Journal of Machine Learning Research 21 (2020) 1-37 Submitted 2/18; Revised 4/20; Published 6/20

Local Causal Network Learning for Finding Pairs of Total and Direct Effects

Yue Liu LY199125@PKU.EDU.CN
Zhuangyan Fang FANGZY_MATH@PKU.EDU.CN
Yangbo He* HEYB@PKU.EDU.CN
Zhi Geng ZHIGENG@PKU.EDU.CN
LMAM, School of Mathematical Sciences, LMEQF, and Center of Statistical Science
Peking University
Beijing, China
Chunchen Liu CHENCANG.LCC@ALIBABA-INC.COM
Damo Academy, Alibaba Group
Beijing, China

Editor: Isabelle Guyon

Abstract

In observational studies, it is important to evaluate not only the total effect but also the direct and indirect effects of a treatment variable on a response variable. In terms of local structural learning of causal networks, we try to find all possible pairs of total and direct causal effects, which can further be used to calculate indirect causal effects. An intuitive global learning approach is first to find an essential graph over all variables representing all Markov equivalent causal networks, and then enumerate all equivalent networks and estimate a pair of the total and direct effects for each of them. However, it could be inefficient to learn an essential graph and enumerate equivalent networks when the true causal graph is large. In this paper, we propose a local learning approach instead. In the local learning approach, we first learn locally a chain component containing the treatment. Then, if necessary, we learn locally a chain component containing the response. Next, we locally enumerate all possible pairs of the treatment's parents and the response's parents. Finally based on these pairs, we find all possible pairs of total and direct effects of the treatment on the response.

Keywords: causal networks, directed acyclic graphs, total effects, direct effects, indirect effects

1. Introduction

In many observational studies such as sociological, epidemiological and biological studies, we may not be dissatisfied with the association or correlation among variables; we, however, are more interested on the total causal effect and the direct causal effect of a treatment or an exposure variable on a response or an outcome variable (Holland, 1986; Greenland et al., 1999; Pearl, 2000). To estimate the total causal effect of the treatment on the response, we

*. Corresponding author.

have to find a confounder set, which blocks all non-causal paths from the treatment to the response (Pearl, 2000). Also, to estimate the direct causal effect of the treatment on the response, we have to find a set of intermediate variables which can block all causal paths from the treatment to the response except for the direct causal edge (Pearl, 2000).

Causal directed acyclic graphs (DAGs) are often used to describe causal relations among variables. Given a causal DAG and a variable in this DAG, the parents of the variable in the graph are called the direct causes of the variable (Pearl, 2000). If the underlying causal DAG is specified, then the parent set of the treatment can be treated as the confounder set, and the parent set of the response can be treated as the intermediate variable set. Therefore, we can estimate the total causal effect and the direct causal effect of the treatment on the response from observational data (Pearl, 2000). However, without prior knowledge, usually we can only learn a set of statistically equivalent DAGs from observational data (Verma and Pearl, 1990; Heckerman et al., 1995; Chickering, 2002a,b). These equivalent DAGs form a Markov equivalence class and are called Markov equivalent, and can be represented by a single graph called a essential graph (Andersson et al., 1997). Essential graphs contain both directed and undirected edges. Unfortunately, since association does not imply causation, those Markov equivalent DAGs may entail different causal relations. Therefore, the parent set of the treatment and the parent set of the response may vary with the different Markov equivalent DAGs. Nevertheless, since we know that the true causal DAG is in the Markov equivalence class, ideally we can enumerate all equivalent DAGs and find the parents of the treatment as well as the parents of the response to evaluate a pair of total and direct causal effects for each of those DAGs (Lauritzen, 1999; Pearl, 2001). Collecting these effect pairs, we obtain a set of all possible pairs of total and direct effects, which should contain the true pair of total and direct causal effects.

There have been several approaches proposed to find the bounds or the set of all possible total effects (Cai et al., 2008; Sjölander, 2009; Maathuis et al., 2009; Nandy et al., 2017), but to the best of our knowledge, there is no work which discusses how to find the set of all possible pairs of total and direct effects. To find the bound or the set of all possible total effects, Maathuis et al. (2009) proposed an approach which first learns an essential graph over all vertices and then locally enumerates all possible parent sets of the treatment. This approach is often called intervention do-calculus when the DAG is absent, or IDA for short. There are also several extensions of the original IDA. Nandy et al. (2017) proposed an approach called joint-IDA which extends IDA to joint interventions. Perković et al. (2017) and Fang and He (2020) extended IDA to dealing with direct causal background knowledge and non-ancestral causal background knowledge. Liu et al. (2020) considered the efficiency problem of the original IDA. However, all those methods need to learn an essential graph over all vertices first.

In this paper, we want to find not only the set of all possible total effects but also the set of all possible pairs of total and direct effects of a treatment on a response. As discussed above, this is equivalent to finding all possible pairs of parent sets of both treatment and response. An intuitive method for enumerating pairs of total and direct effects is to enumerate all possible parent sets of the treatment and parent sets of the response separately, and then consider the Cartesian product of the treatment's parent sets and the response's parent sets. This method is a trivial extension of the IDA framework (Maathuis et al., 2009), and thus we call it IDA-based method. However, the IDA-based method considers all combina-

tions of the treatment's parent sets and the response's parent sets, which may include the pairs of the treatment's parents and the response's parents that are not consistent with any Markov equivalent DAG (see Section 4.2 for more details). To address this problem, we propose an intuitive global learning approach and a local learning approach to find all and only those possible pairs of total and direct effects of a treatment on a response in a given Markov equivalence class. In our approach, we assume that the underlying causal DAG is causal sufficient and faithful to the observed distribution. In the intuitive global learning approach, we first find a whole essential graph over all vertices, and then enumerate all equivalent causal DAGs in the Markov equivalence class represented by the essential graph. Finally, we find a pair of total and direct effects for each equivalent causal DAG. Since it is very inefficient for the global learning approach to learn a whole essential graph with a large number of vertices and enumerate all causal DAGs in the class (Spirtes et al., 2000; He et al., 2015), we propose a local learning approach instead. In the local learning approach, we first learn locally a chain component containing the treatment. Next, if necessary, we learn locally a chain component containing the response. We then present an approach for locally enumerating all possible pairs of the treatment's parent sets and the response's parent sets, which only needs the neighbors of the treatment and the response, respectively. Finally, for each enumerated parent set pair, we find a pair of total and direct effects of the treatment on the response. The local learning approach can avoid not only learning a whole essential graph but also enumerating all possible causal DAGs in that Markov equivalence class.

The remainder of the paper is organized as follows. Section 2 introduces the notation and the definitions. In Section 3, we propose several learning approaches for finding all possible pairs of total and direct effects of a treatment on a response. In Section 4, we illustrate and evaluate the proposed local learning approach on both synthetic data sets and the DREAM4 data sets. Finally, some discussions are given in Section 5.

2. Notation and Definitions

In this section, we briefly introduce DAGs, the Markov equivalence class of DAGs and their representations, and the definitions of total and direct effects of a treatment on a response. A graph $G(V, E)$ consists of a vertex set V and an edge set E, where $V = \{X_1, \ldots, X_n = Y\}$ denotes vertices or variables, specially Y denotes a response variable of interest, and E consists of directed edges and/or undirected edges. If all edges in a graph are directed (undirected), then the graph is called directed (undirected). For simplicity, we use $X_i \to X_j$ and $X_i - X_j$ to denote a directed edge and an undirected edge, respectively. Two vertices are adjacent if they are connected by an edge. For a directed edge $X_i \to X_j$, we say that X_i is a parent of X_j and X_j is a child of X_i. The set of parents and children of X_i are denoted by $pa(X_i)$ and $ch(X_i)$ respectively. For an undirected edge $X_i - X_j$, we say that X_j is a neighbor of X_i and vice versa. The set of neighbors of X_i is denoted by $ne(X_i)$. A v-structure is $X_i \to X_j \leftarrow X_k$ without an edge between X_i and X_k, and X_j is called a collider.

A path from X_i to X_j is a sequence of distinct vertices such that any two consecutive vertices are adjacent. A directed path from X_i to X_j is a path on which all arrows are towards X_j, and an undirected path from X_i to X_j is a path on which all edges are undi-

rected. We say that X_j is a descendant of X_i if there is a directed path from X_i to X_j, and X_i connects X_j if there is an undirected path from X_i to X_j. A cycle is a path from a vertex to itself. We say that a cycle is partially directed if it consists of both undirected edges and directed edges with the same directions. For example, $X_1 \to X_2 - X_3 \to X_1$ is a partially directed cycle. A directed acyclic graph (DAG) is a directed graph without any directed cycle.

The notion of d-separation induces a set of conditional independence relations encoded in a DAG (Pearl, 1988). Two DAGs are Markov equivalent if they induce the same set of conditional independence relations. A Markov equivalence class is denoted by a chain graph or an essential graph which consists of both directed and undirected edges (Andersson et al., 1997; Chickering, 2002b). Undirected edges in a chain graph denote the edges whose directions cannot be determined by observational data. After deleting all directed edges from a chain graph, we obtain several disconnected undirected subgraphs called chain components (Andersson et al., 1997). Each chain component is a chordal graph (Blair and Peyton, 1993; Andersson et al., 1997).

For a given DAG, based on the concept of do-calculus $do(X = x)$ (Pearl, 2000), we define the average total causal effect and the average direct causal effect of a treatment variable X on a response variable Y, which are simply called the total effect and the direct effect of X on Y.

Definition 1 (Total effect) *The total effect of X on Y is defined as*

$$TE(x; Y) = \frac{\partial E[Y|do(X = x)]}{\partial x},$$

for all x.

For a binary X, the total effect of X on Y is defined as $E[Y|do(X = 1)] - E[Y|do(X = 0)]$. The parent set $pa(X)$ of a vertex X is a sufficient confounder set for identifying the total effect. To identify the total effect, we first find $pa(X)$, and then estimate the total effect $TE(x; Y)$ by adjusting for $pa(X)$ (Pearl, 2000). For a Gaussian graphical model described by a DAG and a Gaussian distribution, the total effect $TE(x; Y)$ is the coefficient β_X of X in the following regression of Y on X and $pa(X)$

$$E[Y|x, pa(x)] = \beta_0 + \beta_X x + \beta_{pa(X)} pa(x),$$

where X is not a descendant of Y. The total effect $TE(x; Y)$ is 0 if X is a descendant of Y.

To distinguish the coefficients of X in different regressions of Y, we denote the corresponding conditioning sets $pa(X)$ in the subscript of the coefficients. For example, the above coefficient β_X is denoted by $\beta_{X|pa(X)}$.

Below, blocking all directed paths from X to Y except for $X \to Y$, we define the controlled direct effect of X on Y as follows (Pearl, 2001).

Definition 2 (Direct effect) *Let Z denote the parents of Y except for X (that is, $Z = pa(Y) \setminus X$). The controlled direct effect of X on Y under a setting $do(Z = z)$ is defined as*

$$DE(x; Y|z) = \frac{\partial E[Y|do(X = x, Z = z)]}{\partial x}.$$

The controlled direct effect $DE(x; Y|z)$ is the causal effect of X on Y conditioning on the external intervention $do(Z = z)$ which blocks all directed paths from X to Y except for $X \to Y$. Generally the controlled direct effect $DE(x; Y|z)$ is a function of x and z. For a binary X and a discrete Z, the controlled direct effect of X on Y is defined as the set $\{E[Y|do(X = 1, Z = z)] - E[Y|do(X = 0, Z = z)], \forall z\}$. The set $Z = pa(Y) \setminus \{X\}$ can block all other directed paths from X to Y. Thus, it is suffices to find $pa(Y)$ to estimate the direct effect $DE(x; Y|z)$. Particularly, for a Gaussian graphical model with $X \to Y$ in the corresponding DAG, the controlled direct effect $DE(x; Y|z)$ is the coefficient $\beta_{X|pa(Y)}$ in the following regression of Y

$$E[Y|x, pa(y)] = \beta_0 + \beta_{X|pa(Y)}x + \beta_{Z|pa(Y)}z,$$

where $Z = pa(Y) \setminus \{X\}$. In this case, the controlled direct effect $DE(x; Y|z)$ does not depend on z, and thus can be denoted by $DE(x; Y)$, which is the same as the natural direct effect defined by Pearl (2001); and further, we define the indirect effect as the difference of the total effect and the direct effect, $TE(x; Y) - DE(x; Y)$, which is also the same as the natural indirect effect defined by Pearl (2001).

3. Finding All Pairs of Total and Direct Effects for a Markov Equivalence Class

Let X be a treatment variable of interest and Y be a response variable of interest. For a given DAG, there is a pair of total and direct effects, $(TE(x; Y), DE(x; Y))$. Thus, there is a set of effect pairs for a class of Markov equivalent DAGs. Given an observational data set, our goal is to find all effect pairs from the data set. Below we focus on the algorithms for finding the parent set pairs $(pa(X), pa(Y))$ for all DAGs in the Markov equivalence class represented by a given essential graph G^*, and the pairs of total and direct effects can be estimated according to the definitions introduced in Section 2. A global algorithm and its improved version will be presented first in Section 3.1 and then local learning algorithms will be introduced in Section 3.2.

3.1. The Global Learning Approach for Finding All Pairs of Total and Direct Effects

A global approach for finding the effect pairs has four steps: (1) learning a Markov equivalence class represented by an essential graph G^* from data, (2) enumerating all DAGs in the Markov equivalence class, (3) finding $pa(X)$ and $pa(Y)$ for each equivalent DAG, and (4) estimating an effect pair $(TE(x; Y), DE(x; Y))$ for each parent set pair $(pa(X), pa(Y))$. We present the global algorithm in Algorithm 1.

After finding all possible parent set pairs in Algorithm 1, we need an approach for estimating all possible effect pairs. As discussed in Section 2, If the underlying model is linear Gaussian, then for each possible pair of parent sets $(pa_i(X), pa_i(Y))$, we have $TE_i = \beta_{X|pa_i(X)}$ and $DE_i = \beta_{X|pa_i(Y)}$, which can be estimated with the ordinary least squares (OLS) method.

Because the goal is to find the effect pairs of X on Y only, it may not be necessary to globally orient all undirected edges in the learned essential graph G^*. Given an essential

Algorithm 1 A global algorithm via enumerating all DAGs in the Markov equivalence class

Input: A treatment X, a response Y, and an essential graph G^*.

Output: All parent set pairs $(pa_i(X), pa_i(Y))$'s for DAGs in the class represented by G^*.

1: Enumerate all DAGs in the Markov equivalence class represented by G^*, denoted by $G_1, \ldots, G_m$.
2: **for** $i = 1$ to m **do**
3: Find the parent set pair $(pa_i(X), pa_i(Y))$ in G_i.
4: **end for**
5: **return** The parent set pairs $\{(pa_i(X), pa_i(Y)), \forall i = 1, \ldots, m\}$.

graph G^*, Algorithm 2 is proposed to improve Algorithm 1 by semi-locally orienting the undirected edges in the chain components containing X and Y, respectively, rather than orienting all undirected edges in the essential graph G^* over all vertices in V.

Now we introduce some notation for Algorithm 2. Let $S(Y)$ be a subset of $ne(Y)$. Let $G^*_{S(Y)\to Y}$ be the graph obtained from G^* by orienting $Z - Y$ as $Z \to Y$ for each $Z \in S(Y)$ and orienting $Y - W$ as $Y \to W$ for each $W \in ne(Y) \setminus S(Y)$. We say that a configuration $S(Y) \to Y$ is valid for G^* if there is a DAG G in the Markov equivalence class represented by G^* which has the same directed edges connected to Y as $G^*_{S(Y)\to Y}$ has. Furthermore, let $S(X)$ be a subset of X's neighbors in $G^*_{S(Y)\to Y}$, and $G^*_{(S(Y)\to Y, S(X)\to X)}$ be the graph obtained by orienting $Z - X$ in $G^*_{S(Y)\to Y}$ as $Z \to X$ for each $Z \in S(X)$ and orienting $X - W$ in $G^*_{S(Y)\to Y}$ as $X \to W$ for each remaining neighbor W. We call $(S(Y) \to Y, S(X) \to X)$ a sequential orientation configuration of vertices Y and X in turn to G^*. Similarly, we say that $(S(Y) \to Y, S(X) \to X)$ is valid for G^* if there is a DAG G in G^* which has the same directed edges connected to X or Y as $G^*_{(S(Y)\to Y, S(X)\to X)}$ has. If the orientation of the directed edge between X and Y has been fixed or there is no edge between X and Y, then $G^*_{(S(Y)\to Y, S(X)\to X)}$ is identical to $G^*_{(S(X)\to X, S(Y)\to Y)}$, meaning that the graph is not affected by the order of applying $S(Y) \to Y$ and $S(X) \to X$ to G^*.

Given an essential graph G^*, Algorithm 2 first orients the undirected edges connected to Y or X using the configurations $(S(Y) \to Y, S(X) \to X)$ for all $S(Y) \subseteq ne(Y)$ and

Algorithm 2 An improved algorithm via locally orienting the undirected edges connected to Y or X

Input: A treatment X, a response Y, and an essential graph G^*.

Output: All parent set pairs $(pa_i(X), pa_i(Y))$'s for DAGs in the class represented by G^*.

1: Set $k = 0$.
2: **for** each $S(Y) \subseteq ne(Y)$ and each $S(X) \subseteq ne(X)$ in G^* **do**
3: **if** the configuration $(S(Y) \to Y, S(X) \to X)$ is valid for G^*, **then**
4: Set $k = k + 1$, and save the parent set pair $(pa_k(X), pa_k(Y))$ in $G^*_{(S(Y)\to Y, S(X)\to X)}$.
5: **end if**
6: **end for**
7: **return** The parent set pairs $\{(pa_i(X), pa_i(Y)),\ i = 1, \ldots, k\}$.

$S(X) \subseteq ne(X)$. The algorithm then checks the validity of each orientation configuration. Finally, it outputs all parent set pairs for the valid orientation configurations.

At Step 3 in Algorithm 2, it checks the validity of a configuration $(S(Y) \to Y, S(X) \to X)$. A global method for checking the validity is to enumerate all DAGs in the Markov equivalence class represented by G^* and then to check whether there exists a DAG G in G^* whose all directed edges connected to X or Y are the same as the corresponding edges in $G^*_{(S(Y)\to Y, S(X)\to X)}$. Let p be the number of vertices in all chain components of G^*, and k be the number of vertices of the maximum clique in these components. This global method has a time complexity of $O(k!)$ in the best case and $O(p!)$ in the worst case. Thus it may be time-consuming when p or k is sufficiently large (He et al., 2015). Below we introduce a local approach to check the validity and further to find all pairs of total and direct effects.

3.2. The Local Learning Approach for Finding All Pairs of Total and Direct Effects

In this section, we present a local learning approach for finding all pairs of total and direct effects of X on Y for all DAGs in the Markov equivalence class obtained from the distribution of observed variables. To obtain all of these pairs, we need to check the validity of any given configuration $(S(Y) \to Y, S(X) \to X)$. For an orientation $S(X) \to X$, Maathuis et al. (2009) proposed a local criterion for checking its validity as follows.

Lemma 3 (Maathuis et al., 2009, Lemma 3.1, IDA) *For a vertex X in an essential graph G^*, an orientation configuration $S(X) \to X$ is valid if and only if $S(X) \to X$ does not make any new v-structure in $G^*_{S(X)\to X}$.*

This criterion can be used to find all valid parent sets $pa(X)$ of X. Thus, by using the valid parent sets, the set of all possible total effects of X on Y can be found. For the case with multiple treatments $X_1, \ldots, X_k$, Nandy et al. (2017) proposed a criterion for finding the set of all total effects of these treatments on a response Y. The criterion in Nandy et al. (2017) enumerates all valid local DAGs in each of the chain components containing these treatments and then combines these valid local DAGs together.

Now we propose a local learning approach to improve Algorithm 2 in two ways. One way is first to locally learn the chain component containing X and then the chain component containing Y if necessary. This approach is more efficient than learning a whole essential graph over all vertices in V. The other way is to provide a local criterion, which only depends on the subgraphs of the chain components over the neighbors of X (and Y if necessary), to check the validity of any configuration $(S(Y) \to Y, S(X) \to X)$.

3.2.1. Local Learning Algorithm for Finding Chain Components

Now we present the local learning algorithm for learning a chain component which contains a given target vertex, such as X or Y. Denote by $MB(X)$ the Markov blanket (MB) of X conditioning on which X is independent of other vertices. That is, $X \perp\!\!\!\perp \text{others} \mid MB(X)$. Tsamardinos et al. (2003) proposed the IAMB algorithm to learn the Markov blanket of a given vertex. Wang et al. (2014) proposed the MB-by-MB algorithm which is a sequential local learning algorithm for finding the direct causes and the direct effects of a given target vertex. Let $ChComp(X)$ denote the local structure which consists of the

Algorithm 3 A local learning algorithm for finding the local graph $ChComp(X)$

Input: A vertex X.

Output: The local graph $ChComp(X)$.

1: Set WaitQueue = X (the waiting queue of vertices whose MBs will be found).
2: Set $S = \emptyset$ (to be extended to $ChComp(X)$).
3: **repeat**
4: Pop a node Z from WaitQueue.
5: Find $MB(Z)$ (See the IAMB algorithm in Appendix A).
6: Learn the local structure over $MB(Z) \cup \{Z\}$ using the IC algorithm (Pearl, 2000), denoted by $G_{MB(Z)}$.
7: Update S by adding the new edges connected to Z in $G_{MB(Z)}$ and the v-structures containing Z in $G_{MB(Z)}$ to S.
8: Using Meek's approach to orient undirected edges in S (See Algorithm 9 in Appendix A).
9: Put the vertices in $MB(Z)$ which have never been in WaitQueue to WaitQueue.
10: Remove the vertices from WaitQueue which have no undirected paths to S.
11: **until** WaitQueue = $\emptyset$.
12: Let A be the vertices that have undirected paths to X, and $ChComp(X)$ be the subgraphs of S which consists of all edges with at least one vertex in A.
13: **return** $ChComp(X)$.

undirected subgraph containing X, that is, the chain component containing X, and the directed edges surrounding the undirected subgraph. Algorithm 3 provides a local method to learn the local graph $ChComp(X)$ containing a given target vertex X.

The details of Step 5 for the IAMB algorithm and Step 8 for Meek's approach in Algorithm 3 are given by Algorithms 8 and 9 in Appendix A respectively. The local graph $ChComp(X)$ outputted by Algorithm 3 contains undirected edges and directed edges around the target vertex X. Under the causal sufficiency and faithfulness assumptions (Spirtes et al., 2000), we have the following corollary.

Corollary 4 *Given a causal DAG which is causal sufficient and faithful to a probability distribution. Suppose that all conditional independence relations are correctly checked, then the local graph obtained by Algorithm 3 consists of the chain component which contains vertex X and the directed edges surrounding the chain component in the essential graph representing the Markov equivalence class containing the given causal DAG.*

3.2.2. Local Approach for Finding All Effect Pairs

Given the local structure $ChComp(X)$, we now present the local approach for finding all pairs of total and direct effects of X on Y. For a given Y and $ChComp(X)$, there are four possible cases: (1) $Y \to X$, (2) nonadjacent X and Y in $ChComp(X)$, or Y not in $ChComp(X)$, (3) $X \to Y$, and (4) $X - Y$. Below we discuss the method to find all pairs of total and direct effects of X on Y for each of these four cases.

For the case (1) of $Y \to X$ in $ChComp(X)$, there is neither any total effect nor any direct effect of X on Y. That is, $TE(x; Y) = 0$ and $DE(x; Y) = 0$.

Algorithm 4 for the case (2) of nonadjacent X and Y in $ChComp(X)$

Input: A treatment X, a response Y, and $ChComp(X)$.
Output: All valid parent sets $pa(X)$'s.

1: Set $k = 0$.
2: **for** each $S(X) \subseteq ne(X)$ in $ChComp(X)$ **do**
3: $G_1^* = ChComp(X)_{S(X) \to X}$.
4: **if** X is not a collider of a new v-structure in G_1^*, **then**
5: Set $k = k + 1$, and save $pa_k(X)$ in G_1^*.
6: **end if**
7: **end for**
8: **return** The parent sets $\{pa_i(X),\ i = 1, \ldots, k\}$.

For the case (2) of nonadjacent X and Y in $ChComp(X)$, or Y not in $ChComp(X)$, there is no direct effect of X on Y. That is, $DE(x; Y) = 0$. We only need to find all valid parent sets of X to find the total effect set of X on Y. To check the validity of a parent set $pa_k(X)$ in Algorithm 4, the condition at Step 3 in Algorithm 2 is replaced by the criterion in Lemma 3 at Step 4 in Algorithm 4.

For the case (3) of $X \to Y$ in $ChComp(X)$, we have that there is no undirected path between X and Y since G^* is a chain graph in which no partial directed cycles exist (Andersson et al., 1997). Thus X and Y separately belong to two different chain components. The following theorem shows how to check the validity of orientations for the case where X and Y are not contained in the same chain component.

Theorem 5 *Let G^* be an essential graph, and X and Y be two distinct vertices in G^*. For any orientation configuration $(S_1(Y) \to Y, S_2(X) \to X)$, if X and Y separately belong to two different chain components, we have that the orientation configuration $(S_1(Y) \to Y, S_2(X) \to X)$ is valid with respect to G^* if and only if both orientation configurations $(S_1(Y) \to Y)$ and $(S_2(X) \to X)$ are valid separately with respect to G^*.*

According to Theorem 5, for the case (3) of $X \to Y$ in $ChComp(X)$, we can obtain all valid orientation configurations $(S_1(Y) \to Y, S_2(X) \to X)$ by enumerating all valid orientation configurations $(S_1(Y) \to Y)$ and $(S_2(X) \to X)$ separately in their own chain components. Algorithm 5 outputs all valid parent set pairs of X and Y for the case (3) of $X \to Y$ in $ChComp(X)$.

For the case (4) of $X - Y$ in $ChComp(X)$, we have X and Y in the same chain component. Step 3 in Algorithm 2 can be replaced by checking the validity of the orientation configuration $(S(Y) \to Y, S(X) \to X)$ applied to the chain component. In this case, the result of Theorem 5 no longer holds. That is, a configuration $(S(Y) \to Y, S(X) \to X)$ may not be valid even if both configurations $S(Y) \to Y$ and $S(X) \to X$ are valid separately. Therefore, we can not use IDA criterion in Lemma 3 to check the validity of the orientation configuration $(S(Y) \to Y, S(X) \to X)$. This can be illustrated by the following Example 1.

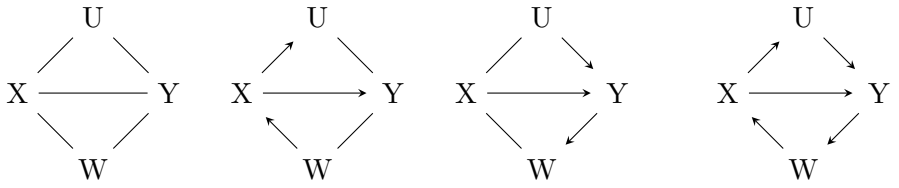
Example 1 *Consider the essential graph G^* in Figure 1(a). When we apply the configuration $(W \to X)$ and the configuration $(\{U, X\} \to Y)$ separately to G^*, both configurations are valid separately for G^* since neither of them generates any new v-structure or*

Algorithm 5 For the case (3) of $X \to Y$ in $ChComp(X)$

Input: A treatment X, a response Y, $ChComp(X)$ and $ChComp(Y)$.
Output: All valid parent set pairs $(pa_i(X), pa_i(Y))$'s.

Set $i = 0$.
for each $S(X) \subseteq ne(X)$ in $ChComp(X)$ **do**
  $G_1^* = ChComp(X)_{S(X)\to X}$.
  if X is not a collider of a new v-structure in G_1^*, **then**
    Set $i = i + 1$, and save the parent $pa_i(X)$ in G_1^*.
  end if
end for
Set $j = 0$.
for each $S(Y) \subseteq ne(Y)$ in $ChComp(Y)$ **do**
  $G_1^* = ChComp(Y)_{S(Y)\to Y}$.
  if Y is not a collider of a new v-structure in G_1^*, **then**
    Set $j = j + 1$, and save the parent $pa_j(Y)$ in G_1^*.
  end if
end for
return The parent set pairs $\{(pa_{i'}(X), pa_{j'}(Y)),\ i' = 1, \ldots, i; j' = 1, \ldots, j\}$.

any cycle, see Figures 1(b) and 1(c) respectively. But when we apply the configuration $(W \to X, \{U, X\} \to Y)$ to G^* simultaneously, it leads to a cyclic as shown in Figure 1(d).



(a) G^* (b) $G^*_{(W\to X)}$ (c) $G^*_{(\{U,X\}\to Y)}$ (d) $G^*_{(W\to X,\{U,X\}\to Y)}$

Figure 1: An example in which both configuration $(W \to X)$ and $(\{U, X\} \to Y)$ are valid separately with respect to G^*, but $(W \to X, \{U, X\} \to Y)$ is not valid.

For the case (4), a simple method for checking the validation is that the configuration $(S(Y) \to Y, S(X) \to X)$ is valid if there is neither new v-structures nor directed cycles in the oriented chain component. It is similar to the semi-local criterion proposed by Nandy et al. (2017). This semi-local criterion is not efficient for a larger chain component. Below we give a local criterion for checking the validity of a configuration $(S(Y) \to Y, S(X) \to X)$ based on the following theorem.

Theorem 6 *Suppose that there is an undirected edge $X - Y$ in the essential graph G^*. Let $neXY = ne(X) \bigcup ne(Y)$. For any orientation configuration $(S(Y) \to Y, S(X) \to X)$ with*

Algorithm 6 For the case (4) of $X - Y$ in $ChComp(X)$

Input: A treatment X, a response Y, and $ChComp(X)$.
Output: All valid parent set pairs $(pa_i(X), pa_i(Y))$'s.

1: Set $k = 0$, and orient $X - Y$ as $X \to Y$ in $ChComp(X)$.
2: **for** each $S(Y) \subseteq (ne(Y) \setminus \{X\})$ in $ChComp(X)$ **do**
3: $G_1^* = ChComp(X)_{S(Y)\to Y}$.
4: **if** Y is not a collider of a new v-structure in G_1^*, **then**
5: **for** each $S(X) \subseteq (ne(X) \setminus \{Y\})$ in $ChComp(X)$ **do**
6: $G_2^* = (G_1^*)_{S(X)\to X}$.
7: **if** X is not a collider of a new v-structure in G_2^*, **then**
8: G'_{neXY} = the subgraph of G_2^* over $ne(X) \bigcup ne(Y)$.
9: **if** No direct cycle in G'_{neXY}, **then**
10: Valid = TRUE.
11: **for** (each partial directed cycle in G'_{neXY}:
$X \to Y \to U - V_1 - \ldots - V_h \to X$ where $h \geq 1$), **do**
12: Valid = Valid $\wedge$ (V_i is adjacent to Y in G'_{neXY}, $\exists i$).
13: **end for**
14: **if** Valid, **then**
15: Set $k = k + 1$, and save the parent set pair $(pa_k(X), pa_k(Y))$ in G'_{neXY}.
16: **end if**
17: **end if**
18: **end if**
19: **end for**
20: **end if**
21: **end for**
22: Return the parent set pairs $\{(pa_i(X), pa_i(Y)),\ i = 1, \ldots, k\}$.

$X \in S(Y)$, define G'_{neXY} be the induced subgraph of $G^*_{S(Y)\to Y, S(X)\to X}$ over $neXY$. Then the orientation configuration $(S(Y) \to Y, S(X) \to X)$ is valid with respect to G^* if and only if the subgraph G'_{neXY} satisfies: (1) no v-structures with collider X or Y, (2) no directed cycle, and (3) for each partial directed cycle containing $X \to Y$, Y is adjacent to at least 3 vertices on the cycle.

By Theorem 6, the validity of an orientation configuration $(S(Y) \to Y, S(X) \to X)$ need not be checked globally in the whole essential graph G^* or semi-locally in the chain component containing X and Y. However, it can be checked locally in the subgraph G'_{neXY} induced by $neXY = ne(X) \bigcup ne(Y)$. Thus by Theorem 6, we give a local algorithm for finding all parent set pairs of X and Y in Algorithm 6.

In Algorithm 6, condition (1) in Theorem 6 is checked at Steps 4 and 7, condition (2) is checked at Step 9, and condition (3) is checked at Steps 11 to 14. At Step 11, a partial cycle must have $Y \to U$ and $V_h \to X$ since $X \to Y$ and the undirected edges connecting Y and the undirected edges connecting X are all oriented at Steps 3 and 6 respectively; h is larger than or equal to 1 since $X \to Y \to U \to X$ for $h = 0$ is checked at Step 9.

Algorithm 7 The local algorithm for finding the effect pairs $(\widehat{TE}_i, \widehat{DE}_i)$'s

Input: : A treatment X, a response Y, and data set D.

Output: Estimates $(\widehat{TE}_i, \widehat{DE}_i)$'s of all effect pairs for DAGs in the class by G^*.

1: Learn $ChComp(X)$ from D via Algorithm 3.

2: For the case (1) $X \leftarrow Y$, set $\widehat{TE} = 0$ and $\widehat{DE} = 0$.

3: For the case (2) without edge between X and Y,
set $\widehat{DE} = 0$, call Algorithm 4 to find $pa_i(X)$'s, and find $\widehat{TE}_i$'s.

4: For the case (3) $X \to Y$,
learn $ChComp(Y)$ from D via Algorithm 3,
then call Algorithm 5 to find $(pa_i(X), pa_i(Y))$'s, and find $(\widehat{TE}_i, \widehat{DE}_i)$'s.

5: For the case (4) $X - Y$, call Algorithm 6 to find $(pa_i(X), pa_i(Y))$'s, and find $(\widehat{TE}_i, \widehat{DE}_i)$'s.

6: At the above steps, for given $pa_i(X)$ and $pa_i(Y)$, we find $\widehat{TE}_i = \hat{\beta}_X^{(i)}$ and $\widehat{DE}_i = \hat{\beta}_{X|pa(Y)}^{(i)}$ from the models

$$E[Y|x, pa_i(X)] = \beta_0^{(i)} + \beta_X^{(i)} x + \beta_{pa(X)}^{(i)} pa_i(X)$$

and

$$E[Y|pa_i(Y)] = \beta_0^{(i)} + \beta_{X|pa(Y)}^{(i)} x + \beta_{Z|pa(Y)}^{(i)} Z,$$

respectively, where $Z = pa(Y) \setminus X$.

In Algorithms 4 to 6, we need not learn a whole essential graph G^*. Instead, we only need to learn one chain component $ChComp(X)$ and learn another chain component $ChComp(Y)$ only for the case of $X \to Y$ in $ChComp(X)$. Summarizing the algorithms for all four cases proposed above, we now present the local learning Algorithm 7 for finding all possible pairs of total and direct effects.

At Step 1 in Algorithm 7, we first learn $ChComp(X)$. According to $ChComp(X)$, we know which one of the four cases (1) to (4) occurs. For the case (1) $Y \to X$ in $ChComp(X)$, both the total effect and the direct effect of X on Y are zero. For the case (2) nonadjacent X and Y in $ChComp(X)$, the direct effect of X on Y is 0, thus we call Algorithm 4 to find all valid parent sets of X, which are used to estimate all possible total effects of X on Y. Only for the case (3) $X \to Y$ in $ChComp(X)$, we need to learn $ChComp(Y)$. Hence we call Algorithm 5 to find all valid parent set pairs of X and Y, which are used to estimate all possible total and direct effect pairs. For the case (4) $X - Y$ in $ChComp(X)$, we call Algorithm 6 to find all valid parent set pairs of X and Y, which are used to estimate all possible total and direct effect pairs. At Step 6, for case (2), we estimate the total effect of X on Y by applying the ordinary least squares method to the regression model of Y on X and $pa(X)$. For cases (3) and (4) we estimate the total effect of X on Y by applying the OLS method to the regression model of Y on X and $pa(X)$. We then estimate the direct effect of X on Y by applying the OLS method to the regression model of Y on X and $pa(Y)$.

There are two advantages for Algorithm 7. One is that it only learns one or two chain components rather than learning the whole essential graph over all vertices for estimating the pairs of total and direct effects of a treatment X on a response Y. The other advantage

for Algorithm 7 is that it locally checks the validity of orientations of the edges connecting X or Y rather than semi-locally checking the validity of orientations of all undirected edges in the chain components.

At Step 1 and Step 4 of Algorithm 7, we use the local method Algorithm 3 to learn the chain components. In fact, we can also use global learning method such as PC and GES to learn an essential graph first, and then directly obtain the chain components of interest. In the following remark, we show that Algorithm 3 is generally more efficient for learning chain components.

Remark 7 *We discuss the first advantage for Algorithm 7 and consider the computational complexity of learning $ChComp(X)$ from a data set D via Algorithm 3 in Step 1 of Algorithm 7. Let $|V|$ be the number of the vertices in V, $|MB|$ be the number of vertices in the maximum Markov blankets and k be the number of vertices in the maximum conditioning set when testing for conditional independence relations. According to Tsamardinos et al. (2006) and Wang et al. (2014), the computational complexities of Steps 5 and 6 in Algorithm 3 are $|MB||V|$ and $|MB|^{2+k}$ respectively. Let r be the number of local structures to be learned in Algorithm 3, we have that the computational complexity of Algorithm 3 is $O(r|MB||V| + r|MB|^{2+k})$. Let $|PC|$ be the number of vertices in the largest set of parents plus children. The complexity of the PC algorithm is $O(|V|^2 \times |PC|^k)$. In general, the local algorithm is faster than the PC algorithm. Consequently, when one just needs to estimate the effect pairs for a given treatment and a given response, the local graph learning in the first Step of Algorithm 7 is usually much faster than the PC algorithm that learns a whole causal graph. When one tries to explore all causal effect pairs among all pairs of every two vertices, it may be better to learn a whole causal graph using a global network learning algorithm instead of the proposed local structure learning algorithm.*

Remark 8 *We consider the other advantage for Algorithm 7. When the treatment X and response Y appear in the different chain components, we can search all valid pairs $(pa(X), pa(Y))$ by finding $pa(X)$ and $pa(Y)$ separately using the IDA algorithm proposed by Maathuis et al. (2009). However, the IDA algorithm is not applicable to the case where X and Y occur in the same chain component. Algorithm 6 is proposed to find all valid pairs $(pa(X), pa(Y))$ by checking the validity in a local subgraph over $N_X \cup N_Y$. When the number of Markov equivalent DAGs in the class represented by the chain components is large, Algorithm 6 is more efficient than Algorithms 1 and 2 since latter two list all possible equivalent DAGs.*

4. Experimental Studies

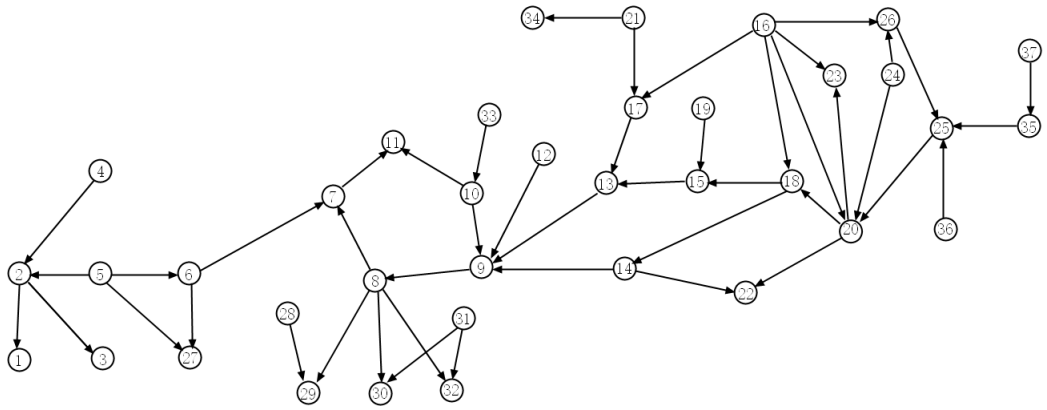
In this section, we illustrate and evaluate the proposed algorithms using some specific causal networks and some causal networks generated randomly. In Section 4.1, we first present a toy example to illustrate how to find all possible parent set pairs of a given treatment and a given response by the proposed local learning algorithm. We then assess the estimates of the pairs of total and direct effects using simulated data. In Section 4.2, we use randomly generated Gaussian graphical models and perfect oracles to test the proposed method and the IDA-based method. The use of perfect oracles rules out the biases caused by learning graphs. The results show that the size of the set of possible effect pairs returned by our

method is in general smaller than that returned by the IDA-based method. In Section 4.3, we further evaluate the effectiveness and efficiency of Algorithm 7 based on randomly generated Gaussian graphical models and finite samples. Finally in Section 4.4, we apply the proposed method on DREAM4 data sets, which are synthetic gene expression data sets and have been widely used in the literature (Hauser and Bühlmann, 2012).

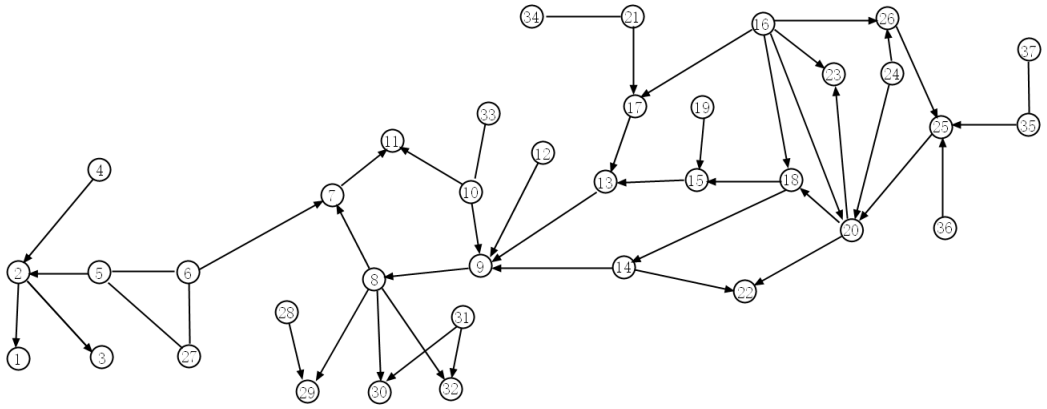
All experiments were run on a computer with Intel 2.5GHz CPU and 8 GB of memory. The experiments in Section 4.1 were implemented in `MATLAB` and all the other experiments were implemented in `R`. The IDA algorithm, the PC algorithm, the stable PC algorithm and the GES algorithm were called from `R package pcalg` (Kalisch et al., 2012). The DREAM4 data sets were called from `R package DREAM4` (Shannon, 2019).

4.1. ALARM Network: A Toy Example

In this section, we use a modified ALARM (Beinlich et al., 1989) network G to illustrate our proposed algorithms. The graph G and its essential graph G^* are shown in Figure 2, where the vertices from 1 to 37 denote the variables from X_1 to X_{37} respectively.



(a) A modified ALARM network G



(b) G^*, the essential graph of G

Figure 2: A modified ALARM network G and its essential graph G^*

Consider the following pairs of treatment and response: (X_{11}, X_{10}), (X_6, X_{11}), (X_{10}, X_{11}) and (X_5, X_{27}), which correspond to the four cases in Algorithm 7: (1) $X_{11} \leftarrow X_{10}$, (2) no edge between X_6 and X_{11}, (3) $X_{10} \rightarrow X_{11}$ and (4) $X_5 - X_{27}$, respectively. Consider a Gaussian graphical model of G defined as follows,

$$X_i = \sum_{X_j \in pa(X_i)} 0.5 \cdot X_j + \varepsilon_i, \tag{1}$$

where $\{\varepsilon_i\}_{1 \le i \le 37}$ are independent variables and

$$\varepsilon_i \sim \begin{cases} N(0,1), & \text{if } pa(X_i) = \emptyset, \\ N(0, 0.1^2), & \text{if } pa(X_i) \neq \emptyset. \end{cases} \tag{2}$$

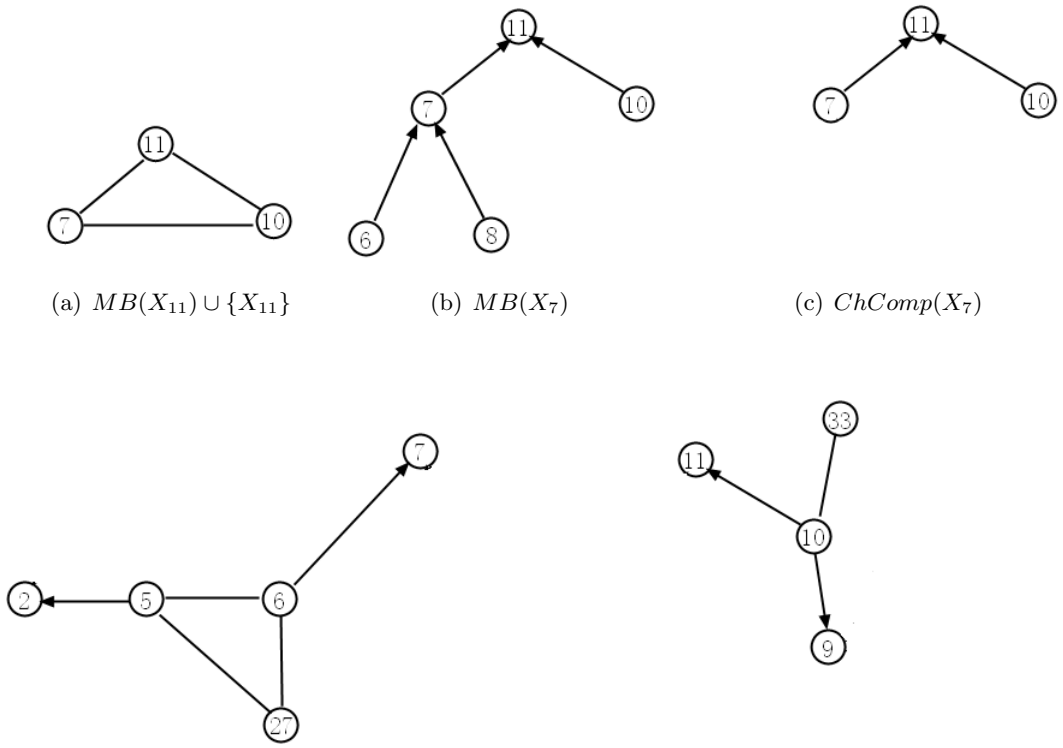
If we know the exact causal graph G, given a treatment X_i and a response X_j, we can obtain the total and direct effects (TE, DE) of X_i on X_j via the path analysis. Below, for each of these four cases, we first show how to learn the local graphs of the essential graph G^* in Figure 2 locally, how to obtain all valid parent sets of treatment and response, and how to give the corresponding pairs of total and direct effects via path analysis using the true parameters. Then we run 100 simulations to learn these parent set pairs and to estimate all pairs of total and direct effects via Algorithm 7 using simulated data.

For the case (1), to find effect pairs of treatment X_{11} on respond X_{10}, we first learn the chain component containing vertex X_{11} by calling Algorithm 3 at Step 1 of Algorithm 7. Algorithm 3 first finds $MB(X_{11}) = \{X_7, X_{10}\}$, and learns the local graph over $MB(X_{11}) \cup \{X_{11}\}$ as shown in Figure 3(a). Then Algorithm 3 finds $MB(X_7) = \{X_6, X_8, X_{10}, X_{11}\}$ and learns the local graph over $MB(X_7) \cup \{X_7\}$ as shown in Figure 3(b). Since the undirected graph around vertex X_{11} is surrounded by directed edges, we obtain that $ChComp(X_{11})$ is $7 \rightarrow 11 \leftarrow 10$ as shown in Figure 3(c). For $11 \leftarrow 10$, it is the case (1) and the effect pair $(TE, DE) = (0, 0)$ at Step 2 of Algorithm 7. The results are shown in the block for the case (1) in Table 1. In Table 1, $\star$ in column 5 indicates that there is no need to find the parent set $pa(R)$ of the response.

For the case (2), to find effect pairs of treatment X_6 on respond X_{11}, we first find the local graphs over $MB(X_6)$, $MB(X_5)$ and $MB(X_{27})$ sequentially and then obtain the local graph $ChComp(X_6)$ at Step 1 of Algorithm 7. The local graph $ChComp(X_6)$ is shown in Figure 3(d). In $ChComp(X_6)$, vertex 6 does not connect vertex 11, and thus at Step 3 of Algorithm 7 we obtain that the direct effect DE of X_6 on X_{11} is 0 (that is, $DE = 0$). Then we call Algorithm 4 to find all possible parent sets of X_6: $\{5\}$, $\emptyset$, $\{27\}$ and $\{5, 27\}$. For each parent set $pa(X_6)$, we obtain the true total effect by path analysis. For example, consider the first parent set $\{X_5\}$. The total effect of vertex 6 on vertex 11 is the conditional expectation of vertex 11 given vertex 6 by adjusting for the vertex 5, which is the coefficient β_X in the following linear regression model

$$E(X_{11}|X_6, X_5) = \beta_0^{(1)} + \beta_X^{(1)} x_6 + \beta_5^{(1)} X_5.$$

Applying the path analysis to the underlying DAG G, we obtain $\beta_X = 0.5 \times 0.5 = 0.25$, and thus the first effect pair is $(TE, DE) = (0.25, 0)$. Similarly, for the other parent sets $\emptyset$, $\{27\}$ and $\{5, 27\}$ of vertex 6, we can get the effect pairs via the path analysis. In Table 1, we show the pairs of total and direct effects in the block for the case (2).



(a) $MB(X_{11}) \cup \{X_{11}\}$ (b) $MB(X_7)$ (c) $ChComp(X_7)$

(d) $ChComp(X_5)$,$ChComp(X_6)$ and $ChComp(X_{27})$ (e) $ChComp(X_{10})$

Figure 3: The local graphs obtained via Algorithm 3

For the case (3), to find effect pairs of X_{10} on X_{11}, we first find the local graphs over $MB(X_{10})$ and $MB(X_{33})$ sequentially. We then obtain the local graph $ChComp(X_{10})$ as shown in Figure 3(e). In the graph $ChComp(X_{10})$, we have $10 \to 11$ and go to Step 4 of Algorithm 7. To obtain the effect pairs, we need to learn $ChComp(X_{11})$ containing the response X_{11}, as shown in Figure 3(a). Then we call Algorithm 5 to find all possible parent set pairs of vertex 10 and vertex 11: $(\{33\}, \{7, 10\})$ and $(\emptyset, \{7, 10\})$. We also get that all possible effect pairs are the same: $(TE, DE) = (0.5625 = 0.5 + 0.5^4, 0.5)$ via the path analysis, as shown in the block for the case (3) in Table 1.

For the case (4), to find effect pairs of X_5 on X_{27}, we first find the local graphs over $MB(5)$, $MB(6)$ and $MB(27)$ sequentially. Then, we obtain $ChComp(X_5)$ in Figure 4 (b). For $5 - 27$ in $ChComp(X_5)$, we go to Step 5 of Algorithm 7 to find all possible parent set pairs with setting $5 \to 27$: $(\emptyset, \{6, 5\})$, $(\{6\}, \{6, 5\})$ and $(\emptyset, \{5\})$. The corresponding effect pairs are $(0.75, 0.5)$, $(0.5, 0.5)$ and $(0.75, 0.75)$, respectively. Additionally, by setting $5 \leftarrow 27$, we have the effect pair $(TE, DE) = (0, 0)$. All parent set pairs and all effect pairs are shown in the block for the case (4) in Table 1.

We have shown how to find the parent set pairs of a given treatment and a given response for four cases in Algorithm 7. It can be seen that we only need to find the chain component containing the treatment for the cases (1), (2) and (4), and further to find the

Case	T	R	No.	$pa(T)$	$pa(R)$	TE	DE	$\widehat{TE}$ $Mean(Std)$	$\widehat{DE}$ $Mean(Std)$
1	11	10	1	$\underline{\{7,10\}}$	$\underline{\star}$	$\underline{0}$	$\underline{0}$	0(0)	0(0)
2	6	11	2	$\underline{\{5\}}$	$\underline{\star}$	$\underline{.25}$	$\underline{0}$	.247(.02)	0(0)
			3	$\emptyset$	$\star$	.25	0	.245(.10)	0(0)
			4	$\{27\}$	$\star$	.25	0	.242(.12)	0(0)
			5	$\{5,27\}$	$\star$	.25	0	.242(.13)	0(0)
3	10	11	6	$\underline{\{33\}}$	$\underline{\{7,\ 10\}}$	$\underline{.0625}$	$\underline{.5}$	.0615(.06)	.500(.01)
			7	$\emptyset$	$\{7,\ 10\}$	.0625	.5	.0623(.01)	.500(.01)
4	5	27	8	$\underline{\emptyset}$	$\underline{\{6,\ 5\}}$	$\underline{.25}$	$\underline{.5}$	.251(.02)	.501(.03)
			9	$\{6\}$	$\{6,\ 5\}$	0	.5	0(0)	.501(.03)
			10	$\emptyset$	$\{5\}$	0	.75	0(0)	.751(.02)
			11	$\{27,\ \star\}$	$\star$	0	0	0(0)	0(0)

Table 1: Results of the modified ALARM by Algorithm 7. T denotes treatment and R response. Each row shows a pair $(pa(T), pa(R))$, a pair (TE, DE), a pair $(\widehat{TE}, \widehat{DE})$. We also underline the true pairs obtained from the underlying causal model.

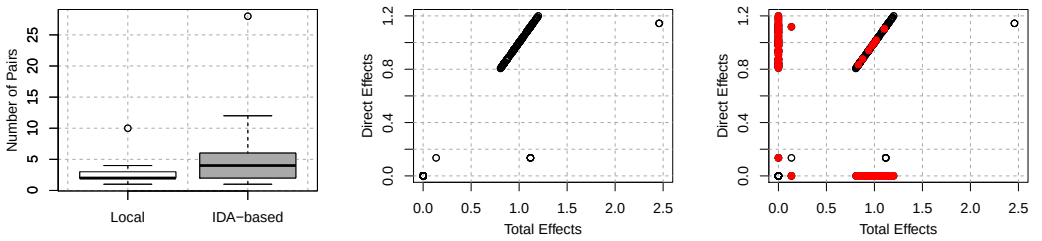
chain component containing the response only for the case (3). Moreover, we need not find the whole essential graph G^* over all vertices for any case.

Now, we evaluate Algorithm 7 via 100 simulations based on the Gaussian graphical model shown at the beginning of this section. The underlying DAG of the given graphical model is the graph G shown in Figure 2. In Table 1, we show the true parent set pairs of treatments and responds and the true effect pairs of treatments on responds with underlines. In each simulation, we first generated a sample of size 1000 from the given Gaussian graphical model and then called Algorithm 7 to find all possible parent set pairs and the estimates of the total and direct effects $(\widehat{TE}, \widehat{DE})$ for each parent set pair via the ordinary least squares method. We give the means and the standard errors (in brackets) of estimates of total effect $\widehat{TE}$ and direct effect $\widehat{DE}$ for each parent set pair via the OLS method. From the estimates shown in Table 1, we can see that they are very close to the true values.

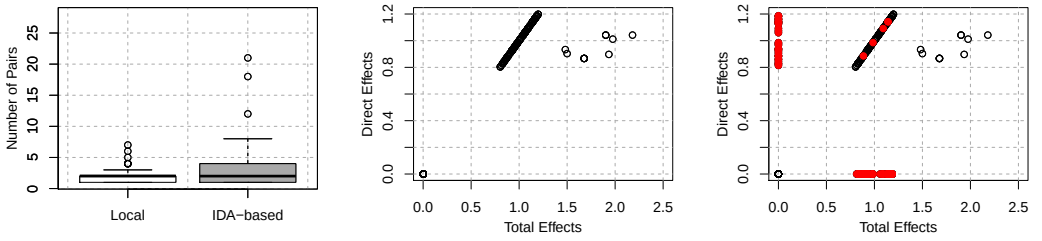
4.2. Evaluation with Randomly Generated Causal Models and Perfect Oracles

In this section, we use randomly generated Gaussian graphical models and perfect oracles to test the proposed method and the IDA-based method. The use of perfect oracles guarantees that the input graph of the IDA-based method and the chain components returned by Algorithm 3 is identical to the underlying true ones, and thus rules out the influence of learning graphs. The experiments were conducted as follows. We first randomly generated causal DAGs with the number of vertices $p = 50, 70$ and average edge degree $\deg = 1, 2, 4$. Next, for each DAG G, we generated a multivariate Gaussian distribution based on the linear structural equation model with respect to G,

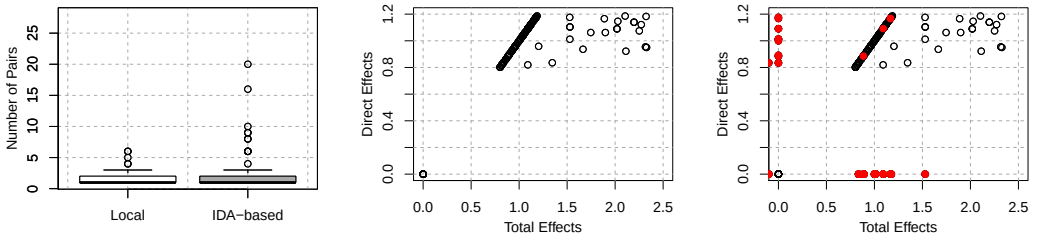
$$X_i = \sum_{X_j \in pa(X_i)} \beta_{j,i} \cdot X_j + \varepsilon_i, \tag{3}$$



(a) Number of pairs, deg = 1 (b) Local method, deg = 1 (c) IDA-based method, deg = 1



(d) Number of pairs, deg = 2 (e) Local method, deg = 2 (f) IDA-based method, deg = 2



(g) Number of pairs, deg = 4 (h) Local method, deg = 4 (i) IDA-based method, deg = 4

Figure 4: Experimental results on randomly generated causal models and perfect oracles, with $p = 50$ and $\deg = 1, 2, 4$. The first column compares the number of possible total and direct effects pairs $(\widehat{TE}, \widehat{DE})$ returned by the local and the IDA-based methods. The second and third columns show all the possible pairs returned by the local and the IDA-based methods, respectively. Note that, the red dots in (c), (f) and (i) are false pairs returned by the IDA-based method.

where the regression coefficients $\beta_{j,i}$'s were independently and uniformly sampled from $[0.8, 1.2]$, and residuals ε_i's were generated from $N(0, 0.1)$ independently. Since we suppose in this simulation that the perfect oracles are available, we directly computed the population covariance matrix from the sampled distribution rather than estimated it from data. Finally, we randomly picked two adjacent vertices X and Y in G, and used the population covariance matrix to estimate the possible pairs of total and direct causal effects of

X on Y with the IDA-based method and the proposed Algorithm 7. For each setting, the above distribution generation procedure was repeated 100 times.

Below we only report the results for DAGs with $p = 50$, since the conclusions are similar for $p = 70$. Figure 4 shows the results on randomly generated causal models and perfect oracles, with $p = 50$ and deg $= 1, 2, 4$. Since perfect oracles were given, the estimated possible total and direct causal effects pairs must include the underlying true pair. Hence, we only studied the number of possible effect pairs estimated by the local method (Algorithm 7) and the IDA-based method. It can be easily seen from Figures 4(a), 4(d) and 4(g) that the number of effect pairs returned by the local method is in general smaller than that returned by the IDA-based method. The more sparse the graph is, the more effect pairs the IDA-based method will return. In fact, as pointed in Section 1, the possible set of effect pairs returned by the IDA-based method may contain effect pairs that never occur in any Markov equivalent DAG. For example, comparing Figures 4(c) and 4(b), there are many effect pairs returned by the IDA-based method are scattered on X-axis and Y-axis. However, those effect pairs could never occur in any Markov equivalent DAG. The effect pairs on Y-axis have zero total effects and positive direct effects, which are impossible since in our settings all regression coefficients are positive. This implies that the total effect of a treatment on a response should be greater than the corresponding direct effect. On the other hand, the effect pairs on X-axis have zero direct effects and positive total effects, which are also impossible since we only considered adjacent variables. We note that, apart from the points on X-axis and Y-axis, there are still many pairs away from X-axis and Y-axis which could never occur in any Markov equivalent DAG. For example, none of the red dots in Figures 4(c), 4(f) and 4(i) is true effect pair, and some of which are not on any axes.

4.3. Evaluation with Randomly Generated Causal Models and Finite Samples

In this section, we use randomly generated Gaussian graphical models and finite samples to test the proposed method and the IDA-based method. The Gaussian graphical models were generated in the same way as described in Section 4.2. For each sampled Gaussian distribution, we further drew samples of size $N = 2000, 10000$. Next, we randomly chose two adjacent vertices X and Y in G, and used the simulated data set to estimate the possible pairs of total and direct causal effects of X on Y with the IDA-based method and the proposed Algorithm 7. For each setting, the above data generation procedure was repeated 100 times.

As discussed in Section 3, the estimation of possible pairs of total and direct causal effects is based on learning essential graphs (which is needed by the IDA-based method) or chain components (which is needed by Algorithm 7). Therefore, we used PC, stable PC and GES to learn essential graphs, and used Algorithm 3 to learn chain components. We also considered to use PC, stable PC and GES to learn essential graphs first, and then obtain the chain components by reading the induced subgraphs of the learned essential graphs. For ease of presentation, we use PC-IDA, PCS-IDA and GES-IDA to denote the IDA-based method combined with the corresponding global learning algorithms, and PC-L, PCS-L and GES-L to denote the local method Algorithm 7 combined with the corresponding global

learning algorithms. The fully local method, which combines two local Algorithms 7 and 3, is denoted by 'Local'.

Apart from CPU time, we define the following three metrics to evaluate and compare the performance of different methods. Let (TE_{true}, DE_{true}) denote the true pair of total and direct effects, $S_{true} = \{(TE_i, DE_i)\}_{i=1}^n$ denote the set of possible pairs of total and direct effects estimated with true essential graph, perfect oracles and Algorithm 2, and $S_{est} = \{(\widehat{TE}_i, \widehat{DE}_i)\}_{i=1}^m$ denote the set of possible pairs of total and direct effects estimated from data with a certain method. The first metric is called the minimum distance, which is defined as

$$\text{minDist}(S_{est}) = \min_{i=1,2,\ldots,m} \sqrt{(\widehat{TE}_i - TE_{true})^2 + (\widehat{DE}_i - DE_{true})^2}.$$

Clearly, the minimum distance equals zero if and only if the true pair is included in the estimated set of total and direct causal effect pairs. The second metric is causal mean square error (CMSE), which is modified from the CMSE defined in Tsirlis et al. (2018) and has the following form,

$$\text{CMSE}(S_{est}) = \frac{1}{m}\sum_{i=1}^{m}\Big[(\widehat{TE}_i - TE_{true})^2 + (\widehat{DE}_i - DE_{true})^2\Big].$$

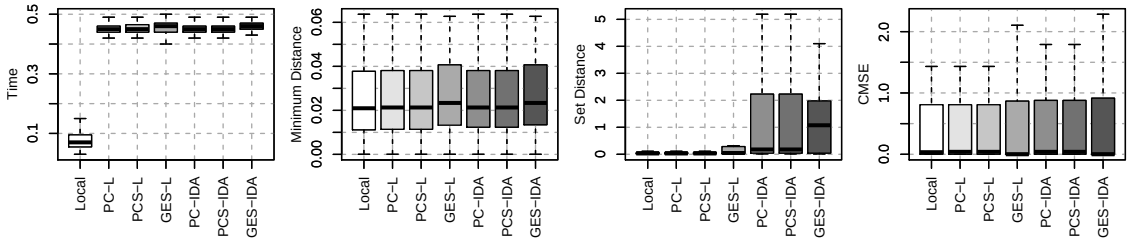
the CMSE can be viewed as the averaged square distance between each pair in S_{est} and the true pair (TE_{true}, DE_{true}). However, due to the non-identifiability of the true causal DAG, the CMSE is generally non-zero since S_{est} may contain more than one total and direct causal effects pair. The last metric is set distance, which measures the distance between the set S_{est} and the set S_{true}. Let S_1 and S_2 be two finite subsets of $\mathbb{R}^2$ with $|S_1| \geq |S_2|$, let $\mathcal{F}$ be the set of all surjections from S_1 to S_2. Then the set distance between S_1 and S_2 is defined as

$$\text{setDist}(S_2, S_1) = \text{setDist}(S_1, S_2) = \min_{f\in\mathcal{F}} \sum_{s_1\in S_1} \|s_1 - f(s_1)\|_2.$$

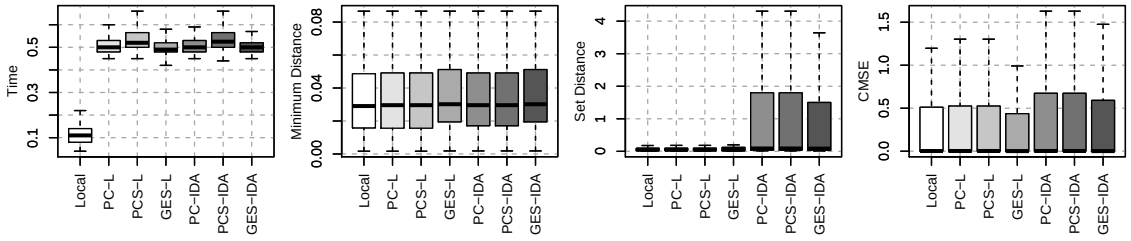
To evaluate a method, we need compute $\text{setDist}(S_{est}, S_{true})$. In Appendix, we transform the above combinatorial minimization problem to a maximum weight matching problem of a bipartite graph, and thus this minimization problem can be easily solved by, for example, Kuhn-Munkres algorithm (Kuhn, 1955; Munkres, 1957).

We only present the results for $p = 50$. Figures 5 and 6 show experimental results on randomly generated causal models and finite samples, with $p = 50$, $\deg = 1, 2, 4$ and $N = 2000, 10000$. The CPU time, minimum distance, set distance, and CMSE of different methods are reported. In order to display the results more clearly, the outliers are not drawn in the figures. Clearly, the CPU time of the fully local method is shorter than that of other methods, especially when the graphs is sparse or the samples size is small. Besides, the CPU time of PC-L (PCS-L, GES-L) is similar to that of PC-IDA (PCS-IDA, GES-IDA), since both of these methods use the PC (stable PC, GES) algorithm to learn an essential graph.

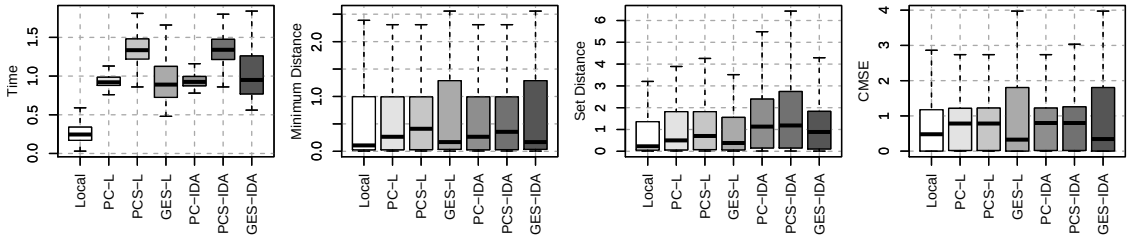
We next consider the effectiveness of different methods. The minimum distance of the fully local method is better than other methods when the sample size is small and similar to



(a) deg = 1



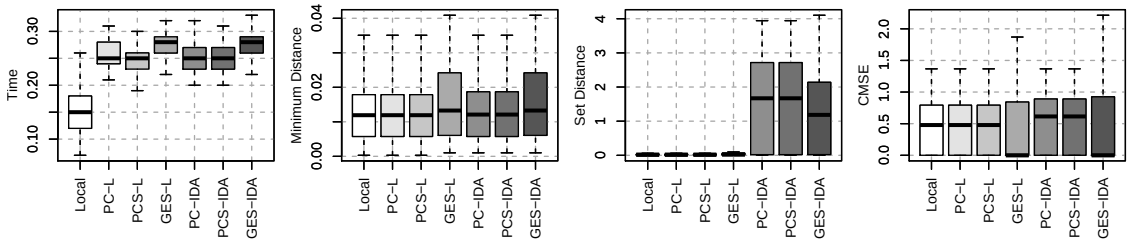
(b) deg = 2



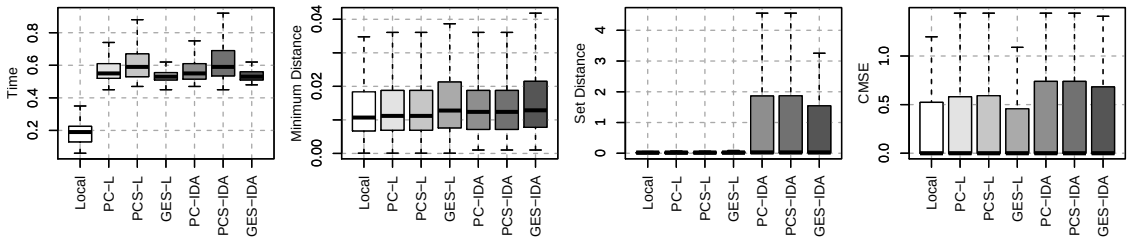
(c) deg = 4

Figure 5: Experimental results on randomly generated causal models and finite samples, with $p = 50$, deg $= 1, 2, 4$ and $N = 2000$. The CPU time, minimum distance, set distance, and CMSE of different methods are reported.

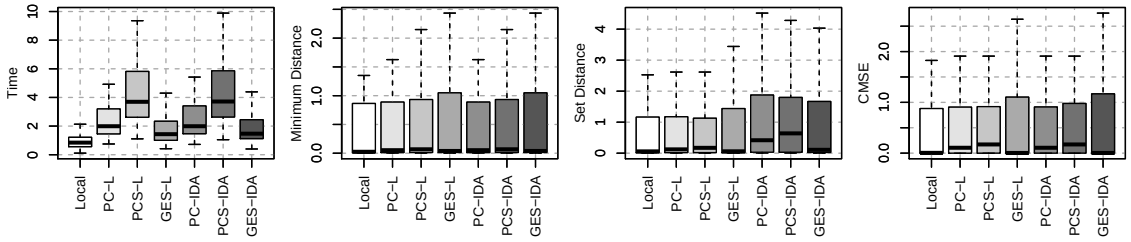
others when the sample size is large. This is because that the fully local method only needs to learn the chain components while others need to learn an entire essential graph. Besides, the minimum distance of PC-L (PCS-L, GES-L) is similar to that of PC-IDA (PCS-IDA, GES-IDA) since they use the same method to learn the graphs and the set of possible effect pairs given by PC-IDA (PCS-IDA, GES-IDA) is a super set of that given by PC-L (PCS-L, GES-L). On the other hand, if we consider the set distance, which includes the influence of the number of the possible effect pairs, the fully local method, PC-L, PCS-L, and GES-L all perform significantly better than PC-IDA, PCS-IDA, and GES-IDA, since the latter three may produce nonexistent total and direct effects pairs. Moreover, among the former four



(a) deg = 1



(b) deg = 2



(c) deg = 4

Figure 6: Experimental results on randomly generated causal models and finite samples, with $p = 50$, deg $= 1, 2, 4$ and $N = 10000$. The CPU time, minimum distance, set distance, and CMSE of different methods are reported.

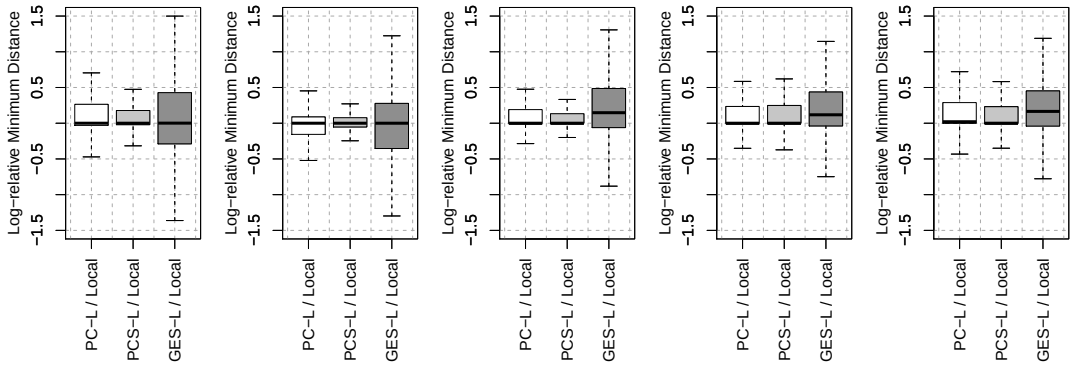
methods, the fully local method is usually the best, since the fully local method only needs to locally learn the chain components.

Finally, let us see the CMSEs of different methods. The CMSEs of the fully local method, PC-L, PCS-L, PC-IDA, PCS-IDA are similar to each other, especially when the graph is sparse or the sample size is small. However, the CMSEs of GES-L and GES-IDA are usually smaller in terms of median, but the standard deviations of CMSE of GES-L and GES-IDA are usually larger. These facts imply that GES-based methods are more sensitive to the sample size, but in general, the estimates are more concentrated than other methods. That is, either the possible effect pairs estimated by GES-based methods are similar to each other, or the number of possible effect pairs are small. However, since the minimum

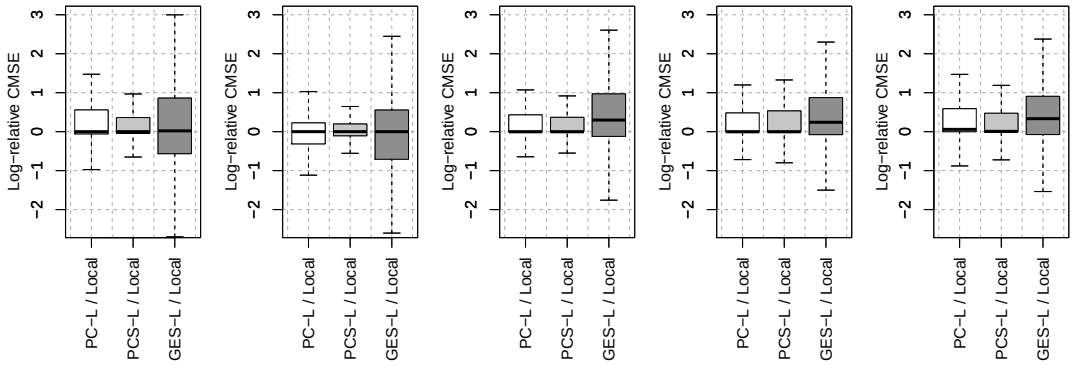
distances of GES-based methods are not the best, it is possible that the estimated effect pairs of GES-based methods do not contain the true one.

4.4. Evaluation with the DREAM4 Data

In this section, we apply our method on synthetic gene expression data sets from the DREAM4 *in silico* challenge. The DREAM4 data provides five data sets with both interventional and observational data simulated from five possibly cyclic gene regulatory networks with 100 genes. The detailed descriptions of the data sets can be found at `http://dreamchallenges.org/project/dream4-in-silico-network-challenge/`. In our experiments, we only used observational data in each data set, which includes 201 observations. We also took the logarithm of each data set and then normalized each data set such that each gene has a sample mean 0 and a sample variance 1.



(a) Data set 1 (b) Data set 2 (c) Data set 3 (d) Data set 4 (e) Data set 5



(f) Data set 1 (g) Data set 2 (h) Data set 3 (i) Data set 4 (j) Data set 5

Figure 7: Experimental results on DREAM4 data sets.

Since the set of possible pairs of total and direct causal effects estimated by the IDA-based method contains some nonexistent pairs, in this section we only apply the proposed local method Algorithm 7. The required chain components were learned by Algorithm 3,

PC, stable PC, and GES, while with later three methods, we first learned an essential graph and then obtained the chain components by reading the induced subgraphs of the learned essential graph.

For each data set, we estimated the set of possible pairs of total and direct causal effects of all pairs of treatment and response, that is, $100 \times 99 = 9900$ treatment-response pairs. To evaluate and compare different methods, we estimated the true pair of total and direct causal effects for each treatment-response pair based on the provided regulatory network, and compared it with the estimated possible set. Two metrics, namely the minimum distance and the CMSE, were used to evaluate the performance of different methods. Note that, since the underlying regulatory networks may contain cycles, the perfect oracles are not available. Thus, the set distance was not considered in the experiments.

Figure 7 reports the log-relative minimum distances and CMSEs of PC-L, PCS-L and GES-L with respect to the fully local method, respectively, on five DREAM4 data sets. The log-relative minimum distance of a method is defined as the log of the minimum distance of the method minus the log of the minimum distance of the fully local method. The log-relative CMSE is defined similarly. In terms of both metrics, on the data sets 1, 3, 4, 5, the fully local method is not worse than PC-L and PCS-L in approximately 70%-75% cases, and on the data sets 3, 4, 5 the fully local method is better than GES-L in approximately 70%-75% cases since the median of 'GES-L / Local' is above zero. On the data set 2, the fully local method does not outperform others. The reason could be that the chain components learned by Algorithm 3 is not accurate enough, since the original network contains directed cycles.

5. Conclusions and Discussions

In this paper, we propose a global learning approach and a local learning approach for finding all possible pairs of the total and direct effects of a given treatment on a given response. For Gaussian graphical models, we first find all possible pairs of parents of treatment and response and then evaluate all pairs of the total and direct effects. We discuss the global learning algorithm and its improved version which first learn an whole essential graph from observed data, then enumerate all possible causal networks in the Markov equivalence class represented by the essential graph, and finally find a pair of the total and direct effects for each DAG in the class. We further propose the local learning approach in which we first learn locally the chain components containing the treatment and the response and then locally enumerate all possible pairs of the parent set of the treatment and the parent set of the response in the Markov equivalence class. To check the validity of any orientation configuration of the neighbors of the treatment and the response, we introduce a local criterion that depends only on the subgraphs of the learned chain components over the neighbors of the treatment and the response.

In our approaches, we require the faithfulness assumption and assume that there are no hidden variables or selection biases. When there exist hidden variables and selection biases, a promising future work is to study the local structure learning and the causal effect estimation under the framework of ancestral graph Markov models (Richardson and Spirtes, 2002; Ali et al., 2005; Zhang, 2008; Malinsky and Spirtes, 2017). The Gaussian assumption is another requirement for our approaches. The Gaussian model is only needed for estimating

total and direct effects via OLS method, but not necessary for the local structure learning. When the variables of interest are not Gaussian, we need to further discuss the definitions of total and direct effects and their estimations.

Acknowledgments

We would like to thank the editor and the three referees for their helpful comments and suggestions that greatly improved the previous version of this paper. This research was supported by National Key R&D Program of China (2018YFB1004300), 973 Program of China (2015CB856000), NSFC (11671020, 11771028, 91630314).

Appendix A. Algorithms

In this appendix, we describe the IAMB algorithm and Meek's orientation algorithm that are used in Algorithm 3. Given a variable Z, Algorithm 8 finds a Markov blanket of a given response variable T; this algorithm can be found in Tsamardinos et al. (2003).

Algorithm 8 The IAMB algorithm (Tsamardinos et al., 2003)

Input: Treatment T, Data $\mathcal{D}$ of the variable set V.
Output: A Market blanket of T.

Phase I (forward)

Set $CMB = \emptyset$.
while CMB has changed, **do**
  Find the variable X in $V - CMB - \{T\}$ that maximizes $f(X; T|CMB)$, where $f(X; T|CMB)$ is the Mutual Information between X and T given CMB.
  if $X \not\perp\!\!\!\perp T|CMB$ **then**
    Add X to CMB.
  end if
end while

Phase II (backwards)

Remove from CMB all variables X, for which $X \perp\!\!\!\perp T|(CMB - \{X\})$.
return CMB.

Algorithm 9 orients some undirected edges in a graph S to directed edges using Meeks' rules (Meek, 1995).

Appendix B. Detailed Proofs

Below, we prove Corollary 4.

Proof Let V be the vertex set of the underlying essential graph G^*, *Donelist* be the subset of vertices visited by Algorithm 3 in Step 4, and S be the final graph updated by $G_{MB(Z)}$ for all $Z \in$ *Donelist*. Let $ChComp(X)$ be a subgraph of S and $ChComp(X)$ consists of all edges with at least one vertex in A, where A is set of the vertices that have undirected paths to X. We have that A is a subset of *Donelist* according to Algorithm 3.

Algorithm 9 Meeks's approach used in the Step 8 in Algorithm 3

Input: A graph S and the independence set, denoted by IndSet, which is used to learn S.
Output: An oriented graph S.

1: **while** S has changed, **do**
2: For any subgraph like $a \to b - c$ in S, if $a \perp\!\!\!\perp c | S_{ac}$ in IndSet, and $b \in S_{ac}$, update S by orienting $b \to c$.
3: For any subgraph like $a \to b \to c - a$ in S, update S by orienting $a \to c$.
4: For any subgraph like $a - b$,$a - c \to b$, and $a - d \to b$ in S, if $c \perp\!\!\!\perp d | S_{cd}$ in IndSet, and $a \in S_{cd}$, update S by orienting $a \to b$.
5: **end while**
6: **return** S.

According to Theorem 1 in Wang et al. (2014), We have that all edges adjacent Z are correct in $G_{MB(Z)}$ regardless of their orientations. Therefore, we have that all edges in $ChComp(X)$ are correct regardless of their orientations since all edges in $ChComp(X)$ are connected to A and A is a subset of $Donelist$. According to Theorem 2 in Wang et al. (2014), we have that all v-structures in S are correct and that all v-structures which have at least one parent contained in DoneList are discovered correctly. According to Meeks orientation approach, we know that the directed edges in S are correctly oriented by checking the absence of edges in S (Meek, 1995). Moreover, when $DoneList$ equals V in Algorithm 3, we have that the final graph S is the same as G^*. Furthermore, $ChComp(X)$ consists of the chain component which contains vertex X and the directed edges surrounding the chain component when $DoneList = V$.

When $DoneList \subsetneqq V$, we continue to update S by the local graph $G_{MB(Z)}$ for every $Z \in V \setminus Donelist$, and denote the final graph as S'. Because $ChComp(X)$ is a subgraph of S and all undirected edges in $ChComp(X)$ have been enveloped by directed edges in S, we have that the undirected edges in $ChComp(X)$ will not be oriented to directed edges by the v-structures found in $G_{MB(Z)}$ for all $Z \in V \setminus Donelist$. Moreover, as shown above, we have that S' is the same as G^*, so $ChComp(X)$ has the same undirected and directed edges connected to A as those in G^*. That is, $ChComp(X)$ consists of the chain component which contains vertex X and the directed edges surrounding the chain component in G^*. ■

Before giving the proof of Theorem 5, we give the following Lemma 9.

Lemma 9 *Let G^* be an essential graph with k chain components $\tau_1, \ldots, \tau_k$, G_1 and G_2 be any two DAGs in the Markov equivalence class represented by G^*, and G_{j,τ_i} be the subgraph of G_j over τ_i for any $i = 1, \ldots, k$. Let G_3 be a graph obtained from G_1 by replacing G_{1,τ_i} in G_1 by G_{2,τ_i}. We have that G_3 is a DAG in Markov equivalence class represented by G^*.*

Proof Because G_1 and G_2 are equivalent DAGs, they have the same skeleton; thus G_3 has the same skeleton as G_1. We just need to show that G_3 has the same V-structures as G_1 and there is no directed cycle in G_3. First, because G_{3,τ_i} is the same as G_{2,τ_i}, and they have the same directed edges surrounding them in G_3 and G_2, respectively, we have that G_3 and G_2 have the same V-structures that contain at least one edge in G_{3,τ_i}. And because G_3 is the same as G_1 except the edges in τ_i, we have that G_3 and G_1 have the same V-structures

that do not contain edges in G_{3,τ_i}. Consequently, G_3 has the same V-structures as G_1 since G_1 and G_2 have the same V-structures.

Suppose that there is a directed cycle in G_3, denoted by $cycle_1$. Since both G_1 and G_{3,τ_i} are DAGs, we have that the subgraph obtained by removing the edges in G_{3,τ_i} from G_3 is also a DAG. Therefore, in the directed cycle $cycle_1$, there exist some directed edges in G_{3,τ_i} and some directed edges out of G_{3,τ_i}. Changing the directed edges of $cycle_1$ that are in chain components in G^* into undirected edges, we obtain a cycle $cycle_2$. Clearly, the cycle $cycle_2$ is partial directed in G^*. Since G^* is a chain graph without partial directed cycles, we have that G_3 is a DAG. ■

Proof of Theorem 5 We only need to prove that $\{S(Y) \to Y, S(X) \to X\}$ is valid if $S(Y) \to Y$ and $S(X) \to X$ are valid, separately, in an essential graph G^*.

Let the chain component containing X and Y be τ_1 and τ_2, respectively. Because both $S(Y) \to Y$ and $S(X) \to X$ are valid, there are two DAGs in the class represented by G^*, denoted by G_1 and G_2, where G_1 has the same directed edges connected to X as $G^*_{S(X)\to X}$ and G_2 has the same directed edges connected to Y as $G^*_{S(Y)\to Y}$. Let G_3 be a graph obtained from G_1 by replacing G_{1,τ_i} in G_1 by G_{2,τ_i}. According to Lemma 9, G_3 is a DAG in Markov equivalence class represented by G^*. Clearly, G_3 has the configuration $\{S(Y) \to Y, S(X) \to X\}$, and so $\{S(Y) \to Y, S(X) \to X\}$ is valid in G^*. ■

In the remainder of this Appendix, we will prove Theorem 6 which gives the sufficient and necessary local conditions for the validity of an orientation configuration $(S(Y) \to Y, S(X) \to X)$ applied to an essential graph G^*. The proof of the validity of $(S(Y) \to Y, S(X) \to X)$ applied to G^* is shown in the following sequential manner.

We first check the validity of $S(Y) \to Y$ applied to G^* using the local criterion given by Lemma 3. If $S(Y) \to Y$ is not valid, then $(S(Y) \to Y, S(X) \to X)$ is not valid. If $S(Y) \to Y$ is valid, then there exist some Markov equivalent DAGs in G^* that have the same directed edges connected to Y as $G^*_{S(Y)\to Y}$. Next we check the validity of $S(X) \to X$ applied to $G^*_{S(Y)\to Y}$.

Theorem 6 in He and Geng (2008) showed that the Markov equivalent DAGs obtained by applying the orientation $S(Y) \to Y$ to G^* can be represented uniquely by an essential graph, denoted by $G^{**}_{S(Y)\to Y}$. Let τ be chain components containing X and Y in G^*, $G^{**}_{S(Y)\to Y}$ can be obtained by applying Meek's the following two rules (Meek, 1995) repeatedly to the undirected edges in $G^*_{S(Y)\to Y}$: for any three vertices Z_1, Z_2 and $Z_3 \in \tau$,

(1) (No new v-structures) if $Z_1 \to Z_2 - Z_3 \in G^*_{S(Y)\to Y}$ and Z_1 and Z_3 are not adjacent, then orient $Z_2 - Z_3$ as $Z_2 \to Z_3$;

(2) (No cycle) if $Z_1 \to Z_2 \to Z_3 \in G^*_{S(Y)\to Y}$ and $Z_1 - Z_3 \in G^*_{S(Y)\to Y}$, then orient $Z_1 - Z_3$ as $Z_1 \to Z_3$.

Thus we can check the validity of $S(X) \to X$ applied to $G^{**}_{S(Y)\to Y}$ using Lemma 3 again.

If $S(Y) \to Y$ applied to G^* is valid and in turn $S(X) \to X$ applied to $G^{**}_{S(Y)\to Y}$ is valid, then we have that $(S(Y) \to Y, S(X) \to X)$ applied to G^* is valid. We give a summary in the following lemma.

Lemma 10 *Let $G^*_{S(Y)\to Y}$ and $G^*_{(S(Y)\to Y, S(X)\to X)}$ be the graphs obtained by applying the orientations $S(Y)\to Y$ and $(S(Y)\to Y, S(X)\to X)$ to an essential graph G^*, respectively, and let $G^{**}_{S(Y)\to Y}$ be the graph obtained by applying Meek's the two rules repeatedly to the undirected edges in $G^*_{S(Y)\to Y}$. We have that the orientation $(S(Y)\to Y, S(X)\to X)$ applied to G^* is valid if*

1. *the orientation $(S(Y)\to Y, S(X)\to X)$ makes no new v-structures in $G^*_{(S(Y)\to Y, S(X)\to X)}$, and*
2. *$S(X)$ is a subset of the neighbor set of X in $G^{**}_{S(Y)\to Y}$.*

Proof Since the orientation $(S(Y)\to Y, S(X)\to X)$ makes no new v-structures in $G^*_{(S(Y)\to Y, S(X)\to X)}$, we have that $S(Y)\to Y$ makes no new v-structures in $G^*_{S(Y)\to Y}$, and thus $S(Y)\to Y$ is valid by Lemma 3. According to Theorem 6 in He and Geng (2008), since $G^{**}_{S(Y)\to Y}$ is an essential graph, $S(X)$ is a subset of the neighbor set of X in $G^{**}_{S(Y)\to Y}$ and the orientation $S(X)\to X$ makes no new v-structures, we have that $S(X)\to X$ applied to $G^{**}_{S(Y)\to Y}$ is valid. Thus there exists a DAG, say D, in the DAG class represented by $G^{**}_{S(Y)\to Y}$, which has the same directed edges connected to X or Y as $G^*_{(S(Y)\to Y, S(X)\to X)}$. Since the class represented by $G^{**}_{S(Y)\to Y}$ is a subset of the class represented by G^*, we have that D is in the class of G^*, and thus the orientation $(S(Y)\to Y, S(X)\to X)$ is valid for G^*. ■

To check the validity of an orientation configuration $(S(Y)\to Y, S(X)\to X)$ for G^*, Theorem 6 presents the local conditions that do not need the essential graph $G^{**}_{S(Y)\to Y}$ obtained by applying Meek's the two rules to $G^*_{S(Y)\to Y}$. Before proving Theorem 6, we show two properties of $G^{**}_{S(Y)\to Y}$ below.

Lemma 11 *Let $S(Y)\to Y$ be a valid orientation configuration containing $X\in S(Y)$ for an essential graph G^*, τ be the chain component containing Y and $G^{**}_{S(Y)\to Y}$ be the essential graph obtained by applying Meek's the two rules repeatedly to $G^*_{S(Y)\to Y}$. We have*

1. *a directed edge $X\to Z$ appears in $G^{**}_{S(Y)\to Y}$ but not in $G^*_{S(Y)\to Y}$ if and only if there is a directed path from Y to Z in the induced subgraph of $G^{**}_{S(Y)\to Y}$ over τ; and*
2. *for any neighbor vertex Z of X in G^*, the directed edge $Z\to X$ does not appear in $G^{**}_{S(Y)\to Y}$.*

Proof For the proof of the property 1, let $H = G^*_{S(Y)\to Y}$, and let H_τ be the induced subgraph of H over τ. To obtain $G^{**}_{S(Y)\to Y}$, we apply Meek's these two rules repeatedly to H_τ.

Below using the inductive method, we prove that for any directed edge (say, $U\to V$) in the induced subgraph of $G^{**}_{S(Y)\to Y}$ over τ, we have either $V = Y$ or that there exists a directed path from Y to V. First, for any directed edge in H_τ (say, $U\to V$), we have either $V = Y$ or that Y is a parent of V. Let $U\to V$ be the first edge oriented by one of the above two rules. We have that there is a directed path from Y to V. That is, there is a directed path from Y to the head of the new oriented edge. Suppose that there exist directed paths

from Y to each of heads of the first k oriented edges. According to Meek's the two rules, we have that there is a directed path from Y to the head of the $(k+1)th$ oriented edge. Therefore, if a directed edge $X \to Z$ appears in $G^{**}_{S(Y)\to Y}$ but not in $G^{*}_{S(Y)\to Y}$, then there is a directed path from Y to Z in $G^{**}_{S(Y)\to Y}$.

Since $G^{**}_{S(Y)\to Y}$ is an essential graph (He and Geng, 2008), there is no partial directed cycle in $G^{**}_{S(Y)\to Y}$. If there is a directed path from Y to Z in $G^{**}_{S(Y)\to Y}$, then $X \to Y$ appears in $G^{**}_{S(Y)\to Y}$, otherwise there is a partial directed cycle from X to itself.

For the proof of the property 2, from the proof of the property 1, in $G^{**}_{S(Y)\to Y}$, there exists a directed path from Y to the head of each directed edge that is oriented by Meek's these two rules. Suppose that $Z \to X$ appears in $G^{**}_{S(Y)\to Y}$. Then there exists a directed path from Y to X in $G^{**}_{S(Y)\to Y}$. That is, the essential graph $G^{**}_{S(Y)\to Y}$ has a directed cycle $X \to Y \to \cdots \to X$ in the essential graph $G^{**}_{S(Y)\to Y}$. This supposition leads to a contradiction. ■

Below we show the proof of Theorem 6.

Proof of Theorem 6 First, we show the necessity. That is, the three conditions in Theorem 6 hold if the orientation configuration $(S(Y) \to Y, S(X) \to X)$ is valid for G^*. According to Lemma 3, the conditions 1 and 2 hold obviously if the orientation configuration $(S(Y) \to Y, S(X) \to X)$ is valid. Consider any partial directed cycle in $G^{*}_{S(Y)\to Y, S(X)\to X}$ (say, $X \to Y \to Z_1 - \cdots - Z_k - Z \to X$), where $Z_1, \cdots, Z_k$ are k distinct vertices. Suppose that the condition 3 does not hold. Then we have that Y is not adjacent to $Z_2, \cdots, Z_k, Z$ in $G^{*}_{(S(Y)\to Y, S(X)\to X)}$. Let $Z_{(1)} = Z_1$, $Z_{(m)} = Z$ and $Z_{(1)} - Z_{(2)} - \cdots - Z_{(m)}$ be the shortest sub-path from Z_1 to Z, in which $\{Z_{(1)}, \cdots, Z_{(m)}\}$ is a subset of $\{Z_1, \cdots, Z_k, Z\}$ and $Z_{(i)}$ is not adjacent to $Z_{(j)}$ in $G^{*}_{(S(Y)\to Y, S(X)\to X)}$ for any $1 \le i, j \le m$ and $j - i > 1$. Since the orientation configuration $(S(Y) \to Y, S(X) \to X)$ is valid, there exists at least a DAG (say D) that has the same directed edges connected to X or Y and the same v-structures as $G^{*}_{(S(Y)\to Y, S(X)\to X)}$. Therefore $Y \to Z_{(1)} \to Z_{(2)} \to \cdots \to Z_{(m)}$ appears in D since $Y \to Z_{(1)}$, Y is not adjacent to $Z_{(2)}$, and $Z_{(i)}$ is not adjacent to $Z_{(i+2)}$ for any $0 \le i \le m-2$ in $G^{*}_{(S(Y)\to Y, S(X)\to X)}$. That is, the DAG D has a directed cycle from X to itself. This leads to a contradiction. Therefore, Y is adjacent to one of $Z_2, \cdots, Z_k, Z$, and thus the condition 3 of Theorem 5 holds.

Now, we prove the sufficiency. That is, an orientation configuration $(S(Y) \to Y, S(X) \to X)$ is valid if the three conditions in Theorem 6 hold. According to Lemma 10, we just need to show that the conditions in Lemma 10 hold. The condition 1 in Lemma 10 holds obviously. If condition 2 in Lemma 10 would not hold. That is, $S(X)$ is not a subset of neighbor set of X in $G^{**}_{S(Y)\to Y}$, then $S(X)$ contains at least one vertex that is not a neighbor of X in $G^{**}_{S(Y)\to Y}$. Let $Z \in S(X)$ and let Z be not a neighbor of X in $G^{**}_{S(Y)\to Y}$. Moreover, since Z is not a neighbor of X in G^*, according to (2) of Lemma 11, we have that Z is not a parent of X in $G^{**}_{S(Y)\to Y}$. Consequently, $X \to Z$ appears in $G^{**}_{S(Y)\to Y}$. By (1) of Lemma 11, there exists a direct path from Y to Z in $G^{**}_{S(Y)\to Y}$, denoted by $Y \to Z_1 \to \cdots \to Z_k \to Z$. Below, we will show that either condition 2 or condition 3 in Theorem 6 does not hold.

Notice that $Z \in S(X)$. So $Z \to X$ occurs in $G^*_{S(Y)\to Y, S(X)\to X}$. If Z is a neighbor of Y in G^*, then $X \to Y \to Z \to X$ in $G^*_{S(Y)\to Y, S(X)\to X}$ forms a directed cycle, and thus condition 2 in Theorem 6 does not hold.

If Z is not a neighbor of Y in G^*, set n be the maximum number from 1 to k such that Z_n is a neighbor of Y in G^*. Since there is a directed path from Y to Z_n in $G^{**}_{S(Y)\to Y}$, we have $Z_n \notin S(Y)$ and that $Y \to Z_n$ appears in $G^*_{S(Y)\to Y}$. Therefore, $X \to Y \to Z_n - \cdots - Z_k - Z \to X$ is a partial directed cycle in $G^*_{S(Y)\to Y, S(X)\to X}$. Let $Z_{(1)} = Z_n$, $Z_{(m)} = Z$ and $Z_{(1)} - \cdots - Z_{(m)}$ be the shortest sub-path of $Z_n - \cdots - Z_k - Z$. Consider the undirected cycle $X - Y - Z_{(1)} - \cdots - Z_{(m)} - X$ in G^*. Because there are no undirected edges among $Y, Z_{(1)}, \cdots, Z_{(m)}$ except those in the cycle, we have that X must be adjacent to each vertex in $\{Y, Z_{(1)}, \cdots, Z_{(m)}\}$. Otherwise, the chain component containing X and Y is not a chordal graph. Considering the partial directed cycle $X \to Y \to Z_{(1)} - \cdots - Z_{(m)} \to X$ in $G^*_{S(Y)\to Y, S(X)\to X}$, we have that all vertices are adjacent to X and that only two vertices (X and $Z_{(1)}$) are adjacent to Y. Thus the condition 3 in Theorem 6 does not hold.

Finally, we have that the conditions in Lemma 10 hold from the conditions in Theorem 6. Thus the orientation configuration $(S(Y) \to Y, S(X) \to X)$ is valid. ■

Appendix C. Implementation Details and Additional Experiments

In this section, we discuss how to efficiently compute the set distance between two sets, and present more simulations to illustrate the estimations of the combinations of direct, indirect and total effects.

C.1. The Computation of Set Distance

Recall that the definition of the set distance between two sets S_1 and S_2 is

$$\text{setDist}(S_2, S_1) = \text{setDist}(S_1, S_2) = \min_{f\in\mathcal{F}} \sum_{s_1\in S_1} \|s_1 - f(s_1)\|_2, \tag{4}$$

where S_1 and S_2 are two finite subsets of $\mathbb{R}^2$ with $|S_1| \geq |S_2|$, and $\mathcal{F}$ is the set of all surjections from S_1 to S_2. We next show that the above combinatorial minimization problem can be transformed to a maximum weight matching problem of a bipartite graph.

Without loss of generality, we can assume that $S_1 = \{s_{1,1}, s_{1,2}, ..., s_{1,m}\}$ and $S_2 = \{s_{2,1}, s_{2,2}, ..., s_{2,n}\}$. If $m > n$, we add some virtual points $\{s_{2,n+1}, s_{2,n+2}, ..., s_{2,m}\}$ to S_2 and construct a new set $S_2^* = S_2 \cup \{s_{2,n+1}, s_{2,n+2}, ..., s_{2,m}\}$, and define the distance between $s_{1,i} \in S_1$ and a virtual point $s_{2,n+j}$ as

$$d(s_{1,i}, s_{2,n+j}) = \min_{k=1,2,...,n} \|s_{1,i} - s_{2,k}\|_2.$$

Note that, the above definition may not be a 'real' distance. That is, one may not find a point on $\mathbb{R}^2$ such that the Euclidean distances between the point and all $s_{1,i}$'s equal to $d(s_{1,i}, s_{2,n+j})$ defined above. That is why we call $s_{2,n+j}$'s virtual points. If $m = n$, then we simply let $S_2^* = S_2$.

Let $M = \{m_{i,j}\}_{m\times m}$ denote the matrix of distance between points in S_1 and S_2^*, where $m_{i,j} = \|s_{1,i} - s_{2,j}\|_2$ if $j \leq n$ and $m_{i,j} = d(s_{1,i}, s_{2,j})$ if $j > n$. Next, we construct a bipartite

graph B with vertices $S_1 \cup S_2^*$ and edges $s_{1,i} - s_{2,j}$ for all $i, j = 1, 2, ..., m$, and the weight of $s_{1,i} - s_{2,j}$ is set to be $-m_{i,j}$. Then we can show that,

Proposition 12 *Let w be the maximum weight of all matchings of B, then the solution of minimization problem (4) is $-w$.*

Proof The case where $m = n$ is simple, thus we assume $m > n$ in the following. Let B' be the bipartite graph resulted from multiplying each edge weight in B by -1. It is suffices to prove that the minimum weight of all matchings of B, denoted by w', equals to the solution of minimization problem (4).

Let g be a matching of B', then g induce a bijection from S_1 to S_2^*. That is, $g(s_{1,i}) = s_{2,j}$ if $s_{1,i}$ and $s_{2,j}$ are connected in the minimum weight matching. Now, consider the inverse image of virtual points in S_2, that is, $\{g^{-1}(s_{2,j})\}_{j>n}$. For any $j > n$, based on the definition of $d(g^{-1}(s_{2,j}), s_{2,j})$, there is a point in S_2, or equivalently $\exists k(s_{2,j}) < n$ such that $d(g^{-1}(s_{2,j}), s_{2,j}) = \|g^{-1}(s_{2,j}) - s_{2,k(s_{2,j})}\|_2$. If we define a map $f : S_1 \to S_2$ such that $f(s_{1,i}) = g(s_{1,i})$ if $s_{1,i} \in S_1 \setminus \{g^{-1}(s_{2,j})\}_{j>n}$, and $f(s_{1,i}) = s_{2,k(g(s_{1,i}))}$ otherwise, then f is a surjection from S_1 to S_2 and the weight of the matching g equals to $\sum_{s_1 \in S_1} \|s_1 - f(s_1)\|_2$. Therefore, $w' \geq \sum_{s_1 \in S_1} \|s_1 - f(s_1)\|_2$.

Conversely, let

$$f^* = \arg\min_{f \in \mathcal{F}} \quad \sum_{s_1 \in S_1} \|s_1 - f(s_1)\|_2 \ ,$$

and let $w = \sum_{s_1 \in S_1} \|s_1 - f^*(s_1)\|_2$, we will construct a matching of B' (or a bijection g from S_1 to S_2^*) such that the weight of the matching is w. Let $L_2 = \{s_{2,j} \in S_2 \mid |f^{*\,-1}(s_{2,j})| > 1\}$. for any $s_{2,j} \in L_2$, we claim that there at most one point in $f^{*\,-1}(s_{2,j})$, denoted by $t(s_{2,j})$, such that

$$\|t(s_{2,j}) - s_{2,j}\|_2 > \min_{k=1,2,...,n} \|t(s_{2,j}) - s_{2,k}\|_2.$$

In fact, if there is another point $t' \in f^{*\,-1}(s_{2,j})$ satisfies the above condition, we can construct another surjection f^{**}, by setting

$$f^{**}(t') = \arg\min_{s_{2,k}, k=1,2...,n} \|t' - s_{2,k}\|_2,$$

and keeping $f^{**}(s_{1,i}) = f^*(s_{1,i})$ if $s_{1,i} \neq t'$. It is easy to verify that $\sum_{s_1 \in S_1} \|s_1 - f^{**}(s_1)\|_2 < w$, which is contradicted to the assumption of f^*. Therefore, to construct the desired bijection g, we only have to map points in $f^{*\,-1}(s_{2,j}) \setminus \{t'\}$ to arbitrary different virtual points who have not been assigned to any point in S_1. Repeat the above procedure for all points in L_2, we will have a matching g such that the weight of the matching is w. Hence, $w' \leq w \leq \sum_{s_1 \in S_1} \|s_1 - f(s_1)\|_2$. This completes the proof. ■

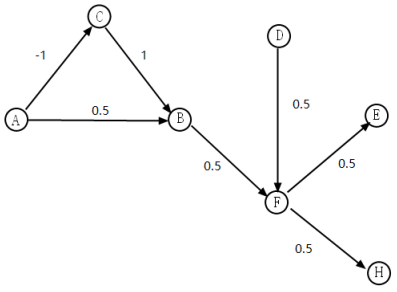
Thus, with the help of Proposition 12, we can easily compute the set distance between two sets.

C.2. Estimating Direct, Indirect and Total Effects

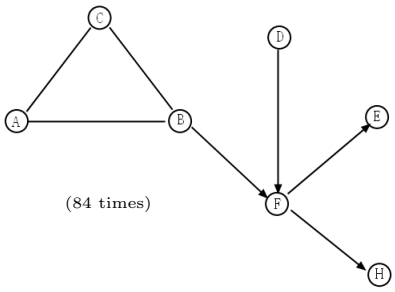
In this section, we use another example to discuss the estimations of direct, indirect and total causal effects and evaluate Algorithm 7 in the case there are errors in structure learning using simulation data. Indirect effects are important to understand the causal mechanism of interest. Given a Gaussian graphical model of a DAG G, X and Y are two distinct variables in G, and the indirect effect of X on Y is the difference of the total effect of X on Y and the direct effect of X on Y. Let $IE_{XY}, TE_{XY}, DE_{XY}$ denote the indirect, total, direct effects of X on Y, respectively. We can estimate IE_{XY} as follows,

$$\widehat{IE}_{XY} = \widehat{TE}_{XY} - \widehat{DE}_{XY}, \tag{5}$$

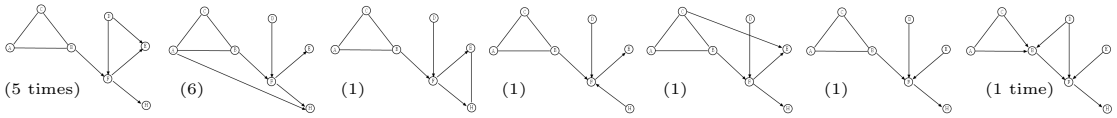
where $\widehat{TE}_{XY}, \widehat{DE}_{XY}$ are estimates of TE_{XY}, DE_{XY}, respectively.



(a) The underlying DAG G



(b) The essential graph G^* of G



(c) The graphs learned wrongly 5,6,1,1,1,1,1 times respectively in 100 simulations

Figure 8: Simulatoins on a Gaussian graphical model of G, the numbers on the edges in G are the coefficients in the regression of a vertex on its parents. The graph G^* is the essential graph of G. The learned graphs and their frequencies (numbers in the brackets) in 100 repetitions are shown.

Consider a DAG G in Figure 8(a), and we generate the samples from a Gaussian graphical model of G according to Equations 1 and 2 except that the coefficients in the regression in Equation 1 are replaced by the numbers on the edges of G. We draw a sample of size 1000 from this Gaussian graphical model, then learn an essential graph with these data, and finally estimate the direct, indirect and total effects of a treatment X on a response Y. Repeating the simulation 100 times, we report the learned graphs in Figures 8(b) and 8(c), and the estimates of causal effects in Figure 9. In 100 repetitions, we learn the underlying essential graph in Figure 8(b) 84 times, and the wrong essential graphs in Figure 8(c) 5,6,1,1,1,1,1 times, respectively.

The estimation of total and direct effects of a treatment on a response only uses their parent set pairs, so the estimation does not depend on the whole structures of the learned

Vertices	A	B	C	D	E	F	H
A	$\star$	99	99	99	99	99	93
B	99	$\star$	99	99	99	92	99
C	99	99	$\star$	99	98	97	99
D	100	99	100	$\star$	95	97	100
E	91	91	91	91	$\star$	93	91
F	97	100	97	100	90	$\star$	90
H	92	92	92	92	92	99	$\star$

Table 2: The numbers of times that the parent set pairs are learned correctly in 100 repetitions for every two distinct vertices in $\{A, B, C, D, E, F, H\}$.

essential graphs in Figures 8(b) and 8(c). In Table 2, we show the numbers of times in 100 repetitions that the correct parent set pairs in the correct essential graph in Figure 8(b) are learned for every two distinct vertices as treatment and response. Although the correct essential graph is learned only 84 times in 100 repetitions, the correct parent set pairs are learned more than 90 times in 100 repetitions. Below we select a specified treatment A and a specified response B to show the sensitivity of the estimates of total, direct and indirect effects to the wrongly learned essential graphs.

We discuss in detail the estimates of the direct, undirect and total effects of a specified treatment A on a specified response B. In this experiment, the true direct, undirect and total effects of A on B are 0.5, -1, and -0.5, respectively. For the underlying essential graph in Figure 8(b), the true effect triples have four combinations of direct, undirect and total effects: $(-0.5, 0, -0.5), (0, 0, 0), (0.5, 0, 0.5)$, and $(0.5, -1, -0.5)$, that are unidentifiable by the underlying statistical distribution but correctly include the true effect combination $(0.5, -1, -0.5)$. We can see that among these wrongly learned essential graphs in Figure 8(c), only the last essential graph has a local subgraph over vertices (A, B, C) different from the underlying essential graph. Thus we can also obtain the correct parent set pairs from the wrongly learned essential graphs except for the last essential graph. For the last learned essential graph, we have two parent set pairs: ($\emptyset$, $\emptyset$) and ($\{C\}$, $\emptyset$), both of which are included in the four parent set pairs of the correct essential graph. Thus, based on the last learned essential graph and the distribution over A, B, C, we can recover two of four possible combinations of causal effects, including $(0.5, 0, 0.5)$ and $(0.5, -1, -0.5)$. That is, in this experiment, regardless of the bias in the estimates of causal effects, we can learn the true combination $(0.5, -1, -0.5)$ in all of 100 repetitions, and recover correctly four possible combinations in 99 of 100 repetitions.

The estimates of the combinations of direct, indirect, and total causal effects of A on B, denoted by $(\widehat{DE}_{AB}, \widehat{IE}_{AB}, \widehat{TE}_{AB})$, are shown in Figure 9. We give a three dimensional scatter plot of all direct, indirect and total causal effects in the left top of Figure 9. We can see that all scatters are distributed in a plane around four points (green stars in the plot), which represent the four possible combinations of direct, indirect, and total effects of A on B. In Figure 9, we also provide three marginal scatter plots of $(\widehat{DE}_{AB}, \widehat{TE}_{AB})$, $(\widehat{IE}_{AB}, \widehat{TE}_{AB})$, and $(\widehat{DE}_{AB}, \widehat{IE}_{AB})$. For each possible combination of causal effects, the mean $(\hat{\mu})$ and the standard deviation $(\hat{\sigma})$ of the corresponding estimates are calculated and

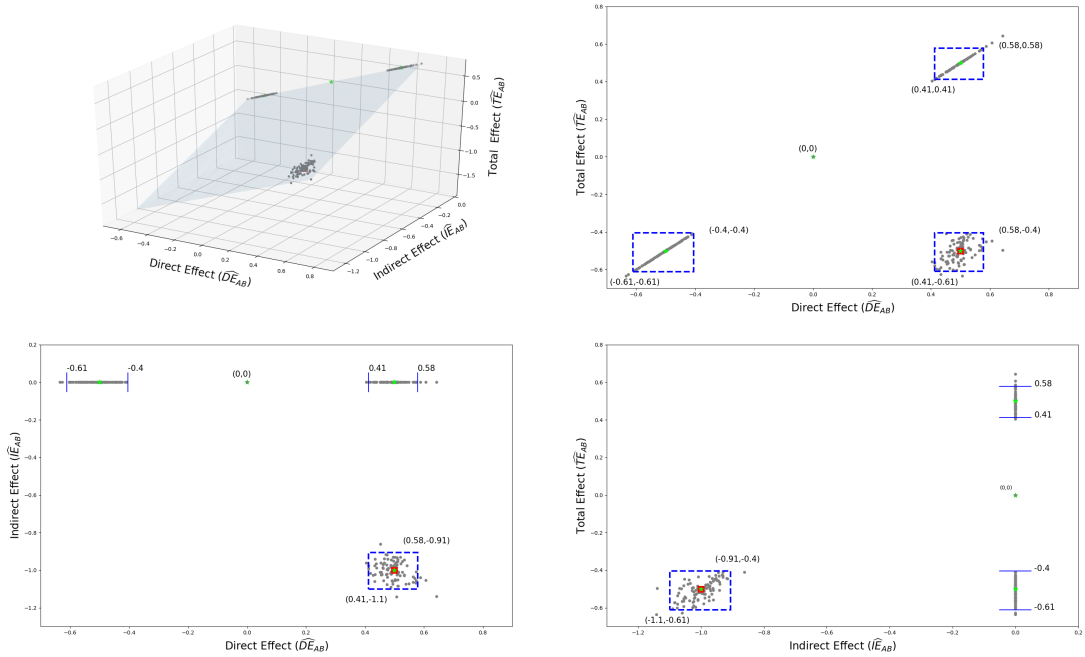


Figure 9: The scatter plots of the estimates of causal effects of A on B in 100 repetitions. The estimates (gray points) and the corresponding interval (mean $\pm$ double standard deviation, blue rectangle or line), the true causal effects (red box), the possible unidentifiable causal effects (green star) are shown.

an estimate interval $\hat{\mu} \pm 2\hat{\sigma}$ is given by blue rectangles or lines. We can see that these intervals are quite small and centered around the possible unidentifiable causal effects.

References

Ayesha R Ali, Thomas S Richardson, Peter Spirtes, and Jiji Zhang. Towards characterizing Markov equivalence classes for directed acyclic graphs with latent variables. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pages 10–17. AUAI press, 2005.

Steen A. Andersson, David Madigan, and Michael D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *The Annals of Statistics*, 25(2):505–541, 04 1997.

Ingo A. Beinlich, H. J. Suermondt, R. Martin Chavez, and Gregory F. Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *AIME 89*, pages 247–256. Springer Berlin Heidelberg, 1989.

Jean R. S. Blair and Barry Peyton. An introduction to chordal graphs and clique trees. In *Graph Theory and Sparse Matrix Computation*, pages 1–29, New York, NY, 1993. Springer New York.

Zhihong Cai, Manabu Kuroki, Judea Pearl, and Jin Tian. Bounds on direct effects in the presence of confounded intermediate variables. *Biometrics*, 64(3):695–701, 2008.

David M. Chickering. Learning equivalence classes of Bayesian-network structures. *Journal of Machine Learning Research*, 2(Feb):445–498, 2002a.

David M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, 3(Nov):507–554, 2002b.

Zhuangyan Fang and Yangbo He. IDA with background knowledge. In *Proceedings of the Thirty-sixth Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020.

Sander Greenland, Judea Pearl, and James M. Robins. Causal diagrams for epidemiologic research. *Epidemiology*, 10:37–48, 1999.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13(Aug):2409–2464, 2012.

Yangbo He and Zhi Geng. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov):2523–2547, 2008.

Yangbo He, Jinzhu Jia, and Bin Yu. Counting and exploring sizes of Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 16:2589–2609, 2015.

David Heckerman, Dan Geiger, and David M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

Paul W. Holland. Statistics and causal inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986.

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.

Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1?2):83–97, 1955.

Steffen L. Lauritzen. Causal inference from graphical models. In *Complex Stochastic Systems*, pages 63–107. Chapman and Hall/CRC Press, 1999.

Yue Liu, Zhuangyan Fang, Yangbo He, and Zhi Geng. Collapsible IDA: Collapsing parental sets for locally estimating possible causal effects. In *Proceedings of the Thirty-sixth Conference on Uncertainty in Artificial Intelligence*. PMLR, 2020.

Marloes H. Maathuis, Markus Kalisch, and Peter Bühlmann. Estimating high-dimensional intervention effects from observational data. *The Annals of Statistics*, 37(6A):3133–3164, 12 2009.

Daniel Malinsky and Peter Spirtes. Estimating bounds on causal effects in high-dimensional and possibly confounded systems. *International Journal of Approximate Reasoning*, 88: 371 – 384, 2017.

Christopher Meek. Causal inference and causal explanation with background knowledge. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 403–410. Morgan Kaufmann Publishers Inc., 1995.

James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

Preetam Nandy, Marloes H. Maathuis, and Thomas S. Richardson. Estimating the effect of joint interventions from observational data in sparse high-dimensional settings. *The Annals of Statistics*, 45(2):647–674, 04 2017.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1988.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.

Judea Pearl. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, pages 411–420. Morgan Kaufmann Publishers Inc., 2001.

Emilija Perković, Markus Kalisch, and Marloes H Maathuis. Interpreting and using CPDAGs with background knowledge. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*. AUAI press, 2017.

Thomas Richardson and Peter Spirtes. Ancestral graph Markov models. *The Annals of Statistics*, 30(4):962–1030, 08 2002.

Paul Shannon. *DREAM4: Synthetic Expression Data for Gene Regulatory Network Inference from the 2009 DREAM4 challenge*, 2019. R package version 1.20.0.

Arvid Sjölander. Bounds on natural direct effects in the presence of confounded intermediate variables. *Statistics in Medicine*, 28(4):558–571, 2009.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, Prediction, and Search*. MIT Press, second edition, 2000.

Ioannis Tsamardinos, Constantin F. Aliferis, and Alexander R. Statnikov. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, pages 376–381. AAAI Press, 2003.

Ioannis Tsamardinos, Laura E. Brown, and Constantin F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65(1):31–78, Oct 2006.

Konstantinos Tsirlis, Vincenzo Lagani, Sofia Triantafillou, and Ioannis Tsamardinos. On scoring maximal ancestral graphs with the max-min hill climbing algorithm. *International Journal of Approximate Reasoning*, 102:74 – 85, 2018.

Thomas Verma and Judea Pearl. Equivalence and synthesis of causal models. In *Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence*, pages 220–227. Elsevier Science Inc., 1990.

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.

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