



# LeCaSiM: Learning Causal Structure via Inverse of M-Matrices with Adjustable Coefficients

Qingsong Cai<sup>1</sup> · Yongchang Zhang<sup>1</sup>

Accepted: 8 December 2023 / Published online: 15 February 2024  
© The Author(s) 2024

## Abstract

The objective of causal discovery is to uncover the causal relationships among natural phenomena or human behaviors, thus establishing the basis for subsequent prediction and inference. Traditional ways to reveal the causal structure between variables, such as interventions or random trials, are often deemed costly or impractical. With the diversity of means of data acquisition and abundant data, approaches that directly learn causal structure from observational data are attracting more interest in academia. Unfortunately, learning a directed acyclic graph (DAG) from samples of multiple variables has been proven to be a challenging problem. Recent proposals leverage the continuous optimization method to discover the structure of the target DAG from observational data using an acyclic constraint function. Although it has an elegant mathematical form, its optimization process is prone to numerical explosion or the disappearance of high-order gradients of constraint conditions, making the convergence conditions unstable and the result deviating from the true structure. To tackle these issues, we first combine the existing work to give a series of algebraic equivalent conditions for a DAG and its corresponding brief proofs, then leverage the properties of the M-matrix to derive a new acyclic constraint function. Based on the utilization of this function and linear structural equation models, we propose a novel causal discovery algorithm called LeCaSiM, which employs continuous optimization. Our algorithm exploits the spectral radius of the adjacency matrix to dynamically adjust the coefficients of the matrix polynomial, effectively solving the above problems. We conduct extensive experiments on both synthetic and real-world datasets, the results show our algorithm outperforms the existing algorithms on both computation time and accuracy under the same settings.

**Keywords** Causal discovery · Observational data · Directed acyclic graph · M-matrix · Acyclicity constraint

---

✉ Qingsong Cai  
caiqs@btbu.edu.cn

Yongchang Zhang  
zhangyongchang@st.btbu.edu.cn

<sup>1</sup> School of Computer Science and Engineering, Beijing Technology and Business University, No. 33, Fucheng Road, Haidian District, Beijing 100048, China

## 1 Introduction

With the emergence of the big data era, machine learning methods have experienced rapid development, and artificial intelligence has achieved significant research breakthroughs in recent years. Its application has rapidly expanded to numerous fields of economic and social development. However, researchers have recognized that most existing machine learning models are primarily based on the study of variable correlations, disregarding the underlying causality in the data. This not only compromises the interpretability of the models but also diminishes their inference capabilities [17]. Understanding the causality between variables and leveraging causal relationships for inference is a crucial aspect of human intelligence. Intelligence that is built upon causal reasoning is a key criterion for assessing the gap between machine models and human intelligence. The fundamental distinction between causality and correlation lies in the inherent structural asymmetry of causality [24]. While correlation primarily focuses on describing the similarity in changing trends between variables, causality emphasizes the directional cause-and-effect relationship among samples. This distinction offers the potential for achieving accurate reasoning. Presently, there is a growing interest among researchers in causal reasoning based on causal relationships. The objective of causal research is to unveil the causal relationships between variables and construct causal models (causal discovery), followed by predicting causal effects and estimating variable changes based on this knowledge (causal inference). Causal research holds wide applicability in various domains, including social computing [10], policy evaluation [12], medical diagnosis [19], and financial market forecasting [11].

Causal discovery, as a crucial prerequisite for causal reasoning, has garnered significant attention in the academic community in recent years [8]. Traditionally, a common approach to causal discovery is through randomized controlled trials (RCTs), which involve interventions or random experiments to obtain sample data and unveil the causal structure between variables. However, RCTs often encounter issues such as high costs and limited practicality [26]. For instance, it is impractical to impose smoking or smoking cessation interventions on the subjects under investigation to study the impact of smoking on health. With the diversification of data acquisition methods and the increasing availability of abundant data, fields like sociology, biomedicine, economics, and environmental science have amassed large amounts of observational data. This data consists of direct observations of facts or phenomena and encompasses the underlying true distribution patterns. Although observational data may contain hidden variables and selection biases, researchers still believe that unaltered observational data can better reflect the true distribution of samples compared to intervention data, making them more generalizable. Consequently, the direct exploration of causal relationships between variables from observational sample data has emerged as a research focal point in statistics and artificial intelligence [4]. In order to address the challenges posed by observational data, researchers have developed various causal relationship models based on a series of fundamental assumptions (such as the sufficiency assumption and Markov assumption) to characterize the causal relationships between random variables [18, 21]. Among these models, the Structural Causal Model (SCM) proposed by J. Pearl has gained wide adoption in the field of explainable artificial intelligence. In SCM, a DAG is utilized to depict the causal relationships between multiple variables, where nodes represent random variables and directed edges signify causal relationships between variables. The objective of causal structure discovery is to learn a DAG that accurately represents the causal relationships between variables from the observed sample data. This task is commonly referred to as DAG structure learning or the Bayesian network structure learning problem [18].

Traditional algorithms for causal discovery based on SCM often approach the problem of learning causal structures as an optimization task involving scoring function values. Acyclicity constraints are introduced on sets of directed graphs, and candidate weight adjacency matrices are iteratively adjusted to optimize the scoring values and determine the final graph structure [5]. However, as the number of nodes increases, the exponential growth of the search space for DAGs makes this problem NP-hard [6], posing a significant challenge for researchers. In the 2018 NIPS conference, Zheng et al. [31] proposed a novel approach that employs continuous optimization under acyclicity constraints to solve the graph structure. This approach cleverly exploits the relationship between the diagonal elements in the  $k$ th power of the adjacency matrix and acyclicity, and by leveraging the convergence properties of matrix exponentiation, a continuous function is introduced to characterize the features of DAG structures. Consequently, the problem of DAG structure learning is recast as a continuous optimization problem under acyclicity constraints. This innovative approach has paved the way for new research directions and has generated considerable interest from researchers, establishing continuous optimization methods as a prominent focus in the field of causal structure discovery. Although this method exhibits elegant mathematical form, the computation of acyclicity constraints requires high-order power operations on the weight adjacency matrix, and the coefficients of high order terms can affect their numerical values. If the coefficients are too large, it may lead to numerical exploding. Therefore, some research efforts [27, 31] have attempted to use very small coefficients for high-order terms to mitigate this issue. However, another study [30] argues that high-order terms contain crucial information about larger cycles. If the coefficients are too small, there is a higher likelihood for the cycle information and gradient information within the higher-order terms to disappear. This leads to unstable convergence conditions and deviations from the true graph structure. Conversely, they argue that using larger coefficients can help detect larger cycles in the graph.

To address the aforementioned contradiction, this paper introduces the M-matrices [15] and its properties to propose a new acyclic constraint continuous function. This function allows for adaptive adjustment of the coefficients of higher-order terms. During the initial stage of optimization, where a significant disparity exists between the candidate graph and the target graph (as indicated by a large spectral radius for the candidate graph and a zero spectral radius for the target DAG), smaller coefficients are assigned to the higher-order terms to prevent numerical instability. As the optimization progresses and the candidate graph approaches the target graph, with the spectral radius approaching zero, larger coefficients are assigned to the higher-order terms to prevent numerical instability and facilitate the identification of larger cycle information in the DAG structure. Based on the aforementioned analysis, the proposed acyclic constraint with adjustable coefficients addresses both the potential issues of numerical instability and gradient disappearance during optimization. Building upon this acyclic constraint, this paper presents a novel causal structure discovery algorithm called LeCaSiM (learning causal structure via inverse of M-matrices with adjustable coefficients). Experimental results on both synthetic and real datasets demonstrate that LeCaSiM effectively learns the causal structure between variables from observational data, surpassing the efficiency and accuracy of existing algorithms. We present the following summary of our contributions:

- In order to precisely characterize the structural features of DAGs, we present a summary of seven algebraic representations that are proven to be equivalent to DAGs. Additionally, concise proofs are provided for each of these representations.
- We present theoretical proofs of the potential issues of numerical exploding and gradient vanishing that can arise in existing acyclicity constraints. To overcome these challenges,

the concept of M-matrices is introduced, and a novel acyclicity constraint function based on the trace of the inverse of M-matrices is proposed. Furthermore, we offer rigorous theoretical analyses to substantiate the effectiveness of the newly proposed constraint function in addressing the aforementioned issues.

- Based on the newly introduced acyclic constraint function and linear structural equation models, we utilize the augmented Lagrangian framework and present the LeCaSiM algorithm to transform the problem of learning DAG structures into a continuous optimization problem.
- To evaluate the accuracy and efficiency of the LeCaSiM algorithm for DAG structure learning, we perform extensive experiments on both synthetic and real-world datasets. The experimental results unequivocally indicate that our proposed algorithm surpasses existing methods in terms of accuracy and efficiency.

The remaining structure of this paper is organized as follows: Sect. 2 introduces the background and related work of causal discovery and DAG structure learning. Section 3 uses algebraic graph theory to provide several equivalent conditions and brief proofs for characterizing the structural characteristics of DAG, and it derives a new continuous optimization constraint function based on the M-matrices. Section 4 introduces and analyzes the proposed causal structure learning algorithm, LeCaSiM. Section 5 presents the experimental verification and result analysis. Section 6 concludes the paper with a brief discussion and outlines possible future research directions.

## 2 Background and Related Work

### 2.1 Symbols and Notations

To ensure clarity, we present Table 1, which provides the symbols used throughout this article along with their corresponding meanings.

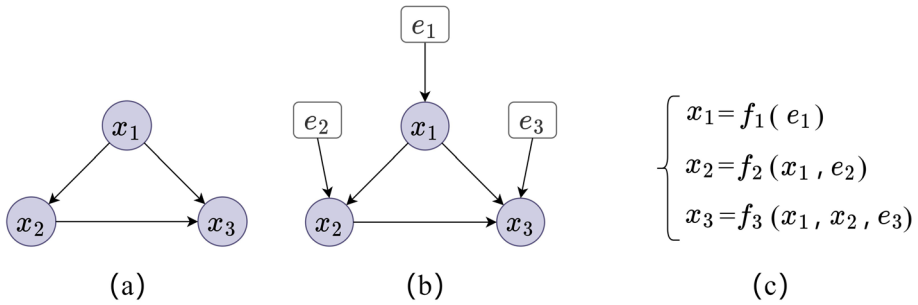
### 2.2 Structural Causal Model

The SCM proposed by J. Pearl comprises two components: the causal graph and the structural equation model (SEM). The causal graph is a DAG used to represent the causal relationships between variables, denoted as  $\mathcal{G} = (\mathbb{V}, \mathbb{E})$ , where  $|\mathbb{V}| = d$ . For a  $d$ -dimensional random vector  $\mathbf{x} = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ , let  $\mathcal{P}(\mathbf{x})$  represents the joint probability distribution of  $\mathcal{G}$ . Node  $i \in \mathbb{V}$  corresponds to the random variable  $x_i \in \mathbf{x}$ , and a directed edge  $(i, j) \in \mathbb{E}$  indicates a causal relationship between  $x_i$  and  $x_j$ . The set of parent nodes of node  $i$  is denoted as  $pa(i)$ . Under the assumptions of sufficiency and faithfulness [21], if all variables satisfy the property of first-order Markov [33],  $\mathcal{P}(\mathbf{x})$  can be decomposed as  $\mathcal{P}(\mathbf{x}) = \prod_{i=1}^d p(x_i | x_{pa(i)})$ . If we consider  $x_i$  as a specific function of  $x_{pa(i)}$ , the causal relationships implied by the DAG correspond to a set of structural equations. Let  $e$  be a noise variable (or exogenous variable), then, the causal relationships between variables can be represented by the following SEM:

$$x_i = f_i(x_{pa(i)}, e_i), e_i \perp x_{pa(i)} \quad (1)$$

**Table 1** Symbols and meanings

Symbol	Meaning	Symbol	Meaning
$\circ$	Hadarnard product operation	$\mathbf{A}$	Adjacency matrix
$\perp$	variables are mutually independent	$\mathcal{G}$	Directed acyclic graph
$d$	Number of nodes	$\mathbb{G}$	Directed graph
$\mathbf{I}$	$d$ -dimensional identity matrix	$\mathbb{V}$	Set of nodes
$\mathbf{e} = (e_1, e_2, \dots, e_d)^T$	$d$ -dimensional noise vector	$\mathbb{E}$	Set of directed edges
$\mathbf{x}_i = (x_1, x_2, \dots, x_d)^T$	$i$ -th $d$ -dimensional random vector	$\lambda(\cdot)$	Matrix eigenvalue
$\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$	Observed data matrix	$\rho(\cdot)$	Matrix spectral radius
$\mathbf{w}_i = (w_{1i}, w_{2i}, \dots, w_{di})^T$	$i$ -th weight vector	$\mathbb{R}$	Set of real numbers
$\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_d)$	Weighted adjacency matrix	$\mathbb{Z}^+$	Set of positive integers



**Fig. 1** **a** The causal graph of the three variables. **b** SCM corresponding to the three variables. **c** The corresponding set of structural equations

where  $i \in \{1, \dots, d\}$ . As previously mentioned, the faithfulness assumption establishes a one-to-one correspondence between  $\mathcal{P}(\mathbf{x})$  and the DAG, which also corresponds to a set of structural equations. Figure 1 illustrates an example of an SCM with three variables.

### 2.3 DAG Structure Learning

As described in Sect. 2.1, the problem of learning the DAG structure based on SCM can be defined as follows: Given the data matrix  $\mathbf{X}$ , the goal is to learn the causal graph  $\mathcal{G} \in \mathbb{S}_{\mathcal{G}}$  under the joint probability distribution  $\mathcal{P}(\mathbf{X})$ .

Following prior research, we also employ the linear structural equation model (LSEM) to depict the causal relationships between variables. Let  $\mathbf{W} \in \mathbb{R}^{d \times d}$  as the weighted adjacency matrix corresponding to graph  $\mathcal{G}$  (the weight values are used to characterize the causal strength), where  $\mathbf{w}_i \in \mathbb{R}^d$ , then in LSEM,  $\mathbf{x} = \mathbf{W}^T \mathbf{x} + \mathbf{e}$ ,  $\mathbf{e} = (e_1, e_2, \dots, e_d)^T$  denotes  $d$  i.i.d. noise terms. Correspondingly, the structural equations of the three nodes in the DAG in Fig. 1 can be expressed as follows ( $w_{ij} \in \mathbb{R}$  represents the weight value of the directed edge between nodes  $i$  and  $j$ ):

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{bmatrix} 0 & w_{12} & w_{13} \\ 0 & 0 & w_{23} \\ 0 & 0 & 0 \end{bmatrix}^T \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

then  $\begin{cases} x_1 = e_1 \\ x_2 = w_{12}x_1 + e_2 \\ x_3 = w_{13}x_1 + w_{23}x_2 + e_3 \end{cases}$

Let  $\mathbb{G}(\mathbf{W})$  denotes the directed graph corresponding to  $\mathbf{W}$ ,  $\mathbb{S}_{\mathcal{G}}$  indicates the search space of the DAG, and the scoring function  $Q(\mathbf{W}, \mathbf{X})$  is utilized to evaluate the similarity between  $\mathbb{G}(\mathbf{W})$  and the observed data  $\mathbf{X}$ . Consequently, the problem of learning the DAG structure can be formalized as the following optimization problem:

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times d}} Q(\mathbf{W}, \mathbf{X}) \quad s.t. \quad \mathbb{G}(\mathbf{W}) \in \mathbb{S}_{\mathcal{G}} \tag{2}$$

### 2.4 Related Work

Algorithms aimed at learning causal structures between variables from observational data can generally be classified into two types: constraint-based algorithms and score-based algo-

rithms. Constraint-based algorithms initiate the process with an undirected complete graph and systematically examine the conditional independence between variable pairs. Eventually, they produce a completed partially directed acyclic graph (CPDAG). The CPDAG represents a Markov equivalence class of the true DAG and does not provide a deterministic causal relationship. Prominent algorithms in this category include the PC (Peter–Clark) algorithm [21] and the IC (inductive causation) algorithm [23].

In this research, we adopt a score-based approach for DAG structure learning. The traditional scoring-function-based algorithms employ combinatorial optimization to optimize the scoring function while adhering to the acyclicity constraint (such as Eq. 2). The scoring function is employed to evaluates the fit of the weighted adjacency matrix to the observed data. To enforce the acyclicity constraint, various implementation strategies involving combinations, search methods, and other techniques are utilized. Commonly used scoring functions include the Bayesian Information Criterion (BIC) [14], Minimum Description Length (MDL) [3], and Bayesian Dirichlet equivalent uniform (BDeu) [9]. Search strategies often encompass the hill climbing algorithm proposed by Heckerman et al. [9], A\* search [29], integer programming [7], and greedy algorithm [5]. However, Since the search space of DAGs grows super-exponentially with the number of nodes, the task of learning specific graph structures using combinatorial optimization-based algorithms is known to be NP-hard [6]. Robinson [20] gives a recursive formula for calculating the number of DAGs that can be constructed from  $d$  nodes:  $|\mathcal{G}_d| = \sum_{i=1}^d (-1)^{i-1} \binom{d}{i} 2^{i(d-i)} |\mathcal{G}_{d-i}|$ , where  $|\mathcal{G}_0| = 1$ . It is evident from Fig. 2 that as the number of nodes increases, both the counts of directed graphs and DAGs exhibit exponential growth, while the proportion of DAGs in the directed graphs decreases rapidly. As a result, the computational effort required to learn the desired DAG in such a vast space of directed graphs using the aforementioned combinatorial optimization algorithms is substantial.

To improve the accuracy and scalability of algorithms for learning directed acyclic graph (DAG) structures, Zheng et al.[31] discover that the trace of the matrix exponential effectively captures cycle information in a directed graph. Leveraging this finding, they develop a continuous acyclicity function denoted as  $h(\mathbf{W})$  that serves as a replacement for the combinatorial constraint  $G(\mathbf{W})$  in the optimization process. This transformative step converts the DAG structure learning problem into a continuous optimization problem, allowing for iterative solving through the utilization of existing numerical optimization algorithms[2].

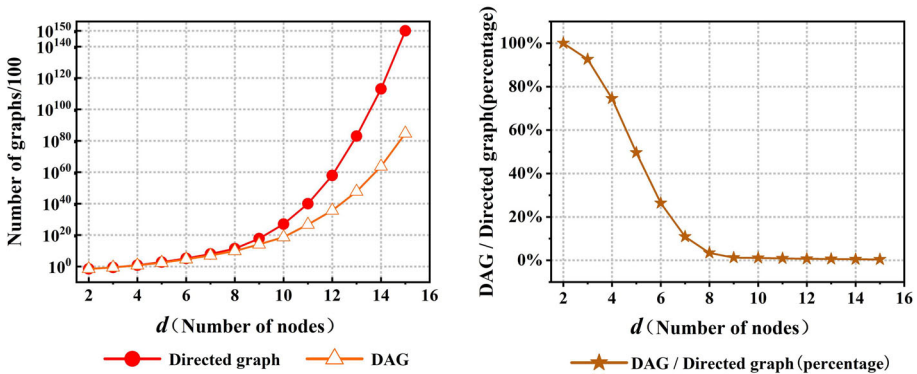


Fig. 2 Left The counts of DAGs and directed graphs for specific nodes. Right The proportion of DAGs within the directed graphs for specific nodes

However, a challenge arises in the form of high-order information loss during the solution process.

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times d}} Q(\mathbf{W}, \mathbf{X}) \quad s.t. \quad h(\mathbf{W}) = 0 \tag{3}$$

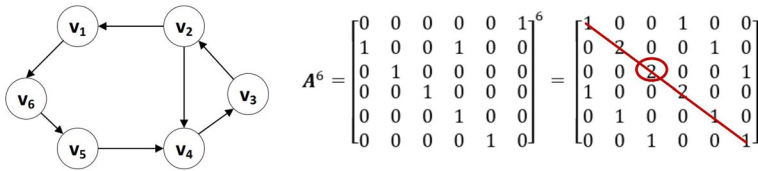
Building upon the aforementioned research, Yu et al. [27] introduce a novel acyclicity constraint for DAG learning that utilizes matrix binomials. Their approach is based on the important observation that a directed graph with  $d$  nodes does not contain any maximal cycles exceeding  $d$  steps. As a result, it is only necessary to compute powers of the weight adjacency matrix up to order  $d$ . This reduction in computational burden enhances the efficiency of the algorithm. However, this approach does not address the problem of high-order information loss. Wei et al. [25] introduce a combined polynomial constraint, replacing the Hadamard product with matrix absolute values, which is utilized for structure learning. Similarly, Lee et al. [13] and Zhu et al. [33] propose alternative constraints based on the spectral radius of the weighted adjacency matrix. These constraints have the notable effect of significantly reducing the computational complexity of the algorithm. In the algorithm proposed by Yu et al. in 2021 [28], there is a departure from previous methods that relied on the relationship between matrix powers and cycles in the graph to enforce acyclicity constraints. Instead, they employ a set of weighted graph potentials with gradients to describe the space of DAGs, enabling direct continuous optimization in algebraic space. Other related works include the extension of Zheng et al.’s work to implement DAG structure learning using time-series data [16] and the utilization of reinforcement learning methods to search for the optimal DAG [34], among other approaches.

### 3 Algebraic Characterization of DAGs

The construction of smooth acyclicity constraint functions plays a crucial role in converting the problem of learning the structure of a DAG from combinatorial optimization to continuous optimization (Eq. 2  $\rightarrow$  Eq. 3). In this section, we present various algebraic characterizations of DAG from the viewpoint of algebraic graph theory. Utilizing these algebraic characterizations, our objective is to devise smooth acyclicity constraint functions that facilitate the creation of continuous optimization algorithms for discovering causal graphs.

Let  $\mathbb{U}_{\mathbb{G}} = (\mathbb{V}, \mathbb{E})$  denote an undirected graph defined on a node set  $\mathbb{V}$ , where  $|\mathbb{V}| = d$  and  $\mathbb{E} \subseteq \binom{\mathbb{V}}{2}$ . The adjacency matrix of  $\mathbb{U}_{\mathbb{G}}$  is represented by  $\mathbf{A} = (a_{i,j})(i, j \in \mathbb{V})$ , where  $a_{i,j} = 1$  indicates the presence of an edge between nodes  $i$  and  $j$ , while  $a_{i,j} = 0$  indicates the absence of an edge. Specifically,  $a_{i,i} = 0$  signifies that there are no self-loops in the graph. A fundamental observation is that  $(\mathbf{A})^k_{ij}$  (where  $k \in \mathbb{Z}^+$ ) characterizes the number of walks of length  $k$  between nodes  $i$  and  $j$ , while  $(\mathbf{A})^k_{ii}$  characterizes the number of closed walks of length  $k$  that start and end at node  $i$ . It is important to note that a walk can pass through the same node multiple times. Correspondingly, in the case of a directed graph  $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ , where  $|\mathbb{V}| = d$  and  $\mathbb{E} \subseteq \mathbb{V} \times \mathbb{V}$ , if  $(\mathbf{A})^k_{ii} \neq 0$ , it indicates the existence of at least one closed walk in  $\mathbb{G}(\mathbf{A})$  that starts from node  $i$ , traverses  $k - 1$  nodes along directed edges, and returns to node  $i$ . Furthermore, the length of the cycle is less than or equal to  $k$ . This observation serves as the foundation for the algebraic characterization of cycles and the derivation of continuous constraint functions using algebraic methods.





**Fig. 3** Left A directed graph with 6 nodes. Right The diagonal elements of the 6th power of the adjacency matrix represent the cycle information of length 6 in the graph

As shown in Fig. 3, there are two cycles originating from node  $v_2$  and returning to itself after 6 steps. The cycles are as follows:  $v_2 \rightarrow v_1 \rightarrow v_6 \rightarrow v_5 \rightarrow v_4 \rightarrow v_3 \rightarrow v_2$  and  $v_2 \rightarrow v_4 \rightarrow v_3 \rightarrow v_2 \rightarrow v_4 \rightarrow v_3 \rightarrow v_2$ .

### 3.1 Algebraic Equivalence Condition for DAGs

This section provides a summary of six algebraic characterizations of equivalent DAG based on cycle information obtained from the adjacency matrix powers. The aim is to derive a continuous function as an acyclicity constraint based on these algebraic characterizations.

**Proposition 1**  $\mathbb{G}(A) \in \mathbb{S}_{\mathcal{G}}$  iff  $A$  satisfies one of the following six conditions:

- (1)  $\sum_{k=1}^d \text{tr}(A^k) = 0$ .
- (2)  $\lambda_i(A) = 0, i \in \{1, 2, \dots, d\}$ ,  $\lambda_i(A)$  denotes the  $i$ -th eigenvalue of matrix  $A$ .
- (3)  $A$  is a nilpotent matrix that  $A^m = \mathbf{0} (m \in \mathbb{Z}^+, m \leq d)$ .
- (4)  $\rho(A) = 0$ .
- (5)  $A$  can be transformed by a similarity transformation into a strictly upper(or lower) triangular matrix.
- (6)  $(I - A)^{-1}$  exists and  $(I - A)^{-1} \geq 0$  (i.e., all elements of the matrix are greater than 0, the same as below).

We provide a brief proof of the equivalence of each item in Proposition 1 to  $\mathbb{G}(A) \in \mathbb{S}_{\mathcal{G}}$  in the ‘‘Appendix A.1’’. Researchers in [27, 31] have introduced acyclicity constraints based on item (1). Similarly, in the study by [1], the acyclicity constraint is grounded in item (2). Moreover, the acyclicity constraints proposed in the research works [13, 33] are derived from item (4).

The key to formulating the DAG structure learning problem as a continuous optimization problem is to find a continuous acyclicity constraint function. However, the current propositions regarding acyclicity rely on the binary adjacency matrix  $A$  (with entries limited to 0 and 1) and cannot facilitate the creation of a continuous acyclicity constraint function. To overcome this limitation, this paper adopts an approach similar to previous studies [27, 31], replacing  $A$  with a weighted adjacency matrix  $W$ .  $W \circ W$  ensures the weighted adjacency matrix remains nonnegative. This operation also maps  $W$  to a positively weighted adjacency matrix while preserving the same structure. In this formulation,  $(W \circ W)_{ij} > 0$  indicates the existence of a directed edge from node  $i$  to node  $j$  in the graph  $\mathbb{G}(W)$ , while  $(W \circ W)_{ij} = 0$  indicates the absence of such an edge. Therefore, the problem of learning DAG structures using continuous optimization methods reduces to constructing a continuous function  $h(W) : \mathbb{R}^{d \times d} \rightarrow \mathbb{R}$  that can serve as an acyclicity constraint.  $h(W)$  should satisfy the following conditions:

- (1)  $h(W)$  quantifies the ‘‘distance’’ or dissimilarity between candidate graphs and the target DAG;

- (2)  $\mathbb{G}(\mathbf{W}) \in \mathbb{S}_{\mathcal{G}}$  iff  $h(\mathbf{W}) = 0$ ;
- (3) The calculation of  $h(\mathbf{W})$  and its gradient  $\nabla h(\mathbf{W})$  is straightforward and convenient.

### 3.2 Analysis of Acyclicity Constraints in Existing Work

In Sect. 3.1, we summarize six equivalent conditions for DAGs based on the adjacency matrix of directed graphs. In the subsequent discussion, we present three continuous acyclicity constraint functions that are constructed in recent research, utilizing the aforementioned propositions. Additionally, we address the associated issues with these functions.

Zheng et al. [31] first define the following continuous acyclicity constraint based on item (1) of Proposition 1 and the trace function of matrices:

$$h_{TI}(\mathbf{W}) = \text{tr}[(\mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-1}] - d \tag{4}$$

where  $\rho(\mathbf{W} \circ \mathbf{W}) < 1$ . The power series of  $(\mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-1}$  is given by  $\sum_{k=0}^{\infty} (\mathbf{W} \circ \mathbf{W})^k, k \in \mathbb{Z}^+$ . However, Zheng et al. find that the function  $h_{TI}(\mathbf{W})$  suffers from numerical explosion and other issues. Therefore, they propose a new acyclicity constraint based on the matrix exponential, which has better convergence properties. The new constraint is as follows:

$$h_{NT}(\mathbf{W}) = \text{tr}(e^{\mathbf{W} \circ \mathbf{W}}) - d \tag{5}$$

where  $e^{\mathbf{W} \circ \mathbf{W}}$  is the matrix exponential, which can be expressed as a power series  $\sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{W} \circ \mathbf{W})^k$ .

To reduce computational complexity, Yu et al. [27] proposed an improved constraint function based on item (1) of Proposition 1 and the matrix binomial to the power of  $d$ . The function is defined as follows:

$$h_{DG}(\mathbf{W}) = \text{tr}\left[\left(\mathbf{I} + \frac{1}{d} \mathbf{W} \circ \mathbf{W}\right)^d\right] - d \tag{6}$$

The power series expansion of  $[\mathbf{I} + \frac{1}{d}(\mathbf{W} \circ \mathbf{W})]^d$  is given by  $\sum_{k=0}^d \binom{d}{k} / d^k (\mathbf{W} \circ \mathbf{W})^k$ . The presence of the  $d$ th power signifies that in a directed graph with  $d$  nodes, there are no cycles of length greater than  $d$ .

However, it should be noted that in order to ensure the convergence of the power series expansion, the constraints defined by Eqs. 4, 5, and 6 can lead to numerical explosion or vanishing gradients for higher-order terms. As a result, when these constraints are utilized in iterative solving, there is a risk of the resulting target graph deviating from the true graph structure. To further illustrate this problem, we will analyze the power series expansions of  $h_{TI}(\mathbf{W}), h_{NT}(\mathbf{W})$  and  $h_{DG}(\mathbf{W})$ , along with their respective gradient terms as follows:

$$h_{TI}(\mathbf{W}) = \text{tr}\left[\sum_{k=1}^{\infty} (\mathbf{W} \circ \mathbf{W})^k\right] \tag{7}$$

$$\nabla h_{TI}(\mathbf{W}) = 2[(\mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-2}]^T \circ \mathbf{W} = 2\left[\sum_{k=0}^{\infty} (k+1) (\mathbf{W} \circ \mathbf{W})^k\right]^T \circ \mathbf{W} \tag{8}$$

$$h_{NT}(\mathbf{W}) = \text{tr}\left[\sum_{k=1}^{\infty} \frac{1}{k!} (\mathbf{W} \circ \mathbf{W})^k\right] \tag{9}$$

$$\nabla h_{NT}(\mathbf{W}) = 2 \left( e^{\mathbf{W} \circ \mathbf{W}} \right)^T \circ \mathbf{W} = 2 \left[ \sum_{k=0}^{\infty} \frac{1}{k!} (\mathbf{W} \circ \mathbf{W})^k \right]^T \circ \mathbf{W} \tag{10}$$

$$h_{DG}(\mathbf{W}) = \text{tr} \left[ \sum_{k=1}^d \frac{\binom{d}{k}}{d^k} (\mathbf{W} \circ \mathbf{W})^k \right] \tag{11}$$

$$\nabla h_{DG}(\mathbf{W}) = 2 \left[ \left( \mathbf{I} + \frac{1}{d} \mathbf{W} \circ \mathbf{W} \right)^{d-1} \right]^T \circ \mathbf{W} = 2 \left[ \sum_{k=0}^{d-1} \frac{\binom{d-1}{k}}{(d-1)^k} (\mathbf{W} \circ \mathbf{W})^k \right]^T \circ \mathbf{W} \tag{12}$$

Applying  $h_{TI}(\mathbf{W})$  as the acyclicity constraint in DAG structure learning, the coefficients of the terms in the expansions of Eqs. 7 and 8 are 1 and  $(k + 1)$ , respectively. However, when  $\rho(\mathbf{W} \circ \mathbf{W})$  is large, there may be numerical explosion issues when computing  $(\mathbf{W} \circ \mathbf{W})^k$  and  $(k + 1)(\mathbf{W} \circ \mathbf{W})^k$  as  $k$  increases. This can adversely affect the convergence stability of the algorithm. In contrast,  $h_{NT}(\mathbf{W})$  and  $h_{DG}(\mathbf{W})$ , along with their gradient expansions, have coefficients of  $1/k!$  and  $\binom{d}{k}/d^k$ , respectively. As the optimization progresses,  $\rho(\mathbf{W} \circ \mathbf{W})$  approaches zero, causing the coefficients of the terms in the expansion to rapidly decrease with increasing  $k$ . Consequently, the  $k$ -th order term in the expansion becomes infinitesimal, leading to the loss of cycle information and gradient information in higher-order terms. This reduction in accuracy negatively impacts the algorithm.

In fact,  $h_{DG}(\mathbf{W})$  is not suitable for discovering DAG structures with a large number of nodes. For example, if we apply  $h_{DG}(\mathbf{W})$  to a DAG with 100 nodes, even for a relatively small value of  $k$  (such as  $k = 10$ ), the coefficient of the  $k$ -th term in the power series expansion becomes  $\binom{100}{10}/100^{10}$ . This can result in the failure to detect cycles in the graph with a length exceeding 10.

### 3.3 New Acyclicity Constraint

#### 3.3.1 Background and Motivation

The acyclicity constraints  $h_{TI}(\mathbf{W})$ ,  $h_{NT}(\mathbf{W})$ , and  $h_{DG}(\mathbf{W})$  can all be expressed as matrix polynomials. They share a general form  $\sum_{k=1}^d l(k) \text{tr}[(\mathbf{W} \circ \mathbf{W})^k]$ , where  $l(k) > 0$ . As mentioned earlier,  $\sum_{k=1}^d l(k) \text{tr}[(\mathbf{W} \circ \mathbf{W})^k] = 0$  signifies the absence of cycles of length less than or equal to  $d$  in  $\mathbb{G}(\mathbf{W})$ . While these three functions achieve the acyclicity constraint in a similar manner, the distinction lies in the coefficient term  $l(k)$ . As previously discussed, selecting a relatively “large” coefficient may lead to numerical explosion, while opting for a relatively “small” coefficient may result in information loss and gradient vanishing. Therefore, in this paper, we utilize item (6) of Proposition 1 to establish a new equivalent condition for DAG and propose an acyclicity constraint function with adaptively adjustable coefficient terms  $l(k)$ .

**Proposition 2**  $\mathbb{G}(\mathbf{W} \circ \mathbf{W}) \in \mathbb{S}_{\mathcal{G}}$  iff  $(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-1}$  exists and  $(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-1} \geq 0$ .

Proposition 2 relaxes the constraint condition from  $\rho(\mathbf{W} \circ \mathbf{W}) < 1$  to  $\rho(\mathbf{W} \circ \mathbf{W}) < \alpha$ . Based on Proposition 2, the premise for defining the continuous acyclicity constraint function is to find a set of weighted adjacency matrices  $\mathbf{W}$ , where Proposition 2 holds true for any  $\mathbf{W}$  in this set. In fact, if  $\mathbf{W}$  is a matrix defined in this set, then  $\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W}$  is characterized as an M-matrix[15] (first introduced by Swiss mathematician Ostrowski in 1937). The definition of an M-matrix is as follows:

**Definition 1** (*M-matrix*) Let  $\mathbf{B} \in \mathbb{R}^{d \times d} \geq 0$ ,  $\mathbf{C} \in \mathbb{R}^{d \times d}$ , if  $\mathbf{C} = \alpha \mathbf{I} - \mathbf{B}$ , where  $\alpha > \rho(\mathbf{B})$ , then  $\mathbf{C}$  is an M-matrix.

If  $\mathbf{C}$  is an M-matrix, it possesses the following properties:

- (1)  $\lambda_i(\mathbf{C}) > 0, i \in \{1, \dots, d\}$ ;
- (2)  $\mathbf{C}^{-1} \geq 0$ .

The mentioned properties are evident: From the equation  $\lambda_i(\mathbf{C}) = \alpha - \lambda_i(\mathbf{B})$  and the fact that  $\alpha > \rho(\mathbf{B})$ , it follows that  $\lambda_i(\mathbf{C}) > 0$ . Moreover, from the power series expansion of  $\mathbf{C}^{-1}$ , it is clear that  $\mathbf{C}^{-1} \geq 0$ . Additionally, it is worth noting that  $\mathbf{B}$  can be equivalently replaced by  $\mathbf{W} \circ \mathbf{W}$ . Thus, when  $\alpha > \rho(\mathbf{W} \circ \mathbf{W})$ ,  $\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W}$  is an M-matrix, ensuring that Proposition 2 holds true. This clever connection between Proposition 2 and M-matrices allows us to determine the desired set of matrices  $\mathbf{W}$  based on the characteristics of M-matrices.

**Definition 2** ( $\mathbb{W}^\alpha$ )  $\mathbb{W}^\alpha = \{ \mathbf{W} \in \mathbb{R}^{d \times d} \mid \alpha > \rho(\mathbf{W} \circ \mathbf{W}) \}$ .

The set  $\mathbb{W}^\alpha$  refers to the aforementioned set that we seek to solve for, and it exhibits the following crucial properties:

**Proposition 3** Let  $\mathbb{W}^{\mathbb{S}_G}$  be the space of weighted adjacency matrices corresponding to  $\mathbb{S}_G$ , when  $\alpha > 0$ :

- (1)  $\mathbb{W}^{\mathbb{S}_G} \subset \mathbb{W}^\alpha$ ;
- (2)  $\mathbb{W}^\alpha \subset \mathbb{W}^t$ , where  $\alpha < t$ .

For any  $\mathbf{W} \in \mathbb{W}^{\mathbb{S}_G}$ , it holds that  $\rho(\mathbf{W} \circ \mathbf{W}) < \alpha$ . item (1) indicates that feasible solutions to the DAG structure learning problem must lie within  $\mathbb{W}^\alpha$ . item (2) suggests that the size of  $\mathbb{W}^\alpha$  can be adjusted by changing the value of  $\alpha$ . As the optimization progresses, gradually decreasing  $\alpha$  narrows down the search space for feasible solutions, which in turn leads to faster convergence of the algorithm.

Based on the analysis above, the proposed continuous constraint function  $h(\mathbf{W}) : \mathbb{W}^\alpha \rightarrow \mathbb{R}$  is constructed as follows:

$$h(\mathbf{W}) = \text{tr} [(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-1}] - d/\alpha \tag{13}$$

### 3.3.2 Theoretical Analysis

In this section, we will analyze the new acyclicity constraint proposed in the previous section (Eq. 13) from three aspects: acyclicity, continuity, and dynamic parameter adjustment. We will provide a brief theoretical analysis and offer a concise proof to support our findings.

**Corollary 1** For any  $\mathbf{W} \in \mathbb{W}^\alpha$ , it holds that  $h(\mathbf{W}) \geq 0$ . Moreover,  $h(\mathbf{W}) = 0$  iff  $\mathbb{G}(\mathbf{W}) \in \mathbb{S}_G$ .

See ‘‘Appendix A.3’’ for detailed proof. Corollary 1 states that for any  $\mathbf{W} \in \mathbb{W}^\alpha$ , we have  $h(\mathbf{W}) \geq 0$ , which implies that  $\mathbb{W}^{\mathbb{S}_G}$  corresponds to the set of local minima of  $h(\mathbf{W})$ . This also helps us prove  $\mathbb{W}^{\mathbb{S}_G} \subset \mathbb{W}^\alpha$  from another perspective. During the optimization process, as the value of  $h(\mathbf{W})$  approaches 0, it indicates that  $\rho(\mathbf{W})$  tends to 0, which means that  $\mathbb{G}(\mathbf{W})$  is getting closer to the optimal solution, satisfying the first condition mentioned in Sect. 3.1.  $h(\mathbf{W}) = 0$  if and only if  $\mathbb{G}(\mathbf{W}) \in \mathbb{S}_G$ , corresponding to the second condition in Sect. 3.1.

**Corollary 2** The gradient of  $h(\mathbf{W})$ , denoted as  $\nabla h(\mathbf{W})$ , is defined as  $\nabla h(\mathbf{W}) = 2[(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-2}]^T \circ \mathbf{W}$ . For any  $i, j \in \mathbb{V}$ , the element  $\nabla h(\mathbf{W})_{ij}$  shares the same sign as  $\mathbf{W}_{ij}$ . If  $\nabla h(\mathbf{W})_{ij} \neq 0$ , it indicates the presence of a cycle in  $\mathbb{G}(\mathbf{W})$ .

See ‘‘Appendix A.4’’ for detailed proof. Corollary 2 states that  $\nabla h(\mathbf{W})$  satisfies the third condition in Sect. 3.1. Since  $2[(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-2}]^T \geq 0$ , the sign of each element in  $\nabla h(\mathbf{W})$  depends on the sign of the corresponding element in  $\mathbf{W}$ . If  $\nabla h(\mathbf{W})_{ij} \neq 0$ , it indicates that  $[(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-2}]_{ij}^T \neq 0$  and  $\mathbf{W}_{ij} \neq 0$ . Here,  $[(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-2}]_{ij}^T = [\sum_{k=0}^{\infty} \frac{k+1}{\alpha^{k+2}} (\mathbf{W} \circ \mathbf{W})^k]_{ij}^T$  represents the sum of weights from node  $j$  to node  $i$ , indicating the existence of a path (possibly through one or more nodes) from node  $j$  to node  $i$  in the graph. Furthermore,  $\mathbf{W}_{ij} \neq 0$  represents the existence of a directed path from node  $i$  to node  $j$ . Therefore, the presence of a cycle in the graph can be inferred as  $i \rightarrow j \rightarrow \dots \rightarrow i$ .

**Corollary 3** *For any  $\mathbf{W} \in \mathbb{W}^\alpha$ , let  $\alpha = \rho(\mathbf{W} \circ \mathbf{W}) + \xi$ , where  $\xi \in \mathbb{R}^+$ . Then, we have  $h(\mathbf{W}) = \text{tr}[(\alpha \mathbf{I} - \mathbf{W} \circ \mathbf{W})^{-1}] - d/\alpha = \sum_{k=1}^{\infty} 1/\alpha^{k+1} (\mathbf{W} \circ \mathbf{W})^k$ . The dynamic adjustment of  $\mathbb{W}^\alpha$  and  $1/\alpha^{k+1}$  in the optimization process ensures that the computation of  $h(\mathbf{W})$  and  $\nabla h(\mathbf{W})$  neither leads to numerical explosion nor loses the cycle information and gradient information in the higher-order terms.*

$h(\mathbf{W})$  shares the same form as  $h_{TI}(\mathbf{W})$ ,  $h_{NT}(\mathbf{W})$ , and  $h_{DG}(\mathbf{W})$ , which is  $\sum_{k=1}^d l(k) \text{tr}[(\mathbf{W} \circ \mathbf{W})^k]$ . The distinction lies in the coefficient  $l(k)$  of the power series in  $h(\mathbf{W})$ , which depends on  $\alpha$ , enabling it to adjust dynamically during the optimization process. Specifically, in the initial stages of optimization when  $\rho(\mathbf{W} \circ \mathbf{W})$  is large, the computation of  $h_{TI}(\mathbf{W})$  and its gradient may encounter numerical explosion. By setting  $\alpha = \rho(\mathbf{W} \circ \mathbf{W}) + \xi$ , the coefficients  $1/\alpha^{k+1}$  in the power series of  $h(\mathbf{W})$  diminish rapidly, effectively avoiding numerical explosion. As the optimization progresses and  $\rho(\mathbf{W} \circ \mathbf{W})$  approaches zero, there is a risk of losing cycle information and gradients in the higher-order terms of  $h_{NT}(\mathbf{W})$  and  $h_{DG}(\mathbf{W})$ . To address this, setting  $\alpha = \rho(\mathbf{W} \circ \mathbf{W}) + \xi$  (where  $\xi = 1$ ) ensures that  $s$  and  $1/\alpha^{k+1}$  are approximately 1. This preserves the true values of each term in the power series and resolves the issue of losing information in the higher-order terms. Additionally,  $\mathbb{W}^\alpha$  adjusts correspondingly as  $\rho(\mathbf{W} \circ \mathbf{W})$  decreases throughout the optimization process, narrowing down the search space for the DAGs. Consequently, the dynamic adjustment of  $\mathbb{W}^\alpha$  and  $1/\alpha^{k+1}$  enhances the efficiency and accuracy of DAG structure learning algorithm.

### 3.3.3 Computational Complexity Analysis

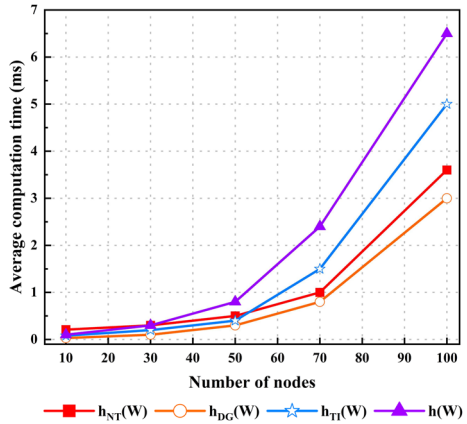
From the previous discussion, it is evident that the proposed acyclicity constraint function  $h(\mathbf{W})$  shares a similar power series form  $\sum_{k=1}^d l(k) \text{tr}[(\mathbf{W} \circ \mathbf{W})^k]$  with  $h_{TI}(\mathbf{W})$ ,  $h_{NT}(\mathbf{W})$ , and  $h_{DG}(\mathbf{W})$ . Therefore, the computational complexity of calculating these acyclicity constraints and their gradients is  $O(d^3)$  for each of them. As shown in Fig. 4,  $h(\mathbf{W})$  and its gradient have slightly higher average computation time, this is primarily due to the additional computation of the spectral radius of the matrix before calculating  $h(\mathbf{W})$ , which incurs extra time. However, in Sect. 5, we will demonstrate through experiments that the algorithm based on  $h(\mathbf{W})$  has the shortest average runtime. This indicates that  $h(\mathbf{W})$  enables the algorithm to converge faster to an approximately optimal solution.

Table 2 provides a concise summary of the acyclicity constraint functions mentioned above, aiming to enhance comparison and comprehension.

## 4 Structure Learning Algorithm

Based on the newly introduced acyclicity constraint function  $h(\mathbf{W})$ , this section presents a causal structure learning algorithm named LeCaSiM.

**Fig. 4** The average computation time required for calculating the acyclicity constraints and their gradients at different node sizes



**Table 2** Comparison of acyclicity constraint functions

Constraint	Function	Gradient	Complexity
$h_{T1}(W)$	$tr[\sum_{k=1}^{\infty} (W \circ W)^k]$	$2[\sum_{k=0}^{\infty} (k+1)(W \circ W)^k]^T \circ W$	$O(d^3)$
$h_{NT}(W)$	$tr[\sum_{k=1}^{\infty} \frac{1}{k!} (W \circ W)^k]$	$2[\sum_{k=0}^{\infty} \frac{1}{k!} (W \circ W)^k]^T \circ W$	$O(d^3)$
$h_{DG}(W)$	$tr[\sum_{k=1}^d \frac{\binom{d}{k}}{d^k} (W \circ W)^k]$	$2[\sum_{k=0}^{d-1} \frac{\binom{d-1}{k}}{(d-1)^k} (W \circ W)^k]^T \circ W$	$O(d^3)$
$h(W)$	$tr[\sum_{k=1}^{\infty} \frac{1}{\alpha^{k+1}} (W \circ W)^k]$	$2[\sum_{k=0}^{\infty} \frac{k+1}{\alpha^{k+2}} (W \circ W)^k]^T \circ W$	$O(d^3)$

### 4.1 LeCaSiM Algorithm

Like previous studies, this paper utilizes the mean squared error (MSE) as the loss function  $q(W, X)$  in the LSEM:

$$q(W, X) = \frac{1}{n} \|X - XW\|_F^2 = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^d (X - XW)_{ij}^2 \tag{14}$$

Considering that practical graphs are often sparse, an  $L_1$ -regularization is incorporated into Eq. 14 to promote sparsity. Thus, the scoring function  $Q(W, X)$  is defined as follows:

$$Q(W, X) = \frac{1}{n} \|X - XW\|_F^2 + \mu \|W\|_1$$

where  $\mu > 0$  is a constant coefficient. Within this framework, the problem of learning DAG structures based on  $h(W)$  can be formulated as the following optimization problem:

$$\min_{W \in \mathbb{W}^\alpha} Q(W, X) \quad s.t. \quad h(W) = 0 \tag{15}$$

To address the optimization problem mentioned above, we utilize the augmented Lagrangian method (ALM)[2] to transform it into the following unconstrained optimization problem:

$$\min_{W \in \mathbb{W}^\alpha} \max_{\varphi \in \mathbb{R}} \ell(W, \varphi) \tag{16}$$

where  $\ell(\mathbf{W}, \varphi) = Q(\mathbf{W}, \mathbf{X}) + \frac{\sigma}{2}h(\mathbf{W})^2 + \varphi h(\mathbf{W})$ ,  $\sigma > 0$  represents penalty parameter, and  $\varphi$  denotes the Lagrange multiplier. The dual ascent method is employed to iteratively solve this unconstrained optimization problem. Assuming we have the solution at the  $k$ -th iteration  $\mathbf{W}^k$  and  $\varphi^k$ , the update process is as follows:

- Substitute  $\varphi = \varphi^k$  into  $\ell(\mathbf{W}, \varphi)$  and solve  $\min_{\mathbf{W} \in \mathbb{W}^\alpha} \ell(\mathbf{W}, \varphi^k)$ . Then, obtain  $\mathbf{W}^{k+1}$ ;
- replace  $\mathbf{W} = \mathbf{W}^{k+1}$  into  $\ell(\mathbf{W}, \varphi)$  and solve  $\max_{\varphi \in \mathbb{R}} \ell(\mathbf{W}^{k+1}, \varphi)$ . Then, acquire  $\varphi^{k+1}$ .  

$$\varphi^{k+1} = \varphi^k + \sigma \cdot \left. \frac{\partial \ell(\mathbf{W}^{k+1}, \varphi)}{\partial \varphi} \right|_{\mathbf{W}=\mathbf{W}^{k+1}, \varphi=\varphi^k}$$

Repeated iterations of the above process are performed until convergence, resulting in an approximate optimal solution. Algorithm 1 outlines our proposed LeCaSiM algorithm, which is based on the analysis mentioned above.

---

**Algorithm 1** LeCaSiM

---

**Require:** data matrix  $\mathbf{X}$ , initial  $\mathbf{W}^0$ , lagrange multiplier  $\varphi^0$ , penalty parameter  $\sigma$ ,  $L_1$  coefficient  $\mu$ , algorithm precision  $\tau$ , weight threshold  $\theta$ , parameter  $\alpha$ , maximum number of iterations  $T$ .

```

1:  $k \leftarrow 0$ 
2: while  $k < T - 1$  do
3:   compute  $\rho(\mathbf{W}^k \circ \mathbf{W}^k)$ , let  $\alpha \leftarrow \rho(\mathbf{W}^k \circ \mathbf{W}^k) + 1$ 
4:   solve  $\mathbf{W}^{(k+1)} \leftarrow \arg \min_{\mathbf{W} \in \mathbb{W}^\alpha} \frac{1}{n} \|\mathbf{X} - \mathbf{X}\mathbf{W}\|_F^2 + \mu \|\mathbf{W}\|_1 + \frac{\sigma}{2} h(\mathbf{W})^2 + \varphi^k h(\mathbf{W})$ 
5:   update using dual ascent:  $\varphi^{(k+1)} \leftarrow \varphi^k + \sigma h(\mathbf{W}^{k+1})$ 
6:   if  $h(\mathbf{W}^{k+1}) < \tau$  then
7:      $\hat{\mathbf{W}} \leftarrow \mathbf{W}^{k+1}$ 
8:   end if
9: end while
10: set the values of elements in  $\hat{\mathbf{W}}$  with absolute values smaller than  $\theta$  to 0
Ensure: near-optimal solution  $\mathbf{W}^*$ 

```

---

To ensure that  $h(\mathbf{W}) = 0$ , a strategy of gradually increasing  $\sigma$  and  $\varphi$  as penalty factors is employed during the optimization process. However, it is important to note that in the continuous optimization process, the value of  $h(\mathbf{W})$  may approach zero indefinitely without reaching exactly zero. Therefore, a precision level for the algorithm is set, and the optimization process terminates when the value of  $h(\mathbf{W})$  becomes smaller than this precision.

In practice, due to factors such as machine precision, the iterative process can only guarantee convergence to a solution near the optimal solution. However, experimental results have consistently shown that this approximate solution is already very close to the optimal solution level.

**4.2 Algorithm Implementation Details**

In the third step of the LeCaSiM, the parameter  $\alpha$  in  $h(\mathbf{W})$  is determined. To enforce acyclicity,  $\alpha$  must be greater than  $\rho(\mathbf{W} \circ \mathbf{W})$ . Thus, we set  $\alpha = \rho(\mathbf{W} \circ \mathbf{W}) + \xi$ , with  $\xi$  taking values from the set  $\{0.1, 0.5, 1, 1.5, 2\}$ . Experimental results indicate that setting  $\xi$  to 1 achieves the best performance. The reason for choosing  $\xi = 1$  is as follows: during the optimization process, as  $\rho(\mathbf{W} \circ \mathbf{W})$  approaches 0, choosing  $\xi < 1$  leads to  $\alpha < 1$ , resulting in a smaller  $\mathbb{W}^\alpha$ . In this case, a smaller learning rate is required to ensure algorithm convergence, which may affect convergence stability. On the other hand, if  $\xi > 1$ , it may cause the coefficients of higher-order terms to approach 0, leading to the loss of information from these terms.

Similar to existing work, we employ the approximate Quasi-Newton method [32] to solve the optimization subproblem in the fourth step. In the tenth step, we apply thresholding to the candidate matrix  $\mathbf{W}$  to filter out elements with absolute values smaller than the weight threshold  $\theta$ . This helps eliminate false causal associations that could induce cycles [31]. The thresholding approach ensures sparsity of the weight matrix and improves the accuracy of the algorithm by removing spurious causal edges from the graph.

## 5 Experiments

This section performs extensive comparative experiments to evaluate the efficiency and accuracy of the LeCaSiM algorithm in learning DAG structures.

### 5.1 Experimental Setup

#### 5.1.1 Comparison Algorithms and Evaluation Metrics

In Sect. 3.2, we analyze the limitations of three acyclicity constraints proposed in existing work. Among them,  $h_{TI}(\mathbf{W})$  is found to be prone to numerical explosion during computation. To demonstrate that the proposed  $h(\mathbf{W})$  can overcome this issue, we replace the  $h(\mathbf{W})$  in the LeCaSiM with  $h_{TI}(\mathbf{W})$  to create the LeCaSiM-TI for comparison. To address the challenges of computing high-order terms in the acyclicity constraint and the problem of gradient vanishing, we also include the NOTEARS [31] algorithm based on  $h_{NT}(\mathbf{W})$  and the DAG-GNN [27] algorithm based on  $h_{DG}(\mathbf{W})$  for comparison. Additionally, to compare the performance of the matrix polynomial constraint proposed in this paper, we included the NOCURL algorithm [28] that employs the weighted gradient set.

We utilize the following four metrics [8] to evaluate the accuracy and efficiency of the algorithm:

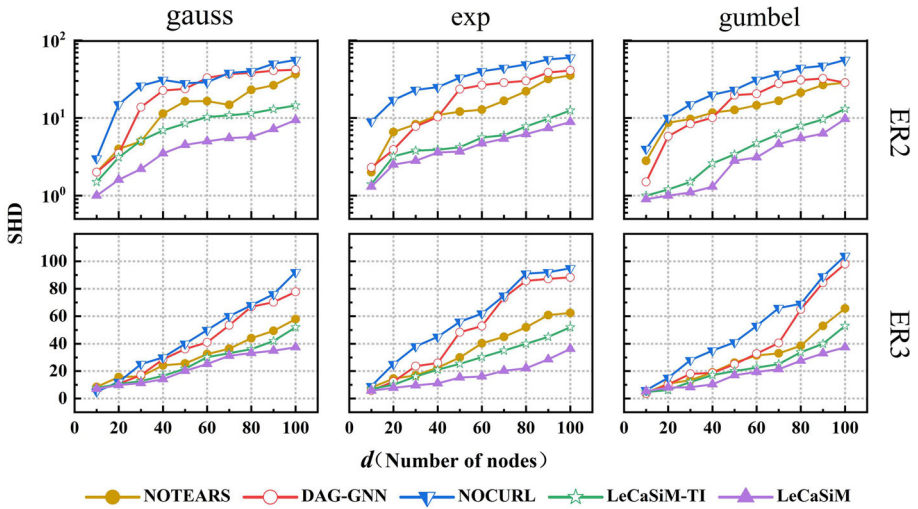
- Structure hamming distance (SHD): SHD calculates the standard distance between the estimated graph and the target graph using their adjacency matrices. A lower value indicates a higher similarity between the estimated and true graphs.
- True positive rate (TPR): TPR measures the proportion of correctly identified edges in the predicted graph compared to the total number of edges in the target graph. A higher value indicates greater accuracy of the algorithm in capturing the true edges.
- False positive rate (FPR): FPR represents the ratio of incorrectly identified edges as true edges in the predicted graph to the total number of edges missing in the target graph. A lower value indicates higher accuracy, as fewer false edges are present.
- Runtime: This metric indicates the time required for the algorithm to execute on a computer, reflecting the convergence speed of the algorithm.

#### 5.1.2 Parameter Settings and Environment Configuration

For the LeCaSiM algorithm, the following parameters are initialized: lagrange multiplier  $\varphi = 0$ , quadratic penalty coefficient  $\sigma = 1$ ,  $L_1$  coefficient  $\mu = 0.1$ , algorithm precision  $\tau = 10^{-6}$ , weight threshold  $\theta = 0.3$ , maximum number of iterations  $T = 100$ .

The experimental environment in this paper was configured with the following hardware: an 8-core 16-thread AMD Ryzen 7 5800H CPU running at a frequency of 3.20 GHz, paired with 16GB of memory.





**Fig. 5** The SHD (lower is better) corresponding to different algorithms regarding different random graph models (ER2, ER3) and noise distributions (gauss, exp and gumbel)

## 5.2 Experimental Results and Analysis

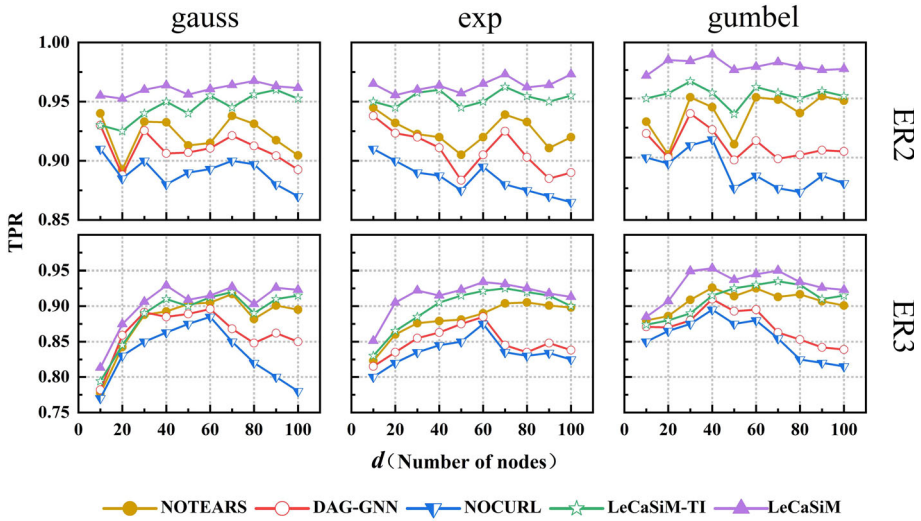
### 5.2.1 Synthetic Datasets

The experimental setup in this paper follows the approach used in previous works [27, 31]. The Erdős-Rényi (ER) random graph model is employed to construct random graphs  $G$  with  $d$  nodes and an anticipated number of  $kd$  edges, known as by  $ER_k$ . The weights of the edges in the graph are uniformly distributed within the ranges of  $[-2.0, -0.5]$  and  $[0.5, 2.0]$ . To obtain the data matrix  $X$ , we generate 1000 samples according to the equation  $X = XW + N$ . The variable matrix  $N \in \mathbb{R}^{n \times d}$  represents the exogenous variables and is generated using one of the following distributions: Gaussian distribution (gauss), exponential distribution (exp), or Gumbel distribution (gumbel).

To ensure consistent and reproducible results, we use the same random seed for each experiment. By doing so, we generate the same  $X$  every time, enabling us to evaluate the performance of different algorithms based on consistent experimental results. For each configuration, we conduct the experiment 100 times and calculate the mean of the evaluation metrics to obtain the final result.

As shown in Fig. 5, the LeCaSiM outperforms the other comparison algorithms in terms of SHD, indicating its superior accuracy in recovering the DAG structure from observed data. For example, when using ER2 random graphs and three different noise distributions, with 100 nodes (200 edges), LeCaSiM achieves an SHD of less than 10. This corresponds to a false discovery rate of approximately 5% (10/200). In comparison, the LeCaSiM-TI exhibits a higher false discovery rate of approximately 7.5%. The NOTEARS displays a significantly higher false discovery rate of 20%, while the other two algorithms have even higher rates.

The TPR reflects the ability of each algorithm to accurately identify true edges and learn DAG structure. The results shown in Fig. 6 demonstrate that the LeCaSiM consistently achieves the highest TPR among the four comparison algorithms in all six experimental settings. For example, in the case of ER2 random graphs, the TPR of the LeCaSiM remains



**Fig. 6** The TPR (higher is better) corresponding to different algorithms regarding different random graph models (ER2, ER3) and noise distributions (gauss, exp and gumbel)

consistently above 0.95 for different numbers of nodes, demonstrating that the LeCaSiM has the highest accuracy in correctly identifying true edges and is capable of learning the true DAG structure more accurately.

The FPR reflects the rate at which algorithms incorrectly identify false edges in the predicted graph. From Fig. 7, the LeCaSiM has the lowest FPR among the compared algorithms. This signifies that the LeCaSiM does not blindly learn a large number of edges in an attempt to achieve a higher TPR. Instead, it accurately learns the true edges without introducing false edges into the predicted graph. In other words, the LeCaSiM is able to identify more true edges of the target graph without increasing the number of false edges in the predicted graph. This capability allows the LeCaSiM to more accurately recover the true graph structure.

As shown in Fig. 4, the calculation time of  $h(W)$  is slightly longer compared to other constraints. However, from Fig. 8, the average runtime of the LeCaSiM is the shortest among the compared algorithms. This indicates that  $h(W)$  possesses certain desirable properties, such as resolving numerical explosion and vanishing gradients. These properties contribute to faster convergence of the algorithm and improve its efficiency.

For the above experiments, the performance of the algorithm is significantly better in the ER2 random graph compared to the ER3 random graph. The reason is that ER3 target graph has more edges (non-sparse graph), which adds difficulty to the optimization of the objective function with the  $L_1$ -regularization, leading to a decrease in the accuracy and efficiency of the algorithm.

Figure 9 shows that the LeCaSiM curve is positioned in the top left part of the other curves, indicating a higher TPR and a lower FPR, i.e. a higher accuracy and a lower error detection rate compared to the other algorithms.

In Table 3, it is evident that the LeCaSiM outperforms the other algorithms in terms of average SHD and average running time. This demonstrates that the LeCaSiM offers the highest accuracy and efficiency among the considered options.

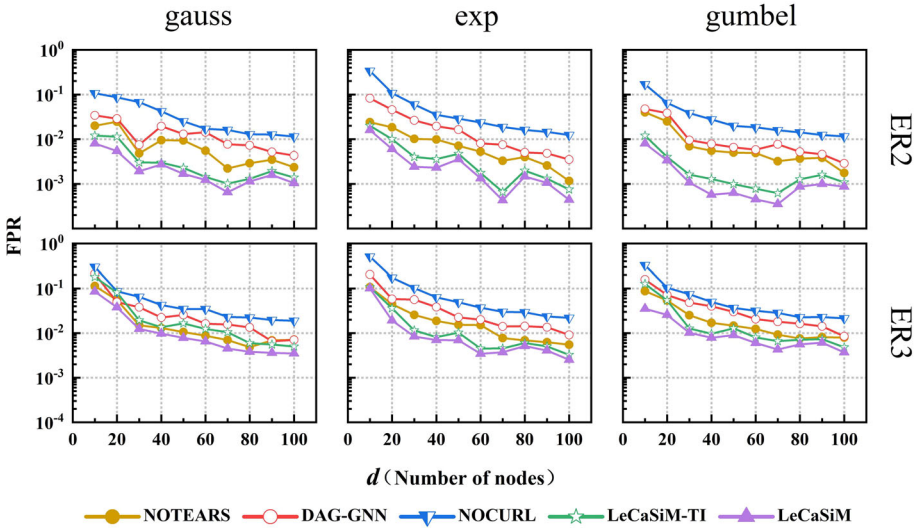


Fig. 7 The FPR (lower is better) corresponding to different algorithms regarding different random graph models (ER2, ER3) and noise distributions (gauss, exp and gumbel)

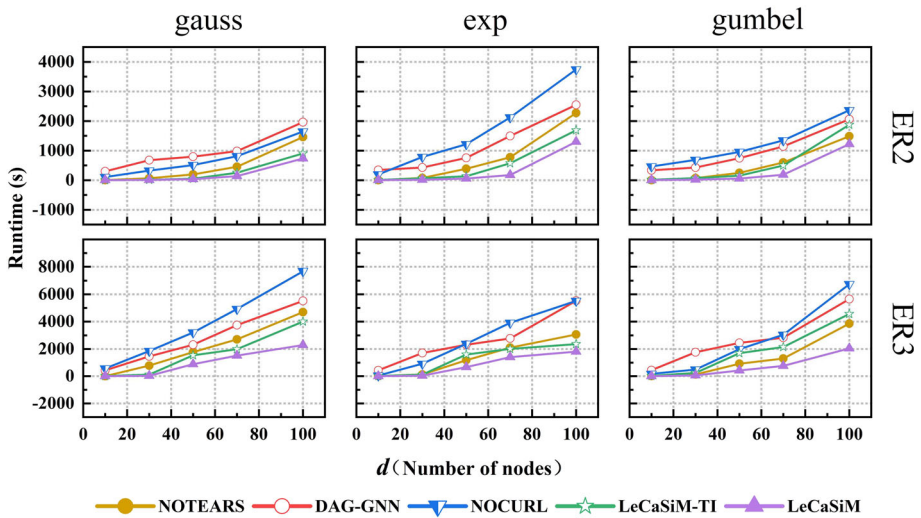
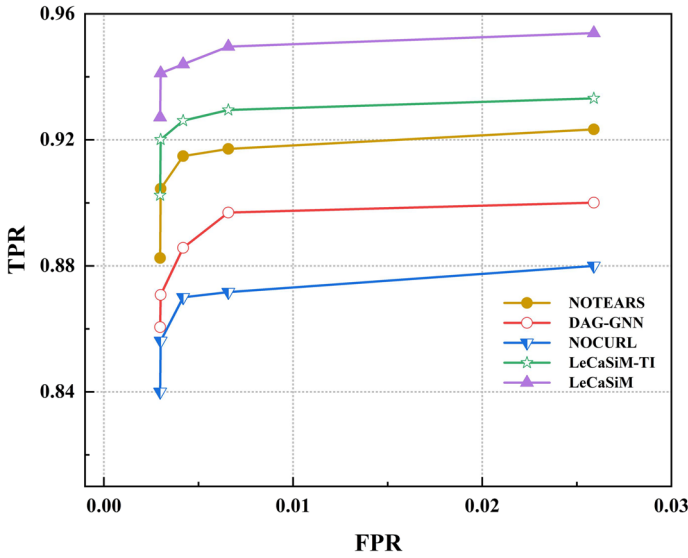


Fig. 8 The runtime (lower is better) corresponding to different algorithms regarding different random graph models (ER2, ER3) and noise distributions (gauss, exp and gumbel)

### 5.2.2 Real Dataset

Similar to previous works [27, 31], we have conducted experiments using the protein signaling networks (PSN) dataset [22], which contains 7466 samples of 11 nodes. There are 20 edges in the ground-truth graph noted by Sachs. The PSN dataset is a well-known dataset in the field of Bayesian network structure learning and has been widely used to evaluate the performance of DAG structure learning algorithms.



**Fig. 9** The average FPR and TPR values plotted for the LeCaSiM and various comparison algorithms with nodes set to 10, 30, 50, 70, and 100, respectively

The proposed LeCaSiM in this paper is based on a linear model that learns the causal structure between variables, while algorithms such as DAG-GNN are based on nonlinear models to learn the causal relationship between variables. In fact, we are unsure whether the causal mechanisms between variables in the PSN dataset are linear or nonlinear. Therefore, we conduct extensive experiments for each algorithm on the PSN dataset, measuring the accuracy of each algorithm’s learning of the causal structure through a comparison of SHD and predicted edge numbers, as shown in Table 4.

The number of edges in the learned graph structure by the algorithm is referred to as the “# Predicted edges”. From Table 4, it can be observed that the predicted edge count for LeCaSiM is 17, consisting of 10 ground-truth edges, 7 erroneous edges, and 1 reverse edges. LeCaSiM achieves the lowest SHD compared to the comparison algorithms, indicating the LeCaSiM can more accurately reconstruct the true causal structure when applied to real-world datasets like the PSN dataset.

## 6 Summary and Future Work

The problem of DAG structure learning has broad applications across various fields, including task scheduling, data flow analysis, dependency management, and more. However, the exponential growth of the DAG search space with the number of nodes poses a major challenge. Recent research has introduced a continuous acyclicity constraint function, which transforms the DAG structure learning problem into a continuous optimization problem. Motivated by this approach, this paper employs concepts from algebraic graph theory to summarize seven equivalent conditions for DAGs and compare existing acyclicity constraints. It is observed that existing methods may encounter issues such as numerical explosion or gradient vanishing during acyclicity constraint computations and optimization iterations. To address these challenges, this paper introduces M-matrices and leverages their properties to propose a novel

**Table 3** The average SHD values and runtimes of LeCaSiM and various comparison algorithms with nodes set to 10, 30, 50, 70 and 100, respectively (The optimal values are shown in bold)

Method	Node	SHD	Runtime(s)
NOTEARS	10	4.48	4.54
	30	11.55	87.55
	50	19.07	465.2
	70	26.88	895.52
	100	47.77	4459.58
DAG-GNN	10	3.65	375.76
	30	14.77	1140.59
	50	27.75	1542.36
	70	43.4	2153.25
	100	63.78	3798.52
NOCURL	10	6	96.33
	30	25.83	613.5
	50	37	1280.67
	70	51.17	2191.67
	100	79.17	5776.17
LeCaSiM-TI	10	3.87	9.63
	30	8.52	101.18
	50	13.85	857.98
	70	19.3	1241.17
	100	32.83	1971.92
LeCaSiM	10	<b>3.5</b>	<b>1.75</b>
	30	<b>6.4</b>	<b>23.72</b>
	50	<b>10.55</b>	<b>590.68</b>
	70	<b>14.95</b>	<b>782.78</b>
	100	<b>23.18</b>	<b>1689.78</b>

**Table 4** Algorithm prediction results based on the real dataset

Method	SHD	# Predicted edges
NOTEARS	22	18
DAG-GNN	19	20
NOCURL	22	19
LeCaSiM-TI	21	18
LeCaSiM	<b>17</b>	18

acyclicity constraint function. By adaptively adjusting the coefficients of high-order terms using the spectral radius of candidate matrices, these issues are effectively addressed. Based on this foundation, the LeCaSiM is developed to solve the DAG structure learning problem. Comprehensive comparative experiments are conducted on both simulated and real-world datasets in this paper. The experimental results demonstrate that the proposed algorithm outperforms existing algorithms in terms of accuracy and efficiency.

While our work contributes valuable insights and advancements, there are still limitations and opportunities for further improvements. Firstly, the assumption of linear causal relationships between variables may not always hold true in practical applications. Future

research should explore the incorporation of deep generative models that can capture nonlinear relationships while adhering to acyclicity constraints. Secondly, our algorithm provides an approximate solution to the unconstrained optimization problem. Further efforts should be directed towards improving the efficiency and convergence stability of the algorithm to enhance its performance in real-world scenarios. Lastly, we intend to explore the seven algebraic propositions of DAGs proposed in this paper further. There is scope for developing acyclicity constraint functions with lower computational complexity while maintaining the desired accuracy and efficiency. These potential research directions and areas for improvement can help advance the field of DAG structure learning and pave the way for more accurate and efficient algorithms in practical applications.

**Author Contributions** QC provided guidance on innovation and methods, as well as made modifications to the papers. YZ was responsible for completing the theoretical derivation, developing formal formulae, designing and conducting experiments, and writing the paper.

**Funding** No funding was received to assist with the preparation of this manuscript.

**Availability of Data and Materials** The data that support the findings of this study are available from the corresponding author upon reasonable request.

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Ethics Approval** The authors assert that no ethical issues are expected to arise following the publication of this manuscript.

**Consent to Participate** Informed consent was obtained from all individual participants included in the study.

**Consent for Publication** All authors agree to publish the paper.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## Appendix A Detailed Proofs

### A.1 Proof of Proposition 1

**Proof Item (1).** According to the previous observation, if the summation of the trace of the powers of the adjacency matrix,  $\sum_{k=1}^d \text{tr}(\mathbf{A}^k) = (\mathbf{A}_{11}^k + \dots + \mathbf{A}_{dd}^k) = 0$ , it suggests that there are no cycles of lengths 1 to  $d$  in the  $\mathbb{G}(\mathbf{A})$ . As a result, it can be inferred that  $\mathbb{G}(\mathbf{A})$  is a DAG.

**Item (2).** According to item (1) of Proposition 1,  $\mathbb{G}(\mathbf{A}) \in \mathbb{S}_{\mathbb{G}}$  is equivalent to  $\sum_{i=1}^d \text{tr}(\mathbf{A}^i) = 0$ . By using the property that the trace of a matrix is equal to the sum of its eigenvalues, we have  $\sum_{i=1}^d \lambda_i(\mathbf{A}) + \sum_{i=1}^d \lambda_i(\mathbf{A}^2) + \dots + \sum_{i=1}^d \lambda_i(\mathbf{A}^d) = 0$ . Therefore, we obtain

$\sum_{i=1}^d \lambda_i(A) + \sum_{i=1}^d \lambda_i(A)^2 + \dots + \sum_{i=1}^d \lambda_i(A)^d = 0$ . Since  $A$  is non-negative, we can conclude that  $\lambda_i(A) = 0$  for  $i \in \{1, 2, \dots, d\}$ .

**Item (3).** We have that a nilpotent matrix is a matrix whose eigenvalues are all zero. According to item (2) of Proposition 1, we can conclude that item (3) of Proposition 1.

**Item (4).** From the definition of the spectral radius,  $\rho(A) = \max_{1 \leq i \leq d} |\lambda_i(A)|$ . We can conclude that  $\rho(A) = 0$  is indeed equivalent to the eigenvalues of  $A$  being all zero. This implies that  $G(A) \in \mathcal{S}_G$ .

Item (4) makes a clever relationship between DAGs and the spectral radius of a matrix. This link permits the application of results from graph theory connected to spectral radius theory to the task of DAG structure learning. As a result, certain research have successfully lowered the time complexity of computing the spectral radius to  $O(d^2)$  or even  $O(d)$ , reducing the computational burden of DAG structure learning methods dramatically.

**Item (5).** If matrix  $A$  can be transformed, via a similarity transformation, into a strictly upper (or lower) triangular matrix, then it follows that the eigenvalues of  $A$  are all 0. Based on item (2) of Proposition 1, we can conclude that  $G(A) \in \mathcal{S}_G$ .

Conversely, if  $G(A) \in \mathcal{S}_G$ , we can perform a topological sorting on  $G(A)$  to obtain at least one vertex sequence. By renumbering the vertices following the rule “if there is a path from vertex  $i$  to  $j$ , then vertex  $i$  has a smaller (or larger) number than vertex  $j$ ”, we obtain the isomorphic graph  $G(A')$  of  $G(A)$ . It is evident that  $A'$  is a strictly upper (lower) triangular matrix. According to the properties of isomorphic graphs, there exists a permutation matrix  $P$  (where  $P$  is an orthogonal matrix,  $P^T = P^{-1}$ ) such that  $P^T A P = A'$ . This implies that  $A$  and  $A'$  are similar, and both matrices are strictly upper (lower) triangular matrices.

**Item (6).** If  $G(A) \in \mathcal{S}_G$ , according to item (3) of Proposition 1, there exists an integer  $k$  ( $k \in \mathbb{Z}^+, k \leq d$ ) make  $A^k = \mathbf{0}$ , let  $U = I + A + A^2 + A^3 + \dots + A^{k-1}$ , then  $(I - A)U = (I - A)(I + A + A^2 + A^3 + \dots + A^{k-1}) = I - A + A - A^2 + A^2 - \dots - A^k = I - A^k = I$ . From this, we can conclude that  $(I - A)^{-1}$  exists and is equal to  $U$  (According to the properties of matrix power series,  $U$  strictly converges when  $\rho(A) < 1$ ). Additionally, since  $A$  is non-negative, we have  $U = (I - A)^{-1} \geq 0$ .

Conversely, let  $U = I + A + \dots + A^{k-1}$  ( $k \rightarrow \infty$ ),  $(I - A)U = I - A^k$ , and  $(I - A)^{-1} \geq 0$ , then  $(I - A)^{-1}(I - A)U = (I - A)^{-1}(I - A^k)$ . This implies  $U = (I - A)^{-1} - (I - A)^{-1}A^k$ . Consequently,  $U + (I - A)^{-1}A^k = (I - A)^{-1}$ , since  $(I - A)^{-1}$  exists, indicating  $\rho(A) < 1$ , we have  $A^k = \mathbf{0}$ , meaning that  $A$  is a nilpotent matrix. This implies the existence of some integer  $m \leq d$  ( $m \in \mathbb{Z}^+$ ) for which  $A^m = \mathbf{0}$ . From item (3) of Proposition 1, we conclude that  $G(A) \in \mathcal{S}_G$ . □

### A.2 Proof of Proposition 2

**Proof** If  $G(W \circ W) \in \mathcal{S}_G$ , according to item (3) of Proposition 1, there exists an integer  $k$  ( $k \in \mathbb{Z}^+, k \leq d$ ) such that  $(W \circ W)^k = \mathbf{0}$ . Let  $U = \frac{1}{\alpha}I + \frac{1}{\alpha^2}W \circ W + \dots + \frac{1}{\alpha^k}(W \circ W)^{k-1}$ , then  $(\alpha I - W \circ W)U = (\alpha I - W \circ W)[\frac{1}{\alpha}I + \frac{1}{\alpha^2}W \circ W + \dots + \frac{1}{\alpha^k}(W \circ W)^{k-1}] = I - \frac{1}{\alpha^k}(W \circ W)^k = I$ . Therefore,  $U = (\alpha I - W \circ W)^{-1}$ , and due to  $W \circ W \geq 0$ , we have  $(\alpha I - W \circ W)^{-1} \geq 0$ .

Let  $U = \frac{1}{\alpha}I + \frac{1}{\alpha^2}W \circ W + \frac{1}{\alpha^3}(W \circ W)^2 + \dots + \frac{1}{\alpha^k}(W \circ W)^{k-1} \geq 0$  ( $k \rightarrow \infty$ ),  $(\alpha I - W \circ W)U = I - \frac{1}{\alpha^k}(W \circ W)^k$ , since  $(\alpha I - W \circ W)^{-1} \geq 0$ ,  $(\alpha I - W \circ W)^{-1}(\alpha I - W \circ W)U = (\alpha I - W \circ W)^{-1}[I - \frac{1}{\alpha^k}(W \circ W)^k]$ . This implies  $U + (\alpha I - W \circ W)^{-1}\frac{1}{\alpha^k}(W \circ W)^k = (\alpha I - W \circ W)^{-1}$ , since  $(\alpha I - W \circ W)^{-1}$  exists, it follows that  $\rho(W \circ W) < \alpha$ , which implies  $(W \circ W)^k = \mathbf{0}$ . Therefore,  $W \circ W$  is a nilpotent

matrix, and there must exist  $m \leq d$  ( $m \in \mathbb{Z}^+$ ) such that  $(W \circ W)^m = \mathbf{0}$ . According to item (3) of Proposition 1, we can conclude that  $\mathbb{G}(W \circ W) \in \mathbb{S}_{\mathcal{G}}$ .  $\square$

### A.3 Proof of Corollary 1

**Proof** For any  $W \in \mathbb{W}^\alpha$ ,  $h(W) = \text{tr}[\sum_{k=1}^\infty \frac{1}{\alpha^{k+1}}(W \circ W)^k]$ , it is obvious that  $\text{tr}[\sum_{k=1}^\infty \frac{1}{\alpha^{k+1}}(W \circ W)^k] \geq 0$ , which means  $h(W) \geq 0$ .

The following paragraph proves that  $\mathbb{G}(W) \in \mathbb{S}_{\mathcal{G}}$  iff  $h(W) = 0$ .

If  $h(W) = 0$ , which implies  $\text{tr}[\sum_{k=1}^\infty \frac{1}{\alpha^{k+1}}(W \circ W)^k] = 0$ , we can conclude that there are no cycles in the graph  $\mathbb{G}(W)$  based on the previous analysis. In other words, it means that  $\mathbb{G}(W) \in \mathbb{S}_{\mathcal{G}}$ .

If  $\mathbb{G}(W) \in \mathbb{S}_{\mathcal{G}}$ , then  $\text{tr}[\sum_{k=1}^\infty \frac{1}{\alpha^{k+1}}(W \circ W)^k] = 0$ . By utilizing the expression  $\text{tr}[(\alpha I - W \circ W)^{-1}] = \text{tr}[\sum_{k=0}^\infty \frac{1}{\alpha^{k+1}}(W \circ W)^k]$ , it can be determined that  $\text{tr}[(\alpha I - W \circ W)^{-1}] = d/\alpha$ . This denotes that  $h(W) = 0$ .  $\square$

### A.4 Proof of Corollary 2

**Proof** Based on Definition 2, we can conclude that  $\alpha I - W \circ W$  is an M-matrix, which implies that  $(\alpha I - W \circ W)^{-1} \geq 0$ . Consequently,  $(\alpha I - W \circ W)^{-2} \geq 0$ . By considering the expression  $\nabla h(W) = 2[(\alpha I - W \circ W)^{-2}]^T \circ W$ , it is evident that when  $\nabla h(W)_{ij} \neq 0$ , the sign of the corresponding element in  $\nabla h(W)$  matches that of the element in  $W$  at the corresponding position. The power series expansion of  $(\alpha I - W \circ W)^{-1}$  is provided as follow:

$$(\alpha I - W \circ W)^{-1} = \sum_{k=0}^\infty \frac{1}{\alpha^{k+1}}(W \circ W)^k = \frac{1}{\alpha}I + \dots + \frac{1}{\alpha^{k+1}}(W \circ W)^k$$

$$(\alpha I - W \circ W)^{-2} = \frac{1}{\alpha^2}I + \dots + \frac{k+1}{\alpha^{k+2}}(W \circ W)^k = \sum_{k=0}^\infty \frac{k+1}{\alpha^{k+2}}(W \circ W)^k$$

$\nabla h(W)_{ij} = 2[(\alpha I - W \circ W)^{-2}]^T_{ij} \circ W_{ij}$  implies that if  $\nabla h(W)_{ij} \neq 0$ , then  $[(\alpha I - W \circ W)^{-2}]^T_{ij} \neq 0$  and  $W_{ij} \neq 0$ . Now, consider  $[(\alpha I - W \circ W)^{-2}]^T_{ij} = [\sum_{k=0}^\infty \frac{k+1}{\alpha^{k+2}}(W \circ W)^k]_{ij}^T \neq 0$ . This implies that there exists at least one term in the polynomial sum for some  $m \in \mathbb{Z}^+$ ,  $(W \circ W)^m_{ji} \neq 0$ , indicating the presence of a cycle of length  $m$  in the graph, such as  $i \rightarrow j \rightarrow \dots \rightarrow i$ . Therefore, if  $\nabla h(W)_{ij} \neq 0$ , it suggests the presence of a cycle in  $\mathbb{G}(W)$ .  $\square$

## References

1. Bello K, Aragam B, Ravikumar P (2022) Dagma: Learning DAGs via m-matrices and a log-determinant acyclicity characterization. *Adv Neural Inf Process Syst* 35:8226–8239
2. Bertsekas DP (2014) *Constrained optimization and Lagrange multiplier methods*. Academic Press, Cambridge
3. Bouckaert RR (1993) Probabilistic network construction using the minimum description length principle. In: *European conference on symbolic and quantitative approaches to reasoning and uncertainty*, Springer, pp 41–48
4. Cai R, Chen W, Zhang K et al (2017) A survey on non-temporal series observational data based causal discovery. *Chin J Comput* 40(6):1470–1490



5. Chickering DM (2002) Optimal structure identification with greedy search. *J Mach Learn Res* 3(Nov):507–554
6. Chickering M, Heckerman D, Meek C (2004) Large-sample learning of Bayesian networks is NP-hard. *J Mach Learn Res* 5:1287–1330
7. Cussens J (2012) Bayesian network learning with cutting planes. arXiv preprint [arXiv:1202.3713](https://arxiv.org/abs/1202.3713)
8. Guo R, Cheng L, Li J et al (2020) A survey of learning causality with data: Problems and methods. *ACM Comput Surv* 53(4):1–37
9. Heckerman D, Geiger D, Chickering DM (1995) Learning Bayesian networks: the combination of knowledge and statistical data. *Mach Learn* 20:197–243
10. Hernán MA (2018) The c-word: scientific euphemisms do not improve causal inference from observational data. *Am J Public Health* 108(5):616–619
11. Imbens GW (2022) Causality in econometrics: choice vs chance. *Econometrica* 90(6):2541–2566
12. Kreif N, DiazOrdaz K (2019) Machine learning in policy evaluation: new tools for causal inference. In: *Oxford research encyclopedia of economics and finance*. Oxford University Press
13. Lee HC, Danieletto M, Miotto R, et al (2019) Scaling structural learning with no-bears to infer causal transcriptome networks. In: *Pacific symposium on biocomputing 2020*, World Scientific, pp 391–402
14. Maxwell Chickering D, Heckerman D (1997) Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. *Mach Learn* 29:181–212
15. Ostrowski A (1937) Über die determinanten mit überwiegender hauptdiagonale. *Commentarii Mathematici Helvetici* 10(1):69–96
16. Pamfil R, Sriwattanaworachai N, Desai S, et al (2020) Dynotears: Structure learning from time-series data. In: *International conference on artificial intelligence and statistics*, PMLR, pp 1595–1605
17. Pearl J (2018) Theoretical impediments to machine learning with seven sparks from the causal revolution. In: *Proceedings of the eleventh ACM international conference on web search and data mining*, pp 3
18. Pearl J et al (2000) *Models, reasoning and inference*. Cambridge University Press, Cambridge, p 3
19. Petersen M, Balzer L, Kwarsiima D et al (2017) Association of implementation of a universal testing and treatment intervention with HIV diagnosis, receipt of antiretroviral therapy, and viral suppression in East Africa. *JAMA* 317(21):2196–2206
20. Robinson RW (1977) Counting unlabeled acyclic digraphs. In: *Combinatorial mathematics V: proceedings of the fifth Australian conference, held at the royal melbourne institute of technology, August 24–26, 1976*, Springer, pp 28–43
21. Rubin DB (1974) Estimating causal effects of treatments in randomized and nonrandomized studies. *J Edu Psychol* 66(5):688
22. Sachs K, Perez O, Pe'er D et al (2005) Causal protein-signaling networks derived from multiparameter single-cell data. *Science* 308(5721):523–529
23. Verma TS, Pearl J (2022) Equivalence and synthesis of causal models. In: *probabilistic and causal inference: the works of Judea Pearl*. pp 221–236
24. Vowels MJ, Camgoz NC, Bowden R (2022) D’ya like DAGs? A survey on structure learning and causal discovery. *ACM Comput Surv* 55(4):1–36
25. Wei D, Gao T, Yu Y (2020) DAGs with no fears: a closer look at continuous optimization for learning Bayesian networks. *Adv Neural Inf Process Syst* 33:3895–3906
26. Yao L, Chu Z, Li S et al (2021) A survey on causal inference. *ACM Trans Knowl Discov Data (TKDD)* 15(5):1–46
27. Yu Y, Chen J, Gao T, et al (2019) Dag-gnn: Dag structure learning with graph neural networks. In: *International conference on machine learning*, PMLR, pp 7154–7163
28. Yu Y, Gao T, Yin N, et al (2021) DAGs with no curl: an efficient dag structure learning approach. In: *International conference on machine learning*, PMLR, pp 12156–12166
29. Yuan C, Malone B, Wu X (2011) Learning optimal Bayesian networks using a\* search. In: *Twenty-second international joint conference on artificial intelligence*
30. Zhang Z, Ng I, Gong D et al (2022) Truncated matrix power iteration for differentiable dag learning. *Adv Neural Inf Process Syst* 35:18390–18402
31. Zheng X, Aragam B, Ravikumar PK, et al (2018) Dags with no tears: Continuous optimization for structure learning. *Adv Neural Inf Process Syst* 31
32. Zhong K, Yen IEH, Dhillon IS, et al (2014) Proximal quasi-newton for computationally intensive 11-regularized m-estimators. *Adv Neural Inf Process Syst* 27
33. Zhu R, Pfadler A, Wu Z, et al (2021) Efficient and scalable structure learning for bayesian networks: Algorithms and applications. In: *2021 IEEE 37th international conference on data engineering (ICDE)*, IEEE, pp 2613–2624
34. Zhu S, Ng I, Chen Z (2019) Causal discovery with reinforcement learning. In: *International conference on learning representations*

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.