Causal Structure Learning With One-Dimensional Convolutional Neural Networks

CHUANYU XU AND WEI XU
Department of Mathematic, University of Science and Technology Beijing, Beijing 100083, China
Corresponding author: Chuanyu Xu (875407999@qq.com)

This work was supported by the National Natural Science Foundation of China under Grant 11801028.

ABSTRACT Causal structure discovery has an important guiding role in explanatory artificial intelligence. In order to discover causal relationships from observed data and restore causal structure graphs, we propose the Directed Acyclic Graph structure learning with Causal Convolutional Neural Networks (DAG-CCNN). First, we employ a nonlinear structural causal model (SCM) generation mechanism and propose an integrated neural network model which combines a fully-connected (FC) layer and one-dimensional convolutional (1D-Conv) layer. Then, a new acyclic algebraic representation is proposed, and a theorem and its proof are given. We use the eigenvalues of the weighted adjacency matrix instead of the Hadamard product of the adjacency matrix to represent the acyclicity of the graph, which essentially avoids the computational complexity associated with matrix multiplication. Finally, compared with DAG-GNN, NOTEARS and GraN-DAG, the experimental results show that the DAG-CCNN method has some advantages in performance on synthetic and real Sach data sets and the results of the restoring causal structure graph are more satisfactory.

INDEX TERMS Causality, convolutional neural network, machine learning, structure learning.

I. INTRODUCTION
Causal structure learning is an emerging topic in machine learning and artificial intelligence, and it has been argued that the study of causality has a crucial role in reasoning about the world. Causal discovery algorithms aim to maximize the reduction of the true directed acyclic graph by observing the data to infer causal relationships between variables, which is of great importance in the medical field [4], business optimization decision field [3], and biological field [10], etc. In recent years, causal discovery algorithms have been proposed with the help of deep learning with powerful non-linear representation capabilities, and a boom in the improvement of causal discovery algorithms has been launched in academia [25].

The traditional method initially proposed was the randomized controlled trial, which focused on intervention data and was therefore difficult to conduct due to high costs and ethical problems. Then, Judea Pearl proposed a theoretical basis for discovering causal relationships through observational data [7]. Subsequently, three algorithms in three general directions have been proposed, which are constraint-based method, causal function model method, and search scoring method. The constraint-based method, represented by the Peter-Clark (PC) algorithm and the Inductive Causation (IC) algorithm, determines the causal structure graph by means of the conditional independence test and orientation rules, but the obtained graph is one of the Markov equivalence graphs, so it is impossible to determine the unique directed acyclic graph (DAG). The method based on the causal function model starts from the causal mechanism of data generation and aims to determine the causal direction. This approach is mainly represented by the linear non-Gaussian acyclic model (LiNGAM) [20], the additive noise model (ANM) [6] and the posterior nonlinear model (PNL) [11]. However, they have strong limitations and are not universal, as they determine the causal direction based on the directional asymmetry of a particular model. The last method is the search scoring method, mainly represented by the greedy search algorithm (GES) [2]. By defining a scoring function, we can search for the highest scoring causally directed acyclic graph in the entire variable space. Obviously, the complexity of the search space grows exponentially as the number of nodes increases, making the problem NP-hard, so it can only be used for the recovery of low-dimensional causal structured graphs [13]. These challenges affect the discovery and estimation of causal relationships.

In recent years, a leapfrog improvement has been achieved in a causal discovery algorithm named NOTEARS [23]. It gives a continuous representation of the acyclic constraints
and successfully converts the combinatorial optimization problem into a continuous constrained optimization problem, cleverly avoiding the super-exponential growth complexity problem faced by traditional algorithms. Although this is a great breakthrough, the algorithm can only be applied to linear SCMs. Recently, new algorithms based on machine learning have been proposed successively to further realize the extension from linear SCM to nonlinear SCM, such as DAG-GNN [26], Gran-DAG [18], DAG-VAE [1], DAG-RL [21], and so on. These algorithms face the difficulty of training deep artificial neural networks and the difficulty of high equipment requirements by building them and making them meet the training requirements by adding mandatory assumptions. Moreover, training deep neural networks requires a sufficiently large training set to achieve reasonable generalization capability. However, the computation of the Hadamard product of the weighted adjacency matrix is more onerous in these algorithms. Therefore, the above methods still face difficulties in estimating causal structure maps for high-dimensional sample data.

In this paper, we propose the Directed Acyclic Graph structure learning with Causal Convolutional Neural Networks (DAG-CCNN), and the method first builds an integrated neural network model combining a one-dimensional convolutional (1D-Conv) layer [22] and a fully-connected (FC) layer [5]. Then a new representation of the acyclic is given. In our model, the 1D-Conv layer performs only linear one-dimensional convolution (including two operations: scalar multiplication and addition) and automatically extracts features, which makes a real-time and low-cost hardware implementation feasible. In addition, we improve the characterization, a new acyclic characterization based on the eigenvalues of the adjacency matrix is constructed, which greatly reduces the complexity difficulties faced by the matrix multiplication operation due to the construction of the adjacency matrix with Hadamard products as mentioned in previous methods. Finally, the experiments show that the proposed method is more effective for both the data set synthesized in this paper and the Sach real data set.

The remainder of the paper is organized as four sections. Section II first introduces the background knowledge about causal discovery. Section III details the model and various methods included in our proposed model. Section IV optimizes the parameters in the proposed model using the augmented Lagrange multiplier method. The experimental results are described in Section V. Finally, the overall conclusion is drawn in the final section.

II. BACKGROUND

A. STRUCTURAL CAUSAL MODEL (SCM)

If all edges are directed, and there are no cycles, we have the well-known class of DAGs. The basic DAG structure learning problem can be summarized as an algebraic representation of SCM, which is formulated as follows. Let $X = (X_1, X_2, \ldots, X_d)$ be a matrix of random vectors, $E = (E_1, E_2, \ldots, E_d)$ be a noise set following the $P(\epsilon)$ distribution. If the causal mechanism $f_i$ is linear, the noise obeys a non-Gaussian distribution (uniform distribution, Laplace distribution), and if the causal mechanism $f_i$ is nonlinear, the noise obeys a Gaussian distribution. $G = (X, E)$ be a DAG and SCM is defined as

$$X_i = f_i(X_{pa_i}, E_i), \quad E_i \perp X_{pa_i}, \quad E_i \sim P(\epsilon), \quad i = 1 \cdots d,$$

we call it SCM, where “$\perp$” represents that variables are independent of each other, $X_{pa_i} \subset (X_1, X_2, \ldots, X_d)$ is the set of the parent nodes of $X_i$, and $f_i$ is a causal mechanism. In addition, $X_j \in X_{pa_i}$ represents that there exists a direct causal relation from $X_j$ to $X_i$, written $X_j \rightarrow X_i$, i.e. there exists a directed edge from $X_j$ to $X_i$. In the following, Fig. 1 shows a SCM based on a DAG.

B. CAUSAL MARKOV ASSUMPTION AND TOPOLOGICAL SORTING

Let $X = (X_1, X_2, \ldots, X_d)$ denote a set of continuous random variables with joint distribution $P(X)$, and $G$ be a DAG. We assume that $P$ can be factorized along $G$

$$P(X) = \prod_{i=1}^{d} P(X_i \mid X_{pa_i}).$$

It is said that the model obeys the causal Markov hypothesis.

Topological sorting for DAG is a linear ordering of vertices such that for every directed edge $X_j \rightarrow X_i$, vertex $X_j$ comes before $X_i$ in the ordering. The first vertex in topological sorting is always a vertex with in-degree as 0 (a vertex with no incoming edges). The Fig. 2 is a simple illustration where a true topological sorting is “2, 1, 0, 3, 4” or “2, 1, 0, 4, 3”.

It can be seen that topological sorting is not unique.
C. CAUSAL IDENTIFIABILITY ASSUMPTION

When the generation mechanism \( f_j \) is a nonlinear function, the joint distribution function \( P(X) \) no longer uniquely satisfies the faithfulness hypothesis. That is, only a set of Markov equivalent directed acyclic graphs can be obtained, and no unique causal structure graph can be determined. Up to now, several special mechanisms have been proposed to satisfy the hypothesis of identifiability, for example, the linear Gaussian case with equal error variances [8], the linear non-Gaussian ANM [14] and nonlinear Gaussian ANM [9]. In the ANM, structural assignments have the form

\[
X_j = f_j(X_{pa_j}) + E_j, \quad E_j \perp X_{pa_j}.
\]

The function \( f_j \) in ANM is a nonlinear function, and \( E_j \) is an additive Gaussian noise. It is claimed that the function has full structure identifiability under this mechanism. In this paper, we consider the nonlinear Gaussian case, that is, we construct special neural networks to fit the nonlinear requirement of the function \( f_j \) and the Gaussian requirement of the fitted noise \( E_j \).

III. DAG STRUCTURE LEARNING (DAG-CCNN)

In this section, we will define our method DAG-CCNN. In \( A \), an integrated neural network model based on 1D-Conv layer and FC layer is proposed. In \( B \), the basic framework of 1D-Conv layer is further given. In \( C \), the theorem of constructing acyclicity based on eigenvalues is given and proved.

A. AN INTEGRATED NETWORK MODEL BASED ON 1D-CONV LAYER AND FC LAYER

The DAG-CCNN method adopts the nonlinear SCM generation mechanism and learns causal structure from the weighted adjacency matrix by using a 1D-Conv layer and using a FC layer. The overall model can be expressed as

\[
X_j = g_j(f_j(X_{\rightarrow j}, E_j))
\]

\[
= \text{Conv1d}_{j,H} \circ \sigma \circ \text{Conv1d}_{j,H-1} \circ \cdots \circ \text{Conv1d}_{j,1}
\]

\[
\circ \sigma \circ \text{FC}_j(X_{\rightarrow j}; A_j^{(1)}, \ldots, A_j^{(h)}, b_j, E_j).
\]  

Each \( f_j \) is the function fitted by the FC model, and each \( g_j \) is the function fitted by the 1D-Conv. \( f_j \) is mainly used for learning the structure of DAG, whereas \( g_j \) is mainly used for feature extraction and further enhancement of learning nonlinear fitting capabilities. \( X_{\rightarrow j} \) represents the \( d \)-dimensional input vector given by variable \( X_j \) with the weight of 0, and \( \text{Conv1d}_{j,H} \) represents the 1D-Conv layer used to fit the variable \( X_j \). \( \text{FC}_j : \mathbb{R}^d \rightarrow \mathbb{R}^{d \times m_1} \) is an affine linear map, and \( \text{FC}_j(x) = A_j x + E_j + b_j \), where \( E_j \) is \( d \times m_1 \) dimensional noise vector, \( b_j \) is \( d \times m_1 \) dimensional bias vector, and \( d \times m_1 \) is the new dimension of the output after transformation. The FC layer is expressed as

\[
\text{FC}(X_{\rightarrow j}; A^{(1)}, \ldots, A^{(h)})
\]

\[
= \sigma(A^{(h)} \sigma(\cdots \sigma(A^{(2)} \sigma(A^{(1)} X_{\rightarrow j}))))
\]

\[
A^{(l)} \in \mathbb{R}^{d \times \zeta^{(l-1)}}, \quad \zeta^{(0)} = d, \quad l = 1 \cdots d.
\]  

In this paper, \( \theta_j = \{A_j^{(1)}, \ldots, A_j^{(h)}, b_j\} \) represents the set of weight matrix and deviation vector of linear causal mechanism \( f_j \) in the neural network model, then \( f_j \) can be converted into non-parameter \( \theta_j \) for the following optimization solution [24].

In the above FC layer model, \( f_j \) mechanism is used to fit \( \hat{X}_j = f_j(X_{\rightarrow j}, E_j) \) and output a 3-dimensional tensor \( \hat{X}_j \). Next, a 1D-Conv layer based on CNN is constructed, and we use \( \hat{X}_j \in \mathbb{R}^{(n,d,m_1)} \) as input in 1D-Conv layer and output a \( d \)-dimensional vector. The Fig. 3 is the model flow chart, and the 1D filter kernels have size 1, 2, \ldots, \( d \).

B. 1D-CONV LAYER

In the literatures, although many studies have developed detection algorithms for causal structure graphs, their results are usually limited to relatively small training/testing data sets. Compared with causal discovery algorithms based on artificial neural networks or graph neural networks, the DAG-CCNN approach proposes an integrated model of FC layer and 1D-Conv layer, where the main idea of 1D-Conv layer is to encapsulate feature extraction and feature selection classification into different modules and directly show the vector data consisting of each feature column of the original adjacency matrix as input, which can be efficiently trained by proper learn the best features.

As a classic structure of deep learning, CNN has made great developments and extensive applications in image recognition, object detection, face recognition, natural language processing, etc. The 1D-Conv layer is often used to process text and sequence data. The convolution operation of 1D-Conv is shown in Fig. 4.

In Fig. 4, we use a 1D-Conv with three layers. All of the \( k \)-th neuron in the \( l \)-th hidden layer first performs a sequence of convolution operations, then its sum is activated by the activation function \( \sigma \) and subsequently subsampling operations are performed. In this paper, the 1D-Conv takes as input the tensor output from the FC layer shown in Fig. 3 and thus learns to extract such features that can be used for the regression task performed. Thus, both feature extraction and regression operations are fused into a single process that can be optimized to maximize the regression performance. Moreover, since the only expensive operation is the one-dimensional convolution sequence, which is nothing more than a linearly weighted sum of two one-dimensional arrays, it guarantees the main advantage of one-dimensional networks that fitting is of low computational complexity. Such linear operations can be efficiently performed during forward and backward propagation.

The main advantages of the 1D-Conv based approach are 1) its compact architectural configuration (instead of a complex deep architecture) which includes only one-dimensional convolution, making it suitable for real-time directed and acyclic detection, 2) its cost effective and practical real-time hardware implementation, 3) its ability to work without any pre-determined transformation, hand-crafted feature
extraction and feature selection, and 4) its ability to provide effective classifier training with limited training data set and limited number of BP iterations. It is able to train the classifier essentially with a limited number of iterations.

C. A NEW ACYCLIC REPRESENTATION BASED ON EIGENVALUES

Constructing a directed acyclic graph includes two processes: constructing a directed graph and an acyclic graph, where acyclic graphs are graphs without directed loops. The study of how to get a suitable acyclic representation has been the direction that scholars are concerned with in the process of causal discovery. Zheng et al. [23] are probably the first to represent the acyclicity of directed graphs by using the regularization of the trace of the exponential form of the matrix, which transforms a discrete combinatorial optimization problem into an equivalent continuous optimization problem that can be solved using numerical methods. However, the product between matrices should be avoided in the numerical algorithm solution. In NOTEARS, the Hadamard product of adjacency matrices has a computational complexity as high as $O(d^3)$.

In this paper, we use the exponential form of the sum of all eigenvalues of the adjacency matrix to represent the acyclicity of the directed graph in place of the trace-characterizing acyclicity of the exponential form of the Hadamard product of the adjacency matrix, so that we avoid the product operation of the matrix and the new acyclic representation is nearer to $O(d)$, and trains between 5 and 15 times faster than NOTEARS. This method significantly reduces the computational complexity from $O(d^3)$ to $O(d)$, where $d$ is the number of edges in the adjacency matrix.

**Proposition 1:** Let $A \in \mathbb{R}^{d \times d}$ be the (possibly negatively) weighted adjacency matrix of a directed graph and the corresponding eigenvalues be $\lambda_1, \lambda_2, \cdots, \lambda_d$, then the following equation holds:

$$\text{tr}(e^A) = \sum_{i=1}^{d} e^{\lambda_i}. \quad (3)$$
Theorem 1: Let $A \in \mathbb{R}^{d \times d}$ be the (possibly negatively) weighted adjacency matrix of a directed graph and the corresponding eigenvalues be $\lambda_1, \lambda_2, \cdots, \lambda_d$, then the graph is acyclic if and only if:

$$
\sum_{i=1}^{d} e^{\lambda_i} - d = 0. \quad (4)
$$

Proof: It essentially boils down to the fact that $tr(A^k)$ counts the number of length-$k$ closed walks in a directed graph. Clearly, an acyclic graph will have $tr(A^k) = 0$ for all $k = 1, \cdots, \infty$.

Then, using Taylor’s expansion

$$
e^A = \sum_{k=0}^{\infty} \frac{A^k}{k!},$$

we have

$$
tr(e^A) = tr(E) + \sum_{k=1}^{\infty} \frac{tr(A^k)}{k!} = d.
$$

So, it can be substituted into Proposition 1 to get

$$
\sum_{i=1}^{d} e^{\lambda_i} - d = 0.
$$

IV. OPTIMIZATION WITH AUGMENTED LAGRANGE MULTIPLIER METHOD

Theorem 1 establishes a smooth, algebraic, computable characterization of acyclicity. After the acyclic representation of the directed graph was given in Theorem 1, we further solve the continuous optimization problem so as to train the relevant parameters of the integrated model at FC layer and 1D-Conv layer. The constrained optimization problem with least-squares loss function $\ell$ can be formulated as:

$$
\min_{\theta} L(f) = \frac{1}{2n} \sum_{j=1}^{d} \ell(X_j, g_j(f(X_j, E_j)))
$$

subject to $h(W(f)) = 0$,

where $W(f) = W(f_1, f_2, \cdots, f_d) \in \mathbb{R}^{d \times d}$, $A_j = (A_j^{(1)}, A_j^{(2)}, \cdots, A_j^{(H)})$, $h(W(f)) = \sum_{j=1}^{d} e^{\lambda_j} - d$, $f_j \sim FC \in \mathcal{H}^t(R^d) \cup \mathcal{T}$, $1D - Conv$, and $f_j$ denotes the usual Sobolev space of square-integrable functions whose derivatives are also square integrable. Let the partial derivative of $f_j$ with respect to $X_k$ be denoted by $\frac{\partial f_j}{\partial X_k}$.

It is easy to show that $f_j$ is independent of $X_k$ if and only if $\left\| \frac{\partial f_j}{\partial X_k} \right\|_{L^2}$ = 0. Let $\left\| \cdot \right\|_p$ represents the $L_p$-norm of the vector, $\left\| \cdot \right\|_{L^p}$ represents the $L^p$-norm of the function, and $\left\| \cdot \right\|_{p,q}$ represents the $(p, q)$-norm of the matrix. Then, we define the weighted adjacency matrix $W(f) \in \mathbb{R}^{d \times d}$ with entries

$$
[W(f)]_{kj} = \left\| \frac{\partial f_j}{\partial X_k} \right\|_{L^2}, \quad (5)
$$

where $[W(f)]_{kj} = 0$ is equivalent to $\left\| kth - column(A_j^{(1)}) \right\|_{L^2}$. That is, when $X_j$ is independent of $X_k$, column $k$ of $A_j^{(1)}$ is 0.

Therefore, the corresponding non-parametric form is defined as

$$
(W(\theta))_{kj} = \left\| kth - column(A_j^{(1)}) \right\|_{2}. \quad (6)
$$

Next, $L_1$ regularization $L_1 = \left\| A_j^{(1)} \right\|_{1,1}$ is used to constrain the adjacency matrix of the directed graph, that is, a sparse regular term with penalty weight $\mu_1 \geq 0$ is added to the scoring function. In order to essentially prevent overfitting, $L_2$ regularization with weight $\mu_2 \geq 0$ is added to all parameters in the 1D-Conv learning process, that is, $L_2 = \left\| W(1D - Conv) \right\|_{2,2}$, and it is proved that the solution result of adding $L_2$ regularization obtained a better solution. Then, the constrained optimization problem are as follows:

$$
\min_{\theta} \quad L(f) + \mu_1 \left\| A_j^{(1)} \right\|_{1,1} + \frac{1}{2} \mu_2 \left\| W(1D - CNN) \right\|_{2,2}^2
$$

subject to $h(W(f)) = 0$.

As in Zheng et al. [23], the standard machinery of augmented Lagrangian can be applied, resulting in a series of unconstrained problems

$$
\min_{\theta} \quad L(f) + \mu_1 \left\| A_j^{(1)} \right\|_{1,1} + \frac{1}{2} \mu_2 \left\| W(1D - CNN) \right\|_{2,2}^2
$$

$$
+ \alpha h(W(f)) + \frac{1}{2} \rho (h(W(f)))^2. \quad (7)
$$

where $\theta = (\theta_1, \ldots, \theta_d)$, $\rho$ is a penalty parameter and $\alpha$ is the Lagrange multiplier. The corresponding non-parametric form is:

$$
\min_{\theta} \quad L(\theta) + \mu_1 \left\| A_j^{(1)} \right\|_{1,1} + \frac{1}{2} \mu_2 \left\| W(1D - CNN) \right\|_{2,2}^2
$$

$$
+ \alpha h(W(\theta)) + \frac{1}{2} \rho (h(W(\theta)))^2. \quad (8)
$$

We choose the L-BFGS-B algorithm [17], and let $f(\cdot) = f^+(\cdot) - f^-(\cdot)$, so $\theta = \theta^+ - \theta^-$ can replace (8):

$$
\min_{\theta} F_{\rho}(\theta, \xi_1, \xi_2, \alpha, \mu_1, \mu_2)
$$

$$
= \min_{\theta^+ - \theta^-} \quad L(\theta^+ - \theta^-) + \mu_1 \left\| A_j^{(1)} \right\|_{1,1}
$$

$$
+ \frac{1}{2} \mu_2 \left\| W(1D - CNN) \right\|_{2,2}^2 + \alpha h(\theta^+ - \theta^-) + \frac{1}{2} \rho (h(\theta^+ - \theta^-))^2, \quad (9)
$$

where $\xi_1$ and $\xi_2$ are both parameters during training. The update rules are as follows

$$
\theta_{t+1}, \xi_{1,t+1}, \xi_{2,t+1} = \text{arg} \min_{\theta} F_{\rho}(\theta, \xi_1, \xi_2, \alpha_t),
$$

$$
\alpha_{t+1} \left\{ \begin{array}{ll}
\alpha_t + \rho_t h(\theta_{t+1}) & \text{if } h(\theta_{t+1}) \geq \gamma h(\theta_t) \\
\alpha_t & \text{otherwise},
\end{array} \right.
$$

$$
\rho_{t+1} \left\{ \begin{array}{ll}
\eta \rho_t, & \text{if } h(\theta_{t+1}) \geq \gamma h(\theta_t) \\
\rho_t & \text{otherwise},
\end{array} \right.
$$

where $t$ is for the t-iteration.
TABLE 1. Comparison of different methods on nonlinear SCMs generated from the ANM with FC. The lower the better for SHD and the higher the better for TPR \((n = 1000)\).

| Method       | ER2 with 10 nodes | SF2 with 10 nodes | ER2 with 40 nodes | SF2 with 40 nodes |
|--------------|-------------------|-------------------|-------------------|-------------------|
|              | TPR       | SHD        | TPR       | SHD        | TPR       | SHD        | TPR       | SHD        |
| DAG-CCNN     | 0.689±0.11 | 6.5±1.2   | 0.759±0.06 | 4.4±2.5   | 0.861±0.00 | 29.3±2.3  | 0.815±0.06 | 17.1±1.4  |
| GraN-DAG [18]| 0.710±0.08 | 8.3±1.7   | 0.660±0.09 | 8.2±1.8   | 0.835±0.09 | 60.3±2.8  | 0.592±0.04 | 75.6±3.9  |
| NOTEARS [23]| 0.685±0.14 | 20.0±2.1  | 0.812±0.14 | 20.1±3.0  | 0.515±0.17 | 64.4±1.3  | 0.725±0.08 | 70.3±2.8  |
| DAG-GNN [26] | 0.712±0.23 | 15.4±2.4  | 0.714±0.07 | 10.4±3.2  | 0.675±0.08 | 62.7±3.7  | 0.534±0.17 | 58.0±2.9  |

TABLE 2. Comparison of different methods on nonlinear SCMs generated from Gaussian processes (GP) with unit independent Gaussian noise. The lower the better for SHD and the higher the better for TPR \((n = 1000)\).

| Method       | ER2 with 10 nodes | SF2 with 10 nodes | ER2 with 40 nodes | SF2 with 40 nodes |
|--------------|-------------------|-------------------|-------------------|-------------------|
|              | TPR       | SHD        | TPR       | SHD        | TPR       | SHD        | TPR       | SHD        |
| DAG-CCNN     | 0.815±0.03 | 12.1±2.2  | 0.798±0.08 | 13.7±2.3  | 0.735±0.03 | 47.1±3.7  | 0.669±0.03 | 63.4±1.8  |
| GraN-DAG [18]| 0.645±0.08 | 10.3±3.0  | 0.501±0.17 | 13.8±2.7  | 0.575±0.08 | 67.2±4.1  | 0.555±0.09 | 72.3±1.6  |
| NOTEARS [23]| 0.810±0.12 | 23.5±1.5  | 0.756±0.05 | 18.2±1.9  | 0.790±0.13 | 75.1±3.9  | 0.655±0.09 | 75.6±2.5  |
| DAG-GNN [26] | 0.732±0.11 | 20.7±3.9  | 0.632±0.21 | 19.2±3.8  | 0.725±0.21 | 75.0±2.5  | 0.711±0.32 | 71.6±3.0  |

TABLE 3. Comparison on nonlinear SCMs generated from ANM with FC and Gaussian processes (GP) with unit independent Gaussian noise. The lower the better for SHD and the higher the better for TPR \((n = 1000, d = 100)\).

| Method       | ER2 with ANM | SF2 with ANM | ER2 with GP | SF2 with GP |
|--------------|--------------|--------------|------------|-------------|
|              | TPR       | SHD        | TPR       | SHD        | TPR       | SHD        | TPR       | SHD        |
| DAG-CCNN     | 0.711±0.19 | 34.0±3.2   | 0.628±0.12 | 43.0±3.8   | 0.631±0.10 | 48.0±1.0  | 0.650±0.17 | 25.5±1.5   |
| GraN-DAG [18]| 0.712±0.18 | 59.0±1.8   | 0.569±0.19 | 78.1±3.2   | 0.432±0.09 | 71.2±2.4  | 0.375±0.34 | 37.0±1.2   |
| NOTEARS [23]| 0.525±0.34 | 47.0±4.1   | 0.730±0.34 | 75.1±2.0   | 0.683±0.14 | 73.4±1.3  | 0.532±0.04 | 41.3±0.8   |
| DAG-GNN [26] | 0.632±0.13 | 60.4±1.4   | 0.532±0.27 | 69.0±0.7   | 0.675±0.38 | 78.7±3.7  | 0.680±0.07 | 42.0±2.3   |

V. EXPERIMENT
In this section, we present a comprehensive set of experiments to demonstrate the validity of the proposed method DAG-CCNN. In Section A, we show the baseline and evaluation metrics used to compare the validity of the methods. In Section B, synthetic data sets are obtained based on two nonlinear data generation mechanisms, namely the ANM with FC mechanism and the nonlinear Gaussian ANM mechanism, and DAG-CCNN is compared with DAG-GNN [26], NOTEARS [23] and GraN-DAG [18] in the synthetic data sets. To further illustrate the usefulness of the proposed approach, in Section C, we apply DAG-CCNN to the protein data set for the discovery of consensus protein signaling networks. Our implementation is based on PyCharm. In extracting the DAG, we used a threshold of 0.3, as suggested by Zheng et al. [23]. As mentioned above, our method named DAG-CCNN includes a FC layer and two 1D-Conv layers.

A. BASELINES AND METRICS
The proposed DAG-CCNN method is compared with three recently proposed deep learning-based DAG structure learning methods, which are DAG-GNN [26], NOTEARS [23], and GraN-DAG [18].

True Positive Rate (TPR) and Structural Hamming Distance (SHD) are commonly used criteria to evaluate the difference between the learned graph and the true graph. SHD is the minimum number of edge additions (E), deletions (D), and reversals (R) required to convert the estimated graph into a true DAG, and a lower SHD indicates a better estimate. TPR represents the correct rate of identifying the number of edges, where both true positive (TP) and true (T) edges are considered, so a higher TPR indicates a better estimate. It is expressed by the formula

\[ SHD = E + D + R, \quad TPR = \frac{TP}{T}. \]

B. SYNTHETIC DATA SETS
Considering graphs with 10, 40 and 100 nodes, the sample size includes \(n \in \{200, 1000\}\). We consider two graph sampling schemes: Erdos-Renyi (ER) and Scale-Free (SF), the expected node degrees for both ER and SF graphs are 1, 2, 4 and let \(ER_x\) represent the expected node degree of \(x\). At this point, the expected number of generated edges is \(x \cdot d\) and \(d\) represents the number of nodes, which also applies to SFx. Then, we synthesize the data sets according to two data generation schemes. The first data generation scheme is ANM with FC (ANM-FC) [15], where the data are sampled as

\[ X_j = \sigma(X_{pa_j} \cdot W_{1j})W_{2j}, \quad j = 1, \ldots, d. \]

Here, \(\sigma\) represents the sigmoid activation function used to encode nonlinear expressions, \(X_{pa_j}\) represents the input variables in the neural network, \(W_{1j}\) is the weight of the first hidden layer, and \(W_{2j}\) is the weight of the second hidden layer. The second data generation scheme is nonlinear Gaussian
FIGURE 5. Structure recovery measured by SHD (lower is better) and TPR (higher is better) to ground truth. Left: the average of SHD for all generation mechanisms. Right: the average of TPR for all generation mechanisms. Rows: four methods. Columns: the values of SHD and TPR. Either DAG-CCNN or GraN-DAG achieves competitive accuracy compared to NOTEARS (n = 200).

ANM (Gauss-ANM) [12], where the data are sampled as

\[ X_j = f_j(W_j, X_{pa_j})W_{2j} + E_j, \quad E_j \sim N(0, \sigma_j^2), \quad j = 1, \ldots, d, \]

where each function \( f_j \) is sampled from a Gaussian process (GP) with a radial basis function (RBF) kernel of bandwidth 1, the noise variables \( E_j \) are independently and normally distributed with uniformly chosen variance, and \( X_{pa_j} \) corresponds to the direct parents of \( X_j \). Thus, both the ANM scheme with FC and the nonlinear Gaussian ANM scheme are sufficient to recover the casual graph structure.

Table 1 and Table 2 show summaries for \( n = 1000 \) and \( d = \{10, 40\} \). Table 3 shows summaries for \( n = 1000 \) and \( d = 100 \).

For the SHD (Taking SHD as a criterion), DAG-CCNN is the best. As we can see from Table 1, no matter for “ER2 with 10 nodes”, “SF2 with 10 nodes”, “ER2 with 40 nodes”, or “SF2 with 40 nodes”, DAG-CCNN obtains the smallest values of SHD. As we can see from Table 2, for “SF2 with 10 nodes”, “ER2 with 40 nodes”, and “SF2 with 40 nodes”, DAG-CCNN also obtains the smallest values of SHD. As we can see from Table 3, the SHD of “ER2 with 100 nodes” and “SF2 with 100 nodes” are larger than that of nodes 10 and 40, which is inevitable because as the node dimension becomes higher, the causal structure graph becomes more complex and the combination space becomes larger making the SHD larger. However, from the overall trend, DAG-CCNN is still the best performer and still has the smallest SHD among the four methods, which indicates that the remaining three methods are less effective in recovering causal structure graphs composed of high-dimensional nodes.

For the TPR (Taking TPR as a criterion), DAG-CCNN is also competitive. As we can see from Table 1, for “ER2 with 40 nodes” and “SF2 with 40 nodes”, DAG-CCNN obtains the largest values of TPR, and for “ER2 with 10 nodes” and “SF2 with 10 nodes”, the TPR value of DAG-CCNN are closed to the largest one. As we can also see from Table 2, “ER2 with 10 nodes”, “SF2 with 10 nodes”, and “SF2 with 40 nodes”, DAG-CCNN obtains the largest values of TPR. As we can see from Table 3, as the node dimension becomes higher, the TPR for “ER2 with 100 nodes” and “SF2 with 100 nodes” are not as high as that for nodes 10 and 40, but, from the overall trend, DAG-CCNN still performs the best and still has the largest TPR among the four methods, which indicates that our method is effective in dealing with high-dimensional causal structure graphs.

Overall, the proposed DAG-CCNN method obtained the best SHD (the lower the better) and TPR (the higher the better) when applied to the simulated data sets generated by the two data generation mechanisms proposed above. When the nodes are 10, the GraN-DAG method also has a lower SHD and higher TPR, but as the nodes grow to 40 and 100, it is clearly observed that it has a correspondingly higher SHD, which indicates that there are more challenges in training gradient-based neural networks as the number of nodes grows. Apart from that, it can be observed that the NOTEARS method does not perform well, which can be partly explained by the fact that it can only be used for linear generation mechanisms, while the synthetic data sets in this paper is modeled with a nonlinear mechanism. Similarly, the poor performance of the DAG-GNN method is perhaps attributed to its use of a strong form of parameter sharing for the neural

| method      | TPR  | SHD  |
|-------------|------|------|
| DAG-CCNN    | 0.868| 12   |
| GraN-DAG    | 0.745| 13   |
| NOTEARS     | 0.797| 19   |
| DAG-GNN     | 0.835| 16   |

FIGURE 6. Causal protein network obtained with DAG-CCNN.

TABLE 4. Comparison of experimental results of four methods to discover causal protein networks. The lower the better for SHD and the higher the better for TPR.
TABLE 5. Comparison of different methods on nonlinear SCMs generated from the ANM with FC. The lower the better for SHD and the higher the better for TPR \((n = 200)\).

| Method          | ER2 with 10 nodes | SP2 with 10 nodes | ER2 with 40 nodes | SP2 with 40 nodes |
|-----------------|-------------------|-------------------|-------------------|-------------------|
|                 | TPR               | SHD               | TPR               | SHD               |
| DAG-CCNN        | 0.732 ± 0.12      | 4.3 ± 1.4         | 0.730 ± 0.13      | 4.5 ± 2.0         |
| GraN-DAG [18]   | 0.750 ± 0.06      | 4.6 ± 1.0         | 0.708 ± 0.12      | 5.6 ± 1.3         |
| NOTEARS [23]    | 0.582 ± 0.08      | 14.4 ± 1.7        | 0.806 ± 0.12      | 15.4 ± 1.0        |
| DAG-GNN [26]    | 0.732 ± 0.18      | 15.0 ± 1.9        | 0.812 ± 0.09      | 6.3 ± 0.9         |

TABLE 6. Comparison of different methods on nonlinear SCMs generated from Gaussian processes (GP) with unit independent Gaussian noise. The lower the better for SHD and the higher the better for TPR \((n = 200)\).

| Method          | ER2 with 10 nodes | SP2 with 10 nodes | ER2 with 40 nodes | SP2 with 40 nodes |
|-----------------|-------------------|-------------------|-------------------|-------------------|
|                 | TPR               | SHD               | TPR               | SHD               |
| DAG-CCNN        | 0.701 ± 0.15      | 10.1 ± 2.0        | 0.738 ± 0.13      | 12.3 ± 2.1        |
| GraN-DAG [18]   | 0.630 ± 0.04      | 8.3 ± 1.0         | 0.651 ± 0.12      | 11.8 ± 2.0        |
| NOTEARS [23]    | 0.761 ± 0.10      | 13.1 ± 1.1        | 0.767 ± 0.08      | 18.8 ± 1.0        |
| DAG-GNN [26]    | 0.810 ± 0.23      | 15.5 ± 3.0        | 0.600 ± 0.12      | 14.4 ± 2.0        |

TABLE 7. Comparison on nonlinear SCMs generated from ANM with FC and Gaussian processes (GP) with unit independent Gaussian noise. The lower the better for SHD and the higher the better for TPR \((n = 200, d = 100)\).

| Method          | ER2 with ANM | SP2 with ANM | ER2 with GP | SP2 with GP |
|-----------------|--------------|--------------|-------------|-------------|
|                 | TPR          | SHD          | TPR         | SHD         |
| DAG-CCNN        | 0.686 ± 0.12 | 38.2 ± 1.0   | 0.647 ± 0.23 | 43.0 ± 0.9  |
| GraN-DAG [18]   | 0.673 ± 0.23 | 47.0 ± 1.8   | 0.530 ± 0.22 | 59.4 ± 1.2  |
| NOTEARS [23]    | 0.545 ± 0.12 | 43.0 ± 1.2   | 0.634 ± 0.20 | 49.9 ± 1.0  |
| DAG-GNN [26]    | 0.532 ± 0.26 | 45.0 ± 1.2   | 0.557 ± 0.43 | 53.0 ± 0.2  |

C. REAL DATA SET

We compared DAG-CCNN with three other methods on the real data set provided by Sachs et al. [10]. This data set includes continuous measurements of protein and phospholipid expression levels in human immune system cells \((n = 7466, d = 11)\). This data set has been widely used in the study of graphical models and its experimental annotations are widely accepted by the biological research community. Sachs’ true causal graph with 11 nodes and 17 edges as shown by Sachs et al. [10].

On this data set, DAG-CCNN achieved the best SHD of 12, whereas GraN-DAG had a SHD of 13. DAG-GNN and NOTEARS close to GraN-DAG in terms of TPR, but they had a lower SHD. In particular, the SHD of DAG-GNN and NOTEARS are 16 and 19, respectively. 10-fold cross-validation was performed for each algorithm, and the average results are reported in Table 3 and the causal protein networks obtained from CCNN are reported in Fig. 6.

VI. CONCLUSION

Maximizing the recovery of cause-effect outcome diagrams and finding causal relationships between variables will help solve many difficult problems in real life. This requires us to continuously improve and innovate the algorithms. In this study, a causal structure learning method based on an integrated network model of FC layer and 1D-Conv layer is proposed. First, the adjacency matrix of the directed graph is obtained by training the FC model, and then the learning capability is further enhanced by 1D-Conv to obtain the directed graph. In addition, the method further gives a new characterization of the acyclicity of the directed graph by using the exponential sum of the eigenvalues of the adjacency matrix, thus avoiding the operational complexity difficulties associated with the product of adjacency matrices.

In future studies, we will explore the causal structure learning method for high-dimensional data samples. On the one hand, we will improve the correct rate of identifying causal directions and further weaken the causality assumption, and on the other hand, we will reduce the computational complexity of neural networks, so as to achieve the accurate discovery of causal structure graphs of high-dimensional data.
APPENDIX

Following the above experimental section, the experimental results obtained under different causal generation mechanisms for sample size \( n = 200 \) are detailed in Tables 5, 6 and 7.

When the sample size \( n = 200 \), the results obtained are consistent with the overall trend when the sample size \( n = 1000 \). The difference is that when the sample size is smaller, the experimental results vary more and there is a chance bias. The same is that, with the increase of sample nodes, all four methods are less effective than those recovered by low-dimensional sample nodes in recovering the causal structure graph. However, it is obvious that among these methods, DAG-CCNN performs the best, with higher TPR and lower SHD in comparison.

ACKNOWLEDGMENT

The authors would like to thank the editor and the anonymous reviewers for the insightful comments and suggestions.

REFERENCES

[1] A. Mnih and K. Gregor, “Neural variational inference and learning in belief networks,” in Proc. 31st Int. Conf. Mach. Learn., Jun. 2014, pp. 1791–1799.
[2] D. M. Chickering, “Optimal structure identification with greedy search,” J. Mach. Learn. Res., vol. 3, no. 1, pp. 507–554, 2002.
[3] G. Borboudakis and I. Tsamardinos, “Towards robust and versatile causal discovery for business applications,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 1435–1444.
[4] G. N. Saxe, S. Ma, L. J. Morales, I. R. Galatzer-Levy, C. Aliferis, and C. R. Marmar, “Computational causal discovery for post-traumatic stress in police officers,” Transl. Psychiatry, vol. 10, no. 1, pp. 1–12, Dec. 2020.
[5] H. Ramchoun, M. A. J. Idrissi, and Y. Ghanou, “Multilayer perceptron: Architecture optimization and training,” Int. J. Interact. Multim. Artif. Intell., vol. 4, no. 1, pp. 26–30, Apr. 2016.
[6] J. M. Mooij, J. Peters, D. Janzing, J. Zscheischler, and B. Schölkopf, “Distinguishing cause from effect using observational data: Methods and benchmarks,” J. Mach. Learn. Res., vol. 17, no. 1, pp. 1103–1204, 2016.
[7] P. J. Models, Reasoning and Inference. Cambridge, U.K.: Cambridge Univ. Press, 2000, p. 19.
[8] J. Peters and P. Bühlmann, “Identifiability of Gaussian structural equation models with equal error variances,” Biometrika, vol. 101, no. 1, pp. 219–228, Mar. 2014.
[9] J. Peters, J. M. Mooij, D. Janzing, and B. Schölkopf, “Causal discovery with continuous additive noise models,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 2009–2053, 2014.
[10] 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, Apr. 2005.
[11] K. Zhang and A. Hyvarinen, “On the identifiability of the post-nonlinear causal model,” in Proc. 25th Conf. Uncertainty Artif. Intell., Jun. 2009, pp. 647–655.
[12] K. Zhang, Z. Wang, J. Zhang, and B. Schölkopf, “On estimation of functional causal models: General results and application to the post-nonlinear causal model,” ACM Trans. Intell. Technol., vol. 7, no. 2, pp. 1–22, Jan. 2016.
[13] D. M. Chickering, D. Heckerman, and C. Meek, “Large-sample learning of Bayesian networks is NP-hard,” J. Mach. Learn. Res., vol. 5, pp. 1287–1330, Oct. 2004.
[14] P.-L. Loh and P. Bühlmann, “High-dimensional learning of linear causal networks via inverse covariance estimation,” J. Mach. Learn. Res., vol. 15, no. 1, pp. 3065–3105, 2014.
[15] P. O. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf, “Non-linear causal discovery with additive noise models,” in Proc. Adv. Neu. Inf. Process. Syst. (NIPS), Dec. 2008, pp. 689–696.
[16] P. Spirites, C. Glymour, and R. Scheines, Causation, Prediction, and Search, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.
[17] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, “A limited memory algorithm for bound constrained optimization,” SIAM J. Sci. Comput., vol. 16, no. 5, pp. 1190–1208, Sep. 1995.
[18] S. Lachapelle, P. Brouillard, and T. Deleu, “Gradient-based neural DAG learning,” in Proc. 80th Int. Conf. Learn. Represent. (ICLR), Apr. 2020, pp. 1–23.
[19] S. Kiranyaz, T. Ince, R. Hamila, and M. Gabbouj, “Convolutional neural networks for patient-specific ECG classification,” in Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC), Aug. 2015, pp. 2608–2611.
[20] S. Shimizu, P. O. Hoyer, A. Hyvärinen, and A. Kerminen, “A linear non-Gaussian acyclic model for causal discovery,” J. Mach. Learn. Res., vol. 7, no. 10, pp. 2003–2030, 2006.
[21] S. Zhu, J. Ng, and Z. Chen, “Causal discovery with reinforcement learning,” in Proc. 80th Int. Conf. Learn. Represent. (ICLR), Apr. 2020, pp. 1–17.
[22] T. Ince, S. Kiranyaz, L. Eren, M. Askar, and M. Gabbouj, “Real-time motor fault detection by 1-D convolutional neural networks,” IEEE Trans. Ind. Electron., vol. 63, no. 11, pp. 7067–7075, Nov. 2016.
[23] X. Zheng, B. Aragam, P. Ravikumar, and E. P. Xing, “DAGs with NO TEARS: Continuous optimization for structure learning,” in Proc. 31st Conf. Neu. Inf. Process. Syst. (NIPS), Dec. 2018, pp. 1–12.
[24] X. Zheng, C. Dan, and B. Aragam, “Learning sparse nonparametric DAGs,” in Proc. 23rd Int. Conf. Artif. Intell. Statist. (AISTATS), Apr. 2020, pp. 3414–3425.
[25] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[26] Y. Yu, J. Chen, and T. Gao, “DAG-GNN: DAG structure learning with graph neural networks,” in Proc. 36th Int. Conf. Mach. Learn. (ICML), Jun. 2019, pp. 7154–7163.

CHUANYU XU received the B.S. degree in information and computing sciences from Linyi University, Shandong, China, in 2019. She is currently pursuing the M.S. degree in mathematics with the University of Science and Technology Beijing, Beijing, China. Her current research interests include machine learning, data mining, and causality discovery.

WEI XU received the B.S. degree from Shandong University, China, in 2007, and the M.S. and Ph.D. degrees from the University of Science and Technology Beijing, China, in 2009 and 2014, respectively. From 2014 to 2016, he did postdoctoral work at the University of Aeronautics and Astronautics, Beijing. Since 2016, he has been working as a Lecturer with the School of Mathematics and Science, University of Science and Technology Beijing. His current research interests include computational complexity, spin glass theory, and constraint satisfaction problem. ** **