Gradient-Based Local Causal Structure Learning

Jiaxuan Liang, Jun Wang, Guoxian Yu, *Member, IEEE*, Carlotta Domeniconi, Xiangliang Zhang, *Senior Member, IEEE*, and Maozu Guo

***Abstract*—Finding the causal structure from a set of variables given observational data is a crucial task in many scientific areas. Most algorithms focus on discovering the global causal graph but few efforts have been made toward the local causal structure (LCS), which is of wide practical significance and easier to obtain. LCS learning faces the challenges of neighborhood determination and edge orientation. Available LCS algorithms build on conditional independence (CI) tests, they suffer the poor accuracy due to noises, various data generation mechanisms, and small-size samples of real-world applications, where CI tests do not work. In addition, they can only find the Markov equivalence class, leaving some edges undirected. In this article, we propose a GradieNt-based LCS learning approach (GraN-LCS) to determine neighbors and orient edges simultaneously in a gradient-descent way, and, thus, to explore LCS more accurately. GraN-LCS formulates the causal graph search as minimizing an acyclicity regularized score function, which can be optimized by efficient gradient-based solvers. GraN-LCS constructs a multilayer perceptron (MLP) to simultaneously fit all other variables with respect to a target variable and defines an acyclicity-constrained local recovery loss to promote the exploration of local graphs and to find out direct causes and effects of the target variable. To improve the efficacy, it applies preliminary neighborhood selection (PNS) to sketch the raw causal structure and further incorporates an l_1-norm-based feature selection on the first layer of MLP to reduce the scale of candidate variables and to pursue sparse weight matrix. GraN-LCS finally outputs LCS based on the sparse weighted adjacency matrix learned from MLPs. We conduct experiments on both synthetic and real-world datasets and verify its efficacy by comparing against state-of-the-art baselines. A detailed ablation study investigates the impact of key components of GraN-LCS and the results prove their contribution.**

***Index Terms*—Acyclicity, causal structural learning, direct causes and effects, feature selection, gradient-based search.**

I. Introduction

CAUSAL structure learning aims to infer the causal relationship among variables from observational data and then generate a causal graph, that is, a directed acyclic graph (DAG). Unlike correlation study, causality analysis reveals the causal mechanism of data generation. In many areas, such as biology [1], economics [2], fairness-aware machine learning [3], [4], and many others [5], [6], causal structure learning plays an essential role in uncovering the domain facts and prediction/decision-making process. Some existing deep learning methods, such as graph neural network, extract feature information that can represent the graph structure but cannot obtain causal direction between variables. For its importance, most recent efforts have been made toward exploring the global causal structure (GCS) of all object variables, which apply regression-based CI tests [7], multilayer perceptron [8], [9], reinforcement learning [10], [11], autoencoder [12], and generative adversarial networks [13] to discover causal structures.

Although it is of great significance to find out the whole causal graph of all variables, in many tasks (i.e., genetics [14], gene regulation network [15], pathway [16]), there maybe no GCS, but several LCSs co-exist in the same system. In addition, it is too time-consuming and even impossible to find the GCS, due to a large number of variables and limited and noisy sample data. A more realistic goal is one that focuses on the local part of the causal graph, which consists of direct causes and direct effects of a given variable. In other words, one can learn the local causal structure (LCS), rather than GCS. Compared with GCS, LCS can be obtained in a more economic way.

There is a very similar study called the Markov blanket (MB) learning that tries to find a minimal set of parents, children, and spouses given a target variable, conditioned on which all other variables are independent of this variable [17]. MB learning plays an essential role in the Bayesian network (BN) skeleton learning and optimal feature selection [18]. Since MBs represent local structures in BN, MB learning can be seen as a subproblem of BN structure learning, but we are interested in the local structure of the network w.r.t. one variable, instead of the entire network. MB can reveal the underlying causal mechanisms, so MB algorithms can be used for causal feature selection, which shows superior performance compared with noncausal ones [19]. Many solutions have been proposed

Manuscript received 12 September 2022; revised 10 November 2022 and 27 December 2022; accepted 14 January 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3502101; in part by NSFC under Grant 62031003, Grant 62272276, and Grant 62072380; and in part by the Qilu Scholar Startup Fund of Shandong University. This article was recommended by Associate Editor J. C. W. Lin. *(Corresponding author: Jun Wang.)*

Jiaxuan Liang, Jun Wang, and Guoxian Yu are with the School of Software, Shandong University, Jinan 250101, China, and also with the Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University, Jinan 250101, China (e-mail: jxliang@mail.sdu.edu.cn; kingjun@sdu.edu.cn; gxyu@sdu.edu.cn).

Carlotta Domeniconi is with the Department of Computer Science, George Mason University, Fairfax, VA 22030 USA (e-mail: carlotta@cs.gmu.edu).

Xiangliang Zhang is with the Department of Computer Science, University of Notre Dame, Notre Dame, IN 46556 USA (e-mail: xzhang33@nd.edu).

Maozu Guo is with the Department of Computer Science, Beijing University of Civil Engineering and Architecture, Beijing 100044, China (e-mail: guomaozu@bucea.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2023.3237635.

Digital Object Identifier 10.1109/TCYB.2023.3237635

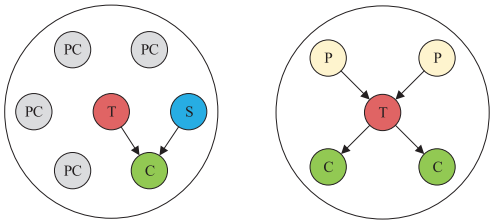


Fig. 1. Difference between canonical MB and the studied local causal graph. For the target variable "T," its MB consists of parent "P," spouse "S," and child "C," "P," and "C" are together referred to as "PC" in MB learning, without the need to distinguish them. In contrast, LCS needs to distinguish the direct causes "P" and effects "C" of "T." Note that "T" and "S" are not adjacent, and "C" has two incoming edges from "T" and "S," so they form a V-structure in the MB.

to explore MBs [20], [21], but they do not distinguish parents from children. In contrast, LCS has to differentiate the direct parents (cause) and children (effect). The differences between MB and LCS are illustrated in Fig. 1. LCS learning faces the following challenges: how to determine neighbors of the target variable from a large number of variables, and how to distinguish parents from children and orient edges. Current LCS algorithms canonically conduct CI tests to obtain MB variables, and orient edges by specific structures [22], [23] or by tracking the CI changes in MB of target variable [24]. Unfortunately, they can only find the Markov equivalence class, leaving some edges undirected. In addition, these CI test-dependent algorithms are very sensitive to the mistakes made by CI tests, which lead to incorrect structure discovery when CI tests do not work. The recent NOTEARS framework [25] and its extensions [9], [26] with the neural network are good solutions to orient edges. A DAG can be induced from a weighted adjacency matrix, which stores coefficients of a structural equation model (SEM). NOTEARS extracts DAG structures from observational data and fits the data generation function by penalizing cycles of any length in the graph. But these NOTEARS-based methods target for exploring GCS. When dealing with the task of searching direct causes and effects, NOTEARS and its extensions cost expensively, as our experiments will show.

In this study, we aim to extend the canonically-used NOTEARS framework [25] to learn LCS. That is, given the target variable, we aim to find out its local causal graph with the parents and children (PC). Two issues need to be addressed when applying NOTEARS to discover LCS. First, searching parents from all variables on high-dimension datasets is too time-consuming. MB-based methods can quickly find out the PC in theory, but they have a poor generalization when facing various data generation mechanisms and they suffer from incorrect CI tests due to noise in reality. Second, NOTEARS targets for GCS, to our knowledge, there is no principal way to extend it to efficiently find out LCS.

We propose a novel algorithm named the GradieNt-based LCS learning approach (GraN-LCS) to address the above issues and to discover LCS. GraN-LCS learns a fully connected multilayer perceptron (MLP) for each variable, and extracts a weighted adjacency matrix from all MLPs. We expect to get parent variables that affect the outputs of each MLP. However, the results may end up with correlated variables, so GraN-LCS incorporates the directed acyclicity constraint of NOTEARS into MLP to determine parents and orient edges between variables simultaneously. To reduce the computational cost, we mask the weighted adjacency matrix of directed acyclicity constraint to make it more effective in local graph search. To further improve the efficiency, GraN-LCS applies preliminary neighborhood selection (PNS) [27] to sketch the initial causal structure and an l_1-norm-based sparse feature selection on the input layer parameter matrix of MLP to reduce the search space and to accelerate the training process. It further defines a local data recovery term on the weighted adjacency matrix of MLPs to guide the gradient going down along the direction of minimizing the local data recovery loss. GraN-LCS finally outputs PC of the target variable as the LCS. The conceptual framework of GraN-LCS is illustrated in Fig. 2. We compare GraN-LCS against gradient-based GCS algorithms (GraN-DAG [8], NOTEARS-MLP [9], DAG-CCNN[28]) and LCS ones (PCD-by-PCD [29], MB-by-MB [30], CMB [24], LCS-FS [22], ELCS [23], and PSL [31]), GraN-LCS manifests a better efficacy than competitive GCS algorithms and outperforms LCS methods on different data generation models. The major contributions of our work are summarized as follows.

1) We propose a novel gradient-based LCS algorithm GraN-LCS that leverages feature selection and neural network to discover LCS in an efficacy way.
2) GraN-LCS introduces a new local data recovery term to induce the weighted adjacency matrix of neural network converging to sparse local causal graph and to explore the causality between variables. It further improves the directed acyclicity constraint by masking weighted adjacency matrix to reduce the computational cost on local causal learning. Unlike existing LCS algorithms, it does not perform model-dependent CI tests and better fits for various data generation models.
3) GraN-LCS is validated on benchmark datasets with different numbers of samples, nodes, and data-generating processes, and it outperforms competitive GCS and LCS algorithms in terms of effectiveness and/or efficiency.

The remainder of this article is organized as follows. Section II discusses related works of MB learning, local and global causality learning, and highlights how our work differs. Section III elaborates on the technical details of the proposed GraN-LCS and its optimization procedure. In Section IV, we present the experimental setup, discuss the results of GraN-LCS and those of competitive algorithms, perform ablation study to investigate the contribution factors of GraN-LCS. Section V summarizes our work and future pursues.

II. Related Works

Our work focuses on learning LCS, it is also closely connected with MB learning and global structural learning. A comprehensive coverage of them is out of the scope of this article, and we just review the most related ones.

MB learning aims to find MB variables of target variable, without differentiating the direct parent and child variables. Current MB learning approaches basically build on CI tests

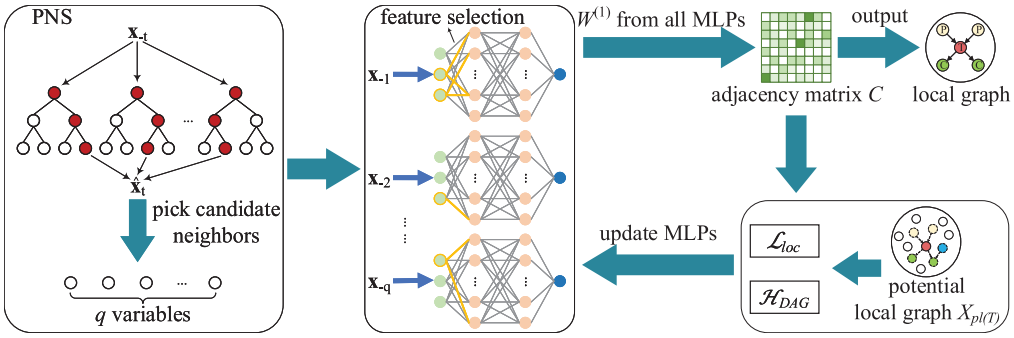


Fig. 2. Conceptual framework of our proposed GraN-LCS. For each variable, GraN-LCS takes PNS sketched neighborhood variables as input (green circles) to train an MLP for reconstructing variable (blue circle) and to find its parents. GraN-LCS incorporates the l_1-norm regularization on the input layer parameter matrix to select variables and speed up the search process (bold yellow line indicates the selected variable), and this matrix quantifies the contribution of each variable to MLP output. Next, GraN-LCS transforms the graph search problem into a score function minimizing problem, further uses local loss $\mathcal{L}_{\text{loc}}$ around the target variable X_T to explore local graph $X_{pl(T)}$, masked acyclicity penalty $\mathcal{H}_{DAG}$ to help orienting edges. Finally, the local graph w.r.t. the target variable in the weighted adjacency matrix is deemed as the LCS.

and heuristic greedy search, which can be roughly divided into nontopology and topology-based ones. The former finds parents, children, and spouses simultaneously by greedily performing CI tests between target variable and other variables, but does not distinguish spouses of target variable from its PC, which needs a large number of samples. GSMB [32] searches candidate MB with two steps of growing and shrinking, which may cause many false positive candidate MBs and, hence, make CI tests less reliable. IAMB [33] deals with this problem by selecting candidate MB variables via heuristic strategy. PFBP [20] proposes a parallel way to improve IAMB and tackle high-dimensional data. To reduce the required sample size, topology-based approaches learn PC and spouses separately, hence, they can identify spouses and part of children when searching spouses, but still have a high-time complexity when the size of MB is large. MMMB [34] is a classical topology-based approach but cannot return correct MB even if the faithfulness assumption is satisfied. PCMB [35] uses a symmetry check to ensure the correctness of the discovered PC set. STMB [21] presents two new strategies of identifying spouses and removing false positives to reduce computational costs. The state-of-the-art MB learning approaches were summarized in a comprehensive review [17]. In brief, MB learning approaches can only obtain a skeleton but cannot tell cause from effect. Nevertheless, they provide feasible strategies for LCS algorithms.

LCS learning aims to discover the direct causes and direct effects of a given target variable. It becomes a worth studying topic in big data era, as exploring GCS from complex data is too time-consuming and even infeasible. Only several algorithms have been proposed for exploring LCS. PCD-by-PCD [29] is the classical work to discover the PCD (parents, children, and some descendants) using max–min PC (MMPC) strategy [34], and it orients directions with V-structure and Meek rules [36]. CMB [24] learns the MB of a target variable, then orients edges by tracking the conditional independence (CI) changes in the MB. LCS-FS [22] and ELCS [23] are two latest LCS algorithms, the former employs a mutual information-based feature selection method for finding PC to accelerate the search process and the latter infers edge directions by finding N-structures to improve the efficiency. PSL [31] discovers local structure of arbitrary depth by different types of found V-structure for orientation. These LCS algorithms find MB of the target variable and further orient the directions to distinguish parents from children by conducting CI tests. These statistical-based algorithms can obtain an LCS in a quite short time but they cannot be flexibly applied to different data generation mechanisms. In some cases, CI tests lead to wrong results, such as PC masking phenomenon [37], result in a low accuracy in real applications. For these reasons, our GraN-LCS takes advantage of MLP, which can approximate any function, to uncover underlying causal relationship and identify local structures. It further makes use of acyclicity constraint to orient edges. In this way, GraN-LCS can adapt to diverse data generation models and deal with a large number of variables.

GCS learning is more widely studied [6], [38], [39]. Early GCS algorithms formulate the directed acyclicity constraint and causality search as a combinatorial problem. As such, these algorithms cannot get the global optimum as the number of variables increases, and most of them adopt heuristic techniques to approximate the DAG [40]. NOTEARS [25] converts the combinatorial problem into a continuous optimization problem and then efficiently solves it by standard numerical algorithms. Particularly, NOTEARS encodes the directed graph as a weighted adjacency matrix, which represents coefficients in a linear SEM, and enforces acyclicity using a constraint that is both efficiently computable and easily differentiable. Although NOTEARS just finds stationary points, the experimental results show that these stationary points are close to global minimal. GAE [12] reconstructs data using graph auto-encoder and extends NOTEARS to nonlinear cases. GraN-DAG [8] fits each variable via an MLP and takes the product of neural network paths as the weight of adjacency matrix. NOTEARS-MLP [9] supposes that if the kth column of the first parameter matrix of MLP consists of all zeros, then variable X_k is independent of this MLP, so it takes l_2-norm of this parameter matrix as weight. DAG-CCNN [28] further introduces convolutional neural networks for feature extraction and nonlinear fitting capability and proposes an acyclicity constraint term to reduce the computational complexity. More related work of GCS can be found in recent survey [6].

However, these GCS algorithms are time-consuming and even infeasible when the network is too large. Even though, they provide a gradient framework to search causality between variables with acyclicity constraints.

In summary, learning LCS is a distinct problem with practical significance. MB-based methods can quickly get the results, but they cannot perform well on various data generation models, and do not distinguish parents from children of target variable. GCS-based methods may waste too much time on insignificant (less focused) local parts. To address these issues, GraN-LCS constructs an MLP for each variable to fit underlying data generation models and find parent variables. With the help of directed acyclicity constraint, GraN-LCS can distinguish parents from children. GraN-LCS further includes a local data recovery term to boost the search of local structure and achieves a better efficacy than these related methods.

III. Proposed Methodology

A. Problem Formulation

Structural causal model (SCM) defined on a set of variables $\mathcal{V} = (X_1, X_2, \ldots, X_d)$ consists of a causal graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represented by a DAG and the structural equations. $\mathcal{E}$ consists of directed and undirected edges. If all edges in a graph are directed/undirected, then this graph is called a directed/undirected graph. For simplicity, we use $X_i \to X_j$ and $X_i - X_j$ to denote directed edges and undirected edges, respectively. A cycle is a path from a variable to itself. A DAG is a directed graph without any cycle. If there is a path from X_i to X_j then X_i has causal impact on X_j. Furthermore, if there is an edge from X_i to X_j then X_i has direct causal impact on X_j, X_i is the parent of X_j. Let $\mathbf{x} = [x_1, x_2, \ldots, x_n]^\top$ be the vector of observational data, $X_{pa(i)}$ denote the set of parents of X_i in $\mathcal{G}$, then the joint distribution $P(X)$ is said to be Markov to $\mathcal{G}$, that is, $P(X) = \prod_{i=1}^{d} P(X_i|X_{pa(i)})$. In addition, $P(X_i|X_{pa(i)}) = P(X_i|X_{-i})$ holds. In this work, we assume that the distribution $P(X)$ is entailed by additive noise model (ANM)

$$X_i = f_i\big(X_{pa(i)}\big) + N_i \tag{1}$$

where f_i is a nonlinear function and N_i is the external noise of X_i. Noises are mutually independent.

LCS algorithms aim to find out direct causes (parents) and effects (children) of the target variable X_T from $\mathcal{V}$ using observational data $\mathbf{x}$.

B. Preliminary

PNS [27]: PNS is a preprocess procedure that gets a set of potential neighbors for each variable, which is achieved by fitting an extremely randomized tree for a variable versus all the other variables and a feature importance score based on the gain of purity is calculated. Only variables that have a feature importance score higher than threshold $\cdot$ mean are kept as candidate PC, where threshold is a predefined parameter and mean is the mean of the feature importance scores of all variables. PNS was applied in [8] and [27] to learn the initial causal structure and proved to be useful.

Theorem 1 (DAG Constraint): An adjacency matrix $U \in \mathbb{R}^{d\times d}$ is a DAG if and only if [25]

$$h(U) = \operatorname{tr}\big(e^{U\circ U}\big) - d = 0 \tag{2}$$

where $\operatorname{tr}(\cdot)$ represents the trace of a matrix. Theorem 1 computes the matrix exponential of Hadamard product of the weighted adjacency matrix U to obtain paths of any length between two variables in the d-variable graph. If the trace is greater than d, there is a circle in the graph.

C. GraN-LCS

We propose a novel algorithm for exploring LCS from $P(X)$. First, we apply the PNS procedure on target variable and get candidate PC set. In this way, the search space of target variable's local graph is reduced to q-dimension. Next, for each variable $X_k (k = 1, 2, \ldots, q)$, we construct an MLP to fit a nonlinear function f_k with input from samples $\mathbf{x}_{-k} \in \mathbb{R}^{n\times q}$, that is, the vector $\mathbf{x}$ with the component k masked as zero. Consider an $L+1$ layers MLP with q input variables, $\{H_1, H_2, \ldots, H_L\}$ hidden nodes and an output node. The weight vectors $W_j^{(1)} = [W_{1,j}^{(1)}, W_{2,j}^{(1)}, \ldots, W_{H_1,j}^{(1)}]^\top (j = 1, 2, \ldots, q)$ connect the jth input variable with hidden layer. Let $W^{(1)} = (W_1^{(1)}, W_2^{(1)}, \ldots, W_q^{(1)}) \in \mathbb{R}^{H_1\times q}$ denote the first layer parameter matrix of neural network. Similarly, we denote the parameters of MLP as $W = (W^{(1)}, W^{(2)}, \ldots, W^{(L)}, W^{(L+1)})$ and parameterize the MLP as follows:

$$\hat{\mathbf{x}}_k = \operatorname{MLP}(\mathbf{x}_{-k}, W_k) \tag{3}$$

where W_k is the parameter set of the kth MLP. $W_{h,j}^{(1)} (h = 1, 2, \ldots, H_1)$ contains all weights between the jth variable and hidden nodes of the first layer, so if $W_{h,j}^{(1)}$ consists of 0 or close to 0, then the jth variable makes a trivial contribution to the output and could be eliminated. In other words, X_j cannot be the parent of X_k.

Given that, we can perform *feature selection* on each column to obtain a sparse matrix and speed up the selection process of parent variables. Different from the preliminary screening of PNS which reduces the scale of the problem, or denoising methods [41], here, we integrate the feature selection into the optimization process and confirm parent variables. For this purpose, we introduce an l_1-norm-based regularization on all $W^{(1)}$ to select sparse and associated variables represented as follows:

$$\delta_k = \sum\nolimits_{j=1}^{q} ||W_j^{(1)}||_1. \tag{4}$$

We penalize $W^{(1)}$ so as to exclude the expected nonparent variable of X_k by forcing the redundant weights to zero. Other sparsity regularization terms (i.e., l_0-norm, $l_{1/2}$-norm [42], and others [43]) for feature selection can also be adopted, here, our choice of l_1-norm is for its simplicity, intuition, and efficacy.

In this way, we reduce the scale of candidate parents. In order to find out exact variables around target variable, we minimize local data recovery loss $\mathcal{L}_{\text{loc}}$. Before defining $\mathcal{L}_{\text{loc}}$, we need to determine the scope of a local graph. Our LCS contains target variables, parent variables, and child variables. To recover the local structure more accurately, it is necessary to appropriately define the local search area. Here, we use $\mathcal{M}$ to denote a unit of measurement for the size of a local graph, $X_{pl(T)}$ to denote the potential local graph of given variable X_T, when $\mathcal{M}$ increases by 1, $X_{pl(T)}$ will add variables which make $X_{pl(T)}$ conditional independent of other variables in weighted adjacency matrix. For example, local graph with $\mathcal{M} = 0$ only

contains the target variable. If $\mathcal{M} = 1$, local graph consists of the target variable, parents, children, and spouses. If $\mathcal{M} = 2$, local graph includes variables when $\mathcal{M} = 1$ and their MBs.

Proposition 1: When the predicted values of $\mathbf{x}_{pl(T)}$ output by MLP are close to the true values of $\mathbf{x}_{pl(T)}$, then we can say that MLP approximates the complicated function f and the resulting local graph is an LCS. The local loss term is concreted as

$$\mathcal{L}_{\text{loc}} = \ell\big(\mathbf{x}_{pl(T)}, \text{MLP}\big(\mathbf{x}_{pl(T)}; W_{pl(T)}\big)\big) \tag{5}$$

where $\ell(\cdot)$ is the least square objective function.

Each MLP aims to find the parents of a corresponding variable, but simply data fitting by MLP might just get correlation variables without direction, which cannot distinguish between parent and child variables. Here, we include the directed acyclicity constraint to reduce the search space and to direct edges between variables. In addition, we define a local graph search problem via a minimization problem as follows:

$$\begin{aligned} \min_{W} \mathcal{F} &= \alpha \sum\nolimits_{k=1}^{q} \delta_k + \beta \mathcal{L}_{\text{loc}} \\ \text{s.t.} \quad & h(C) = 0 \end{aligned} \tag{6}$$

where $h(C)$ is the directed acyclicity constraint borrowed from NOTEARS (see Theorem 1). $\mathcal{L}_{\text{loc}}$ is the *local loss* of recovering the local graph.

As the weighted adjacency matrix of NOTEARS can only apply in linear model, but we need to extract weights from each MLP. Here, we construct a weighted adjacency matrix C using nonlinear conditions similar to [9]. For each entry of C, we define it as follows:

$$C(j,k) = \begin{cases} \sqrt{\sum_{i=1}^{H_1} \left(W_{i,j}^{(1)}\right)^2}, & j \neq k \\ 0, & j = k \end{cases} \tag{7}$$

where $W^{(1)}$ is the first layer parameter matrix of the kth MLP. The weighted adjacency matrix C can be converted into a DAG by low-weight pruning and acyclicity checking.

In this work, we concentrate on discovery of local graph, Actually, we only need to pay attention to the acyclicity around target variable, so for element of $C(j,k)$, if $X_j \notin X_{pl(T)}$ and $X_k \notin X_{pl(T)}$ hold, then we will mask it as 0. We define a *mask* matrix A whose jth row and the kth column elements are all zero when $X_j, X_k \notin X_{pl(T)}$ and the others are all one. Note that A will change during iterations. Therefore, the formulation of directed acyclicity constraint can be defined as follows:

$$h(A \circ C) = 0. \tag{8}$$

We further optimize the gradient descent process as follows, and provide more details in the supplementary file:

$$\begin{aligned} W^{t+1} \leftarrow \arg\min \Bigg\{ & \alpha \sum_{k=1}^{q} \delta_k\Big(W^{(1)t}\Big) + \beta \mathcal{L}_{\text{loc}}\big(W^t\big) \\ & + \frac{\mu}{2} |h\big(A \circ C\big(W^t\big)\big)|^2 + \lambda^t h\big(A \circ C\big(W^t\big)\big) \Bigg\} \end{aligned} \tag{9}$$

$$\mu^{t+1} \leftarrow \begin{cases} \eta \mu^t, & h\big(A \circ C\big(W^t\big)\big) > \gamma h\big(A \circ C\big(W^{t-1}\big)\big) \\ \mu^t, & \text{otherwise} \end{cases} \tag{10}$$

$$\lambda^{t+1} \leftarrow \lambda^t + \mu\big(h\big(A \circ C\big(W^t\big)\big)\big). \tag{11}$$

Algorithm 1 GraN-LCS Algorithm

Input: observed data $\mathbf{x}$, target variable X_T, initial parameters thr_{PNS}, $\mathcal{M}$, α, β, λ, μ
Output: Adjacency matrix C

1: $\mathcal{R}_T \leftarrow \text{PNS}(X_T, \mathbf{x}, thr_{PNS}) \cup \{X_T\}$
2: Initialize parameters of MLPs for variable $X_k \in \mathcal{R}_T$
3: **for** $t = 1, 2, \ldots$ **do**
4: $\quad X_{pl(T)}, A \leftarrow getLocalGraph(C, \mathcal{M})$
5: $\quad$ Optimize Eq. (9) by L-BFGS-B algorithm
6: $\quad$ Update parameters μ and λ by Eq. (10) and Eq. (11)
7: $\quad$ **if** $h(A \circ C) \leq 10^{-8}$ or $\mu^{t+1} \geq 10^{16}$ **then**
8: $\quad\quad$ break
9: $\quad$ **end if**
10: **end for**

The whole procedure of GraN-LCS is summarized in Algorithm 1. Particularly, line 1 applies PNS on target variable X_T and returns candidate neighbors, which together with X_T constitute the candidate local variable set $\mathcal{R}_T$. Line 2 constructs an MLP for each variable in $\mathcal{R}_T$ and initializes their parameters. Lines 3–10 perform LCS learning by iteratively optimizing the objective function. Specifically, line 4 treats the variables and edges of the local graph with size $\mathcal{M}$ in C as $X_{pl(T)}$ and A separately. Line 5 solves the objective function minimization problem using the L-BFGS-B algorithm. Line 6 updates the penalty coefficient μ and dual variable λ, along with the acyclicity constraint; and lines 7–9 determine whether to stop the iteration in advance or not. Line 5 is time-costing but it can converge to the approximate optimal value after several iterations.

Although GraN-LCS needs to train q MLPs, we can run them in parallel with GPU acceleration. Alike NOTEARS, GraN-LCS requires the evaluation of matrix exponential of $A \circ C$ at each iteration, which costs $\mathcal{O}(d^3)$ for a dense $d \times d$ matrix. Taking GraN-LCS with one hidden layer of H_1 neurons as an example, the computational complexity of evaluating the objective function and the gradient is $\mathcal{O}(nd^2H_1 + d^2H_1 + d^3)$. Actually, matrix C after being masked by A is quite sparse, which significantly reduces the time consumption. The space complexity of GraN-LCS is $\mathcal{O}(d^2H_1)$ for d MLPs. Note that the optimization problems inherit the difficulties of nonconvexity, which means they can only be stationary solved. Nevertheless, experimental results show that GraN-LCS takes much less time than GraN-DAG and NOTEARS-MLP, with better (or comparable) performance than them on learning LCS.

IV. Experimental Results and Analysis

In this section, we conduct experiments to investigate the efficacy of GraN-LCS and compare it against competitive baselines on different datasets. The baselines include three gradient-based GCS algorithms, GraN-DAG [8], NOTEARS-MLP [9], and DAG-CCNN [28]; and six LCS algorithms, PCD-by-PCD [29], MB-by-MB [30], CMB [24], LCS-FS [22], ELCS [23], and PSL [31]. We do not take the original NOTEARS for experiments, since it targets for linear models, while its extensions GraN-DAG, NOTEARS-MLP, and DAG-CCNN adopt neural network and can fit for nonlinear models.

The configurations of compared methods are given in Table S1 of the Supplementary file. These parameters are optimized as suggested in the respective literature or shared codes.

We evaluate the quality of found causal structures using three typical metrics. True positive rate (TPR) is the ratio between correct edges given by an algorithm and all edges. False discovery rate (FDR) is the ratio between wrong edges and all edges given by the algorithm. Structural hamming distance (SHD) is the smallest number of edge additions, deletions, and reversals to convert the estimated graph into the true LCS. In addition, we take the actual runtime of each algorithm as another metric. We record these metric values on each LCS and report the average. LCS-based algorithms may output undirected edges, if there is an undirected edge between X_i and X_j, we compute it twice.

A. Results on Synthetic Datasets

We generated different synthetic datasets, which are with different data generation processes, the number of nodes, and samples. First, we generate a ground truth DAG using the Erdős-Rényi (ER) scheme. We consider graphs of $d(\in \{50, 100, 200, 500, 800, 1500\})$ nodes and the number of edges is $2d$. For each dataset, we generate $n \in \{1000, 5000\}$ samples with ten independent repeats, and report the average and standard deviations. To evaluate the performance under different data generation mechanisms, we consider three data generating processes for f_i: 1) *ANM with the Gaussian processes (GPs)*; 2) *Additive model with GP*; and 3) *Index model* ($M = 3$). The noise N_i is sampled from the normal Gaussian distribution $\mathcal{N}(0, 1)$.

Besides, to simulate a situation usually happened in reality that the causal network is composed of multiple unconnected subgraphs, we generate another synthetic dataset named *multi-subgraph* dataset. We consider graph sizes $d \in \{50, 100, 200\}$ and $2d$ edges. Each graph consists of several subgraphs with 5–20 nodes.

Table I records the experimental results on Synthetic datasets with ANM with GP. Table S2 in the supplementary material reports the results with an additive model with GP and Table S3 lists the results with Index model ($M = 3$) in the Supplementary file. Learning causal structure on nonlinear models is a challenging task due to complex structural equations. This becomes more difficult with limited samples and numerous nodes. As such, SHD shows an upward trend with the number of nodes increasing for all algorithms. Since our task is to discover LCS, we report the average runtime of LCS algorithms on all nodes and the total runtime of GCS methods on the same nodes. If an algorithm needs $\geq$ 100 h for one run, its runtime is denoted as "−."

1) GraN-LCS Versus GCS Algorithms: GraN-DAG, NOTEARS-MLP, and DAG-CCNN need to learn the causal relation of all variables, they all need an extreme runtime. More specifically, as shown in Table I, for graph of 800 nodes, GraN-LCS is 568 times faster than GraN-DAG with 1000 samples, and about 711 times faster with 5000 samples. Compared with NOTEARS-MLP, GraN-LCS is more than 2695 times faster with 1000 samples (the running time of NOTEARS-MLP exceeds 100 h) and 257 times faster with 5000 samples. When $d \geq 1500$, these GCS algorithms can hardly give a result within 100 h and suffer too expensive time cost on high-dimensional datasets. This is uneconomic because only part of the outcomes are what we need. DAG-CCNN takes less time compared with other GCS algorithms because it replaces the calculation of matrix exponential with eigenvalue to reduce the computational complexity. However, the excessive conditions for the weighted adjacency matrix make it difficult to produce valid results when applied to large graphs. For clarification, we set TPR, FDR, and SHD as "/" when DAG-CCNN obtains an empty graph. GraN-DAG performs well on small network with sufficient samples (i.e., $n = 5000$ samples with $d = 50$ variables). It needs much more runtime on $n = 5000$ than $n = 1000$, since its PNS preprocess on each node is very time-consuming [8]. Even though, it still loses to GraN-LCS by a large margin. As the network expands, the performance of GraN-DAG declines rapidly. That is because it takes the product of whole neural network paths as the weighted adjacency matrix, which makes it difficult to get an optimal solution. Therefore, GraN-DAG is more suitable for learning small causal graph with sufficient samples. NOTEARS-MLP performs competitively on the three data generation models, especially, on Index model. GraN-LCS achieves a similar or better performance than GraN-DAG and NOTEARS-MLP with $n = 1000$ ($n = 5000$), while GraN-LCS needs much less runtime. GraN-LCS takes at most 1000 s to obtain an LCS even with $d = 1500$. These results confirm the efficacy advantage of GraN-LCS to GCS methods and justify the rationality of extending NOTEARS for exploring LCS.

2) GraN-LCS Versus LCS Algorithms: GraN-LCS achieves a much better performance than other LCS algorithms. Most MB-based LCS algorithms cost much fewer time to find directed causes and effects, owing to well-designed search process for conducting fewer CI tests. Among them, LCS-FS has a higher efficiency on all datasets and performs competitively. MB-by-MB cannot adapt to most samples, it often has a high SHD and FDR. In terms of TPR, PCD-by-PCD, CMB, LCS-FS, ELCS, and PSL perform stably as the number of nodes changes. But they mistake many causal relations. It is worth noting that LCS algorithms perform worse on $n = 5000$ than $n = 1000$, which is contrary to the conventional impression. GraN-LCS adopts a gradient-optimization strategy to obtain LCS, as such it needs some iterations to find out LCS and costs more runtime than other LCS algorithms. Even so, compared with MB-by-MB, CMB, ELCS, and PSL, GraN-LCS spends similar or less time on high-dimensional datasets, especially, when $d = 1500$. We want to remark that GraN-LCS can find out more correct causal relations and make fewer mistakes than these LCS algorithms within an acceptable runtime.

3) Impact of Different Models: We observe that GraN-LCS performs competitively on the three data models, especially, on nonlinear models with additive GPs. Our GraN-LCS loses to NOTEARS-MLP on Index model, but on high-dimensional graph, especially, when $n = s1000$, its SHD is close or even better than NOTEARS-MLP. GraN-DAG performs better on Index model but still loses to GraN-LCS. This proves that

TABLE I
RESULTS ON *Nonlinear Models With GPs*. ↑ (↓) MEANS THE HIGHER(LOWER) THE VALUE, THE BETTER THE PERFORMANCE IS. THE BEST RESULT IS HIGHLIGHTED IN **BOLD** FONT. GRAN-LCS WITH THE SECOND BEST RESULT IS UNDERLINED

		Size=1000				Size=5000			
#Nodes		TPR(%)↑	FDR(%)↓	SHD↓	Runtime(s)↓	TPR(%)↑	FDR(%)↓	SHD↓	Runtime(s)↓
50	GraN-DAG [8]	23.6±7.7	80.0±5.8	8.28±1.61	851.68	48.4±11.1	**0.0±0.0**	2.28±0.40	2486.35
	NOTEARS-MLP [9]	47.3±4.8	**8.0±2.7**	**2.33±0.21**	254.20	50.9±3.9	6.5±2.9	2.25±0.15	98.02
	DAG-CCNN [28]	5.6±3.0	90.3±4.6	7.90±1.44	53.44	5.1±2.6	84.3±18.4	7.44±1.66	49.47
	PCD-by-PCD [29]	29.3±2.9	50.3±5.8	3.37±0.16	0.69	42.5±3.6	54.3±4.1	3.33±0.20	2.03
	MB-by-MB [30]	32.1±4.6	62.9±7.5	4.63±0.59	1.67	48.1±5.8	76.3±4.1	8.04±0.88	5.80
	CMB [24]	32.0±2.8	50.5±3.5	3.44±0.16	6.66	40.9±4.3	57.1±5.8	3.41±0.22	13.74
	LCS-FS [22]	21.5±4.9	25.8±3.2	3.91±0.10	**0.12**	17.3±3.3	21.4±2.9	3.95±0.06	**0.17**
	ELCS [23]	28.4±5.1	44.0±5.5	4.02±0.13	3.33	26.0±4.0	51.5±8.5	3.80±0.17	6.38
	PSL [31]	29.6±4.1	53.2±5.8	3.56±0.17	11.68	39.4±6.6	52.5±5.6	3.41±0.26	22.73
	GraN-LCS	**57.7±4.7**	17.5±4.0	2.38±0.22	21.58	**72.5±3.6**	14.8±3.6	**1.71±0.19**	47.85
100	GraN-DAG	2.3±2.2	98.3±2.2	15.04±4.53	2583.10	36.1±8.4	**0.0±0.0**	2.81±0.33	7901.11
	NOTEARS-MLP	**41.8±5.1**	**6.7±2.3**	**2.65±0.18**	1832.14	48.6±3.7	6.2±3.1	2.36±0.15	687.45
	DAG-CCNN	3.3±1.0	25.3±11.5	4.34±0.21	109.46	2.8±1.4	25.9±17.6	4.36±0.27	126.67
	PCD-by-PCD	28.0±2.4	55.3±2.6	3.56±0.10	1.57	39.0±1.9	56.9±1.7	3.44±0.15	3.26
	MB-by-MB	31.0±4.5	71.0±2.6	5.94±0.43	3.57	45.5±2.8	77.4±2.3	8.89±0.60	10.01
	CMB	28.1±3.5	57.0±5.5	3.68±0.18	11.84	37.2±2.9	59.8±2.5	3.60±0.18	27.86
	LCS-FS	26.1±2.3	23.9±3.6	3.87±0.07	**0.20**	19.7±1.6	20.1±2.5	3.95±0.04	**0.30**
	ELCS	22.2±3.0	44.3±4.9	4.03±0.15	6.11	23.9±2.5	51.9±3.7	3.83±0.09	11.62
	PSL	29.2±3.7	56.1±4.9	3.70±0.17	22.97	37.2±3.2	57.6±3.9	3.63±0.18	45.16
	GraN-LCS	31.6±4.4	9.4±3.2	3.03±0.16	18.32	**67.1±1.0**	13.8±1.8	**1.97±0.09**	70.06
200	GraN-DAG	3.5±1.6	85.7±12.8	10.37±3.28	7484.14	23.5±2.9	**0.0±0.0**	3.21±0.10	25952.86
	NOTEARS-MLP	**33.7±1.9**	12.1±2.5	**2.90±0.06**	10559.34	47.2±1.9	9.1±1.7	2.37±0.07	3386.69
	DAG-CCNN	0.5±0.2	**6.2±3.5**	4.14±0.12	680.72	0.5±0.1	8.5±5.8	4.19±0.16	883.84
	PCD-by-PCD	27.5±2.2	62.4±2.0	3.93±0.12	4.01	39.7±1.5	62.2±1.9	3.99±0.14	6.60
	MB-by-MB	30.9±2.0	83.8±2.4	9.14±0.67	9.71	45.0±2.5	86.9±1.6	14.45±1.32	30.22
	CMB	27.6±2.1	64.6±2.8	4.11±0.09	33.18	41.0±2.6	64.6±2.3	4.20±0.15	67.14
	LCS-FS	25.6±2.3	28.6±1.7	3.89±0.05	**0.44**	20.6±2.0	24.3±2.6	3.94±0.03	**0.64**
	ELCS	19.0±2.0	53.8±4.4	4.22±0.11	24.74	22.7±2.4	59.1±3.8	4.06±0.15	31.38
	PSL	29.1±1.5	64.0±1.9	4.19±0.11	68.21	38.5±2.6	63.2±2.1	4.22±0.15	139.64
	GraN-LCS	27.9±0.8	13.5±0.3	3.08±0.06	34.54	**64.3±2.2**	15.3±1.4	**2.02±0.13**	114.88
500	GraN-DAG	3.7±0.8	30.2±16.4	4.54±0.47	37351.26	13.5±2.2	5.5±5.1	3.65±0.07	133312.95
	NOTEARS-MLP	23.3±1.5	21.8±2.7	3.55±0.02	137135.91	38.8±2.2	10.5±1.0	2.68±0.07	50406.85
	DAG-CCNN	1.6±0.0	**0.5±0.6**	4.01±0.01	2802.65	1.6±0.0	**1.2±1.9**	4.02±0.04	4209.12
	PCD-by-PCD	26.6±0.8	70.3±0.9	4.64±0.11	8.98	36.2±0.9	71.1±1.4	4.90±0.14	15.36
	MB-by-MB	**28.4±1.3**	94.2±0.5	20.74±0.90	59.34	40.5±1.3	94.2±0.4	28.41±1.18	162.41
	CMB	26.3±0.9	75.1±1.4	5.22±0.15	170.41	37.2±1.3	73.8±1.0	5.63±0.07	379.00
	LCS-FS	25.2±1.9	26.3±1.6	3.84±0.03	**0.98**	18.8±0.9	21.4±0.6	3.91±0.03	**1.54**
	ELCS	15.0±1.4	69.2±2.1	4.69±0.08	93.43	20.7±1.4	70.4±1.1	4.83±0.05	108.85
	PSL	28.0±1.7	74.7±0.9	5.35±0.06	155.44	35.8±1.2	72.7±0.6	5.50±0.08	299.82
	GraN-LCS	20.1±2.7	11.4±3.1	**3.40±0.12**	72.87	**56.6±2.5**	14.8±1.4	**2.32±0.11**	259.79
800	GraN-DAG	5.2±2.4	21.8±19.4	4.33±0.64	75935.95	11.6±2.8	33.0±20.7	4.63±1.01	338457.15
	NOTEARS-MLP	-	-	-	-	32.5±1.5	**11.1±1.2**	2.92±0.06	122324.56
	DAG-CCNN	/	/	/	2606.80	/	/	/	10825.85
	PCD-by-PCD	25.7±0.5	75.8±0.9	5.34±0.07	21.31	36.1±0.5	75.4±0.6	5.66±0.10	39.96
	MB-by-MB	**27.7±0.6**	97.0±0.1	36.48±1.01	277.93	40.6±0.8	96.5±0.1	46.50±1.20	496.20
	CMB	25.1±0.7	80.6±0.7	6.26±0.08	488.97	38.5±1.5	77.7±0.8	6.66±0.10	954.63
	LCS-FS	24.7±1.0	26.2±0.9	3.87±0.03	**1.47**	18.3±1.2	21.1±1.3	3.93±0.03	**1.52**
	ELCS	13.2±0.6	77.0±0.9	5.24±0.09	151.10	19.5±1.1	76.4±1.4	5.43±0.10	203.67
	PSL	27.0±0.7	78.8±0.6	6.24±0.06	464.26	36.5±1.4	77.0±0.5	6.64±0.06	618.78
	GraN-LCS	15.8±0.8	**9.9±0.7**	**3.52±0.01**	133.60	**48.7±1.4**	16.8±0.6	**2.49±0.04**	476.33
1500	GraN-DAG	-	-	-	-	-	-	-	-
	NOTEARS-MLP	-	-	-	-	-	-	-	-
	DAG-CCNN	/	/	/	42830.35	/	/	/	89840.17
	PCD-by-PCD	25.5±0.4	81.3±0.6	6.45±0.09	81.47	36.0±0.3	81.3±0.4	7.43±0.11	196.56
	MB-by-MB	**28.1±1.7**	99.3±0.6	91.50±2.77	2066.27	39.7±3.0	98.6±0.1	107.14±3.07	4107.57
	CMB	24.5±0.3	85.1±0.5	7.97±0.10	1767.19	36.4±2.1	79.9±1.6	8.54±0.14	3380.98
	LCS-FS	24.7±0.7	26.4±0.6	**3.88±0.03**	**4.92**	19.6±0.3	21.9±0.5	3.93±0.02	**6.51**
	ELCS	12.7±0.8	84.2±0.7	6.12±0.06	576.39	19.5±0.7	82.5±0.4	6.69±0.07	570.15
	PSL	25.8±1.0	84.3±0.4	7.92±0.04	1390.24	33.4±1.6	83.8±1.8	9.22±0.11	2613.88
	GraN-LCS	12.6±0.5	**14.6±1.5**	3.96±0.11	313.51	**42.1±1.6**	**19.3±0.9**	**2.73±0.06**	1068.02

GraN-LCS is superior to other gradient-based methods on large graphs. Most LCS algorithms clearly lose to GraN-LCS in terms of SHD, and LCS-FS has similar results with GraN-LCS on additive GPs model with $n = 1000$. PCD-by-PCD is another LCS algorithm that performs well in terms of SHD and runtime on Index model.

Fig. 3 reports the results of GraN-LCS and other compared methods on synthetic datasets with multiple LCSs. GraN-LCS achieves a much-improved performance over compared methods on important metrics. That is because multiple causal graphs co-exist lead to difficulties for GCS methods. GraN-DAG keeps FDR near zero with 5000 samples, so if there are enough samples, GraN-DAG can find out few wrong edges on multiple subgraphs dataset, this is probably due to its stringent sparsity constraint. Although LCS methods spend fewer time, they get more than three wrong edges at each variable on average. A high FDR indicates that CI tests fail to obtain the smallest set, which makes the target variable and others conditional independent. In other words, too many wrong variables are introduced into PC set. The poor performance signifies that they are hard to be applied in practice. In contrast, GraN-LCS is a good alternative, it can find out LCS with an acceptable time cost.

In addition, we conduct the Wilcoxon signed-rank test [44] between GraN-LCS and other GCS algorithms across these metrics and datasets. Wilcoxon signed-rank test is a nonparametric statistical hypothesis test, it is widely used to contrast the performance of two methods on multiple datasets. The test results show that all the p-values are smaller than 0.005. We also apply the Wilcoxon signed rank test to check the statistical difference between GraN-LCS and other LCS algorithms in terms of TPR, FDR, or SHD, we have similar conclusions with all p-values smaller than 0.001, which means GraN-LCS significantly outperforms other algorithms.

In summary, although existing LCS algorithms can obtain results in a short time, their results are unreliable in most cases. GraN-LCS needs more time than several LCS algorithms to obtain an LCS, but its runtime is generally acceptable given the performance improvement. GCS algorithms may obtain effective results but spend too much unnecessary time. GraN-LCS can make LCSs in an economics way for practical use.

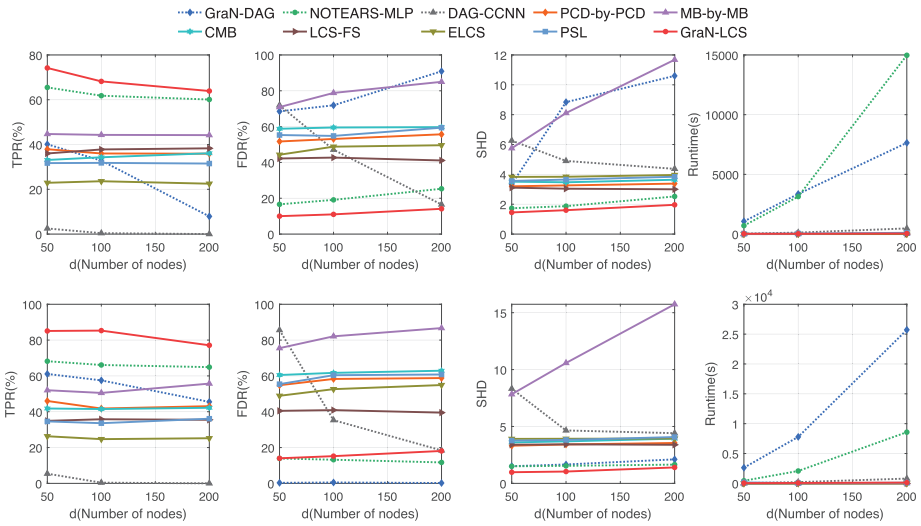


Fig. 3. Results on Datasets With Multiple Subgraphs *Generated With Additive GPs*. First Row: $n = 1000$; Second Row: $n = 5000$.

B. Real and Pseudo-Real Data

To evaluate the performance of these compared algorithms for real applications, we consider a well-known dataset provided in [1], which measures the level of different expressions of protein and phospholipids in human cells. The ground truth causal graph of this dataset consists of 11 nodes and 17 edges. In this work, we test on observational data with $n = 853$ samples. We also consider a pseudo-real dataset generated by SynTReN generator [45], which creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximates experimental data. We generate ten datasets of 20 nodes with 500 samples using the default parameters, except that the probability for complex 2-regulator interactions was set to 1 and the random seeds were 0 to 9.

As shown in Table II, except our GraN-LCS, all other LCS methods achieve poor performance in terms of FDR and SHD on the Sachs' protein signaling dataset. Too many wrong edges result in high SHD. NOTEARS-MLP also makes too many mistakes, this might be due to lack of identifiability guarantees in real applications. GraN-DAG gets a small number of edges but all of them are correct, that is why it performs well in terms of SHD. GraN-LCS outperforms these rivals in terms of TPR and SHD with much less time than GraN-DAG. This result proves that GraN-LCS is also competitive in reality and justifies its effectiveness.

On the SynTReN dataset, GraN-DAG achieves the lowest FDR in average, but its standard deviation is too high to get a stable result. In contrast, GraN-LCS performs a little higher in terms of average but much lower in terms of standard deviation than GraN-DAG, which means in most cases GraN-LCS obtains fewer wrong edges than GraN-DAG. LCS-FS conducts the fewest CI tests (the number of CI tests of LCS algorithms is presented in the Supplementary file) among MB-based LCS algorithms, but it spends too much time on feature selection. As a result, it takes an average of nearly 700 s per variable. GraN-LCS achieves a much lower SHD than other LCS algorithms without too much runtime.

TABLE II
RESULTS ON REAL AND PSEUDO-REAL DATA

	Sachs' protein signaling dataset			
	TPR(%)↑	FDR (%)↓	SHD↓	Runtime(s)↓
GraN-DAG [8]	29.4	**0.0**	2.18	1357.78
NOTEARS-MLP [9]	35.3	84.2	6.73	1119.69
DAG-CCNN [28]	17.6	25.0	2.73	2.30
PCD-by-PCD [29]	35.3	57.1	3.09	0.23
MB-by-MB [30]	35.3	57.1	3.09	0.40
CMB [24]	35.3	57.1	3.09	**0.18**
LCS-FS [22]	41.2	65.0	3.64	15.39
ELCS [23]	26.5	50.0	3.09	0.36
PSL [31]	29.4	58.3	3.09	0.23
GraN-LCS	**47.6**	38.2	**1.93**	42.71
	SynTReN dataset (20 nodes)			
	TPR(%)↑	FDR (%)↓	SHD↓	Runtime(s)↓
GraN-DAG	9.5±12.6	**78.3±24.5**	2.77±0.72	438.30
NOTEARS-MLP	29.9±18.3	93.1±3.4	10.41±2.96	1411.56
DAG-CCNN	**50.2±17.1**	93.6±1.2	13.46±2.60	2.67
PCD-by-PCD	27.3±8.6	83.9±5.7	4.42±0.71	4.18
MB-by-MB	37.4±14.0	88.0±2.0	7.13±1.14	**2.11**
CMB	33.5±12.5	82.6±8.1	4.69±0.92	26.54
LCS-FS	8.4±8.6	91.5±9.1	3.84±0.62	690.79
ELCS	8.3±2.9	89.4±4.0	3.30±0.47	4.12
PSL	27.1±8.8	85.6±4.4	4.77±0.64	36.18
GraN-LCS	34.2±3.5	**81.6±1.9**	**2.41±0.26**	34.54

C. Ablation Study

We conduct three ablation studies to investigate the contribution of the size $\mathcal{M}$ of local graph, l_1-norm for feature selection and PNS for preliminary screening. In addition, we plot the convergence curve of GraN-LCS, whose loss generally decreases with the increase of iterations, and PNS has a positive impact on the performance of GraN-LCS. We also carry out parameter sensitivity studies w.r.t. α, β, and PNS threshold. More details about the study of PNS, convergence analysis, and parameter sensitivity analysis are given in the Supplementary file.

We take $\mathcal{M} \in \{0, 1, 2, 3\}$ with $n = 1000$ ($n = 5000$) samples to investigate the impact of different values of $\mathcal{M}$ on the results. Note the search space is not expanded as the increase of $\mathcal{M}$. Fig. 4 shows that GraN-LCS needs more time to generate results when $\mathcal{M}$ takes a larger value, and the SHD manifests an improved trend. GraN-LCS takes less time when

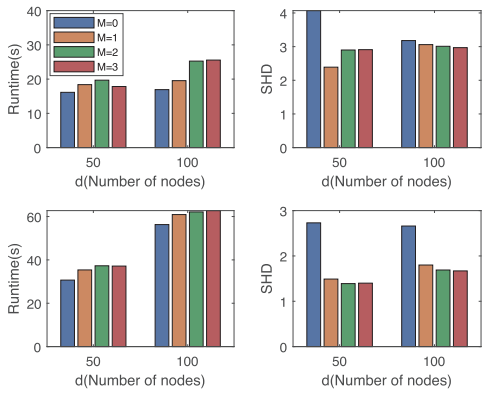


Fig. 4. Runtime and SHD of GraN-LCS with $\mathcal{M} = \{0, 1, 2, 3\}$. First row: $n = 1000$; Second row: $n = 5000$.

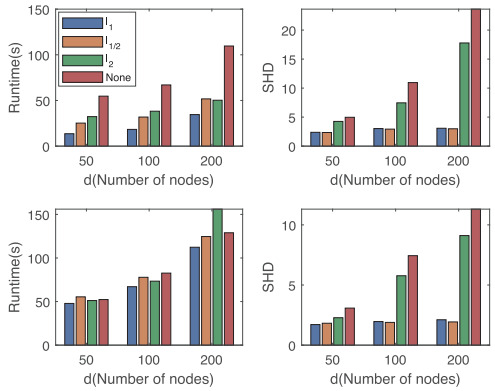


Fig. 5. Runtime and SHD of GraN-LCS with l_1, $l_{1/2}$, l_2-norm and without feature selection. First row: $n = 1000$; Second row: $n = 5000$.

$\mathcal{M} = 1$ than $\mathcal{M} = 2$ and 3, while holds a similar SHD. GraN-LCS achieves economic efficacy with $\mathcal{M} = 1$. This is because when $\mathcal{M} = 1$, the local graph already includes the parent and children nodes of the target node, while $\mathcal{M} = 0$ only includes the target node. $\mathcal{M} = 2$ or 3 includes not only parent and children nodes but also too many irrelevant ones, which results in a long runtime but not necessarily an improved performance.

We also conduct experiments to further investigate the impact of different norms for feature selection. We testify the typical l_1, $l_{1/2}$, l_2-norm and without feature selection on datasets with $n = 1000/5000$ generated samples with nonlinear models with GP. As shown in Fig. 5, l_1-norm holds a lead in time cost. The advantage is more obvious in SHD as the graph increases except $l_{1/2}$-norm, which even leads to a smaller SHD on $d = 100/200$. This is because l_1-norm can make the matrix sparse, and, thus, reduce the computation complexity and candidate variables. $l_{1/2}$-norm can produce more sparse solutions than l_1-norm, so it outputs fewer false positive edges when dealing with high-dimensional data, but it costs more time than l_1-norm. l_2-norm cannot induce a sparse graph and has a higher SHD than l_1-norm with sparse feature selection. GraN-LCS without feature selection has the largest time cost and highest SHD. This observation confirms the necessity and efficacy advantage of feature selection.

V. Conclusion

This article studies how to explore LCS of a target variable in an economic way. Existing LCS algorithms canonically based on CI tests suffer the poor accuracy due to noises, various data generation mechanisms and small-size samples; while GCS methods pursue a global causal graph, which is too time-consuming and may include many noninteresting parts. We design a novel algorithm named GraN-LCS that explores the LCS of target variable in a gradient-descent way. It leverages masked directed acyclicity constraints, feature selection, and PNS procedure to determine neighbors and orient edges. Experimental results on both synthetic and real-world datasets prove that GraN-LCS is robust and competitive on learning LCS. There are some future pursues for GraN-LCS: 1) the weight in the adjacency matrix may not be causal effect, so some correct edges might be pruned when converting the weight adjacency matrix into a DAG and 2) prior knowledge of real-world applications is not used. In the future, we will study more effective strategies for processing weighted adjacency matrix and merging prior knowledge.

References

[1] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan, "Causal protein-signaling networks derived from multiparameter single-cell data," *Science*, vol. 308, no. 5721, pp. 523–529, 2005.

[2] J. Pearl, *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[3] R. Xu et al., "Algorithmic decision making with conditional fairness," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2020, pp. 2125–2135.

[4] S. Cong, Y. Guoxian, W. Jun, Y. Zhongmin, and C. Lizhen, "A review on causality-based fairness machine learning," *Intell. Robot.*, vol. 2, no. 3, pp. 244–274, 2022.

[5] B. H. Kim, S. Jo, and S. Choi, "ALIS: Learning affective causality behind daily activities from a wearable life-log system," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13212–13224, Dec. 2022.

[6] M. J. Vowels, N. C. Camgoz, and R. Bowden, "D'ya like DAGs? A survey on structure learning and causal discovery," *ACM Comput. Surveys*, vol. 55, no. 4, pp. 1–36, 2023.

[7] H. Zhang et al., "Learning causal structures based on divide and conquer," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3232–3243, May 2022.

[8] S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien, "Gradient-based neural DAG learning," in *Proc. Int. Conf. Learn. Rep.*, 2020, pp. 1–9.

[9] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. Xing, "Learning sparse nonparametric dags," in *Proc. Int. Conf. Artif. Intell. Stat.*, 2020, pp. 3414–3425.

[10] S. Liu, Y. Feng, K. Wu, G. Cheng, J. Huang, and Z. Liu, "Graph-attention-based casual discovery with trust region-navigated clipping policy optimization," *IEEE Trans. Cybern.*, early access, Oct. 19, 2021, doi: 10.1109/TCYB.2021.3116762.

[11] X. Wang et al., "Ordering-based causal discovery with reinforcement learning," in *Proc. 13th Int. Joint Conf. Artif. Intell.*, 2021, pp. 3566–3573.

[12] I. Ng, S. Zhu, Z. Chen, and Z. Fang, "A graph autoencoder approach to causal structure learning," in *Proc. NeurIPS Workshop*, 2019, pp. 1–9.

[13] Y. He, P. Cui, Z. Shen, R. Xu, F. Liu, and Y. Jiang, "DARING: Differentiable causal discovery with residual independence," in *Proc. 27th ACM SIGKDD Conf. Knowl. Disc. Data Min.*, 2021, pp. 596–605.

[14] W. T. C. C. Consortium et al., "Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls," *Nature*, vol. 447, no. 7145, p. 661, 2007.

[15] I. S. Peter and E. H. Davidson, "A gene regulatory network controlling the embryonic specification of endoderm," *Nature*, vol. 474, no. 7353, pp. 635–639, 2011.

[16] J. Wang, Z. Yang, C. Domeniconi, X. Zhang, and G. Yu, "Cooperative driver pathway discovery via fusion of multi-relational data of genes, miRNAs and pathways," *Briefings Bioinf.*, vol. 22, no. 2, pp. 1984–1999, 2021.

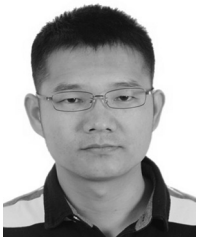
[17] K. Yu et al., "Causality-based feature selection: Methods and evaluations," *ACM Comput. Surveys*, vol. 53, no. 5, pp. 1–36, 2020.
[18] J. Zhou, Q. Wu, M. Zhou, J. Wen, Y. Al-Turki, and A. Abusorrah, "LAGAM: A length-adaptive genetic algorithm with Markov blanket for high-dimensional feature selection in classification," *IEEE Trans. Cybern.*, early access, Nov. 14, 2022, doi: 10.1109/TCYB.2022.3163577.
[19] K. Yu, L. Liu, and J. Li, "A unified view of causal and non-causal feature selection," *ACM Trans. Knowl. Disc. Data*, vol. 15, no. 4, pp. 1–46, 2021.
[20] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, and V. Christophides, "A greedy feature selection algorithm for big data of high dimensionality," *Mach. Learn.*, vol. 108, no. 2, pp. 149–202, 2019.
[21] T. Gao and Q. Ji, "Efficient Markov blanket discovery and its application," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1169–1179, May 2017.
[22] Z. Ling, K. Yu, H. Wang, L. Li, and X. Wu, "Using feature selection for local causal structure learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 4, pp. 530–540, Aug. 2021.
[23] S. Yang, H. Wang, K. Yu, F. Cao, and X. Wu, "Towards efficient local causal structure learning," *IEEE Trans. Big Data*, vol. 99, no. 1, pp. 1–14, Dec. 2022.
[24] T. Gao and Q. Ji, "Local causal discovery of direct causes and effects," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2512–2520.
[25] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing, "DAGs with NO TEARS: Continuous optimization for structure learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9492–9503.
[26] R. Zhu et al., "Efficient and scalable structure learning for Bayesian networks: Algorithms and applications," in *Proc. IEEE Int. Conf. Data Eng.*, 2021, pp. 2613–2624.
[27] P. Bühlmann, J. Peters, and J. Ernest, "CAM: Causal additive models, high-dimensional order search and penalized regression," *Ann. Stat.*, vol. 42, no. 6, pp. 2526–2556, 2014.
[28] C. Xu and W. Xu, "Causal structure learning with one-dimensional convolutional neural networks," *IEEE Access*, vol. 9, pp. 162147–162155, 2021.
[29] J. Yin, Y. Zhou, C. Wang, P. He, C. Zheng, and Z. Geng, "Partial orientation and local structural learning of causal networks for prediction," in *Proc. Causation Prediction Challenge*, 2008, pp. 93–105.
[30] C. Wang, Y. Zhou, Q. Zhao, and Z. Geng, "Discovering and orienting the edges connected to a target variable in a DAG via a sequential local learning approach," *Comput. Stat. Data Anal.*, vol. 77, pp. 252–266, Sep. 2014.
[31] Z. Ling, K. Yu, L. Liu, J. Li, Y. Zhang, and X. Wu, "PSL: An algorithm for partial Bayesian network structure learning," *ACM Trans. Knowl. Disc. Data*, vol. 16, no. 5, pp. 1–25, 2022.
[32] D. Margaritis and S. Thrun, "Bayesian network induction via local neighborhoods," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 505–511.
[33] I. Tsamardinos and C. F. Aliferis, "Towards principled feature selection: Relevancy, filters and wrappers," in *Proc. Int. Workshop Artif. Intell. Stat.*, 2003, pp. 300–307.
[34] I. Tsamardinos, C. F. Aliferis, and A. Statnikov, "Time and sample efficient discovery of Markov blankets and direct causal relations," in *Proc. 9th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2003, pp. 673–678.
[35] J. M. Pena, R. Nilsson, J. Björkegren, and J. Tegnér, "Towards scalable and data efficient learning of Markov boundaries," *Int. J. Approx. Reason.*, vol. 45, no. 2, pp. 211–232, 2007.
[36] C. Meek, "Causal inference and causal explanation with background knowledge," in *Proc. 11th Conf. Uncertainty Artif. Intell.*, 1995, pp. 403–410.
[37] X. Wu, B. Jiang, K. Yu, c. Miao, and H. Chen, "Accurate Markov boundary discovery for causal feature selection," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4983–4996, Dec. 2020.
[38] P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman, *Causation, Prediction, and Search*. Cambridge, MA, USA: MIT Press, 2000.
[39] D. M. Chickering, "Optimal structure identification with greedy search," *J. Mach. Learn. Res.*, vol. 3, no. 11, pp. 507–554, 2002.
[40] C. Heinze-Deml, M. H. Maathuis, and N. Meinshausen, "Causal structure learning," *Annu. Rev. Stat. Appl.*, vol. 5, no. 1, pp. 371–391, 2018.
[41] C. Tian, M. Zheng, W. Zuo, B. Zhang, Y. Zhang, and D. Zhang, "Multi-stage image denoising with the wavelet transform," *Pattern Recognit.*, vol. 134, Feb. 2023, Art. no. 109050.
[42] Y. Wang, D. Meng, and M. Yuan, "Sparse recovery: From vectors to tensors," *Nat. Sci. Rev.*, vol. 5, no. 5, pp. 756–767, 2018.
[43] X. Li, Y. Wang, and R. Ruiz, "A survey on sparse learning models for feature selection," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1642–1660, Mar. 2022.
[44] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.
[45] T. Van den Bulcke et al., "SynTReN: A generator of synthetic gene expression data for design and analysis of structure learning algorithms," *BMC Bioinf.*, vol. 7, no. 1, pp. 1–12, 2006.

Jiaxuan Liang received the B.Eng. degree from the School of Software, Shandong University, Jinan, China, in July 2021, where he is currently pursuing the M.Phil. degree with the School of Software.

His research interests include artificial intelligence and its application in bioinformatics.

Jun Wang received the Ph.D. degree in artificial intelligence from the Harbin Institute of Technology, Harbin, China, in 2010.

She is a Professor with the Joint SDU-NTU Centre for Artificial Intelligence Research, Shandong University, Jinan, China. Her current research interests include machine learning, data mining and their applications in bioinformatics.



Guoxian Yu (Member, IEEE) received the Ph.D. in computer science from the South China University of Technology, Guangzhou, China, in 2013.

He is a Professor with the School of Software, Shandong University, Jinan, China. His research interests include data mining and bioinformatics.

Dr. Yu has served as an Associate Editor for *Interdisciplinary Sciences: Computational Life Sciences, BioMed Research International*.

Carlotta Domeniconi received the Ph.D. degree in computer science from the University of California at Riverside, Riverside, CA, USA, in 2002.

She is an Associate Professor with the Department of Computer Science, George Mason University, Fairfax, VA, USA. Her research interests include machine learning, pattern recognition, and data mining, with applications in text mining and bioinformatics.

Ms. Domeniconi is an Associate Editor of IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and *Knowledge and Information Systems*.

Xiangliang Zhang (Senior Member, IEEE) received the Ph.D. degree in computer science from INRIA-University Paris-Sud 11, Bures-sur-Yvette, France, in 2010.

She is an Associate Professor with the Department of Computer Science and Engineering, University of Notre Dame, Notre Dame, IN, USA. Her main research interests are in diverse areas of machine learning and data mining.

Dr. Zhang is an Associated Editor of IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, *Information Sciences*, and *International Journal of Intelligent Systems*.

Maozu Guo received the Ph.D. degree in computer science and technology from the Harbin Institute of Technology, Harbin, China, in 1997.

He is a Professor with the College of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, China. His research interests include bioinformatics, machine learning, and data mining.