prior knowledge to initialize causal discovery, but learn a DAG through heuristics or continuous optimization, instead of a MPDAG with theoretical guarantees. Hiremath et al. [49] and Ejaz and Bareinboim [50] use BK during causal discovery to prioritize edge deletions and additions, instead of avoiding them.

5 Experiments

We evaluate the effects of utilizing BK for all methods described in Sec. 3. For all algorithms except PC, we randomly sample a treatment-outcome target pair where the treatment is an ancestor of the outcome in the true graph. Since we focus on BK about direct causal relations, we do not provide this ancestral relation as BK. Following Schubert et al. [11], we extend MB-by-MB-BK and LDECC-BK

to also determine the ancestral relation of the targets, denoted as MB-by-MB$^+$-BK and LDECC$^+$-BK. As baselines, we compare our approach to the post-processing alternative. For PC and SNAP(∞) we use Perković et al. [12], for MB-by-MB$^+$ and LOAD we use [23], and for LDECC$^+$ we add or remove parents and children to its output based on the BK. We provide additional discussion and results on post-processing in App. G. We publish our code at `https://github.com/matyasch/bk`.

We report the number of CI tests as an implementation independent measure of computational efficiency, while execution time provides a practical metric. Since not all methods output a causal graph, we only report structural metrics for PC-BK and SNAP(∞)-BK in App. F and instead measure the intervention distance [8], which compares the true causal effect between the targets and the causal effect estimated based on the output of the algorithm using 1000 new data points [51]. This aligns better with the purpose of many of these algorithms, i.e., downstream causal effect estimation. We repeat each experiment 100 times and average results over 90 with the best and worst 5 results removed. We provide additional details on libraries and hardware used for all experiments in App. E.

Synthetic Data. We evaluate all algorithms on data sampled from random Erdős–Rényi causal models with 1000 nodes, expected degree of 3, and maximum degree of 10. We evaluate on three different data domains and corresponding CI tests: we use oracle d-separation CI tests to demonstrate the theoretical advantages of our framework, and also run experiments with 10000 samples of linear Gaussian and binary data. For the linear Gaussian data we sample edge weights from the range of $[-3, 0.5] \cup [0.5, 3]$ and use standard Gaussian deviation, and employ Fisher-Z CI tests. The binary data is generated from a conditional probability table based on the causal graph with uniform probabilities, and we use G^2 CI tests. We use a significance level of $\alpha = 0.01$ or all CI tests.

For each experiment, we sample BK between a fraction of all variable pairs, according to the *rate* of BK. If two variables are non-adjacent and selected for BK, then we register it as a known gap. Otherwise, it can be registered in $\mathcal{B}$ either as an adjacency or as an orientation with uniform probability. This means that a BK rate of 1 provides information about all variable pairs, but not necessarily all orientations, and hence may not identify the true causal DAG. We do not show results for LDECC with Fisher-Z CI tests since its running times exceeded our threshold of 24 hours.

In Fig. 3 we show the difference in each metric between our extended methods and each original method combined with postprocessing. A lower value than zero indicates a decrease in the number of tests, execution time or distance between the true and estimated causal effects, while a higher value than zero indicates an increase. The results demonstrate that our strategy of utilizing BK can improve both computational requirements compared to post-processing, while maintaining the same output quality over most data types and algorithms. The first two rows show the number of CI tests and running time, and indicate that BK can consistently reduce the cost of all causal discovery algorithms (see Fig. 12 for SNAP(∞)-BK and Fig. 15 for LOAD-BK). These improvements are the most beneficial for PC-BK and LDECC$^+$-BK compared to post-processing.

The third row of Fig. 3 shows our results for the intervention distance, where our approach maintains the performance of post-processing. In line with our theoretical proofs, the prediction accuracy of our approach is identical to post-processing in the oracle d-separation setting for all methods except LDECC, which slightly improves. Orientation BK can constrain the equivalence class of the true graph, and thus the set of possible adjustment sets, and sometimes turn a causal effect fully identifiable. This benefit can be clearly observed from the third row of Fig. 3 with oracle d-separation CI tests, where all methods obtain more accurate estimates of the true causal effect. Since LDECC-BK is not complete, its intervention distance is slightly higher than for MB-by-MB-BK. However, it achieves a lower distance compared to the post-processing baseline as it can identify more children during discovery. On linear Gaussian data, similar benefits can be observed for PC-BK, SNAP(∞)-BK and LDECC$^+$-BK. On binary data, the intervention distance is quite low throughout BK rates.

Ablations. In App. I we study which type of BK benefits our methods the most by providing each type separately. Our results show that PC-BK and SNAP(∞)-BK benefit much more from known gaps to reduce computational costs than local methods, which benefit the most from orientation BK. As expected, orientation BK improves the intervention distance for all algorithms and data modalities.

We consider *imperfect BK* in App. J by introducing errors into 10% of adjacencies and orientations, and 5% of gaps to keep errors balanced between BK types, and show the results in Fig. 16. While MB-by-MB$^+$-BK, and LDECC$^+$-BK still maintain decreasing computational costs as BK increases

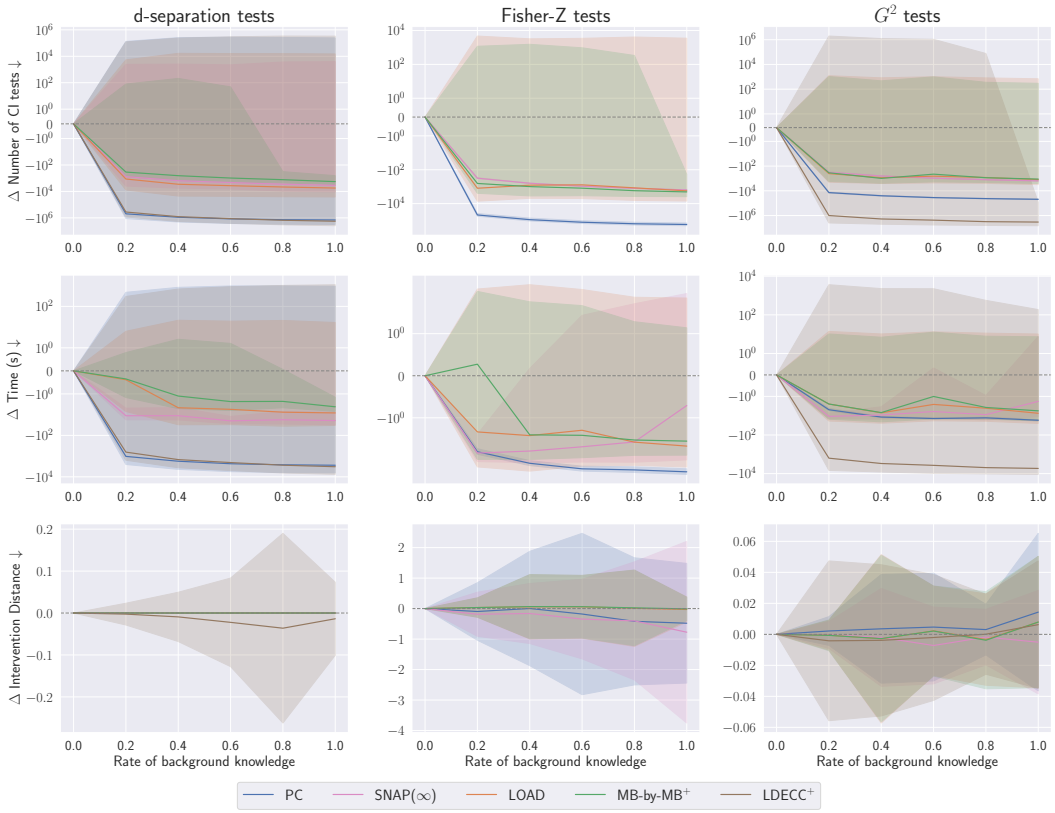


Figure 3: Difference between our methods and baselines combined with postprocessing over rate of BK between variable pairs, where lower means better. The shadow area denotes standard deviation.

despite errors, SNAP(∞)-BK becomes increasingly slow at higher BK rates. Intervention distances increase with more imperfect BK due to the growing number of incorrect orientations, accompanied by larger variance. We also vary the error rate from 10%, 20% and 30% for a fixed BK rate of 0.2. Our results in Fig. 17 show that for most methods the computational cost and intervention distance increase with higher error rates, although MB-by-MB$^+$-BK, which gets slightly cheaper. Fig. 8 shows that our approach remains comparable against post-processing under erroneous BK.

Realistic Data. We also evaluate our approach on data generated by the MAGIC-NIAB, ECOLI70, ARTH150 and ANDES networks from the `bnlearn` repository [52]. Our results in App. K show the same benefits of integrating BK as for synthetic data. The computational costs of most methods improve over all settings, especially on linear Gaussian data. The main exception is SNAP(∞) which shows a constant cost over all rates of BK Without any background knowledge, the PC(-BK) and LDECC$^+$(-BK) algorithms took prohibitively long to run on the ARTH150 network, demonstrating that utilizing BK during discovery can be necessary in some scenarios.

6 Conclusions

In this work, we develop methods to integrate background knowledge about direct causal relations *during* the causal discovery process. This allows us to improve not only the accuracy and identifiability of the learned causal relations, but also the scalability of current methods. We implement our insights in two global causal discovery methods, three local causal discovery methods and two Markov blanket discovery algorithms from literature. We prove that our extensions are sound in terms of propagating the background knowledge, and that all, except LDECC-BK, are also complete. Our empirical results show that integrating background knowledge can reduce the computational cost of these methods in several settings, with gains that depend on the method, CI test and BK type. We plan to extend our results to settings with latent confounding and selection bias in future work.

Acknowledgements

We thank SURF (www.surf.nl) for the support in using the National Supercomputer Snellius.

Funding Transparency Statement

The authors did not receive third-party support in relation to this project and declare no competing interests. In particular, while TA works at Uber, this work was performed during his Master thesis at the University of Amsterdam. Similarly, while SM had previously worked part-time for the MIT-IBM Watson AI Lab, this work is not related to the topics and projects she contributed to there.

References

[1] Judea Pearl. *Causality*. Cambridge university press, 2009.

[2] Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00524.

[3] Emilija Perković, Johannes Textor, Markus Kalisch, and Marloes H. Maathuis. Complete graphical characterization and construction of adjustment sets in markov equivalence classes of ancestral graphs. *Journal of Machine Learning Research*, 18(220):1–62, 2018. URL http://jmlr.org/papers/v18/16-319.html.

[4] Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):2553, 2019.

[5] Jacqueline Maasch, Kyra Gan, Violet Chen, Agni Orfanoudaki, Nil-Jana Akpinar, and Fei Wang. Local causal discovery for structural evidence of direct discrimination. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):19349–19357, Apr. 2025. doi: 10.1609/aaai.v39i18.34130. URL https://ojs.aaai.org/index.php/AAAI/article/view/34130.

[6] David S Watson and Ricardo Silva. Causal discovery under a confounder blanket. In *Uncertainty in Artificial Intelligence*, pages 2096–2106. PMLR, 2022.

[7] Jacqueline R. M. A. Maasch, Weishen Pan, Shantanu Gupta, Volodymyr Kuleshov, Kyra Gan, and Fei Wang. Local discovery by partitioning: Polynomial-time causal discovery around exposure-outcome pairs. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024.

[8] Mátyás Schubert, Tom Claassen, and Sara Magliacane. Snap: Sequential non-ancestor pruning for targeted causal effect estimation with an unknown graph. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan, editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 3340–3348. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/schubert25a.html.

[9] Changzhang Wang, You Zhou, Qiang Zhao, and Zhi Geng. Discovering and orienting the edges connected to a target variable in a dag via a sequential local learning approach. *Computational statistics & data analysis*, 77:252–266, 2014.

[10] Shantanu Gupta, David Childers, and Zachary Chase Lipton. Local causal discovery for estimating causal effects. In *Conference on Causal Learning and Reasoning*, pages 408–447. PMLR, 2023.

[11] Mátyás Schubert, Tom Claassen, and Sara Magliacane. Local causal discovery for statistically efficient causal inference. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026. URL https://openreview.net/forum?id=FlWl20PFd7.

[12] Emilija Perković, Markus Kalisch, and Marloes H. Maathuis. Interpreting and using cpdags with background knowledge. In Gal Elidan, Kristian Kersting, and Alexander Ihler, editors, *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI 2017, Sydney, Australia, August 11-15, 2017*. AUAI Press, 2017. URL http://auai.org/uai2017/proceedings/papers/120.pdf.

[13] Zhuangyan Fang, Ruiqi Zhao, Yue Liu, and Yangbo He. On the representation of pairwise causal background knowledge and its applications in causal inference. *Journal of Machine Learning Research*, 26(229):1–73, 2025. URL http://jmlr.org/papers/v26/23-0624.html.

[14] Christine W Bang and Vanessa Didelez. Do we become wiser with time? on causal equivalence with tiered background knowledge. In *Uncertainty in Artificial Intelligence*, pages 119–129. PMLR, 2023.

[15] Bryan Andrews, Peter Spirtes, and Gregory F. Cooper. On the completeness of causal discovery in the presence of latent confounding with tiered background knowledge. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 4002–4011. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/andrews20a.html.

[16] Sara Magliacane, Tom Claassen, and Joris M Mooij. Ancestral causal inference. *Advances in Neural Information Processing Systems*, 29, 2016.

[17] Kristjan Greenewald, Dmitriy Katz, Karthikeyan Shanmugam, Sara Magliacane, Murat Kocaoglu, Enric Boix Adsera, and Guy Bresler. Sample efficient active learning of causal trees. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5ee5605917626676f6a285fa4c10f7b0-Paper.pdf.

[18] Guangya Wan, Yunsheng Lu, Yuqi Wu, Mengxuan Hu, and Sheng Li. Large language models for causal discovery: Current landscape and future directions. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pages 10687–10695. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/1186. URL https://doi.org/10.24963/ijcai.2025/1186. Survey Track.

[19] Tao Feng, Lizhen Qu, Niket Tandon, Zhuang Li, Xiaoxi Kang, and Gholamreza Haffari. On the reliability of large language models for causal discovery. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9565–9590, 2025.

[20] Wei Wang, Gangqiang Hu, Bo Yuan, Shandong Ye, Chao Chen, Yayun Cui, Xi Zhang, and Liting Qian. Prior-knowledge-driven local causal structure learning and its application on causal discovery between type 2 diabetes and bone mineral density. *IEEE Access*, 8:108798–108810, 2020. doi: 10.1109/ACCESS.2020.2994936.

[21] Xinpeng Shen, Sisi Ma, Prashanthi Vemuri, and Gyorgy Simon. Challenges and opportunities with causal discovery algorithms: application to alzheimer's pathophysiology. *Scientific reports*, 10(1):2975, 2020.

[22] Ronja Foraita, Janine Witte, Claudia Börnhorst, Wencke Gwozdz, Valeria Pala, Lauren Lissner, Fabio Lauria, Lucia A. Reisch, Dénes Molnár, Stefaan De Henauw, Luis Moreno, Toomas Veidebaum, Michael Tornaritis, Iris Pigeot, and Vanessa Didelez. A longitudinal causal graph analysis investigating modifiable risk factors and obesity in a european cohort of children and adolescents. *Scientific Reports*, 14(1):6822, 2024. doi: 10.1038/s41598-024-56721-y. URL https://doi.org/10.1038/s41598-024-56721-y.

[23] Qingyuan Zheng, Yue Liu, and Yangbo He. Local Causal Discovery with Background Knowledge . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pages 1–12, February 2026. ISSN 1939-3539. doi: 10.1109/TPAMI.2026.3667409. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2026.3667409.

[24] Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.

[25] Markus Kalisch, Martin Mächler, Diego Colombo, Marloes H. Maathuis, and Peter Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, 47(11):1–26, 2012. doi: 10.18637/jss.v047.i11.

[26] Yujia Zheng, Biwei Huang, Wei Chen, Joseph Ramsey, Mingming Gong, Ruichu Cai, Shohei Shimizu, Peter Spirtes, and Kun Zhang. Causal-learn: Causal discovery in python. *Journal of Machine Learning Research*, 25(60):1–8, 2024.

[27] Joseph D Ramsey, Kun Zhang, Madelyn Glymour, Ruben Sanchez Romero, Biwei Huang, Imme Ebert-Uphoff, Savini Samarasinghe, Elizabeth A Barnes, and Clark Glymour. Tetrad—a toolbox for causal discovery. *8th international workshop on climate informatics*, 2018.

[28] Antti Hyttinen, Frederick Eberhardt, and Matti Järvisalo. Constraint-based causal discovery: Conflict resolution with answer set programming. In Jin Tian and {Nevin L.} Zhang, editors, *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence (UAI 2014)*, pages 340–349, International, 2014. AUAI Press. Conference on Uncertainty in Artificial Intelligence ; Conference date: 01-01-1800.

[29] Sofia Triantafillou and Ioannis Tsamardinos. Constraint-based causal discovery from multiple interventions over overlapping variable sets. *Journal of Machine Learning Research*, 16(66): 2147–2205, 2015. URL `http://jmlr.org/papers/v16/triantafillou15a.html`.

[30] Joris M. Mooij, Sara Magliacane, and Tom Claassen. Joint causal inference from multiple contexts. *Journal of Machine Learning Research*, 21(99):1–108, 2020. URL `http://jmlr.org/papers/v21/17-123.html`.

[31] Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, page 403–410, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc. ISBN 1558603859.

[32] Ankur Ankan and Johannes Textor. pgmpy: A python toolkit for bayesian networks. *Journal of Machine Learning Research*, 25(265):1–8, 2024. URL `http://jmlr.org/papers/v25/23-0487.html`.

[33] Diego Colombo, Marloes H Maathuis, Markus Kalisch, and Thomas S Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.

[34] Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. *Advances in neural information processing systems*, 12, 1999.

[35] Jean-Philippe Pellet and André Elisseeff. Using markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9(7), 2008.

[36] Christine W Bang, Janine Witte, Ronja Foraita, and Vanessa Didelez. Improving finite sample performance of causal discovery by exploiting temporal structure. *arXiv preprint arXiv:2406.19503*, 2024.

[37] Christine W Bang and Vanessa Didelez. Constraint-based causal discovery with tiered background knowledge and latent variables in single or overlapping datasets. *Proceedings of Machine Learning Research*, 275:1–31, 2025.

[38] Tobias Ellegaard Larsen, Claus Thorn Ekstrøm, and Anne Helby Petersen. Score-based causal discovery with temporal background information. *arXiv preprint arXiv:2502.06232*, 2025.

[39] Tian-Zuo Wang, Tian Qin, and Zhi-Hua Zhou. Sound and complete causal identification with latent variables given local background knowledge. *Artificial Intelligence*, 322:103964, 2023. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2023.103964. URL `https://www.sciencedirect.com/science/article/pii/S0004370223001108`.

[40] Aparajithan Venkateswaran and Emilija Perković. Towards complete causal explanation with expert knowledge. *arXiv preprint arXiv:2407.07338*, 2024.

[41] Tian-Zuo Wang, Wen-Bo Du, and Zhi-Hua Zhou. Polynomial-delay MAG listing with novel locally complete orientation rules. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=70voOlSPos.

[42] Tom Claassen and Tom Heskes. A logical characterization of constraint-based causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 135–144, 2011.

[43] Christophe Gonzales, Axel Journe, and Ahmed Mabrouk. A hybrid algorithm for learning causal networks using uncertain experts' knowledge. In Antonio Salmerón and Rafael Rumí, editors, *Proceedings of The 11th International Conference on Probabilistic Graphical Models*, volume 186 of *Proceedings of Machine Learning Research*, pages 241–252. PMLR, 05–07 Oct 2022. URL https://proceedings.mlr.press/v186/gonzales22a.html.

[44] Lyuzhou Chen, Taiyu Ban, Xiangyu Wang, Derui Lyu, and Huanhuan Chen. Mitigating prior errors in causal structure learning: A resilient approach via bayesian networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[45] Stephanie Long, Alexandre Piché, Valentina Zantedeschi, Tibor Schuster, and Alexandre Drouin. Causal discovery with language models as imperfect experts. In *ICML 2023 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 2023.

[46] Masayuki Takayama, Tadahisa Okuda, Thong Pham, Tatsuyoshi Ikenoue, Shingo Fukuma, Shohei Shimizu, and Akiyoshi Sannai. Integrating large language models in causal discovery: A statistical causal approach. *Transactions on Machine Learning Research*, 2025.

[47] Shiv Kampani, David Hidary, Constantijn van der Poel, Martin Ganahl, and Brenda Miao. Llm-initialized differentiable causal discovery. In *Causality and Large Models@ NeurIPS 2024*, 2024.

[48] Xiangyu Wang, Taiyu Ban, Lyuzhou Chen, Derui Lyu, Qinrui Zhu, and Huanhuan Chen. Large-scale hierarchical causal discovery via weak prior knowledge. *IEEE Transactions on Knowledge and Data Engineering*, 37(5):2695–2711, 2025. doi: 10.1109/TKDE.2025.3537832.

[49] Sujai Hiremath, Dominik Janzing, Philipp Faller, Patrick Blöbaum, Elke Kirschbaum, Shiva Prasad Kasiviswanathan, and Kyra Gan. From guess2graph: When and how can unreliable experts safely boost causal discovery in finite samples? *arXiv preprint arXiv:2510.14488*, 2025.

[50] Adiba Ejaz and Elias Bareinboim. Less greedy equivalence search. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 123068–123109. Curran Associates, Inc., 2025. doi: 10.52202/085713-4104. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b26dfe705b549365ccbdbfdec74b1117-Paper-Conference.pdf.

[51] Paula Gradu, Tijana Zrnic, Yixin Wang, and Michael I. Jordan. Valid inference after causal discovery. *Journal of the American Statistical Association*, 120(550):1127–1138, 2025. URL https://EconPapers.repec.org/RePEc:taf:jnlasa:v:120:y:2025:i:550:p:1127-1138.

[52] Marco Scutari. Learning bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010. doi: 10.18637/jss.v035.i03.

[53] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

[54] F Richard Guo, Emilija Perković, and Andrea Rotnitzky. Variable elimination, graph reduction and the efficient g-formula. *Biometrika*, 110(3):739–761, 2023.

[55] Janine Witte, Leonard Henckel, Marloes H. Maathuis, and Vanessa Didelez. On efficient adjustment in causal graphs. *Journal of Machine Learning Research*, 21(246):1–45, 2020. URL http://jmlr.org/papers/v21/20-175.html.

[56] Zhuangyan Fang, Yue Liu, Zhi Geng, Shengyu Zhu, and Yangbo He. A local method for identifying causal relations under markov equivalence. *Artificial Intelligence*, 305:103669, 2022. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2022.103669. URL https://www.sciencedirect.com/science/article/pii/S0004370222000091.

[57] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[58] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[59] Diego Colombo and Marloes H Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, 15(1):3741–3782, 2014.

A Markov blanket discovery with Background Knowledge

Markov blanket discovery algorithms input a target variable T and return the set of nodes in their Markov blanket, denoted as $MB(T)$. We can use background knowledge to initialize the MB of T in the beginning of MB discovery algorithms with all variables adjacent to T according to $\mathcal{B}$, as well as known spouses, i.e. all variables V such that $T \to V'$ and $V \to V'$ are in $\mathcal{B}$, as described in Alg. 9. This allows the algorithms to skip testing whether these variables are in the MB of T. We implement this for the Grow-Shrink [34] algorithm in Alg. 10 and Total Conditioning [35] in Alg. 11.

Algorithm 9 PossMB($\mathbf{V}, T, \mathcal{B}$)

Require: Variables $\mathbf{V}$, target $T \in \mathbf{V}$, background knowledge $\mathcal{B}$
1: $PossMB(T) \leftarrow \{V \in \mathbf{V} | V - T \in \mathcal{B}\}$
2: $PossMB(T) \leftarrow PossMB(T) \cup \{V \in \mathbf{V} | V \to T \in \mathcal{B}\}$
3: $PossMB(T) \leftarrow PossMB(T) \cup \{V \in \mathbf{V} | V \leftarrow T \in \mathcal{B}\}$
4: $PossMB(T) \leftarrow PossMB(T) \cup \{V \in \mathbf{V} | \exists V' \in \mathbf{V} : T \to V' \in \mathcal{B} \wedge V' \leftarrow V \in \mathcal{B}\}$
Ensure: Initial Markov blanket of $T : MB(T)$ given $\mathcal{B}$

Algorithm 10 Grow-Shrink-BK

Require: Variables $\mathbf{V}$, target $T \in \mathbf{V}$, background knowledge $\mathcal{B}$
1: MB(T) $\leftarrow$ PossMB($\mathbf{V}, T, \mathcal{B}$)
2: **while** $\exists V \in \mathbf{V} \setminus (\{T\} \cup MB(T)) : T \not\perp\!\!\!\perp V | MB(T)$ **do**
3: $\quad MB(T) \leftarrow MB(T) \cup \{V\}$
4: **while** $\exists V \in MB(T) : T \perp\!\!\!\perp V | MB(T)$ **do**
5: $\quad MB(T) \leftarrow MB(T) \setminus \{V\}$
Ensure: Markov blanket of $T : MB(T)$

Theorem A.1. *Given oracle CI tests and consistent background knowledge $\mathcal{B}$, Alg. 10 is sound and complete in finding the true Markov blanket of a target T.*

Proof. Background knowledge is utilized in Alg. 9, where variables known by background knowledge to be in the Markov blanket of T by definition (neighbors and spouses of T) are added to $MB(T)$. These variables would also be added to $MB(T)$ by the original Grow-Shrink algorithm (Fig. 2 of [34]) starting at line 2, as they are always dependent on T. Since the Grow-Shrink algorithm does not specify an order for considering variables in the grow phase, we can assume an ordering where it starts with exactly the variables that are added to $MB(T)$ in the first four lines of Alg. 10. Then from this point on, the execution of the two algorithms are equivalent. Then, the soundness and completeness of Alg. 10 is given by the soundness and completeness of the original Grow-Shrink algorithm as shown by Margaritis and Thrun [34]. □

Algorithm 11 Total Conditioning-BK

Require: Variables $\mathbf{V}$, target $T \in \mathbf{V}$, background knowledge $\mathcal{B}$
1: MB(T) $\leftarrow$ PossMB($\mathbf{V}, T, \mathcal{B}$)
2: **for** $V \in \mathbf{V} \setminus (\{T\} \cup MB(T))$ **do**
3: $\quad$ **if** $T \not\perp\!\!\!\perp V | \mathbf{V} \setminus \{T, V\}$ **then**
4: $\quad\quad MB(T) \leftarrow MB(T) \cup \{V\}$
Ensure: Markov blanket of $T : MB(T)$

Theorem A.2. *Given oracle CI tests and consistent background knowledge $\mathcal{B}$, Alg. 11 is sound and complete in finding the true Markov blanket of a target T.*

Proof. Background knowledge is utilized in Alg. 9, where variables known by background knowledge to be in the Markov blanket of T by definition (neighbors and spouses of T) are added to $MB(T)$. The rest of the algorithm adds all variables $V \in \mathbf{V}$ to $MB(T)$ if and only if $T \not\perp\!\!\!\perp V | \mathbf{V} \setminus \{T, V\}$, which is a sound and complete criterion as shown by Property 7 of [35]. □

B Additional Algorithms

Alg. 12 implements the orientation rules of Meek [31].

Algorithm 12 Meek [31]

Require: Partially directed acyclic graph $\hat{G}$
repeat
  if $Z \to X - Y$ in $\hat{G}$ and $Z \notin Adj_{\hat{G}}(Y)$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
  if $X \to Z \to Y$, and $X - Y$ in $\hat{G}$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
  if $V \to Y \leftarrow Z, V - X - Z$ and $X - Y$ in $\hat{G}$, and $V \notin Adj_{\hat{G}}(Z)$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
  if $X - Y, X - V \to Z$ and $X - Z \to Y$ in $\hat{G}$, and $V \notin Adj_{\hat{G}}(Y)$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
until no edges can be oriented
Ensure: Completed partially directed acyclic graph $\hat{G}$

Alg. 13 shows the OrientUndirectedEdges procedure from the original MB-by-MB algorithm in [9], with the addition of the fourth Meek rule.

Algorithm 13 OrientUndirectedEdges [9]

Require: Partially directed acyclic graph $\hat{G}$
repeat
  if $Z \to X - Y$ in $\hat{G}$ and $\exists\ sepset(Z, Y)$ s.t. $X \in sepset(Z, Y)$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
  if $X \to Z \to Y$ and $X - Y$ in $\hat{G}$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
  if $X - Y$ and $X - V \to Z$ and $X - Z \to Y$ in $\hat{G}$ and $\exists\ sepset(X, Y)$ s.t. $V \in sepset(X, Y)$ **then**
    Orient $V - Z$ as $V \to Z$ in $\hat{G}$
  if $X - Y$ and $X - V \to Z$ and $X - Z \to Y$ in $\hat{G}$, and $V \notin Adj_{\hat{G}}(Y)$ **then**
    Orient $X - Y$ as $X \to Y$ in $\hat{G}$
until no edges can be oriented
Ensure: Completed partially directed acyclic graph $\hat{G}$

Algorithms 14 and 15 correspond to orienting v-structures as done in the PC algorithm [24] and the RFCI algorithm [33] algorithms. Both algorithms use the meta-symbol $*$ as a wildcard to denote both edge tails and arrowheads. This allows edges to be oriented as bi-directed.

Algorithm 14 VstructPC: Orienting v-structures in the PC algorithm [24]

Require: Skeleton $\hat{U}$, separating sets $sepset$
$\hat{G} \leftarrow \hat{U}$
for $X - Z - Y$ in $\hat{G}$ such that $X \notin Adj_{\hat{G}}(Y)$ **do**
  if $Z \notin sepset(X, Y)$ **then**
    Orient $X \ast\!\ast\ Z \ast\!\ast\ Y$ as $X \ast\!\!\to Z \leftarrow\!\!\ast Y$ in $\hat{G}$
Ensure: Partially oriented DAG $\hat{G}$

We implement LocalPC-BK in Alg. 16 according to Sec A.1.1 in [11] by conditioning on all subsets of the Markov blanket when trying to separate variables.

C Proofs

C.1 Proof for Thm. 3.1

Theorem 3.1. *Given oracle CI tests and consistent BK, PC-BK outputs the true MPDAG.*

Algorithm 15 VstructRFCI: Orienting v-structures in the RFCI algorithm [33]

Require: Skeleton $\hat{U}$, separating sets $sepset$

$M \leftarrow \{(X, Z, Y)$ such that $X - Z - Y$ in $\hat{\mathbf{E}}$ and $X \notin Adj_{\hat{U}}(Y)\}$
$L \leftarrow \{\}$
repeat
  $X, Z, Y \leftarrow$ choose an unshielded triple from M
  if $Z \notin sepset(X, Y)$ **then**
    if $X \not\perp\!\!\!\perp Z | sepset(X, Y)$ and $Z \not\perp\!\!\!\perp Y | sepset(X, Y)$ **then**
      $L \leftarrow L \cup \{(X, Y, Z)\}$
    else
      for $V \in \{X, Y\}$ **do**
        if $V \perp\!\!\!\perp Z | sepset(X, Y)$ **then**
          $\mathbf{S} \leftarrow sepset(X, Y)$
          done $\leftarrow$ False
          while not done **do**
            done $\leftarrow$ True
            for $S \in \mathbf{S}$ **do**
              if $V \perp\!\!\!\perp Z | \mathbf{S} \setminus \{S\}$ **then**
                $\mathbf{S} \leftarrow \mathbf{S} \setminus \{S\}$
                done $\leftarrow$ False
                break
          $sepset(V, Z) \leftarrow sepset(Z, V) \leftarrow \mathbf{S}$
          $M \leftarrow M \cup$ all triangles $(\min(V, Z), \cdot, \max(V, Z))$ in $\hat{U}$
          $M \leftarrow M \setminus$ all triples in M of the form $(V, Z, \cdot), (Z, V, \cdot), (\cdot, V, Z)$ and $(\cdot, Z, V)$
          $L \leftarrow L \setminus$ all triples in L of the form $(V, Z, \cdot), (Z, V, \cdot), (\cdot, V, Z)$ and $(\cdot, Z, V)$
          Delete the edge $V - Z$ from $\hat{U}$
until M is empty
$\hat{G} \leftarrow \hat{U}$
for $(X, Z, Y) \in L$ **do**
  Orient $X \ast\!\!\ast Z \ast\!\!\ast Y$ as $X \ast\!\!\rightarrow Z \leftarrow\!\!\ast Y$ in $\hat{G}$

Ensure: Partially oriented DAG $\hat{G}$ and separating sets $sepset$

Algorithm 16 LocalPC-BK

Require: Target T, Markov blanket $MB(T)$ and background knowledge $\mathcal{B}$

$Ne(T) \leftarrow MB(T)$
$sepset \leftarrow \emptyset$
for $i \in 0, \ldots, |MB(T)|$: **do**
  for $V \in Ne(X)$ s.t. $V \ast\!\!\rightarrow \notin \mathcal{B}$ **do**
    for $\mathbf{S} \subseteq MB(X) \setminus \{V\}$ s.t. $|\mathbf{S}| = i$ **do**
      if $V \perp\!\!\!\perp T | \mathbf{S}$ **then**
        $Ne(T) \leftarrow Ne(T) \setminus \{V\}$
        $sepset(\{T, V\}) \leftarrow \mathbf{S}$

Ensure: Neighbors $Ne(T)$ and separating sets $sepset$

Proof. Given a true DAG D and consistent background knowledge $\mathcal{B}$, let G^* be the true MPDAG and let $\hat{G}$ be the graph discovered by Alg. 3 using oracle CI tests. We now prove that $\hat{G}$ is identical to G^*.

We first prove that the skeleton (i.e., adjacencies) of $\hat{G}$ is identical to the skeleton of G^*. The adjacencies of $\hat{G}$ are established in two consecutive procedures, using Alg. 1 at line 4 and Alg. 2 at line 6. Alg. 1 is the same as the skeleton search of the standard PC algorithm with two differences:

1. At line 1, all CI tests for pairs of $(X, Y) \in \mathcal{B}$ are skipped, and thus kept adjacent.
2. At line 2, the candidate sets to separate X from Y are the $PossPa_{\hat{G}}(X, \mathcal{B}) = Adj_{\hat{G}}(X) \setminus (\{V \in \mathbf{V} | X \to V \in \mathcal{B} \vee X \neq V \in \mathcal{B}\})$.

The first difference impacts all pairs $(X, Y) \in \mathcal{B}$. If $X - Y \in \mathcal{B}$ or $X \to Y \in \mathcal{B}$, then X and Y are adjacent in G^*, thus leaving them adjacent without any CI tests keeps their adjacency correct. The gaps $X \neq Y \in \mathcal{B}$ are made non-adjacent in $\hat{G}$ later on by Alg. 2 at line 6.

The second difference impacts all other pairs $(X, Y) \notin \mathcal{B}$. In particular, this filters from the candidate sets to separate X from some other variable Y the known children of X or variables known to be non-adjacent to X. As shown by Lemma 3.3.9 in [24], if X and Y are not adjacent, then they can be d-separated by the parents of X or Y in the underlying causal graph. Since we usually do not know the true parents during the skeleton search, in standard PC we test subsets of the currently adjacent nodes, which will be a superset of the true parents. In our setting, we assume that the background knowledge $\mathcal{B}$ is consistent, so this allows us to filter these nodes and reduce the number of separating sets to be tested. Thus, if $(X, Y) \notin \mathcal{B}$, and X and Y are still adjacent in $\hat{G}$ by line 4 then X and Y are also adjacent in G^*.

This means that by line 4, the adjacencies of $\hat{G}$ and G^* are the same except for $X \neq Y \in \mathcal{B}$ which are still adjacent in $\hat{G}$ and not adjacent in G^*. These superfluous adjacencies are taken care of by Alg. 2 at line 6, which finds a corresponding separating set for each of them. Similarly to Alg. 1, the candidate separating sets do not involve known children or non-adjacencies, which is correct for the same reasoning as described earlier. Thus, this procedure finds a separating set for every gap $X \neq Y \in \mathcal{B}$, which are then also made non-adjacent in $\hat{G}$.

In case Alg. 2 does not find a corresponding separating set for a known gap $X \neq Y \in \mathcal{B}$, either due to incorrect CI test results or inconsistent background knowledge, then it simply ignores the known gap and keeps the edge $X - Y$ in $\hat{G}$. This ensures that all non-adjacencies have a corresponding separating set, which is crucial for correct v-structure orientation. However, this situation should never arise given oracle CI tests and consistent background knowledge.

This means that by line 6, the skeleton of $\hat{G}$ and G^* and all non-adjacencies have a corresponding separating set. At line 7, we orient v-structures the same way as standard PC. Since we found a separating corresponding for every non-adjacency, $sepset(\{X, Y\})$ is always well-defined. Then, the correctness of orienting v-structures is given by Lemmas 5.1.2 and 5.1.3 in [24].

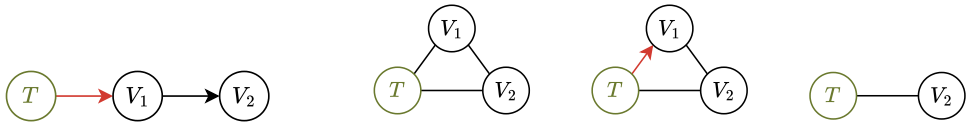
Finally, we maximally orient $\hat{G}$ using Alg. 12, which is the same procedure as described by Algorithm 1 in [31] and [12], except that we do not check for cases when $\mathcal{B}$ and $\hat{G}$ disagree, since this can never happen assuming oracle CI tests and consistent background knowledge $\mathcal{B}$.

Thus, the final output $\hat{G}$ is identical to the true MPDAG G^*. □

C.2 Proof for Thm. 3.2 and Cor. 3.1

A crucial step of SNAP(k)-BK (Alg. 4) is pruning variables at every iteration. We prune variables at line 13 that have no possibly directed paths to the targets $\mathbf{T}$, i.e., we keep their possible ancestors. This step is identical to the original algorithm. We show why it is not enough to keep only the b-possible ancestors of the targets at lower orders in Ex. C.1.

Example C.1. *Consider the DAG in Fig. 4a with target T and background knowledge $\mathcal{B} = \{T \to V_1\}$. The corresponding true MPDAG is identical to the DAG, and the induced subgraph of the MPDAG over the b-possible ancestors of T is the graph with the single node of T. However, applying Algorithm 1 of [12] at every iteration of SNAP would yield an incorrect MPDAG as follows.*



(a) DAG with $\mathcal{B} = \{T \to V_1\}$. (b) Skeleton. (c) Oriented graph. (d) Pruned graph.

Figure 4: Fig. 4a: Example DAG with target T where utilizing background knowledge $\mathcal{B}$, denoted in red, and naively pruning variables that are not b-possible ancestors of T can lead to incorrect results in SNAP(k)-BK. Fig. 4b: The skeleton discovered at order 0. Fig. 4c: The graph after orienting v-structures and orienting edges according to $\mathcal{B}$. Fig. 4d: The graph after pruning the variables that are not b-possible ancestors of T.

1. *The skeleton step at order 0 produces the graph shown in Fig. 4b, since all variables are marginally dependent.*
2. *The graph after orienting v-structures (none) and Algorithm 1 of [12] is shown in Fig. 4c.*
3. *In this graph, only V_2 is a b-possible ancestor of T, hence V_1 is pruned, resulting in the graph shown in Fig. 4d.*
4. *At this point, T and V_2 can never be separated, resulting in an **incorrect** final MPDAG shown in Fig. 4d.*

Ex. C.1 shows why keeping only b-possible ancestors of the targets is not enough at lower orders. Instead, similarly to possibly ancestral sets in [8], we define b-possibly ancestral sets as $\mathbf{V}^* \subseteq \mathbf{V}$ such that b-PossAn$_G(\mathbf{V}^*) \subseteq \mathbf{V}^*$, and prune variables that are not in this set. Lem. C.1 shows that the b-possibly ancestral set is exactly the same as the ancestral set, and thus the pruning step at line 13 should remain the same.

Lemma C.1. *Let G be a full MPDAG over variables $\mathbf{V}$ and Let $\mathbf{V}^* \subseteq \mathbf{V}$ be a b-possibly ancestral set of nodes, i.e. b-PossAn$_G(\mathbf{V}^*) \subseteq \mathbf{V}^*$. If $V \in \mathbf{V}^*$ then all nodes with a possibly directed path to V is also in $\mathbf{V}^*$.*

Proof. Let $V^0 = V \in \mathbf{V}^*$. Since b-PossAn$_G(\mathbf{V}^*) \subseteq \mathbf{V}^*$, then b-PossAn$_G(V) \subseteq \mathbf{V}^*$. This means that all of its parents $V^1 \to V$ and siblings $V^1 - V$ are also elements of $\mathbf{V}^*$. Since $V^1 \in \mathbf{V}^*$, then all of its parents $V^2 \to V^1$ and siblings $V^2 - V^1$ are also elements of $\mathbf{V}^*$. We have to continue adding all elements to $\mathbf{V}^*$ like this until all nodes with possible directed paths $V^n \to\!* V^{n-1} \to\!* \cdots \to\!* V^1 \to\!* V^0$ are in $\mathbf{V}^*$. Hence, all nodes with possibly directed paths to V are elements of $\mathbf{V}^*$. □

Lem. C.2 shows that unlike b-possible ancestors, b-possibly ancestral sets form a sound and complete objective, i.e., the induced subgraph of a full MPDAG over a b-possibly ancestral set retains all edges and orientations. This is analogous to Lem. 3.1 in [8] for possibly ancestral sets and CPDAGs.

Lemma C.2. *Let G be a full MPDAG over variables $\mathbf{V}$ with background knowledge $\mathcal{B}$. Let $\mathbf{V}^* \subseteq \mathbf{V}$ be a b-possibly ancestral set of nodes, i.e. b-PossAn$_G(\mathbf{V}^*) \subseteq \mathbf{V}^*$. Let $G|_{\mathbf{V}^*}$ be the induced subgraph of G over $\mathbf{V}^*$ and let $G(\mathbf{V}^*)$ be the MPDAG over variables $\mathbf{V}^*$ with the same background knowledge $\mathcal{B}$. Then we have $G|_{\mathbf{V}^*} = G(\mathbf{V}^*)$.*

Proof. Let D be the true underlying causal DAG, and C the corresponding CPDAG of the MPDAG G. $\mathbf{V}^*$ is a b-possibly ancestral set of nodes, i.e. for every $V \in \mathbf{V}^*$ it holds that b-PossAn$_G(V) \subseteq \mathbf{V}^*$. By definition, An$_D(V) \subseteq$ b-PossAn$_G(V)$. Hence, $\mathbf{V}^*$ is an ancestral set in D. Then, Lauritzen [53] (Proposition 3.22) and Guo et al. [54] (Lemma D.1) show that if $C|_{\mathbf{V}^*}$ is the induced subgraph of C over $\mathbf{V}^*$ and $C(\mathbf{V}^*)$ is the CPDAG over variables $\mathbf{V}^*$, then it holds that $C|_{\mathbf{V}^*} = C(\mathbf{V}^*)$. The MPDAG corresponding to $C(\mathbf{V}^*)$ with background knowledge $\mathcal{B}$ (between pairs in $\mathbf{V}^*$) is exactly $G(\mathbf{V}^*)$.

Now we show that the MPDAG corresponding to $C|_{\mathbf{V}^*}$ with background knowledge $\mathcal{B}$, denoted as G', equals $G|_{\mathbf{V}^*}$.

Since G is a MPDAG corresponding to the CPDAG C and G' is a MPDAG corresponding to CPDAG $C|_{\mathbf{V}^*}$, the adjacencies in $G|_{\mathbf{V}^*}$ and G' have to be identical, and every edge oriented in $C|_{\mathbf{V}^*}$ is oriented the same in $G|_{\mathbf{V}^*}$ and G'. Thus, $G|_{\mathbf{V}^*}$ and G' can only differ in edge orientations due to background knowledge. Since $G|_{\mathbf{V}^*}$ and G' are closed under orientation by definition correspond to

the same background knowledge, $G|_{\mathbf{V}^*}$ and G' cannot have contradicting orientations, i.e., $X \to Y$ in one and $X \leftarrow Y$ in the other. Furthermore, since $G|_{\mathbf{V}^*}$ is the induced subgraph of the full MPDAG G over variables $V \supseteq \mathbf{V}^*$, and thus its edges can be oriented by at least as much or more background knowledge in $\mathcal{B}$, it also cannot happen that the edge is oriented in G' but not in $G|_{\mathbf{V}^*}$.

Hence, if there is a differing edge between $G|_{\mathbf{V}^*}$ and G', it has to be oriented in $G|_{\mathbf{V}^*}$ but not in G', due to some orientation background knowledge in $\mathcal{B}$ that is not in $\mathcal{B}$. This orientation background knowledge has to be between variables where at least one is not in $\mathbf{V}^*$, which then propagated through G via orientation rules shown in Alg. 12 and oriented an edge in $G|_{\mathbf{V}^*}$, but it could not appear in G'. In the following, we show that this cannot happen, because if an edge $X \to Y$ such that $X, Y \in \mathbf{V}^*$ is oriented via orientation rules, then all other variables appearing in the orientation also have to be in $\mathbf{V}^*$. Note, that Lem. C.1, if $X, Y \in \mathbf{V}^*$, then every node with a possibly directed path to X or Y is also in $\mathbf{V}^*$.

1. The first rule involves the structure $Z \to X - Y$. Then, because $Z \to X$, it follows that $Z \in \mathbf{V}^*$.
2. The second rule involves the structure $X \to Z \to Y$. Then, because $Z \to Y$, it follows that $Z \in \mathbf{V}^*$.
3. The third rule involves the structures $V \to Y \leftarrow Z, V - X - Z$ and $X - Y$. Then, due to $V \to Y \leftarrow Z$, it follows that $V, Z \in \mathbf{V}^*$.
4. The fourth rule involves the structures $X - Y, X - V \to Z$ and $X - Z \to Y$. Then, because $X - V$ and $Z \to Y$, it follows that $V, Z \in \mathbf{V}^*$.

Thus, no edge in $G|_{\mathbf{V}^*}$ could have been oriented by orientations between edges involving variables not in $\mathbf{V}^*$. This means that $G|_{\mathbf{V}^*} = G'$.

We have shown earlier that $C|_{\mathbf{V}^*} = C(\mathbf{V}^*)$, and that the MPDAG corresponding to $C(\mathbf{V}^*)$ with background knowledge $\mathcal{B}$ is $G(\mathbf{V}^*)$. If two CPDAGs are equal, then their corresponding MPDAG with the same background knowledge $\mathcal{B}$ is also equal. Thus, $G(\mathbf{V}^*) = G'$. Then, it follows that $G|_{\mathbf{V}^*} = G(\mathbf{V}^*)$. □

With the objective of recovering $G|_{\mathbf{V}^*}$ with possibly ancestral set $\mathbf{V}^*$ established, we use the following lemmas to prove Thm. 3.2. Lem. B.1. in [8] is directly transferrable to SNAP(k)-BK.

Lemma C.3 (Lemma B.1 in [8])**.** *Given oracle conditional independence tests, at any iteration $i = 0, .., k$ of Alg. 4, the undirected graph $\hat{U}$ is a supergraph of $U|_{\hat{\mathbf{V}}}$, the induced subgraph of the true skeleton U over $\hat{\mathbf{V}}$.*

Proof. In Alg. 4 we only remove edges in $\hat{U}$ between two nodes $X, Y \in \hat{\mathbf{V}}$ for which we can find a separating set. By faithfulness, we assume that these nodes are therefore also non-adjacent in the ground truth skeleton U. Hence, the resulting skeleton $\hat{U}$ over the variables $\hat{\mathbf{V}}$ is a supergraph of the induced subgraph of the true skeleton, denoted by $U|_{\hat{\mathbf{V}}}$. □

Next, we show that no variable from the possibly ancestral set containing the targets is wrongly eliminated at any order i, starting with v-structures. The only difference between SNAP(k) and SNAP(k)-BK for orienting v-structures is that SNAP(k)-BK already applies the RFCI v-structure orientation rules (Alg. 15) starting at iteration $i = 1$ instead of $i = 2$. Ex. C.2 shows how orienting v-structures with the PC v-structure orientation rules (Alg. 14) can lead to wrong results.



(a) DAG with $\mathcal{B} = \{V_1 \neq V_2, V_2 \neq V_3\}$. (b) DAG with $\mathcal{B} = \{V_1 \to V_2, V_2 \neq V_3\}$.

Figure 5: Example DAGs where utilizing $\mathcal{B}$, denoted in red, during skeleton search can lead to incorrect results in SNAP. In both cases, the skeleton step up to order 1 never finds $V_1 \perp\!\!\!\perp V_3 | V_2$ due to the utilization of background knowledge, keeping V_1 and V_3 adjacent. On the other hand, it may find $V_0 \perp\!\!\!\perp V_1 | V_2$ before $V_0 \perp\!\!\!\perp V_1 | V_3$. In that case, at order 1, SNAP would orient a v-structure $V_1 \to V_3 \leftarrow V_0$, which contradicts $V_3 \to V_0$ in the true DAGs.

Example C.2. *Consider the DAGs in Fig. 5 with background knowledge $\mathcal{B}$ denoted in red. A naive implementation of SNAP(k)-BK that uses the PC v-structure orientation rules at order $i = 1$ may conclude that V_3 is a non-ancestor of V_0 at, even though $V_3 \to V_0$ in both DAGs.*

The incorrect naive implementation would proceed as follows:

1. *At order 0, no variables can be made marginally independent.*
2. *At order 1, the skeleton step may find $V_0 \perp\!\!\!\perp V_1 | V_2$ before $V_0 \perp\!\!\!\perp V_1 | V_3$, resulting in $sepset(V_0, V_1) = \{V_2\}$.*
3. *$\mathcal{B}$ indicates that $V_2 \notin PossPa(V_1)$ and $V_2 \notin PossPa(V_3)$, and thus V_2 is not required to separate V_1 from V_3 and $V_1 \perp\!\!\!\perp V_3 | V_2$ is never tested, rendering V_1 and V_3 still adjacent after the skeleton step at order 1.*
4. *At order 1, the v-structure step orients $V_1 \to V_3 \leftarrow V_0$, since $V_3 \notin sepset(V_0, V_1) = \{V_2\}$.*
5. *At order 1, SNAP(k)-BK **incorrectly** concludes that V_3 is a definite non-ancestor of V_0.*

The reason why using the PC v-structure orientation rules at order $i = 1$ in SNAP(k)-BK can fail is because the case of order $i = 1$ in Lem. B.4. of [8] does not hold anymore.

We now show in Lem. C.4 that correctly orienting v-structures using PC v-structure orientation rules only at order $i = 0$ at line 8 and using RFCI v-structure orientation rules at line 10 otherwise, will never contradict ancestry in the true causal DAG D, i.e., they orient $X \to Z \leftarrow Y$ only if $Z \notin An_D(X)$ (and $Z \notin An_D(Y)$).

Lemma C.4 (Analogous to Lemmas B.2 and B.4 in [8])**.** *Given the true MPDAG G with background knowledge $\mathcal{B}$, at any iteration $i = 0, \dots, k$ of SNAP(k)-BK (Alg. 4), lines 8 and 10 orient $X *\!\!\to Z \leftarrow\!\!* Y$ only if $Z \notin$ b-PossAn$_G(X)$.*

Proof. Our proof follows the general structure of Lemmas B.2 and B.4 in [8]. We first show that at every iteration, $X *\!\!\to Z \leftarrow\!\!* Y$ is oriented only if the following three conditions hold: $Z \notin sepset(X, Y)$, $X \not\perp\!\!\!\perp Z | sepset(X, Y)$ and $Z \not\perp\!\!\!\perp Y | sepset(X, Y)$.

At iteration $i = 0$ SNAP(k)-BK orients v-structures using `VstructPC` (Alg. 14), which orients unshielded triples as $X \to Z \leftarrow Y$ if and only if $Z \notin sepset(X, Y) = \emptyset$. At this point, every pair of variables that may be separated (not known as adjacent by $\mathcal{B}$) has been tested for independence with the empty conditioning set during the skeleton step. Thus, if there is still an edge between X and Z and between Z and Y, then they are dependent given the empty set, or in other words, $X \not\perp\!\!\!\perp Z | \emptyset$ and $Y \not\perp\!\!\!\perp Z | \emptyset$ always hold.

At every iteration $i > 0$, SNAP(k)-BK orients v-structures using `VstructRFCI` (Alg. 15), which explicitly tests the three conditions of $Z \notin sepset(X, Y)$, $X \not\perp\!\!\!\perp Z | sepset(X, Y)$ and $Z \not\perp\!\!\!\perp Y | sepset(X, Y)$ at lines 5 and 6.

Then, the following proof by contradiction is directly reproduced from Lem B.2 in [8]. For the sake of contradiction, let us assume that there exists a DAG D in the MEC of G such that $Z \in An_D(X)$. Then, there is a directed path $Z \to \cdots \to X$ in D. Since $Z \not\perp\!\!\!\perp Y | sepset(X, Y)$, there exists at least one open path between Y and Z not blocked by $sepset(X, Y)$ in D. However, since $X \perp\!\!\!\perp Y | sepset(X, Y)$, there has to be a node $W \in sepset(X, Y)$ on the directed path from $Z \to \dots W \cdots \to X$ that blocks this directed path in D. Otherwise, the open path between Y and Z and the directed path from Z to X would not be blocked by $sepset(X, Y)$. However, we also know that $X \not\perp\!\!\!\perp Z | sepset(X, Y)$ implying that there is some other path between X and Z that is not blocked by $sepset(X, Y)$. Thus, there is a path between X and Z and between Z and Y that are not blocked by $sepset(X, Y)$ in D. In this case, $X \perp\!\!\!\perp Y | sepset(X, Y)$ can only hold if Z is a collider when connecting these two paths, because $Z \notin sepset(X, Y)$. However, this path is then unblocked by W since it is a descendant of Z and it is in $sepset(X, Y)$. It follows that $X \perp\!\!\!\perp Y | sepset(X, Y)$ cannot hold in D, where $Z \in An_D(X)$, hence we get a contradiction. Since no such DAG D in the MEC of G can exist where $Z \in An_D(X)$, this proves that $Z \notin$ b-PossAn$_G(X)$. □

Using the lemmas above, we can prove the main result of Thm. 3.2 and consequently Cor. 3.1.

Theorem 3.2. *Given oracle CI tests and consistent BK, at each step $i = 0, \dots, k$ of SNAP(k)-BK for the remaining variables $\hat{\mathbf{V}}$ that are output by SNAP, it holds that $\mathbf{T} \subseteq \hat{\mathbf{V}}$ and b-PossAn$_G(\hat{\mathbf{V}}) \subseteq \hat{\mathbf{V}}$.*

Proof. The proof is analogous to the proof of Lem B.2 in [8]. When pruning variables at line 13, all edges are oriented either by v-structures, which are shown to be sound by Lem. C.4, or by consistent orientation background knowledge $\mathcal{B}$. From this and Lem. C.3 it follows that if there is a possibly directed path in the true MPDAG G from some $V \in \hat{\mathbf{V}}$ to $V' \in \hat{\mathbf{V}}$, then a possibly directed path with the same sequence of nodes also exists in $\hat{G}$ and it is also possibly directed. Since $\hat{\mathbf{V}}$ is constructed by taking all nodes with a possibly directed path to $\mathbf{T}$, then by Lem. C.1 it holds that $\hat{\mathbf{V}}$ forms a b-possibly ancestral set that contains $\mathbf{T}$. □

Corollary 3.1. *Given true MPDAG G, oracle CI tests and consistent BK, let $\hat{G}$ be the output graph of SNAP(∞)-BK for targets $\mathbf{T}$. Then, SNAP(∞)-BK is sound and complete over the b-possible ancestors $\mathbf{T}$, i.e. $\hat{G} = G|_{\textit{b-PossAn}(\mathbf{T})}$, and $\hat{G}$ is an MPDAG.*

Proof. We show that the graph $\hat{G}$ by the end of SNAP(k)-BK running with $k = |\mathbf{V}| - 2$ has the same skeleton and v-structures as the induced sub-graph of the true MPDAG $G|_{\hat{\mathbf{V}}}$. This part of the proof is analogous to the proof of Lem. B.5 in [8]. Lem. C.3 shows that the discovered skeleton $\hat{U}$ is always a supergraph of the skeleton of $G|_{\hat{\mathbf{V}}}$. Furthermore, since SNAPk-BK only removes an edge if a corresponding separating set is found, then by faithfulness, every node is adjacent to its parents. Furthermore, since the background knowledge $\mathcal{B}$ is consistent, the parents of each node V are a subset of its possible parents $PossPa_{\hat{\mathbf{G}}}(V, \mathcal{B})$. Then, by line 13 with $k = |\mathbf{V}| - 2$, all subsets of the parents have been tried as conditioning sets when trying to separate a variable from another. Hence, by faithfulness, if two nodes are still adjacent in $\hat{G}$ then they are also adjacent in $G|_{\hat{\mathbf{V}}}$. From this and Lem. C.3 it follows that the skeleton of $\hat{G}$ and $G|_{\hat{\mathbf{V}}}$ are identical.

Then, their unshielded triples are also identical and by Lem. C.4 it follows that their v-structures are also identical.

Given $\hat{G}$ obtained from SNAP(k)-BK with $k = |\mathbf{V}| - 2$, the final step of the proof is analogous to the proof of Thm. 3.2 in [8]. If $\hat{G}$ and $G|_{\hat{\mathbf{V}}}$ have the same skeleton and v-structures, then they are Markov equivalent. Then, as shown by Perković et al. [12], applying consistent orientation background knowledge in $\mathcal{B}$ and orientation rules in Alg. 12 on $\hat{G}$ yields exactly the true MPDAG $G|_{\hat{\mathbf{V}}}$. The final output graph is obtained at line 3, where the final b-possibly ancestral set is obtained according to the $\hat{G} = G|_{\hat{\mathbf{V}}}$. By Lem. C.2, $\hat{G}$ is an MPDAG. □

C.3 Proof for Thm. 3.3

Before proving Thm. 3.3, we provide a detailed an example for why a naive application of PC-BK (Alg. 3) results in wrong local structures discovered in MB-by-MB-BK

Example C.3. *Consider the example DAG/MPDAG shown in Fig. 2a with background knowledge $\mathcal{B} = \{T \to V_2, V_2 \not\to X\}$ denoted in red. Naively using PC-BK when when discovering the local structure over the Markov blanket of T can incorrectly identify X as adjacent to T, as shown in Fig. 2b. In particular, the MB-by-MB-BK algorithm would then proceed as follows:*

1. *In the beginning of the algorithm WaitList = [T] and $\hat{G} = (\{T, X, V_1, V_2, V_3, V_4\}, \emptyset)$.*
2. *In the first iteration, at line 6, MB-by-MB-BK correctly correctly finds $MB(T) = \{X, V_1, V_2, V_3\}$*
3. *Then, at line 14, MB-by-MB proceeds to discover the local structure L_T over $MB^+(T)$ naively using PC-BK:*
   (a) *When trying to separate nodes from T, PC-BK does not find $T \perp\!\!\!\perp X | \{V_2\}$ because $V_2 \notin PossPa_{L_T}(T, \mathcal{B})$ due to $T \to V_2 \in \mathcal{B}$.*
   (b) *When trying to separate nodes from X, PC-BK also does not find $X \perp\!\!\!\perp T | \{V_2\}$ because $V_2 \notin PossPa_{L_T}(X, \mathcal{B})$ due to $X \not\to V_2 \in \mathcal{B}$.*
   (c) *The only other way to separate T and X would be $T \perp\!\!\!\perp X | \{V_4\}$, but $V_4 \notin MB^+(T)$ and thus it is not considered in the local structure.*
   (d) *For the same reason, $V_3 \perp\!\!\!\perp X | \{V_4\}$ is also not found.*
   (e) *At the end of the skeleton search by line 6, T and X (as well as V_3 and X) are still **incorrectly** adjacent.*
   (f) *At line 7, PC-BK orients the v-structure $T \to X \leftarrow V_3$ (along with $T \to V_1 \leftarrow V_3$),*

4. *MB-by-MB-BK updates $\hat{G}$ with edges and v-structures containing T in L_T at line 15 including $T \to X \leftarrow V_3$.*
5. *However, the edge between X and V_1 in L_T is discarded as it is not part of a v-structure, due to the extra edge between T and X.*
6. *MB-by-MB-BK applies background knowledge $T \to V_2$ at line 16, resulting in the graph shown in Fig. 2b*
7. *As all neighbors of T are oriented, MB-by-MB-BK terminates and **incorrectly** returns X as a child of T.*

Instead, if PC-BK uses the adjusted definition of possible parents defined in Equation (1)*, then the MB-by-MB-BK algorithm would proceed as follows:*

1. *In the beginning of the algorithm WaiList = [T] and $\hat{G} = (\{T, X, V_1, V_2, V_3, V_4\}, \emptyset)$.*
2. *In the first iteration, at line 6, MB-by-MB-BK correctly correctly finds $MB(T) = \{X, V_1, V_2, V_3\}$*
3. *Then, at line 14, MB-by-MB proceeds to discover the local structure L_T over $MB^+(T)$ using PC-BK:*
   (a) *When trying to separate nodes from T, PC-BK does not find $T \perp\!\!\!\perp X | \{V_2\}$ because $V_2 \notin PossPa^*_{\hat{G}}(T, \mathcal{B})$ due to $T \to V_2 \in \mathcal{B}$.*
   (b) *When trying to separate nodes from X at $i = 1$ of the skeleton search of PC-BK, at this point X and V_2 are adjacent, and so $V_2 \in PossPa^*_{\hat{G}}(X, \mathcal{B}) = Adj_{\hat{G}}(X)$. Thus PC-BK can **correctly** find $X \perp\!\!\!\perp T | \{V_2\}$.*
   (c) *$V_3 \perp\!\!\!\perp X | \{V_4\}$ is again not found, but in this case it is not an issue.*
   (d) *At the end of the skeleton search by line 6, T and X are **correctly** non-adjacent.*
   (e) *At line 7, PC-BK only orients v-structure $T \to V_1 \leftarrow V_3$.*
4. *MB-by-MB updates its global graph with edges and v-structures containing T at line 15, including $T \to V_1 \leftarrow X$.*
5. *MB-by-MB-BK applies background knowledge $T \to V_2$ at line 16, resulting in the graph shown in Fig. 2c.*
6. *As all neighbors of T are oriented, MB-by-MB-BK terminates and **correctly** returns X as non-adjacent of T.*

Theorem 3.3. *Given oracle CI tests and consistent BK, MB-by-MB-BK is sound and complete in identifying the parents, children and siblings of the target.*

Proof. Alg. 6 utilizes background knowledge in Markov blanket discovery, learning the local structure over the Markov blanket, and orienting edges. Markov blanket discovery is performed by GS-BK (Alg. 10) at line 6, which is sound and complete by Thm. A.1.

The local structure over Markov blankets is learned at line 14 using PC-BK (Alg. 3) with the updated definition of possible parents given in Equation (1). An example for why this adjustment is necessary is shown in Ex. C.3. In particular, the adjustment definition in Equation (1) make sure that the limiting of candidate separating sets in Alg. 1 and Alg. 2 is only applied for the variable X when discovering the local structure over $MB(X)$. Limiting the candidate separating for X is sound, because all of its neighbors are in its Markov blanket, including its possible parents. However, this might be not true for any other node in $MB(X)$, hence we do not limit their separating sets (as shown by Ex. C.3).

Hence both the Markov blankets as well as the adjacencies and v-structures learned in the local structures are identical to the ones MB-by-MB would learn without background knowledge.

Finally, orientation background knowledge is applied on the learned graph at line 16. This is the same step as line 3 of Algorithm 1 of [23], another approach for MB-by-MB with background knowledge. As the other steps of the two algorithms are the same the original MB-by-MB algorithm [9], Alg. 6 is equivalent to Algorithm 1 of [23] in terms of orientations. Thus, the soundness and completeness of Alg. 6 follows from Lem. 11 of [23]. □

C.4 Example of when LDECC-BK (Sec. 3.4) is not complete

The following example shows how LDECC-BK fails to identify a parent of a target even though it is oriented in the true MPDAG.

Example C.4. *Consider the true DAG $X \to Y \to T$ with BK $\mathcal{B} = \{X \to Y\}$. The MPDAG of this DAG equals to the true DAG, since after applying the orientation $X \to Y$ to the corresponding*

CPDAG $X - Y - T$, as $X \to Y - T$, the edge $Y - T$ is oriented by the first Meek rule into $Y \to T$, resulting in a fully oriented MPDAG $X \to Y \to T$.

Applying LDECC-BK in Alg. 7 on the example above with target T however cannot identify Y as a parent of T because it has no way of propagating orientation BK outside of the Markov blanket of T. In particular, the algorithm executes as follows.

1. *At line 1, LDECC-BK finds $MB(T) = \{Y\}$.*
2. *At line 2, LDECC-BK finds $Ne(T) = \{Y\}$.*
3. *At line 3, LDECC-BK identifies $Ch(T) = \{\}$.*
4. *The loop at line 7 does not find more independences due to $X \ast\!\!-\!\!\ast\, Y) \in \mathcal{B}$ and all other CI tests including target T.*
5. *Finally, LDECC-BK **incorrectly** returns Y as a sibling of T.*

Note, that LDECC-BK would also return Y as a sibling of T if it did not skip CI tests for X and Y. In that case, it would simply find $X \not\perp\!\!\!\perp Y|\emptyset$ and $X \not\perp\!\!\!\perp Y|\{T\}$, which do not allow it to orient any edges.

C.5 Proof for Thm. 3.4

To prove Thm. 3.4, we first adapt Def. D.1 and Lem 4.4 in [11] to MPDAGs.

Definition C.1 (Projection for MPDAGs with single treatment and outcome given $\mathbf{F}$, analogous to Def. D.1 in [11])**.** *Let G be a MPDAG with nodes $\mathbf{V}$ and background knowledge $\mathcal{B}$, and let $T, O \in \mathbf{V}$, and $\mathbf{F} \subset \mathbf{V} \setminus \{T, O\}$. A projection $\hat{G}^{T,O}$ of G is a graph with nodes $\mathbf{V} \setminus \mathbf{F}$ and edges as follows. For distinct nodes $W_i, W_j \in \mathbf{V} \setminus \mathbf{F}$,*

1. *$\hat{G}^{T,O}$ contains a directed edge $W_i \to W_j$ if and only if G contains a directed path $W_i \to \cdots \to W_j$ on which all non-endpoint nodes are in $\mathbf{F}$,*
2. *$\hat{G}^{T,O}$ contains a bi-directed edge $Wi \leftrightarrow Wj$ if and only if G contains a path, with at least one non-endpoint node, of the form $W_i \leftarrow \cdots \to W_j$ on which all non-endpoints are non-colliders and in $\mathbf{F}$,*
3. *$\hat{G}^{T,O}$ contains an undirected edge $W_i - W_j$ if and only if G contains $W_i - W_j$.*

The following lemma, analogous to Lem. 4.4 in [11], shows that the modified forbidden projection in LOAD-BK results in an MPDAG in which the optimal adjustment set are the parents of the outcome.

Lemma C.5 (Analogous to Lem. 4.4 in [11])**.** *For treatment T and outcome O in an amenable MPDAG G with background knowledge $\mathcal{B}$, let $\mathbf{D} = PossDe_G(T)) \setminus \{T, O\}$ and $G^{T,O}$ be the modified forbidden projection with nodes $\mathbf{V} \setminus \mathbf{D}$ and background knowledge on adjacencies and orientations over $\mathbf{V} \setminus \mathbf{D}$ in $\mathcal{B}$. We show that $G^{T,O}$ is a MPDAG and the optimal adjustment set $Oset_G(X, Y)$ is given by*

$$Oset_G(T, O) = Pa_{G^{T,O}}(O) \setminus \{T\}.$$

Proof. We first note, that the background knowledge on adjacencies and orientations over $\mathbf{V} \setminus \mathbf{D}$ in $\mathcal{B}$ is valid under the modified projection. This holds because the modified forbidden projection does not remove any adjacencies between nodes that are not projected out, hence their direct causal relations remain the same as in the original MPDAG G over all variables.

Then, the only difference between the modified forbidden projection of LOAD-BK and LOAD arises from orientation BK, which may orient some additional undirected neighbors of T and parents, thus reducing the set of projected variables $\mathbf{D}$.

Now, the same proof applies as for em. 4.4 in [11], which we also present here for completeness.

As shown in Prop. 22 by Witte et al. [55] projecting out the forbidden nodes $\mathbf{F} = PossDe_G(PossCn_G(T, O)) \setminus \{T, O\}$, where $PossCn_G(T, O)$ are the nodes on possibly directed paths from T to O, excluding T, from G results in an MPDAG $\tilde{G}^{T,O}$, where optimal adjustment set are the parents of O except T in $\tilde{G}^{T,O}$.

We can further project out the remaining variables in $\tilde{G}^{T,O}$, $\Delta = PossDe_G(T) \setminus PossDe_G(PossCn_G(T, O)) \cup \{T, O\}$. These will be still the remaining possible descendants of T in this new graph, i.e., $PossDe_{\tilde{G}^{T,O}}(T)$, which are definite non-ancestors of O and its possible ancestors, e.g., its parents, in both graphs, since otherwise they would be possible causal nodes in the

original graph G and removed in the first projection. As such, following Schubert et al. [8], they can be safely marginalized out in $G^{T,O}$ without changing any orientation in the graph regarding O, its possible ancestors, e.g., parents, i.e., without changing the parent set of O, and hence the optimal adjustment set.

To prove that $G^{T,O}$ is still a MPDAG, we first show that it does not contain bi-directed edges, and then show that we do not add any other edge to the remaining variables with respect to the forbidden projection $\tilde{G}^{T,O}$.

In order for a bi-directed edge to appear in $G^{T,O}$, there have to exist two nodes $W_i, W_j \notin \mathbf{D}$ such that $W_i \leftarrow \cdots \rightarrow W_j$ in G and all non-endpoint nodes on this path are non-colliders and in $\mathbf{D}$. This means that both W_i and W_j have to be possible descendants of a node in $\mathbf{D}$. By transitivity, possible descendants of nodes in $\mathbf{D}$ are also in $\mathbf{D}$, except for T and O, which are then the only possible candidates for W_i and W_j. On the other hand, there cannot exists such a path $W_i = T \leftarrow \cdots \rightarrow O = W_j$ in G such that all non-endpoint nodes on this path are possible descendants of T due to the definition of $\mathbf{D}$, since this would be a contradiction based on the orientation of the path.

We now show that the modified projection does not add any other directed or undirected edge compared to the original forbidden projection $\tilde{G}^{T,O}$. A fully undirected path or (possibly) directed path from T to other variables $V \in \mathcal{V} \setminus \{\Delta \cup O, T\}$ that passes through some variable $D \in \Delta$ would imply that V is also in Δ by definition of possible descendants, which is a contradiction. Moreover, it is not possible to have a directed path from V to T through some $D \in \Delta$, since this would contradict the fact that Δ are possible descendants of T.

We will therefore consider the case when there is a possibly directed path $p = (V, \ldots, D_1, \ldots, D_k, \ldots, T)$ in $\mathcal{G}$ for $V \in \mathcal{V} \setminus \{\Delta \cup \{O, T\}\}$ where all non-endpoint nodes $D_1, \ldots, D_k$ are in Δ. The subpath p' from $D_1 - D_2 \cdots - D_k - T$ has to undirected, since it cannot be possibly directed from any D_i to T by definition of Δ and it cannot be possibly directed in the other direction by assumption. Additionally, the subpath p'' from $V \ldots D_1$ cannot be completely undirected, otherwise V would be part of Δ. In order for p' to stay undirected, we need that the variable V_j on p'' closest to D_1 that has an orientation is part of a shielded triple $V_i \rightarrow V_j - D_1$ with additionally $V_i - D_1$ or $V_i \rightarrow D_1$, otherwise the orientation will propagate further due to Meek's rules [31], but we cannot use $V_i - D_1$, since this would mean that $V_i \in \Delta$, because it is connected by an undirected path to T, which is a contradiction. So the only possible shield is $V_i \rightarrow D_1$, but then since we have $V_i \rightarrow D_1 - D_2$ this would imply that there needs to be another shield $V_i \rightarrow D_2$, on which we can apply the same argument on each node D_i, until we would get a shield $V_i \rightarrow T$. This means that in the projection we would not need to add any edge from V_i to T. A simplified case is when we only have three variables $V \rightarrow D - T$, which would require $V - T$, which is again a contradiction, or $V \rightarrow T$ which would mean we do not need to add a new edge after the projection. The only possible path from T to other variables $V \in \mathcal{V} \setminus \{\Delta \cup O, T\}$ that passes through some variable $D \in \Delta$ would have D as a collider, which means we can safely remove this variable and still have a valid MPDAG, which means we do not need to add any edge to the original $\tilde{G}^{T,O}$. □

Theorem 3.4. *Given oracle CI tests and consistent BK, LOAD-BK is sound and complete in finding optimal adjustment set for a pair of target variables.*

Proof. Thm. 3.3 it follows that every step of LOAD-BK uses sound and complete information according to the true MPDAG. Then, the soundness and completeness of Steps 1 and 2 follows from Lem 4.1. and Cor 4.1 of [11], while Step 4 follows from Thm. 2 by [56]. Finally, Lem. C.5 shows that step 4 of LOAD-BK is also sound and complete. □

D Limitations and Extensions

In this work we focus on background knowledge about direct causal relationships between pairs of variables, and how it can be utilized during causal discovery. However, our approach is also compatible with any type of post-processing approach that is appropriate to the output structures, e.g. MPDAG for PC-BK/SNAP(∞)-BK or local structures for MB-by-MB-BK/LDECC-BK and LOAD-BK.

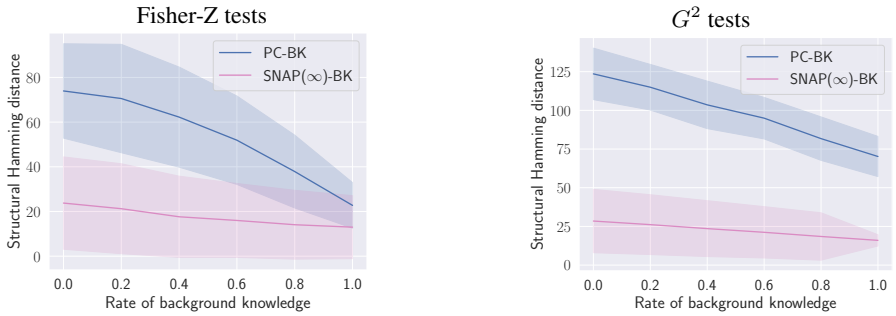


Figure 6: SHD for PC-BK and SNAP(∞)-BK over rate of BK between variable pairs. The shadow area denotes standard deviation.

D.1 Ancestral relations.

Other types of BK, such as ancestral relationships, often have implications in terms of direct relations that we can still exploit in our methods. For example, if X is an ancestor of Y, this obviously implies that Y is not a direct cause of X. Additionally, if we know that X and Y are adjacent, then in this case we can also infer that X is a direct cause of Y. Finally, if X and Y are not ancestors of each other then we know that there is a gap between them.

Utilizing ancestral BK in a more complete way during discovery is not trivial, as Fang et al. [13] showed that ancestral BK cannot even be fully represented by MPDAGs.

E Experimental Details

For our experiments, we used the following libraries: igraph [57] (GNU GPL version 2 or later), networkx [58] (3-Clause BSD License), bnlearn [52] (MIT License), causal-learn [26] (MIT License). In particular, we used the CI test implementations from the causal-learn package [26].

All experiments were performed on AMD Rome CPUs, using 48 CPU cores and 84 GiB of memory. We let each experiment run for at most 24 hours. If an experiment over 100 seeds did not finish in the given time or did not fit in the given memory, then we do not report any results for it.

F Results on Structural Hamming Distance

The various methods we extend recover different types of subgraphs with different sets of nodes or even just local structures, such as LOAD-BK and LDECC$^+$-BK. Because of this it would not be fair or even possible to compare all algorithms using structural metrics. Instead, we generally compare all methods based on the downstream task we are interested in, i.e., causal effect estimation for a set of target variables, which we measure through the intervention distance.

However, as PC-BK and SNAP(∞)-BK do recover MPDAGs, for completeness in this section we report their SHD with 100 nodes and otherwise under the same finite data settings as in Fig. 3. In Fig. 6 we measure the SHD compared to the true MPDAG for PC-BK, and compared to the induced subgraph of the true MPDAG over the b-possible ancestors of the targets in the true MPDAG for SNAP(∞)-BK, so these results are not directly comparable across the methods, but these results can be used to evaluate how the SHD changes based on the BK rate. As expected, our results show a clear trend of decreasing (improving) SHD with increasing rate of BK for both methods in both data settings.

G Comparison to Post-Processing

While we propose methods for integrating the BK as soon as possible, so either before or during the learning process, our work is compatible with other approaches for BK in postprocessing, such as [12] or [23]. So in principle one could use different sets of BK for our approaches and for postprocessing.

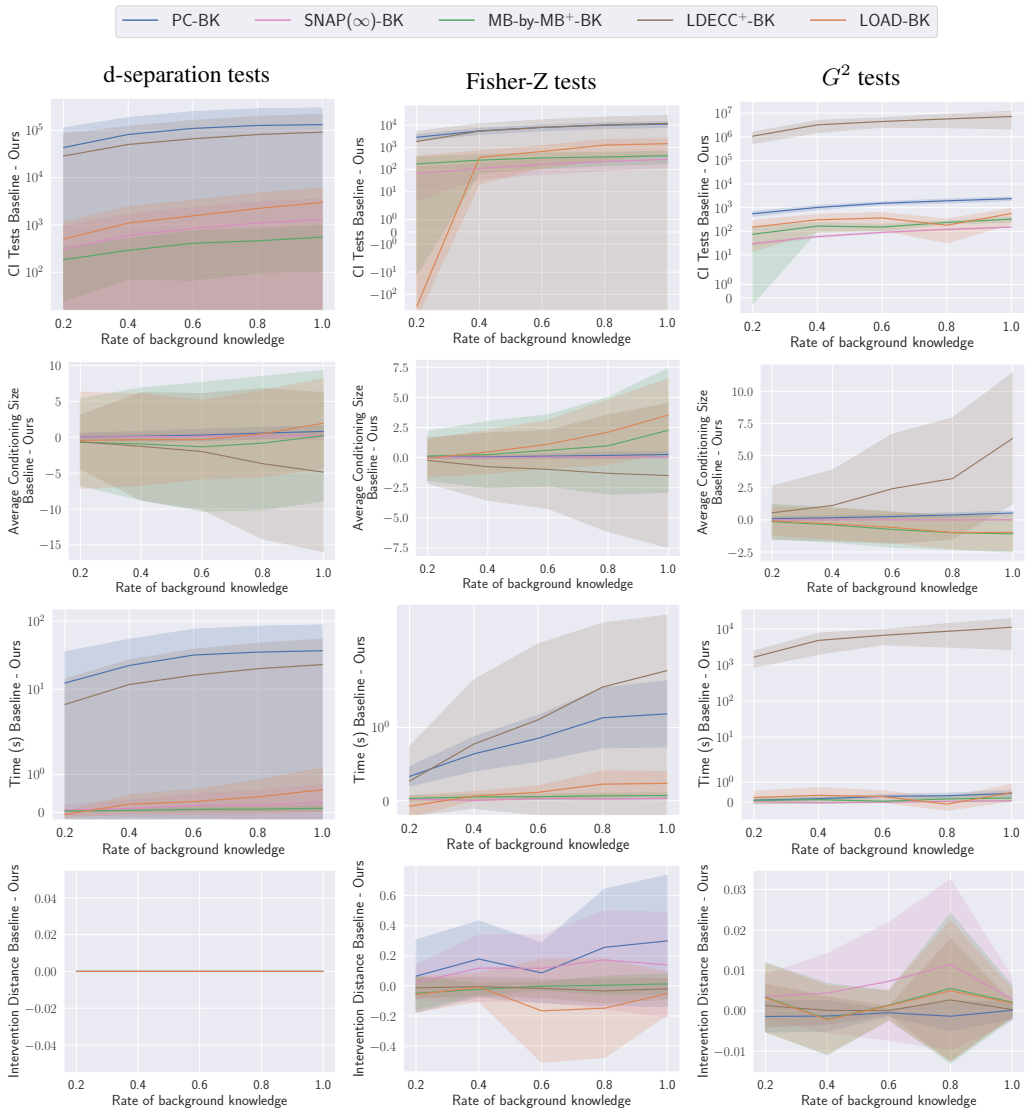


Figure 7: Results for showing the *improvement* of our approach compared to the post-processing baselines described in App. G, quantified as the difference between the baselines and our approach, in the same setting as Fig. 3 but on 100 nodes. The shadow area denotes standard deviation. Higher is better for each metric.

If we consider perfect BK and oracle d-separation tests, the outputs should be the same, as shown by our theoretical results. In terms of pre-processing, we already integrate as much as possible of the knowledge before starting the learning procedure and show that doing so naively (like in the current implementations described in App. H.1) might create errors.

In general, the computational efficiency of using BK in postprocessing would be very similar, if not in some cases identical, to the results shown in Fig. 3 for a BK rate of zero. For PC and SNAP(∞), integrating BK in postprocessing with the approach by Perković et al. [12] gives exactly the same CI tests as these results, since there are no CI tests in the postprocessing. The execution time is also dominated by the tests, so this is quite similar. For local methods like MB-by-MB and LOAD, we can follow [23] for postprocessing to orient the local structures, which also gives similar results. For LDECC, we can simply add or remove parents and children to its output based on the BK. While this strategy is sound, it may identify fewer children than our approach, as using orientations during discovery can orient additional children at line 19 of LDECC (Fig. 5 in [10]).

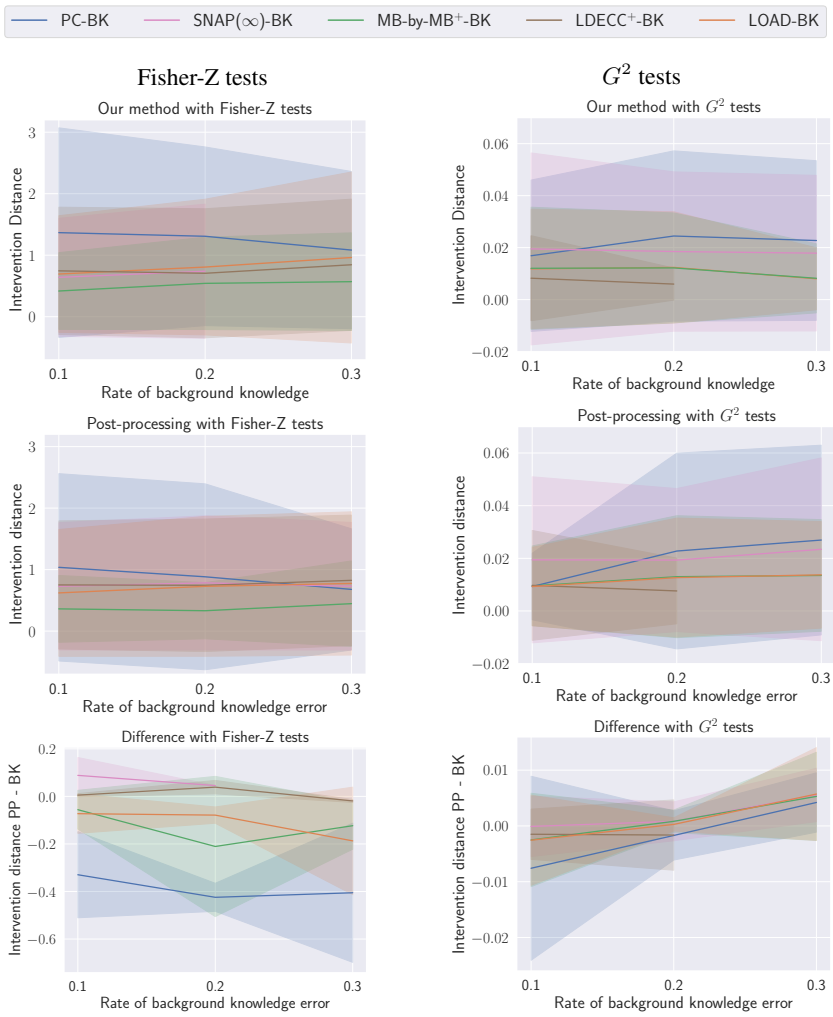


Figure 8: Results comparing our approach to post-processing baselines described in App. G under different amounts of background knowledge error in the same setting as Fig. 3 but on 100 nodes. The shadow area denotes standard deviation. Higher is better for each metric.

To make this point even clearer, we provide empirical results, showing the improvement of our methods in comparison with using BK in postprocessing. In Fig. 7 we report the *improvement* of our approach compared to the aforementioned baselines, quantified as the difference between the baselines and our approach, in the same setting as Fig. 3, where higher is better. The results show that as the rate of BK increases our methods increasingly reduce the number of CI tests compared to post-processing. Similarly, they increasingly reduce the average conditioning set size for PC-BK and SNAP(∞)-BK, while for MB-by-MB$^+$-BK, LDECC$^+$-BK and LOAD-BK this depends on the data setting and the rate of BK. Similar to the number of tests, the running time is also consistently reduced. While our main focus is on computational efficiency, the intervention distance in finite sample data seems to improve for PC-BK and SNAP(∞)-BK in Fisher-Z tests, while the trend is less clear for other methods and settings.

Additionally, in Fig. 8 we also compare how sensitive the two stages of applying BK are to noisy data by measuring the difference in intervention distance over different rates of BK error (as in Fig. 17). This figure shows that the intervention distance gets worse with a higher rate of BK error for both types of methods (first two rows), but as the rate of BK error increases, for G^2 tests there seems to be a higher improvement in intervention distance for our methods vs postprocessing (last row), while for Fisher-Z this is less clear.

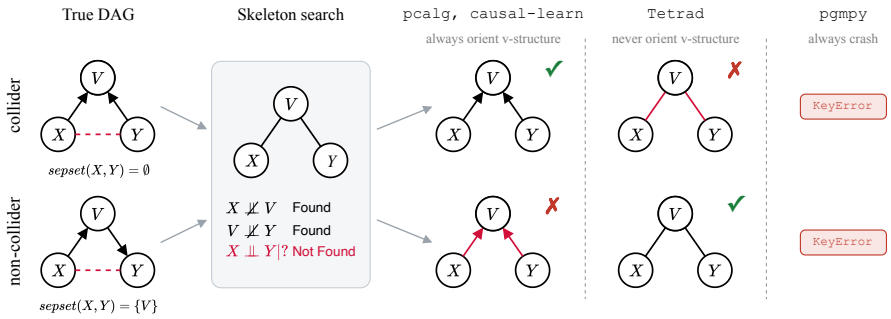


Figure 9: Given BK $\mathcal{B} = \{X \not\to Y\}$, current implementations of PC [25, 27, 26, 32] completely skip tests to find a separating set for X and Y, resulting in incorrect orientations or crashes.

H Comparison to Unsound Methods

As discussed in Sec. 3, current implementations of BK in PC are *not sound* and can only handle a *subset* of pairwise BK during discovery. In particular:

- `pcalg` [25] can only integrate BK on gaps and adjacencies, but *not orientations*, where they become fixed in the skeleton. Fixing gaps directly leads to incorrect v-structure orientations, i.e., the *soundness issues* we discuss in Sec. 3.
- `causal-learn` [26] can only integrate gaps and orientations, i.e., it cannot handle BK that two variables are adjacent with unknown orientation. Similarly to `pcalg`, gaps and adjacencies get fixed in the skeleton, which leads to the *same soundness issues*. The orientations are used to obtain an MPDAG in the orientation phase, *after* all CI tests have been performed, so they do not improve the computational efficiency of the method.
- `tetrad` [27], similarly `causal-learn`, can only handle gaps and orientations which get fixed in the skeleton and lead to the *same soundness issues*. Nonetheless, `tetrad` is the only implementation that utilizes BK during graph learning by avoiding possible-parents in separating sets, which we also discuss in Sec. 3.

These soundness issues are based on the intuitive but wrong assumption that BK should always take priority against CI tests. Fig. 9 provides minimal examples on how these implementations fail. Additionally, none of these implementations use all of the BK we consider (gaps, adjacencies and orientations) during the causal discovery process, which we also prove that we do in a *sound and complete* way. We also note that using order-invariant versions of PC [59] do not solve this problem, since the issue does not stem from the order of the CI tests.

H.1 Code Examples for Handling Gaps

The following examples shows how, for the true DAG $X \to Y \to Z$ with consistent background knowledge $\mathcal{B} = \{X \not\to Z\}$ and oracle CI tests, the PC algorithm is implemented by different causal discovery libraries.

In the `pcalg` library [25], PC returns an incorrect CPDAG $X \to Y \leftarrow Z$ instead of the correct CPDAG $X - Y - Z$.

```r
library(pcalg)

# Define true DAG as X -> Y -> Z
dag <- new("graphNEL",
    nodes = c("X", "Y", "Z"),
    edgeL = list("X" = "Y", "Y" = "Z", "Z" = character(0)),
    edgemode = "directed"
)
# Define background knowledge with forbidden edge between X and Z
fixedGaps <- matrix(FALSE, nrow = 3, ncol = 3)
fixedGaps[1, 3] <- TRUE
```

```r
fixedGaps[3, 1] <- TRUE
# Run PC algorithm with background knowledge
pc_res <- pc(
    suffStat = list(g = dag, jp = RBGL::johnson.all.pairs.sp(dag)),
    indepTest = pcalg::dsepTest,
    alpha = 0.05,
    labels = c("X", "Y", "Z"),
    fixedGaps = fixedGaps
)
# Testing if result is X -- Y -- Z fails!
stopifnot(
    "Y" %in% graph::adj(pc_res@graph, "X") &&
    "X" %in% graph::adj(pc_res@graph, "Y") &&
    "Y" %in% graph::adj(pc_res@graph, "Z") &&
    "Z" %in% graph::adj(pc_res@graph, "Y")
)
```

The following code snippet shows the same behavior in the `causal-learn` library [26].

```python
from causallearn.graph.GraphClass import CausalGraph
from causallearn.search.ConstraintBased.PC import pc
from causallearn.utils.PCUtils.BackgroundKnowledge import BackgroundKnowledge
import networkx as nx
import numpy as np

# Define true DAG as 0 -> 1 -> 2
dag = nx.DiGraph([(0, 1), (1, 2)])
# Define background knowledge with forbidden edge between 0 and 2
nodes = CausalGraph(3).G.get_nodes()
bk = BackgroundKnowledge()
bk.add_forbidden_by_node(nodes[0], nodes[2])
bk.add_forbidden_by_node(nodes[2], nodes[0])
# Run PC algorithm with background knowledge
pc_res = pc(
    data=np.zeros((10, 3)),
    alpha=0.05,
    indep_test="d_separation",
    true_dag=dag,
    background_knowledge=bk,
)
# Testing if result is 0 -- 1 -- 2 fails!
assert pc_res.G.is_undirected_from_to(
    nodes[0], nodes[1]
) and pc_res.G.is_undirected_from_to(nodes[1], nodes[2])
```

In the pgmpy library [32], PC fails (i.e., raises an exception) if the BK is utilized during skeleton search (`enforce_expert_knowledge=True`) due to the missing separating set $sepset(X, Z)$. Otherwise, PC returns the correct CPDAG $X - Y - Z$.

```python
import networkx as nx
from pgmpy.base import DAG
from pgmpy.estimators import ExpertKnowledge, PC
import pandas as pd


# Define true DAG as X -> Y -> Z
dag = DAG([("X", "Y"), ("Y", "Z")])
data = pd.DataFrame({"X": [0] * 10, "Y": [0] * 10, "Z": [0] * 10})
# Define background knowledge with forbidden edge between X and Z
bk = ExpertKnowledge(forbidden_edges=[("X", "Z"), ("Z", "X")])
# Run PC algorithm with background knowledge
```

```
ci_test = lambda x, y, S, **kwargs: float(
    nx.is_d_separator(dag, x, y, set(S) if S else set())
)
pc_res = PC(data).estimate(
    ci_test=ci_test, expert_knowledge=bk, enforce_expert_knowledge=True
)
```

Unlike the previous implementations of PC, the `tetrad` library [27] instead fails to orient v-structures $X \to Y \leftarrow Z$ when the consistent BK $\mathcal{B} = \{X \not\sim Z\}$ is provided.

```
package edu.cmu.tetrad;

import edu.cmu.tetrad.data.Knowledge;
import edu.cmu.tetrad.graph.*;
import edu.cmu.tetrad.search.Pc;
import edu.cmu.tetrad.search.test.IndependenceTest;
import edu.cmu.tetrad.search.test.MsepTest;

public class Main {
    public static void main(String[] args) {
        // Define true DAG as X -> Y <- Z
        Graph dag = GraphUtils.convert("X-->Y,Z-->Y");
        // Define background knowledge with forbidden edge between X and Z
        Knowledge knowledge = new Knowledge();
        knowledge.setForbidden("X", "Z");
        knowledge.setForbidden("Z", "X");
        // Run PC algorithm with background knowledge
        IndependenceTest ciTest = new MsepTest(dag);
        Pc pc = new Pc(ciTest);
        pc.setKnowledge(knowledge);
        Graph pcRes = null;
        try {
            pcRes = pc.search();
        } catch (InterruptedException e) {
            throw new RuntimeException(e);
        }
        // Testing if result is X -> Y <- Z fails!
        if (!pcRes.equals(dag)) {
            throw new AssertionError();
        }
    }
}
```

H.2 Empirical comparison

Regardless of these issues, in Fig. 10 we provide an experimental comparison to `pcalg` (with postprocessing to get orientations) and `causal-learn`, showing their computational performance, but also SHD in comparison to our methods. We use the same setting as in Fig. 3, but only on 50 nodes because on larger graphs the running time exceeded our limit of 24 hours for each graph. This happened for `causal-learn` with any rate of BK, and most graphs using `pcalg` with lower rates of BK (partly due to the implementations of these libraries). We include stable PC implementations for completeness. Since all PC variants output an MPDAG, we can also compare them using SHD. We do not report results for `causal-learn` with BK rate 0.4-1.0 due to the time limit.

Our results show the advantages of our approach. Since PC from `pcalg` skips all CI tests corresponding to known gaps, it can perform slightly fewer CI tests compared to our approach. However, this comes with a *much higher SHD* with any rate of BK. Crucially, *all baselines have a non-zero SHD for d-separation* with BK due to incorrectly handling gaps.

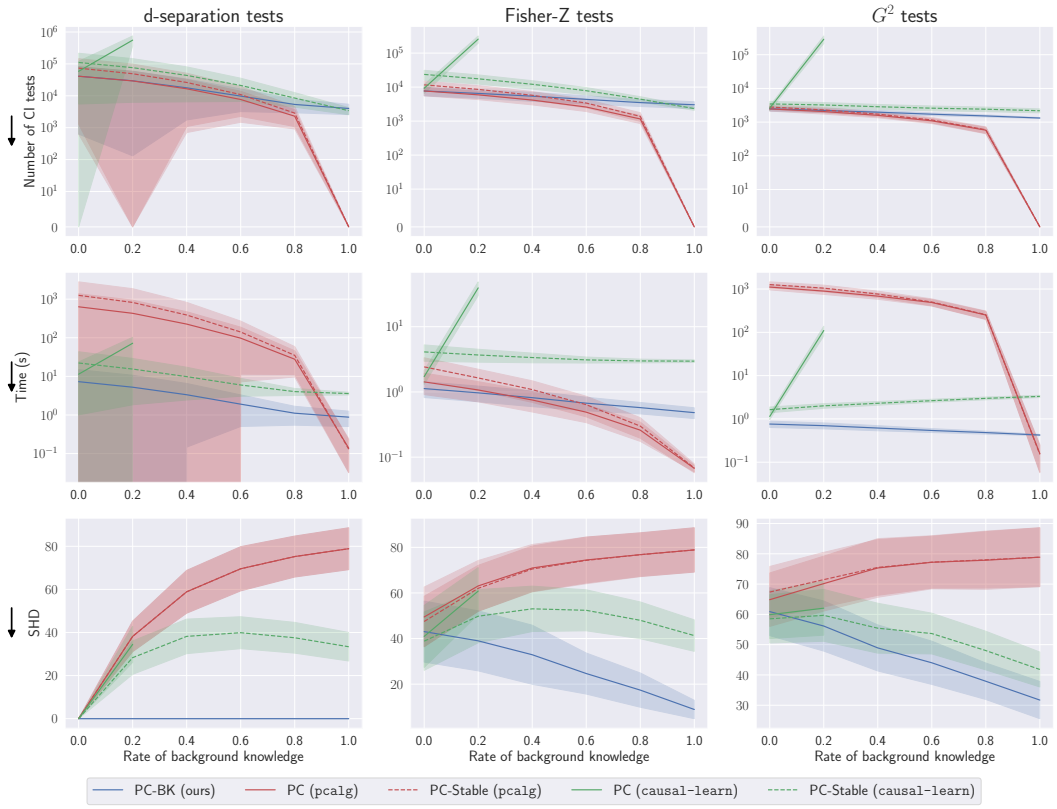


Figure 10: Results over rate of BK between variable pairs, comparing our approach to pre-processing approach for PC available in `pcalg` and `causal-learn`. The shadow area denotes standard deviation.

I Types of Background Knowledge

In this section we study the importance of different types of background knowledge, i.e., adjacencies, orientations and gaps. We show our results for each algorithm separately.

Our results for PC-BK in Fig. 11 show that all types of BK are beneficial. On almost all figures we can see the same trend of more BK implying better performance both in terms of speed and quality. The only exceptions are on binary data with G^2 CI tests, where more known gaps BK increase computation time and known adjacencies increase intervention distance.

Results for SNAP(∞)-BK are shown in Fig. 12. The number of CI tests steadily decreases over all BK types and data domains except for gaps on binary data with G^2 CI tests. As expected, orientation BK consistently helps the most to push down intervention distance. While running time generally decreases when utilizing all types of BK at once over all domains, the BK types separately show less consistency.

The results for MB-by-MB$^+$-BK in Fig. 13 show that all types of BK over all domains, except for gaps, decrease computational costs. As usual, orientation BK helps the most with intervention distance, where results are less regular for other types of BK.

LDECC$^+$-BK, shown in Fig. 14, shows the most clear trends over all metrics with increasing amount of BK over all BK types, metrics and data domains.

Results for LOAD-BK in Fig. 15 are very similar to MB-by-MB$^+$-BK, the local causal discovery sub-routine it uses.

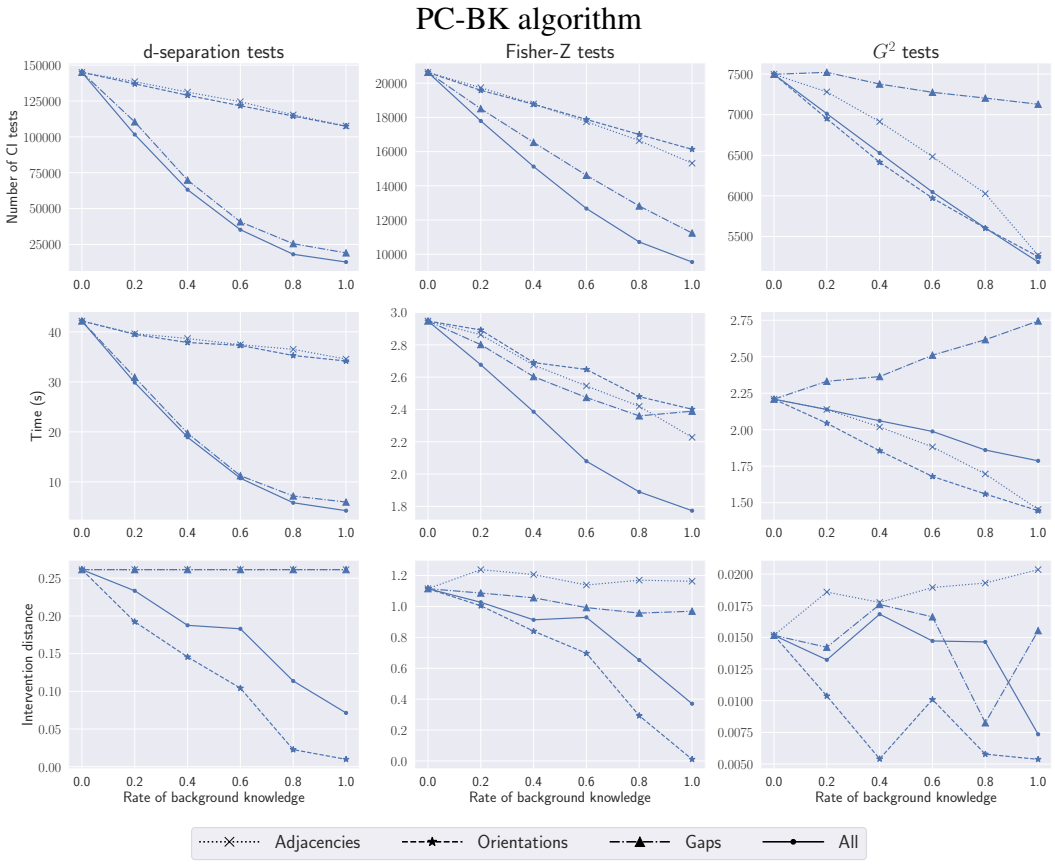


Figure 11: Results for the PC-BK algorithm over rates of different types background knowledge between variable pairs.

J Imperfect Background Knowledge

In this section we experiment with imperfect BK by deliberately introduce errors into it. For a given pair $(X, Y) \in \mathcal{B}$, we generate an erroneous BK as follows. If X and Y are adjacent, then, we $\mathcal{B}$ will either indicate a gap between them, or an orientation that contradicts the true DAG, with uniform probability. If X and Y are non-adjacent, then $\mathcal{B}$ will indicate that they are adjacent or oriented in one of two directions, with uniform probability over all three options. Throughout our experiments we select half the amount of gaps to have errors so that the errors between BK types are more balanced.

In Fig. 16 we conduct experiments with 100 nodes and otherwise the same settings as for the main results shown in Fig. 3 but with a fixed rate of errors by select 10% of adjacencies and orientations, and 5% of gaps to have errors. Our results show that different algorithms display different amounts of sensitivity to BK errors. Even with errors, the computational costs of MB-by-MB$^+$-BK, LDECC$^+$-BK and LOAD-BK are decreasing with more BK, along with all domains, along with PC-BK with d-separation CI tests. On the other hand, SNAP(∞)-BK is much more sensitive to errors, resulting in prohibitive execution times for higher rate of BK across all domains. With a higher rate of BK and a constant rate of errors, the number of incorrect orientations grows. Hence, most algorithms generally exhibit a growing intervention distance with larger variance. Interestingly, different algorithms achieve different intervention distances with the exact same BK errors and oracle d-separation CI tests, further demonstrating that error sensitivity is highly algorithm dependent. Nonetheless, the achieved intervention distances stay reasonably low across algorithms and domains, with local methods MB-by-MB$^+$-BK, LDECC$^+$-BK and LOAD-BK, staying the most robust. In general, the performance of most algorithms, except for SNAP(∞)-BK, degrades gradually rather than sharply, demonstrating that they are resilient against higher rates of BK error.

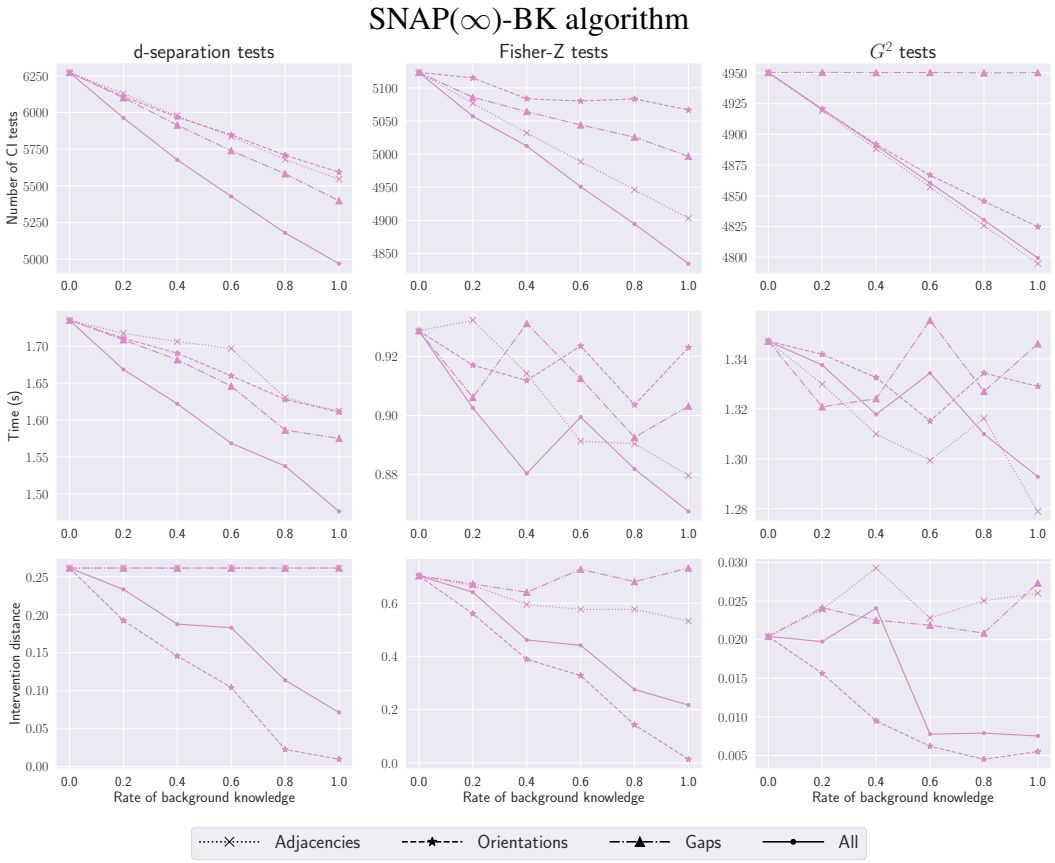


Figure 12: Results for the SNAP(∞)-BK algorithm over rates of different types background knowledge between variable pairs.

In our experiments shown in Fig. 17, we instead keep the BK rate fixed at 0.2 and change the error rate between 0.0 to 0.3, corresponding to errors in 30% of adjacencies and orientations, and 15% of gaps. As expected, the number of CI tests and computation time required by most algorithms increase, but not substantially. One exception is SNAP(∞)-BK which, similarly to the results with fixed error shown in Fig. 16, is much more sensitive to errors compared to other algorithms. The other exception is MB-by-MB$^+$-BK, which gets slightly cheaper as the rate of errors grow. As expected, intervention distance generally grows with higher error rate due to incorrect orientations. PC-BK is a notable exception, as its intervention distance seems to drop for a BK error of 0.3 both with d-separation and Fisher-Z CI tests. However, the variance of the intervention distance achieved by PC-BK is so large that this trend may be only due to noise.

K Realistic Data

We perform experiments on realistic semi-synthetic data sampled according to MAGIC-NIAB and ANDES causal graphs from the `bnlearn` package [52] using the `pgmpy` library [32]. We use the same experimental setup as described in Sec. 5. Our results are shown in Fig. 18.

The MAGIC-NIAB network is causal graph with 44 nodes and 66 edges that generates linear Gaussian data, hence we use Fisher-Z CI tests for all algorithms. Results for MAGIC-NIAB are shown in the first column, where more BK improves all metrics, especially the number of CI tests and computation time. For intervention distance, a difference between no BK and a BK rate of 0.2 leads to a much smaller variance.

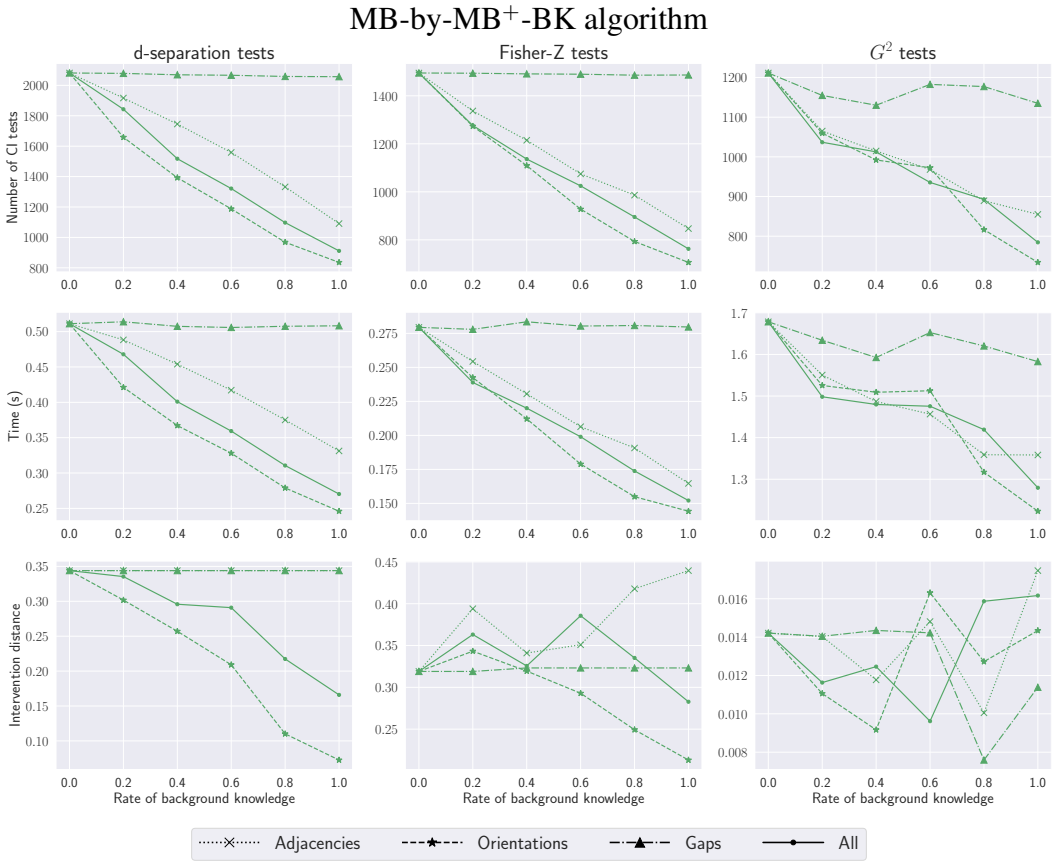


Figure 13: Results for the MB-by-MB$^+$-BK algorithm over rates of different types background knowledge between variable pairs.

The ECOLI70 is another causal graph generating linear Gaussian data over 46 variables and 70 edges. Our results in the second column of Fig. 18 show that more BK improves all metrics, especially in terms of computational efficiency, where the improvements are exponential (shown in log-scale).

The largest linear Gaussian causal graph is ARTH150 with 107 variables 150 edges. Without any background knowledge, the original PC and LDECC algorithms (corresponding to PC-BK and LDECC$^+$-BK with BK rate of 0.0) took prohibitively long to run. Utilizing some amount of BK makes it possible to apply these algorithms on ARTH150, demonstrating that the utilization of BK can be necessary is some scenarios. Computational costs improve for all other algorithms as well (it is difficult to see for SNAP(∞) due to scale).

The ANDES network generates discrete binary data over 223 variables with 338 direct causal relations. We were not able to run LDECC$^+$-BK due to memory issues. The results for the ANDES network, shown in the fourth column, are less straightforward. The computational effort of all methods decrease with more BK except for SNAP(∞)-BK, which decreases only a tiny amount. The intervention distance remains negligible throughout rates of BK.

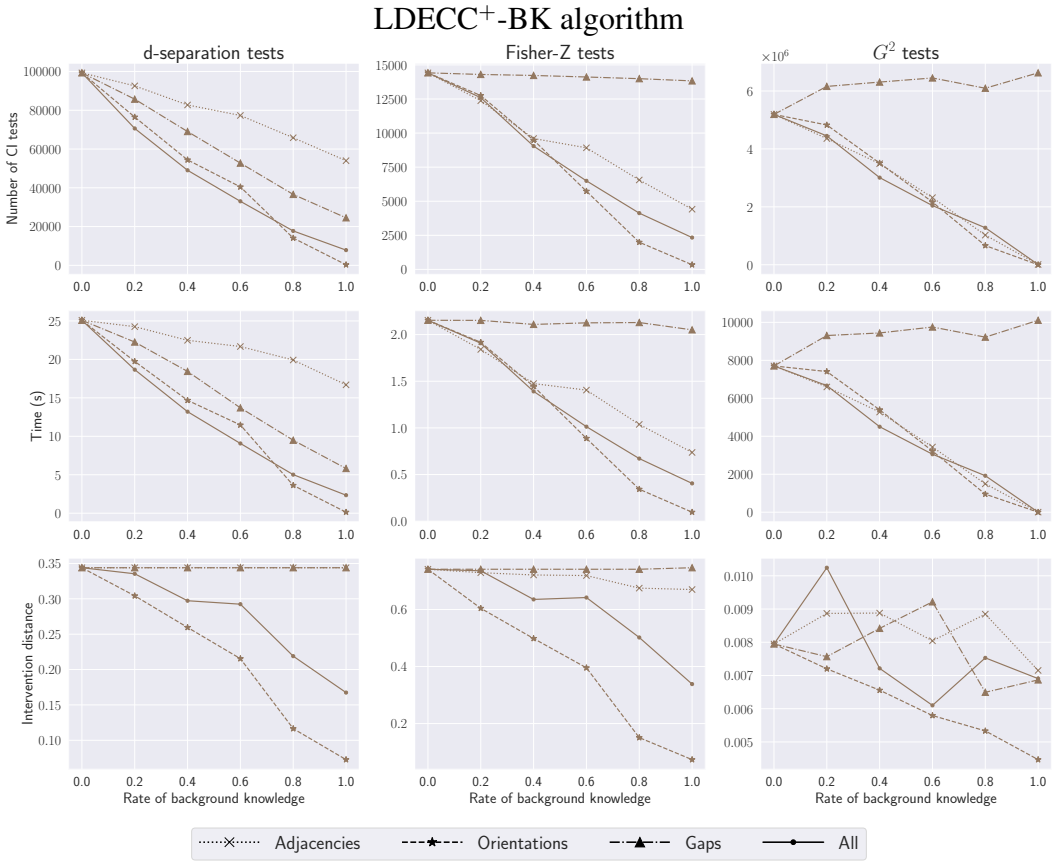


Figure 14: Results for the LDECC$^+$-BK algorithm over rates of different types background knowledge between variable pairs.

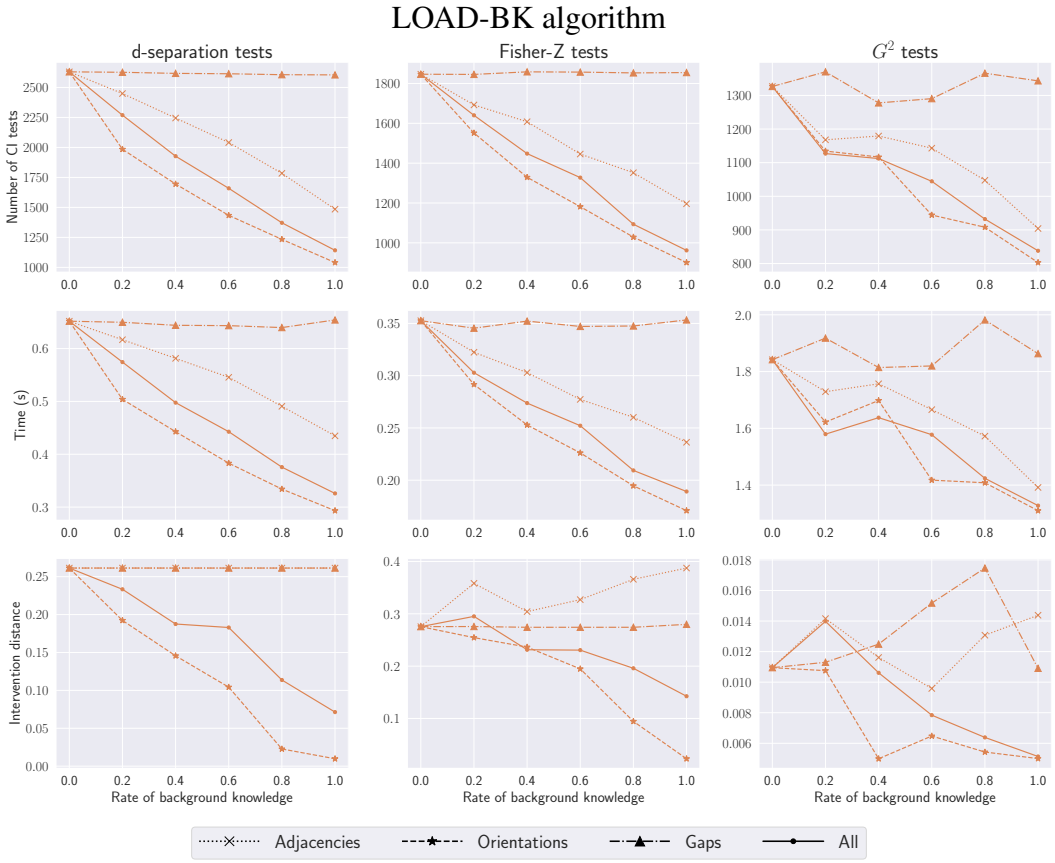


Figure 15: Results for the LOAD-BK algorithm over rates of different types background knowledge between variable pairs.

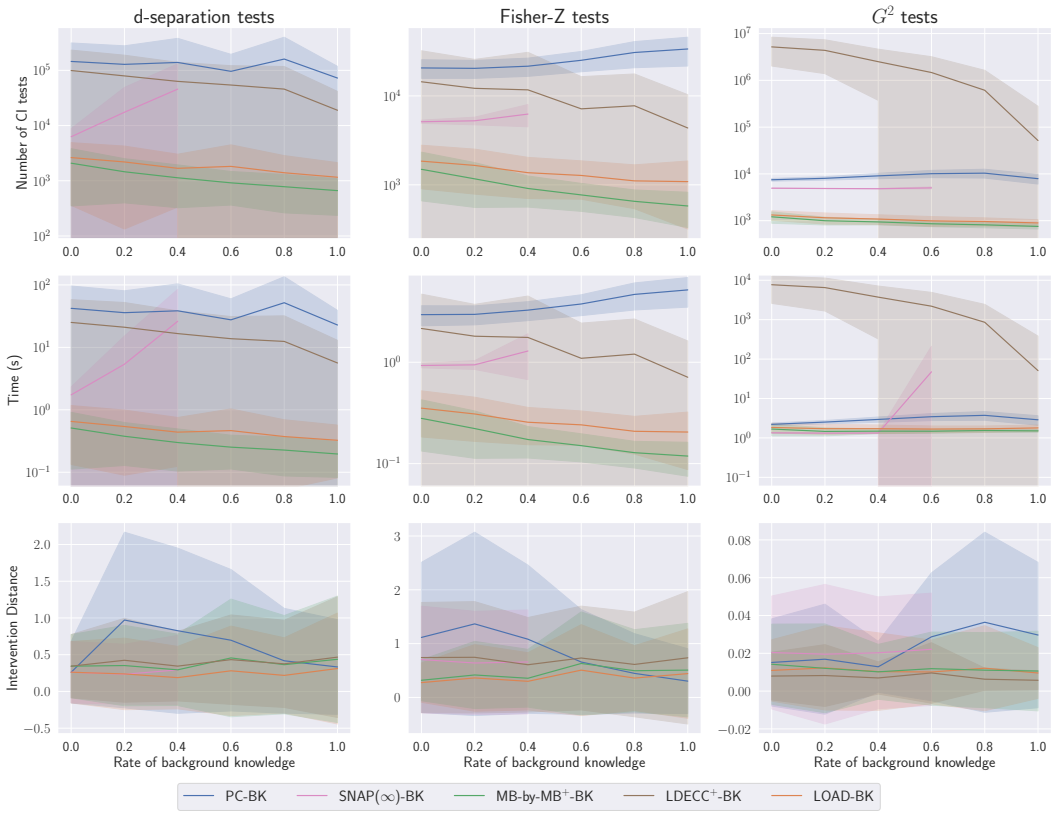


Figure 16: Results over rate of BK between variable pairs, where BK contains errors at a fixed rate of 10% for adjacencies and orientations, and 5% for gaps. The shadow area denotes standard deviation.

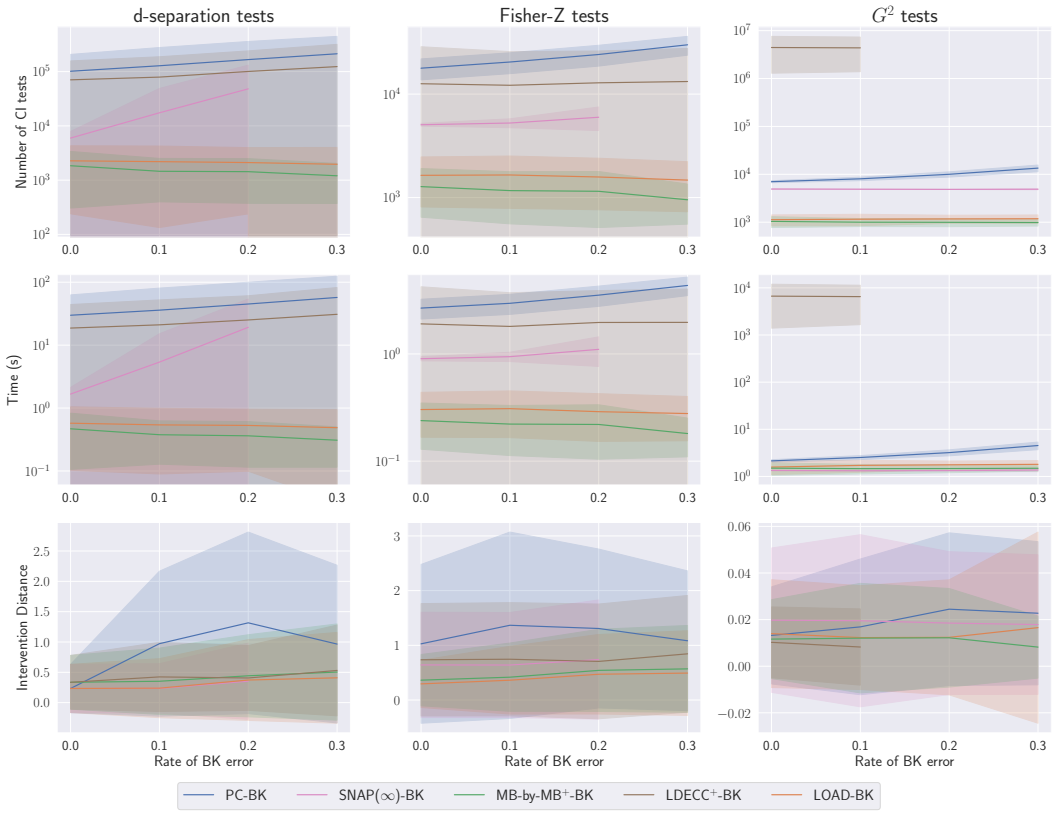


Figure 17: Results over rate of BK error, with BK rate kept fixed at 0.2. The error rate for gaps is always a half of the reported number (e.g., at a rate of 0.1, BK contains errors for 10% of adjacencies and orientations, and for 5% of gaps) to keep the errors balanced between error types. The shadow area denotes standard deviation.

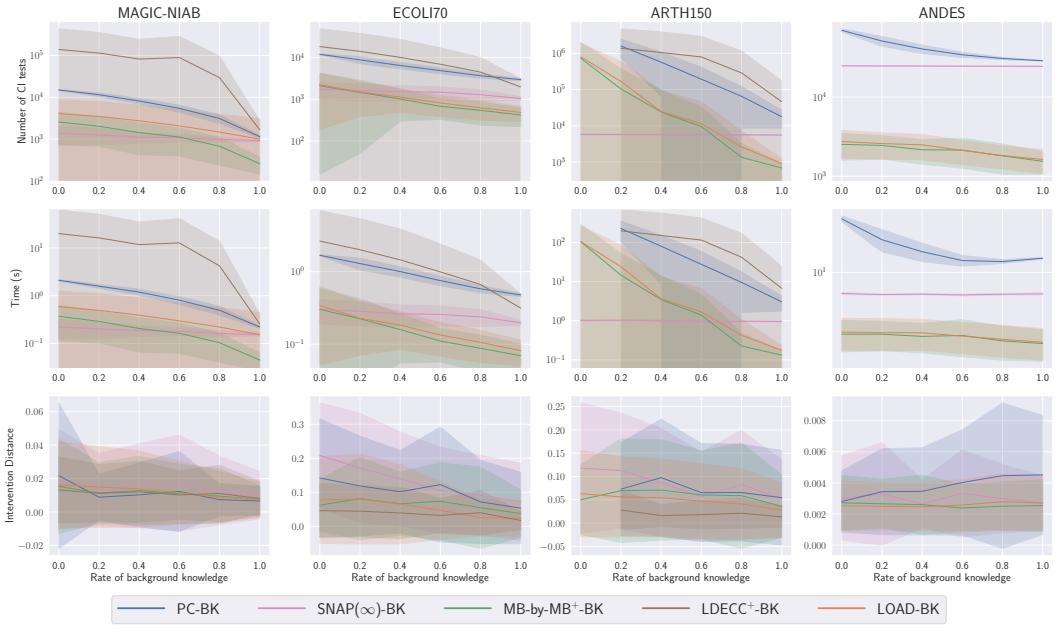


Figure 18: Results on semi-synthetic data according to various causal graphs from the `bnlearn` package [52].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: All claims made in the abstract and the introduction are supported by theoretical results with proofs and empirical evidence including several ablations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We state limitations of our work in the conclusions and extensively discuss them in App. D.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We clearly state our assumptions in Sec. 2 and provide proofs for all theorems in App. C, which we always reference in the main paper.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We explicitly state the parameters of our experiments in Sec. 5 and App. E, and provide several pseudocode for every evaluated algorithm in Sec. 3, App. A and App. B. Additionally, we provide the source code to replicate all results in the supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Our evaluation data is synthetic in nature and can be reproduced using our source code as well as the cited open source library utilized for semi-synthetic datasets.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We specify all learning parameters in Sec. 5 as well as in the provided source code.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All experimental results include standard deviations in both figures and tables, except for a single ablation study in App. I.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the exact computer resources used to run the experiments in App. E

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The paper contains foundational research. It is not tied to particular applications and there is no direct path to any negative applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper contains foundational research with synthetically generated data, and poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite each library along with their corresponding license that are used in the experiments in App. E.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The source code is documented and included in the submission in an anonymized zip file.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.