Efficient Nonlinear DAG Learning under Projection Framework

Naiyu Yin[1][0009−0001−0120−4852], Yue Yu[2][0000−0002−9150−3986], Tian Gao[3][0000−0002−0337−6682], and Qiang Ji[1][0000−0002−4302−288]

[1] Rensselaer Polytechnic Institute, Troy NY 12180, USA
{yinn2, jiq}@rpi.edu
[2] Lehigh University, Bethlehem PA 18015, USA
{yuy214}@lehigh.edu
[3] IBM Research, Yorktown Heights NJ 10598, USA
{tgao}@us.ibm.com

Abstract. Directed Acyclic Graphs (DAGs) are foundational in machine learning, causal inference, and probabilistic modeling. Recovering the underlying DAG structure from observational data is crucial in these areas. The DAG learning can be approached as a constrained optimization problem with a continuous acyclicity constraint, often solved iteratively through sub-problem optimization. A recent breakthrough has shown that the set of DAGs can be represented as the weighted gradients of graph potential functions. Hence, one may search for a DAG in the equivalent space, whereby the acyclicity constraint is guaranteed to be satisfied. However, the original work, DAG-NoCurl, is limited to (generalized) linear structural equation models (SEMs) where explicit weighted adjacency matrices are defined. Herein, we theoretically derive a nonlinear projection formulation and propose an efficient two-step nonlinear DAG learning method, which we coined DAG-NCMLP. The proposed approach first obtains a non-acyclic graph and then projects it to the equivalent space of DAGs to obtain the acyclic graph. Experimental studies on benchmark datasets demonstrate that our proposed method provides similar accuracy, if not better, compared to state-of-the-art non-parametric DAG learning methods with hard-constrained optimization, while substantially reducing the computational time.

Keywords: Causal Discovery · Structure Learning · Directed Acyclic Graphs.

1 Introduction

Directed Acyclic Graphs (DAGs) are foundational in numerous fields, including machine learning [19, 28], causal inference [20], and probabilistic modeling. Their acyclic nature provides a clear directionality, making them ideal for representing causal relationships among variables within a system. Learning the DAG structure from observational data is crucial for uncovering causal mechanisms, making predictions, and understanding complex systems. However, the DAG

learning problem is NP-hard, and the DAG space grows super-exponentially with the number of variables [1]. Zheng et al. (2018) [37] proposes a continuous DAG constraint, transforming the combinatorial optimization problem of DAG learning into a constrained continuous optimization problem. This formulation opens the door to employing various continuous optimization techniques from deep learning [12, 13, 15, 16, 32].

While achieving state-of-the-art accuracy on synthetic and real data, methods developed using the continuous optimization framework with the continuous DAG constraint face challenges in scaling to large datasets with thousands of variables due to their time-consuming nature. One of the primary reasons for this inefficiency is the use of the augmented Lagrangian method to enforce the continuous DAG constraint, as proposed by Zheng et al. (2018) [37]. This procedure transforms the constrained optimization problem into a sequence of soft-constrained optimization sub-problems, which are solved iteratively. To address this efficiency issue, Yu et al. (2021) [34] propose a novel approach that learns the DAG without any explicit acyclicity constraint. Their method projects the DAGs into an equivalent set and optimizes the solution for the DAG parameters within this admissible set. Consequently, the DAG learning problem can be formulated as a continuous optimization problem without an explicit acyclicity constraint, avoiding the need to directly solve the constrained optimization problem using the time-consuming augmented Lagrangian method.

While Yu et al. (2021) [34] demonstrates significant efficiency improvements, it is built upon the linear Structure Equation Model (SEM), where parameters are represented as a weighted adjacency matrix. This formulation cannot be directly applied to nonlinear SEMs with the non-parametric formulation, which uses a gradient-based adjacency matrix representation. Consequently, its performance in terms of accuracy may suffer when applied to complex nonlinear SEMs. To address this limitation, we propose applying the concept of DAG projection to nonlinear SEMs. Specifically, we theoretically establish that an equivalent set of gradient-based adjacency matrices exists and introduce a novel two-step approach to optimizing the solution within this equivalent set search space. Empirical studies demonstrate that our proposed approach achieves a significant efficiency gain over other state-of-the-art nonparametric DAG learning models.

Main Contributions. This paper presents three contributions. 1) We theoretically derive a non-parametric projection formulation for gradient-based adjacency matrices, thereby extending the projection framework's applicability beyond weighted adjacency matrix representation. 2) Building on this non-parametric projection formulation, we introduce a two-step DAG learning approach, referred to as DAG-NCMLP. 3) We empirically demonstrate the effectiveness of our proposed project-based nonparametric DAG learning algorithm on benchmark synthetic and real datasets. Our method significantly enhances computational efficiency while maintaining comparable accuracy to state-of-the-art DAG learning methods.

2 Related Work

The gold standard for establishing causality between variables in an intelligent system is intervention through controlled experiments. However, conducting such experiments is often impractical due to cost or feasibility constraints. As a result, recent studies have focused on recovering causal relationships solely from observational data. Causal discovery involves identifying causal relationships among a set of random variables in the form of DAGs using observational data.

The traditional causal discovery algorithms can be broadly categorized into two groups: constraint-based methods and score-based methods. Constraint-based methods estimate the DAG by conducting independent tests between variables. Popular algorithms in this category include PC [27], FCI [28, 36], and IC [21]. On the other hand, score-based methods involve pre-defining a score function and searching the DAG space for a DAG with the optimal score. The differences among score-based methods lie in their search procedures, which can include hill-climbing [9, 30], forward-backward search [1], dynamic programming [26], A^* [35], and integer programming [2, 11]. Other widely used DAG learning methods include topological order-based search [4, 6, 25, 29] and sampling [3, 5, 7, 8, 14, 18, 31].

Structure equation model-based methods encode statistical and causal dependencies through SEMs. Zheng et al. (2018) [37] introduced a continuous DAG constraint and the NOTEARS algorithm, which reformulates the original combinatorial DAG learning problem as a constrained continuous optimization. This conversion enables the use of continuous optimization techniques, as demonstrated in subsequent works such as [12], [15], and [32]. Since then, several studies have extended the continuous DAG-constrained optimization formulation from linear models to nonlinear and nonparametric models [6, 13, 32, 38]. To address the efficiency issues in these methods arising from the time-consuming augmented Lagrangian method used to enforce acyclicity, Ng et al. (2020) [16] and Yu et al. (2021) [34] have investigated learning frameworks that do not require an iterative process. Ng et al. (2020) [16] proposes training the framework with a soft acyclicity constraint, while Yu et al. (2021) [34] suggests projecting the DAG into an equivalent set that guarantees acyclicity. However, both works focus on the linear SEM setting. To the authors' best knowledge, this paper is the first attempt at developing an efficient continuous optimization approach without the iterative process for the nonlinear SEM setting.

3 DAG Projection under Nonparameteric SEM

In this section, we provide the theoretical results of the DAG projection framework under the nonlinear SEM. These theoretical results will serve as the fundamental for developing the proposed algorithm in Section 4. With basic and necessary concepts introduced in Section 3.1, our theoretical contribution will be entailed in Section 3.2.

3.1 Preliminary

Nonlinear SEM. Let X denotes a set of d numbers of random variables, $X = (X_1, \cdots, X_d) \in \mathbb{R}^d$. The causal relations between a variable $X_j \in X$ and its parents can be modeled via SEM:

$$X_j = f_j(X_{\pi_j}) + E_j, j = 1, 2, \cdots, d \tag{1}$$

where $f_j(\cdot)$ is the nonlinear structural causal function. X_{π_j} are the parent variables of X_j. E_j is the exogenous noise variable corresponding to variable X_j. Together they account for the effects from all the unobserved latent variables and are assumed to be mutually independent [22].

DAG Learning under Nonlinear SEM. To learn a DAG $\mathcal{G}$ from a given joint distribution, X is modeled via SEMs defined by a set of continuous parameters $\boldsymbol{A} = (A_1, A_2, \cdots, A_d)$ that encode all the causal relations, as outlined in Eq. (2),

$$X_j = f_j(X; A_j) + E_j, j = 1, 2, \cdots, d \tag{2}$$

where A_j are the parameters in the nonlinear SEM for selecting parent variables X_{π_j} for variable X_j. Similar to prior works [38, 13], we employ neural networks, in particular MLPs, to parameterize the nonlinear causal functions $f = (f_1, f_2, \cdots, f_d)$. For f_j, we have

$$f_j(X; A_j) = A_j^{(H)} \sigma \left(\cdots \sigma \big(A_j^{(2)} \sigma (A_j^{(1)} X) \big) \cdots \right) \tag{3}$$

where A_j^h represents the parameters for h^{th} layer in the MLP for X_j. We denote $A_j := (A_j^{(1)}, A_j^{(2)}, \cdots, A_j^{(h)}, \cdots, A_j^{(H)})$. Since $\boldsymbol{A}$ in the nonlinear SEM is not a weighted adjacency matrix with d by d dimensions, the DAG learning formulation that satisfies Eq. (2) and Eq. (3) is also known as **nonparametric SEM** according to [38]. We denote the $\boldsymbol{A}$ in the nonlinear SEM as the **gradient-based adjacency matrices**. We encode the causal dependencies in the first layers of MLPs, i.e., $A_1^{(1)}, A_2^{(1)}, \cdots, A_d^{(1)}$. We can obtain a weighted adjacency matrix $W(\boldsymbol{A}) \in \mathbb{R}^{d \times d}$ using the first layer weights, i.e., $W(\boldsymbol{A})[k, j] = \sqrt{\sum_b (A_j^{(1)}[b, k])^2}$. If there exists a causal link from variable X_k to X_j, then $W(\boldsymbol{A})[k, j] > 0$. Otherwise, we have $W(\boldsymbol{A})[k, j] = 0$ and equivalently $A_j^{(1)}[b, k] = 0$ for all b.

Given n observations of X, denoted as input data matrix $\boldsymbol{X}^{d \times n}$, the DAG learning problem can be formulated as follow

$$\begin{aligned} \boldsymbol{A}^* = \arg\min_{\boldsymbol{A}} & \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{d} \mathcal{L}\Big(X_j(i), f_j(\boldsymbol{X}(i); A_j) \Big) \\ & \text{subject to } h(W(\boldsymbol{A})) = 0 \end{aligned} \tag{4}$$

where $\boldsymbol{X}(i) \in \mathbb{R}^d$ is the i^{th} observation of variables X. $X_j(i)$ is the i^{th} observation of variable X_j. $h(W(\boldsymbol{A})) = tr(e^{W(\boldsymbol{A}) \circ W(\boldsymbol{A})}) - d = 0$ is the continuous acyclicity constraint following [38]. $\mathcal{L}(\cdot)$ is the least squared loss.

The SEM we employ in Eq. (2) has following assumptions: firstly, $f = (f_1, f_2, \cdots, f_d)$ represents a set of nonlinear causal functions; and secondly, $E_1, E_2, \cdots, E_d$ are independent noise variables. According to [23], given a distribution over random variables $p(X)$, a unique causal graph $\mathcal{G}$ can be identified.

We then briefly introduce notation for graph calculus in the following section.

Graph Calculus: Let $\widehat{\mathcal{G}} = (V, E)$ be a **complete undirected** graph where $V := \{1, \cdots, d\}$ is the set of vertices and E is the set of undirected edges. On each vertice, there is a real-valued function $f : V \to \mathbb{R}$, which is also known as the potential function. We denote the space of all potential functions as $L^2(V)$. We also define real-valued functions on edges $E = \{(i, j), i, j \in V\}$ with the requirement that these functions are alternating, i.e., $E[i, j] = -E[j, i]$. We denote the space of all alternating edge functions as $L^2_\wedge(E)$. Here we note that $p \in L^2(V)$ corresponds to a real vector $p = [p(1), \cdots, p(d)] \in \mathbb{R}^d$, and any $Y \in L^2_\wedge(E)$ corresponds to a skew-symmetric real matrix $Y \in \mathbb{R}^{d\times d}$ with $[Y]_{ij} = Y[i, j]$ and $Y = -Y^T$. We will use the same letter to denote a vector/matrix and the corresponding function on vertices/edges. We introduce graph calculus operators gradient, divergence, and the graph laplacian in **Definition 1**.

Definition 1. *The gradient (*$\text{grad} : L^2(V) \to L^2_\wedge(E)$*) is an operator defined on any function p on vertices:*

$$(\text{grad } p)[i, j] = p(j) - p(i), \quad \forall (i, j) \in E$$

The divergence ($\text{div} : L^2_\wedge(E) \to L^2(V)$*) is defined on any alternating function Y on edges:*

$$(\text{div } Y)(i) = \sum_{j=1}^{d} Y[i, j], \quad \forall i \in V.$$

The graph Laplacian ($\triangle_0 : L^2(V) \to L^2(V)$*) is an operator on any function p on vertices:*

$$(\triangle_0 p)(i) = -(\text{div grad } p)(i) = dp(i) - \sum_{j=1}^{d} p(j), \quad \forall i \in V.$$

Given a function $Y \in L^2_\wedge(E)$, with ReLU denoting the rectified linear unit function, we can find a weighted adjacency matrix $\text{ReLU}(Y) \in \mathbb{R}^{d\times d}$ as:

$$\text{ReLU}(Y)[i, j] = \begin{cases} Y[i, j], & \text{if } Y[i, j] > 0, \\ 0, & \text{else,} \end{cases}$$

We define a weighted directed graph $\mathcal{G}_{\text{ReLU}(Y)}$ from $\text{ReLU}(Y)$ in **Definition 2**:

Definition 2. *Consider a complete undirected graph* $\widehat{\mathcal{G}}(V, E)$ *and* $Y \in L^2_\wedge(E)$*, a directed graph* $\mathcal{G}_{\text{ReLU}(Y)}(V, E_{\text{ReLU}(Y)})$ *is defined such that there is a directed edge from vertex i to vertex j in* $\mathcal{G}_{\text{ReLU}(Y)}$ *if and only if* $Y[i, j] > 0$*, i.e., the set of directed edges* $E_{\text{ReLU}(Y)} = \{(i, j) | Y[i, j] > 0\}$*.* $\text{ReLU}(Y)$ *is a weighted adjacency matrix of* $\mathcal{G}_{\text{ReLU}(Y)}$*.*

Building on **Definition 1** and **Definition 2**, [34] offers an equivalent representation of a DAG under linear SEM, whereby a DAG $\mathcal{G}$ with d nodes is characterized by a weighted adjacency matrix $W \in \mathbb{R}^{d\times d}$. This formulation is supported by **Theorem 1** as presented in [33, 34].

Theorem 1. *[33] For any weight matrix $S \in \mathbb{R}^{d\times d}$ and potential function $p \in L^2(V)$, $S \circ \mathrm{ReLU}(\mathrm{grad}(p))$ is the weighted adjacency matrix of a DAG. On the other hand, let $W \in \mathbb{R}^{d\times d}$ be the weighted adjacency matrix of any DAG with d nodes, then there exists a weight matrix $S \in \mathbb{R}^{d\times d}$ and a function $p \in L^2(V)$ such that $W = S \circ \mathrm{ReLU}(\mathrm{grad}(p))$. Hence, $\{\mathcal{G}_{S\circ\mathrm{ReLU}(\mathrm{grad}(p))}\}$ is equivalent to the DAG space.*

3.2 An Equivalent Model for DAG

Theorem 1 can only be applied to linear SEMs because it requires the usage of the square-weighted adjacency matrices. **Our key theoretical contribution is to derive the equivalent theorem in Theorem 2 for nonlinear (nonparametric) SEMs to remove this limitation and handle the gradient-based adjacency matrix representation in Eq.** (3)**.**

Theorem 2. *The acyclicity holds for the neural network formulation in Eq.* (3) *if and only if there exists a function $p \in L^2(V)$ and weight matrices $S_j \in \mathbb{R}^{m_1\times d}$, $j = 1, \cdots, d$, such that*

$$A_j^{(1)}[b,k] = S_j[b,k]\,\mathrm{ReLU}(\mathrm{grad}(p))[k,j]. \tag{5}$$

Here m_1 is the number of hidden units in the first layer of MLP.

Proof. As shown in [38], $W(\boldsymbol{A})[k,j] = \sqrt{\sum_b (A_j^{(1)}[b,k])^2}$ encodes the dependency structure amongst the X_j and the neural network formulation in Eq. (3) satisfies the acyclicity constraint if and only if $W(\boldsymbol{A})$ is acyclic. Assuming $A_j^{(1)}$ satisfies Eq. (5) for all j, we note that

$$W(\boldsymbol{A})[k,j] = \sqrt{\sum_b (S_j[b,k])^2}\,\mathrm{ReLU}(p(j)-p(k)) = \tilde{S} \circ \mathrm{ReLU}(\mathrm{grad}(p)),$$

where $\tilde{S}[k,j] = \sqrt{\sum_b (S_j[b,k])^2}$ and $\tilde{S} \in \mathbb{R}^{d\times d}$. **Theorem 1** then immediately indicates that $W(\boldsymbol{A})$ is acyclic. On the other hand, if $W(\boldsymbol{A})$ satisfies the acyclicity constraint, **Theorem 1** guarantees that one can find $\tilde{S} \in \mathbb{R}^{d\times d}$ and $p \in L^2(V)$ satisfying

$$W(\boldsymbol{A})[k,j] = \tilde{S}[k,j]\,\mathrm{ReLU}(p(j)-p(k)).$$

Notice that when $\mathrm{ReLU}(p(j)-p(k)) = 0$, we have $W(\boldsymbol{A})[k,j] = \sqrt{\sum_b (A_j^{(1)}[b,k])^2} = 0$ and hence $A_j^{(1)}[b,k] = 0$ for all $b \in \{1, \cdots, m_1\}$. Therefore Eq. (5) can be satisfied by setting

$$S_j[b,k] = \begin{cases} 0, & \text{if } p(j) \le p(k) \\ \frac{A_j^{(1)}[b,k]}{p(j)-p(k)}, & \text{if } p(j) > p(k). \end{cases} \tag{6}$$

The dependency structure $W(\boldsymbol{A})$ obtained by performing projection is a non-maximum acyclic graph that minimizes $\|W(\boldsymbol{A}) - \tilde{S}\|_2$.

Let $C(M)$ denote the connectivity matrix [17] of a directed graph M such that $[C(M)]_{ij} = 1$ only if a directed path exists from vertex i to vertex j. **Theorem 3** provides an efficient approach to calculate p and S_j from $\boldsymbol{A}$:

Theorem 3. *Let $\boldsymbol{A}$ be a set of parameters in Eq. (3) which satisfying the acyclicity constraint, then*

$$p = -\triangle_0^{\dagger} \operatorname{div}\left(\frac{1}{2}(C(W(\boldsymbol{A})) - C(W(\boldsymbol{A}))^T)\right), \tag{7}$$

preserves the topological order in $W(\boldsymbol{A})$ such that $p(j) > p(i)$ if there is a directed path from vertex i to j. Here $\dagger$ denotes the Moore-Penrose pseudo-inverse. Moreover, with S_j defined in Eq. (6) we have

$$A_j^{(1)}[b,k] = S_j[b,k] \operatorname{ReLU}(\operatorname{grad}(p))[k,j]$$

We refer interested readers to Appendix A for a detailed proof.

Theorem 2 and **Theorem 3** allow us to find the equivalent search space for the gradient-based adjacency matrix representations in Eq. (3). We introduce a two-step DAG learning algorithm that optimizes parameters within the equivalent search space, thereby circumventing the need for enforcing the computationally intensive acyclicity constraint.

4 Proposed Algorithm: DAG-NCMLP

Guided by theoretical insights from **Theorem 2** and **Theorem 3**, we propose a nonparametric project-based DAG learning algorithm, named DAG-NCMLP, which employs MLP as the gradient-based weighted adjacency matrix representation. To avoid the strict enforcement of the DAG constraint, we propose learning the neural network parameters, $\boldsymbol{S}$ and the potential function p, instead of directly optimizing a gradient-based weighted adjacency matrix representation $\boldsymbol{A}$ that must satisfy the DAG constraint. Given the increasing complexity of optimizing both $\boldsymbol{S}$ and p, we employ a two-step procedure. In Step 1, we derive an initial solution $\hat{\boldsymbol{A}}$ without strictly adhering to the DAG constraint. This step aims to obtain a good initial solution from which a stable, informative estimate of the potential function, p^{pre}, can be extracted. In Step 2, we focus on optimizing $\boldsymbol{S}$ and p, guided by p^{pre}, to ultimately learn the optimal DAG. The algorithm is outlined in Algorithm 1. We will detail each step as follows.

Step 1. This step aims to yield an estimation of $\boldsymbol{A}$ that produces a stable and preferably informative potential function p. To obtain such an initial estimate, we propose to solve a penalized formulation of the original constrained optimization problem as shown in Eq. (10), by employing the standard augmented Lagrangian method and gradually increase the penalization parameter ρ. Instead of continuing the iterative procedure till convergence as in the original

augmented Lagrangian, here we only solve the sub-optimizations for a few iterations. As a result, the solution is not guaranteed to fully satisfy the acyclicity constraint. To be more specific, we denote the objective function in Eq. (10) as $L_\rho(\boldsymbol{A}, \alpha)$. Initially, we update the penalization parameter ρ by gradually increasing its value while holding $(\alpha, \boldsymbol{A})$ constant. Then, we update α using Eq. (8) for $K = 5$ iterations. For each pair of given ρ and α, we solve the sub-optimization problem in Eq. (9) for $T = 10d$ iterations[4]. Further details of the choices of K and T can be found in Section 5.

$$\alpha^{k+1} = \alpha^k + \rho_{k+1} h(W(\boldsymbol{A}^k)). \tag{8}$$

$$\boldsymbol{A}^{k+1} = \arg\min_{\boldsymbol{A}} L_{\rho_{k+1}}(\boldsymbol{A}^k, \alpha_{k+1}) \tag{9}$$

DAG-NCMLP utilizes a distinct Step 1 procedure compared to [34]. In [34], the optimization is solved with fixed values of α and ρ, akin to the augmented Lagrangian method with only one step of optimization. This approach is sufficient to yield a stable potential vector p under linear SEM with simple linear relationships between variables. However, for nonlinear models, solutions obtained with fixed coefficients are often inadmissible for estimating the potential function.

Step 2. This step aims to optimize the parameters $\boldsymbol{S}$ and p within the equivalent DAG space, using the potential function p^{pre} derived from the initial solution $\hat{\boldsymbol{A}}$. After Step 1, the resulting $W(\hat{\boldsymbol{A}})$ is typically non-acyclic since the DAG constraint is not satisfied. To obtain a DAG solution, we first approximate the potential function p^{pre} using Eq. (7) from **Theorem 3**. Next, we derive an initial graph solution $W(\boldsymbol{A}^{pre})$ in Eq. (12) by optimizing over $\boldsymbol{A}$ with p^{pre} fixed. Finally, we obtain the optimal DAG solution $W(\boldsymbol{A}^*)$ in Eq. (13) by jointly optimizing over $\boldsymbol{A}$ and p. Both in Step 1 and Step 2, we apply the standard thresholding procedure [37] to $W(\hat{\boldsymbol{A}})$ and $W(\boldsymbol{A}^*)$, respectively. The outcome of Step 1 directly impacts Step 2. A more accurate estimation of W^{pre} in Step 1 results in a better approximation of the potential function p^{pre}. This, in turn, encodes more accurate partial ordering information, aiding the algorithm in converging to an accurate estimation of $\boldsymbol{A}^*$ in Step 2c. Here we note that whether W^{pre} satisfies the acyclic constraint does not affect the algorithm's ability to obtain an effective p^{pre}, since p^{pre} can preserve the partial ordering information of a non-acyclic W^{pre}. Our proposed method involves optimizing over both $\boldsymbol{A}$ and p, with each affecting the estimation of the other during the optimization process. Step 2b simplifies the optimization process by fixing p to p^{pre}, allowing $\boldsymbol{A}$ to achieve a good initial estimation. The accuracy of $\boldsymbol{A}^*$ obtained by DAG-NCMLP is greatly compromised if Step 2b is omitted. We also point out that the objective functions in Eq. (12) and Eq. (13) are non-convex. Consequently, only stationary solutions can be guaranteed, a characteristic shared with all continuous optimization-based DAG algorithms.

[4] "Solving the sub-optimization problem for $T = 10d$ iterations" means the optimizer stops when it performs $T = 10d$ gradient descent steps.

Algorithm 1 DAG-NCMLP Algorithm

Step 1: Within fixed numbers of iterations, solve for initialization $\hat{\boldsymbol{A}}$.

$$\begin{aligned}\hat{\boldsymbol{A}} = \quad & \underset{\boldsymbol{A}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\mathcal{L}\Big(X_j(i), f_j(\boldsymbol{X}(i), A_j)\Big) + \lambda||A_j^{(1)}||_{1,1} \\ & + \alpha h(W(\boldsymbol{A})) + \frac{\rho}{2}\|h(W(\boldsymbol{A})\|^2.\end{aligned} \tag{10}$$

Threshold $W(\hat{\boldsymbol{A}})$ to obtain W^{pre}.

Step 2: Obtain an acyclic graph solution $W(\boldsymbol{A}^*)$

2a) Obtain initial guess of potential vector p^{pre}:

$$p^{pre} = -\triangle_0^{\dagger}\,\mathrm{div}\left(\frac{1}{2}(C(W^{pre}) - C(W^{pre})^T)\right), \tag{11}$$

which preserves the variable ordering of W^{pre}.

2b) Solve for the initial guess of DAG $W(\boldsymbol{A}^{pre})$ with fixed potential vector p^{pre} and initialization $\hat{\boldsymbol{A}}$:

$$\boldsymbol{A}^{pre} = \underset{\{\boldsymbol{A},S\}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\mathcal{L}\Big(X_j(i), f_j(\boldsymbol{X}(i), A_j)\Big) + \lambda||A_j^{(1)}||_{1,1} \tag{12}$$

where $A_j^{(1)}[b,k] = S_j[b,k]\mathrm{ReLU}(p^{pre}(j) - p^{pre}(k))$.

2c) Solve for $W(\boldsymbol{A}^*), p^*$ with initialization $\boldsymbol{A}^{pre}$:

$$\boldsymbol{A}^*, p^* = \underset{\{\boldsymbol{A},S,p\}}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{d}\mathcal{L}\Big(X_j(i), f_j(\boldsymbol{X}(i), A_j)\Big) + \lambda||A_j^{(1)}||_{1,1} \tag{13}$$

where $A_j^{(1)}[b,k] = S_j[b,k]\mathrm{ReLU}(p^*(j) - p^*(k))$. Threshold $W(\boldsymbol{A}^*)$ to obtain W^{est} as output.

5 Experiments

We perform empirical evaluations on both synthetic and real data to demonstrate the effectiveness of our proposed DAG-NCMLP algorithm in improving efficiency while maintaining comparable accuracy.

Synthetic Datasets. We evaluate DAG-NCMLP on synthetic nonlinear datasets, generated using the same method as in prior work [38]. The ground truth DAGs are generated from Erdo-Renyi (ER) and Scale-Free (SF) graph models, with an expected edge degree set to 2 and 4. The synthetic data are generated from three-layer MLPs, which are universal nonlinear estimators, following the approach in [38]. To demonstrate the robustness of our proposed method across different data models, we also generate data using the Gaussian Process (GP) SEM. We create 10 graphs for each graph setting (ER2-MLP, ER4-MLP, SF2-MLP, SF4-MLP, ER2-GP, ER4-GP, SF2-GP, and SF4-GP), and test with varying numbers

of variables $d = 10, 20, 40, 50, 100$. For each setting, we simulate 10 trials with $n = 1000$ i.i.d. data observations.

Real Dataset. We further assess the performance of DAG-NCMLP using real-world flow cytometry data from Sachs et al. (2005) [24] for modeling protein signaling pathways. The dataset comprises continuous measurements of 11 phosphoproteins in individual T-cells. We specifically selected 853 observations corresponding to the first experimental condition outlined in Sachs et al. (2005) [24] as our dataset $\mathcal{D}$. For our reference graph (ground truth), we utilize the provided DAG, which consists of 11 nodes and 17 edges. It is important to note that this consensus graph may not provide a comprehensive or entirely accurate representation of the system under study.

Evaluation Metrics. We employ the Structural Hamming Distance (SHD) and runtime to evaluate the accuracy and efficiency of the estimated DAGs respectively. We report the average SHD with its standard deviation across 10 trials, and the average time (in seconds) with its standard deviation in Table 1, 2, and 3. The SHD metric we use doesn't consider Markov Equivalence since the non-linear SEM in our formulation is fully identifiable.

Baselines. We mainly compare our method with following SEM-based baselines: GraN-DAG [13], DAG-GNN [32], GS-GES[10] and NOTEARS-MLP [38]. We use the default parameters for these baselines. For the baseline NOTEARS-MLP, we use the hyper-parameters that are reported in Zheng et al. (2020) [38]. The experiments for all the baselines and the proposed method, DAG-NCMLP are computed on a computing node with twenty 3.1 GHz CPU cores[5].To provide a more comprehensive comparison, we also compared causal discovery methods from different categories, including MMHC [30] and DAG-NoCurl [34] and show the empirical results in Appendix B.

The choice of K and T. The hyperparameters K and T control the accuracy of the potential function p^{pre}, and consequently, the accuracy of the final output DAG. Ideally, we want to select relatively small values for K and T to enhance the algorithm's efficiency by reducing the number of optimization steps. However, K and T should also be large enough to allow p^{pre} to capture as much information as possible. A reasonable approach to selecting the hyper-parameters K, T is through empirical evaluation. The K, T are empirically selected on ER2 datasets when values of p do not change substantially (note that we do not use accuracy or SHD as the selection criterion). We observe that the algorithm performance is not sensitive to the values of K, T, Hence we fix the values of $K = 5$ and $T = 10d$.

[5] Due to the complexity of the neural networks used in methods like DAG-GNN and GraN-DAG, these models are typically run on a GPU to reduce runtime. However, to ensure a fair comparison of efficiency, we run experiments for these two baselines on a CPU, as with the other baselines. GPU acceleration is a standard technique and not a unique contribution of these two baselines; it can be applied to all the algorithms, including our DAG-NCMLP.

5.1 Empirical Results on Synthetic Data

Table 1. Comparison of Different Algorithms on Nonlinear **Multi-Layer Perceptron** Synthetic datasets: results (mean ± standard error over 10 trails) on SHD and Run time(in seconds), where bold number s highlight the best method for each case.

ER2: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	15.0 ± 6.0	13.3 ± 5.5	10.5 ± 3.9	5.7 ± 3.2	$\mathbf{5.5 \pm 2.5}$
20	22.7 ± 1.8	25.7 ± 3.6	19.4 ± 5.6	$\mathbf{13.0 \pm 3.8}$	13.5 ± 4.0
40	57.5 ± 8.6	56.1 ± 6.7	40.5 ± 9.5	$\mathbf{27.7 \pm 5.1}$	27.8 ± 5.8
50	68.9 ± 13.3	65.8 ± 7.8	50.6 ± 8.4	$\mathbf{36.0 \pm 9.7}$	36.9 ± 10.3
100	$> 60h$	144.8 ± 7.1	$> 60h$	$\mathbf{77.3 \pm 4.0}$	80.5 ± 6.0
SF2: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	9.0 ± 4.5	9.5 ± 3.4	8.5 ± 3.2	$\mathbf{1.9 \pm 1.2}$	2.2 ± 1.1
20	19.7 ± 2.1	22.9 ± 3.4	22.7 ± 4.4	8.0 ± 3.2	$\mathbf{7.8 \pm 2.9}$
40	48.6 ± 4.6	52.4 ± 3.1	51.3 ± 7.0	18.9 ± 6.5	$\mathbf{18.5 \pm 6.2}$
50	52.3 ± 11.9	58.6 ± 6.5	65.1 ± 5.6	$\mathbf{24.5 \pm 6.2}$	25.5 ± 5.5
100	$> 60h$	149.2 ± 7.6	149.5 ± 7.2	81.4 ± 9.9	$\mathbf{79.0 \pm 7.1}$
ER4: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	22.4 ± 4.8	27.1 ± 3.4	24.10 ± 7.1	$\mathbf{8.0 \pm 1.9}$	9.9 ± 2.4
20	71.2 ± 16.2	65.5 ± 8.1	50.2 ± 9.7	$\mathbf{29.1 \pm 4.7}$	32.7 ± 7.1
40	96.7 ± 18.4	130.4 ± 10.2	87.7 ± 12.8	$\mathbf{47.7 \pm 9.3}$	55.0 ± 25.9
50	121.0 ± 16.9	161.1 ± 10.8	115.70 ± 21.8	$\mathbf{68.7 \pm 14.0}$	70.9 ± 15.3
100	$> 60h$	332.2 ± 12.6	$> 60h$	$\mathbf{134.5 \pm 13.4}$	144.1 ± 38.0
SF4: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	14.0 ± 1.5	18.1 ± 3.3	18.3 ± 5.6	$\mathbf{3.5 \pm 2.3}$	4.8 ± 2.9
20	29.7 ± 2.1	48.2 ± 5.5	44.3 ± 3.9	$\mathbf{12.4 \pm 4.2}$	$\mathbf{12.4 \pm 4.1}$
40	78.6 ± 4.2	119.9 ± 6.8	111.0 ± 7.2	$\mathbf{47.2 \pm 5.5}$	48.7 ± 6.5
50	132.3 ± 11.0	158.6 ± 6.5	148.3 ± 7.6	$\mathbf{62.1 \pm 21.2}$	77.7 ± 17.4
100	$> 60h$	323.1 ± 9.6	$> 60h$	211.1 ± 11.8	$\mathbf{202.0 \pm 10.8}$
ER2: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	$2.6e3 \pm 5.2e3$	$6.6e2 \pm 2.3e2$	$3.1e2 \pm 5.3e1$	$3.1e2 \pm 1.1e2$	$\mathbf{1.2e2 \pm 6.0e1}$
20	$2.5e3 \pm 5.8e2$	$1.5e3 \pm 2.3e2$	$1.5e3 \pm 1.2e2$	$7.6e2 \pm 1.4e2$	$\mathbf{3.9e2 \pm 8.6e1}$
40	$8.6e3 \pm 1.8e3$	$8.0e3 \pm 1.9e2$	$6.8e3 \pm 1.4e3$	$1.6e3 \pm 2.2e2$	$\mathbf{1.1e3 \pm 2.1e2}$
50	$1.4e4 \pm 1.5e3$	$1.2e4 \pm 1.1e2$	$1.0e4 \pm 9.4e2$	$8.5e3 \pm 2.3e3$	$\mathbf{1.6e3 \pm 3.9e2}$
100	$> 60h$	$2.0e4 \pm 6.4e2$	$> 60h$	limit to $60h$	$\mathbf{6.5e3 \pm 1.3e3}$
SF2: Run Time					
d	GraN-DA	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	$1.2e3 \pm 1.3e2$	$1.1e3 \pm 1.7e2$	$2.6e2 \pm 2.7e1$	$2.4e2 \pm 8.2e1$	$\mathbf{7.5e1 \pm 3.5e1}$
20	$4.6e3 \pm 1.7e3$	$1.2e3 \pm 2.2e2$	$1.1e3 \pm 1.4e2$	$1.8e3 \pm 3.9e2$	$\mathbf{3.4e2 \pm 5.9e1}$
40	$1.5e4 \pm 1.3e4$	$3.2e3 \pm 2.8e2$	$4.4e3 \pm 3.9e2$	$2.8e3 \pm 6.6e3$	$\mathbf{1.1e3 \pm 1.3e2}$
50	$2.2e4 \pm 5.2e2$	$2.1e4 \pm 1.5e2$	$7.0e3 \pm 5.0e2$	$4.5e3 \pm 1.2e3$	$\mathbf{1.8e3 \pm 5.3e2}$
100	$> 60h$	$> 60h$	$> 60h$	limit to $60h$	$\mathbf{9.3e3 \pm 2.7e3}$
ER4: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	$1.2e3 \pm 2.3e2$	$8.6e2 \pm 8.2e1$	$6.7e2 \pm 3.6e2$	$1.2e3 \pm 4.4e2$	$\mathbf{1.5e2 \pm 9.2e1}$
20	$7.1e3 \pm 8.2e2$	$9.6e2 \pm 7.5e1$	$5.5e3 \pm 9.9e3$	$2.7e3 \pm 6.8e2$	$\mathbf{6.4e2 \pm 2.7e2}$
40	$7.6e3 \pm 1.0e3$	$7.2e3 \pm 9.7e2$	$9.6e3 \pm 2.1e3$	$7.4e3 \pm 1.6e3$	$\mathbf{1.5e3 \pm 2.5e2}$
50	$1.9e4 \pm 7.8e2$	$2.1e4 \pm 2.1e2$	$1.9e4 \pm 2.1e2$	$1.0e4 \pm 2.1e3$	$\mathbf{2.3e3 \pm 4.6e2}$
100	$> 60h$	$> 60h$	$> 60h$	limit to $60h$	$\mathbf{7.4e3 \pm 2.2e3}$
SF4: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	$1.2e3 \pm 1.8e2$	$8.4e2 \pm 1.1e2$	$3.2e2 \pm 3.2e1$	$8.4e2 \pm 3.7e2$	$\mathbf{1.2e2 \pm 4e1}$
20	$1.3e3 \pm 2.7e2$	$8.2e2 \pm 1.5e2$	$1.3e3 \pm 1.8e2$	$1.5e3 \pm 4.4e2$	$\mathbf{3.6e2 \pm 9.9e1}$
40	$9.8e3 \pm 1.5e2$	$7.8e3 \pm 1.9e2$	$5.4e3 \pm 5.8e2$	$8.4e3 \pm 3.7e2$	$\mathbf{1.3e3 \pm 3.2e2}$
50	$2.2e4 \pm 6.8e3$	$1.6e4 \pm 2.0e3$	$9.1e3 \pm 1.9e3$	$6.8e3 \pm 3.3e3$	$\mathbf{2.7e3 \pm 6.7e2}$
100	$> 60h$	$> 60h$	$> 60h$	limit to $60h$	$\mathbf{7.1e3 \pm 8.6e2}$

In Tables 1 and 2, the top four sub-tables present the accuracy results in terms of the SHD. The bottom four sub-tables display the computational efficiency measured in CPU runtime in seconds. Given the complexity of the data, we imposed a 60-hour time limit for each method and then evaluated the inter-

Table 2. Comparison of Different Algorithms on Nonlinear **Gaussian Process** Synthetic datasets: results (mean ± standard error over 10 trails) on SHD and Run time(in seconds), where bold number s highlight the best method for each case.

ER2: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	12.3 ± 1.5	17.3 ± 0.9	9.6 ± 2.4	**7.2 ± 2.1**	7.5 ± 2.2
20	34.3 ± 8.8	36.0 ± 1.7	**19.3 ± 6.5**	30.0 ± 0.7	30.5 ± 10.8
40	48.4 ± 4.4	73.2 ± 2.1	**33.7 ± 10.0**	43.2 ± 7.2	42.7 ± 7.9
50	71.2 ± 12.4	93.1 ± 3.1	**47.8 ± 8.2**	62.1 ± 9.6	62.2 ± 8.9
100	> 60h	185.6 ± 3.7	**94.5 ± 7.3**	125.7 ± 2.5	128.3 ± 3.3
SF2: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	16.4 ± 2.0	15.4 ± 0.9	11.4 ± 3.7	**7.5 ± 1.9**	7.7 ± 2.3
20	33.1 ± 5.4	34.8 ± 1.2	31.2 ± 3.9	**28.6 ± 2.7**	29.1 ± 4.4
40	62.3 ± 6.5	72.4 ± 1.6	64.8 ± 6.7	**58.6 ± 5.0**	58.7 ± 4.5
50	94.8 ± 10.8	91.7 ± 2.3	82.1 ± 8.2	**77.0 ± 4.4**	79.0 ± 5.0
100	> 60h	185.8 ± 1.4	> 60h	**171.2 ± 2.1**	173.0 ± 3.3
ER4: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	20.8 ± 3.8	38.3 ± 1.1	32.3 ± 2.6	**19.3 ± 3.4**	21.1 ± 2.7
20	68.1 ± 9.7	78.7 ± 1.0	60.5 ± 3.6	56.4 ± 4.2	**56.3 ± 4.3**
40	154.9 ± 7.0	140.8 ± 4.7	**119.0 ± 7.2**	138.1 ± 6.0	138.2 ± 9.0
50	186.6 ± 16.2	195.7 ± 1.7	**154.9 ± 5.6**	188.3 ± 12.7	189.5 ± 15.4
100	> 60h	391.0 ± 2.7	**309.0 ± 9.9**	350.7 ± 2.1	354.3 ± 4.2
SF4: SHD					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	27.1 ± 4.4	28.8 ± 1.0	24.3 ± 2.7	**15.1 ± 3.1**	16.1 ± 3.9
20	63.1 ± 1.9	68.2 ± 1.2	61.0 ± 4.0	60.7 ± 2.3	**60.5 ± 2.4**
40	139.9 ± 6.1	145.8 ± 1.7	133.4 ± 2.8	**129.4 ± 4.0**	131.1 ± 4.3
50	184.6 ± 4.2	185.1 ± 1.4	**169.8 ± 6.5**	171.8 ± 4.5	170.4 ± 4.4
100	> 60h	379.2 ± 3.2	> 60h	**351.7 ± 4.1**	356.0 ± 5.0
ER2: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	5.3e2 ± 7.5e1	5.3e2 ± 3.7e1	1.8e2 ± 1.8e1	1.7e2 ± 7.8e1	**6.5e1 ± 1.6e1**
20	7.6e2 ± 7.7e1	5.5e2 ± 4.7e1	8.0e2 ± 9.0e1	1.1e3 ± 2.7e2	**7.8e1 ± 1.9e1**
40	2.1e3 ± 1.4e2	6.6e2 ± 3.2e1	4.1e3 ± 3.6e2	2.7e3 ± 7.5e2	**5.4e2 ± 4.9e1**
50	2.4e3 ± 2.1e2	7.6e2 ± 4.6e1	5.5e3 ± 5.2e2	4.1e3 ± 8.8e2	**7.0e2 ± 4.4e1**
100	> 60h	3.6e3 ± 4.1e2	2.1e4 ± 1.1e3	limit to 60h	**1.2e3 ± 3.1e1**
SF2: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	4.2e2 ± 3.4	4.3e2 ± 4.0e1	1.3e2 ± 1.4e1	1.6e2 ± 4.7e1	**2.1e1 ± 4.5**
20	9.1e2 ± 1.4e2	5.3e2 ± 2.3e1	4.9e2 ± 6.3e1	5.4e2 ± 1.7e2	**8.6e1 ± 1.4e1**
40	2.0e3 ± 1.7e2	6.8e2 ± 5.5e1	2.3e3 ± 2.5e2	1.5e3 ± 4.2e2	**5.5e2 ± 6.2e1**
50	2.8e3 ± 5.2e2	8.3e2 ± 8.8e1	7.0e3 ± 5.0e2	2.8e3 ± 7.0e2	**6.6e2 ± 3.2e1**
100	> 60h	2.3e3 ± 1.8e2	> 60h	limit to 60h	**1.2e3 ± 3.0e1**
ER4: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	3.4e2 ± 4.4e1	5.0e2 ± 2.1e1	2.5e2 ± 4.0e1	1.5e2 ± 2.5e1	**6.6e1 ± 1.1e1**
20	6.8e2 ± 9.8e1	5.5e2 ± 3.5e1	9.6e2 ± 8.5e1	5.8e2 ± 1.4e2	**2.1e2 ± 2.3e1**
40	1.8e3 ± 1.2e2	6.9e2 ± 9.7e1	4.0e3 ± 1.7e2	2.5e3 ± 5.5e2	**6.3e2 ± 1.4e2**
50	2.3e3 ± 1.7e2	1.2e3 ± 3.0e2	6.3e3 ± 3.6e2	3.6e3 ± 6.2e2	**1.1e3 ± 2.3e2**
100	> 60h	4.2e3 ± 3.1e2	> 60h [6]	6.5e3 ± 2.3e2	**3.3e3 ± 2.0e2**
SF4: Run Time					
d	GraN-DAG	DAG-GNN	GS-GES	NOTEARS-MLP	DAG-NCMLP
10	3.4e2 ± 3.5e1	5.0e2 ± 3.4e1	1.3e2 ± 1.2e1	2.3e2 ± 4.8e1	**4.9e1 ± 9.5**
20	7.0e2 ± 1.1e2	5.7e2 ± 4.2e1	5.2e2 ± 7.3e1	2.7e2 ± 9.6e1	**1.9e2 ± 3.3e1**
40	1.8e3 ± 1.7e2	6.9e2 ± 3.3e1	2.9e3 ± 2.4e2	1.4e3 ± 2.7e2	**6.5e2 ± 4.3e1**
50	2.6e3 ± 4.3e2	9.6e2 ± 2.0e2	4.7e3 ± 3.5e2	2.1e3 ± 3.7e2	**8.6e2 ± 4.6e1**
100	> 60h	1.8e3 ± 4.2e2	> 60h	3.9e3 ± 1.4e2	**1.3e3 ± 5.7e2**

mediate or final learned DAGs. The data is generated under a non-linear SEM assumption, rendering linear SEM-based methods ineffective in capturing the complex non-linear relationships present in the data. Consequently, we compare our DAG-NCMLP only with baselines developed under non-linear SEMs.

Table 1 demonstrates that NOTEARS-MLP consistently outperforms other advanced methods across most settings, aligning with previous observations. Our proposed DAG-NCMLP method shows significant accuracy improvements compared to the baselines (GraN-DAG, GS-GES, and DAG-GNN) across all graph

settings. While DAG-NCMLP's accuracy is comparable to NOTEARS-MLP, it surpasses NOTEARS-MLP in accuracy in 6 out of 20 graph settings and falls slightly behind within an acceptable range of differences in the remaining settings. In terms of efficiency, DAG-NCMLP requires significantly less computational time compared to the baselines, particularly NOTEARS-MLP. It typically completes computations in approximately half to 10% of the time required by NOTEARS-MLP.

Despite the universal nonlinear estimation capability of the 3-layer MLP model used to generate the synthetic data in Table 1, we aim to demonstrate the effectiveness of our proposed methods across different nonlinear SEM assumptions. Therefore, we present empirical evaluation results on Gaussian Process data in Table 2. Table 1 showcases DAG-NCMLP outperforming GraN-DAG and DAG-GNN, achieving results comparable to NOTEARS-MLP. However, in contrast to the results in Table 1, GS-GES outperforms NOTEARS-MLP in 8 out of 20 graph settings, achieving the highest accuracy. The differences between the accuracy of NOTEARS-MLP and DAG-NCMLP are minimal, with SHDs of DAG-NCMLP typically within a 2.6% variation of those of NOTEARS-MLP, except in extreme cases. In terms of efficiency, DAG-NCMLP is significantly more computationally efficient, requiring only 15.97% to 70.37% of the time required by NOTEARS-MLP, with greater gains for larger d. This observation in Table 2 aligns with the findings in Table 1, demonstrating that DAG-NCMLP substantially improves efficiency while maintaining comparable accuracy compared to NOTEARS-MLP. Additionally, DAG-NCMLP outperforms other state-of-the-art nonlinear SEM-based methods in terms of accuracy. Comparing the runtime of DAG-NCMLP in both tables, it is faster on GP data in Table 2 than on MLP data in Table 1. This difference is due to the simpler data generation process for GP data, which uses fewer parameters. As a result, DAG-NCMLP finds it easier to model the data distribution of GP data compared to MLP data.

Empirical results in Appendix B indicate that although some popular causal discovery methods have good efficiency, however, they suffer from poor accuracy issues. Our proposed DAG-NCMLP achieves good accuracy as the nonlinear SEM-based baselines while significantly improving the efficiency.

5.2 Empirical Results on Real Data

Table 3 presents the results of applying the DAG-NCMLP and 4 other baseline methods on the real dataset. The table reports the accuracy of the SHD, the number and the ratio of correctly estimated edges, and the computational efficiency in terms of the runtime in seconds. Table 3 shows that NOTEARS-MLP achieves an SHD of 15 in $4.4e2$ seconds, while DAG-NCMLP achieves an SHD of 15 in $1.5e2$ seconds. Two methods correctly estimate the same number of edges. Hence, on the real dataset, DAG-NCMLP can achieve a comparable accuracy with substantially reduced efficiency compared to NOTEARS-MLP. This is consistent with our observation on synthetic datasets.

Table 3. Comparison of different algorithms on Real Data: results on SHD, number of edges, and runtime.

Dataset	SHD	# Correct Edges	Ratio of Correct Edges	Runtime
GraN-DAG	**13**	6/17	0.353	$6.1e2$
DAG-GNN	19	**8/17**	**0.471**	$5.3e2$
GS-GES	17	6/17	0.353	$5.0e2$
NOTEARS-MLP	15	7/17	0.412	$4.4e2$
DAG-NCMLP	15	7/17	0.412	**1.5e2**

6 Conclusion

In this paper, we introduce an efficient DAG learning algorithm that utilizes a projection formulation on nonlinear SEMs, enabling better capture of complex nonlinear relationships between variables. We theoretically derive nonlinear projection formulations for gradient-based adjacency matrix representations. Leveraging these formulations, we propose a novel nonlinear DAG learning algorithm, DAG-NCMLP, designed to efficiently solve the unconstrained optimization problem inherent in the formulation and learn the DAG structure. Our empirical results demonstrate that DAG-NCMLP significantly enhances computational efficiency, particularly in scenarios with a large number of variables. Importantly, DAG-NCMLP achieves comparable accuracy to state-of-the-art nonparametric or nonlinear DAG learning methods. We believe that DAG-NCMLP presents a promising framework for DAG learning.

Acknowledgements This work was supported in part by the National Science Foundation award IIS 2236026 and in part by IBM through the IBM-Rensselaer Future of Computing Research Collaboration. Y. Yu is grateful for support from the AFOSR grant FA9550-22-1-0197.

References

1. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research (2002)
2. Cussens, J.: Bayesian network learning with cutting planes. In: UAI (2011)
3. Eaton, D., Murphy, K.: Bayesian structure learning using dynamic programming and MCMC. arXiv preprint arXiv:1206.5247 (2012)
4. Friedman, N., Koller, D.: Being bayesian about network structure. In: Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence. pp. 201–210. Morgan Kaufmann Publishers Inc. (2000)
5. Friedman, N., Koller, D.: Being Bayesian about network structure. a Bayesian approach to structure discovery in Bayesian networks. Machine learning **50**(1-2), 95–125 (2003)

[6] We allow for $60h$ on 10 graphs. If the average runtime is longer than $2.16e4$ seconds, then we will mark the runtime as $> 60h$ in the table. For example, the average runtime for GS-GES on ER4 graphs is $2.8e4 \pm 2.2e3$.

6. Gao, M., Ding, Y., Aragam, B.: A polynomial-time algorithm for learning nonparametric causal graphs. arXiv preprint arXiv:2006.11970 (2020)
7. Grzegorczyk, M., Husmeier, D.: Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. Machine Learning **71**(2-3), 265 (2008)
8. He, R., Tian, J., Wu, H.: Structure learning in Bayesian networks of a moderate size by efficient sampling. Journal of Machine Learning Research **17**(1), 3483–3536 (2016)
9. Heckerman, D., Geiger, D., Chickering, D.M.: Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning **20**(3), 197–243 (1995)
10. Huang, B., Zhang, K., Lin, Y., Schölkopf, B., Glymour, C.: Generalized score functions for causal discovery. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1551–1560 (2018)
11. Jaakkola, T., Sontag, D., Globerson, A., Meila, M.: Learning Bayesian network structure using LP relaxations (2010)
12. Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., Sebag, M.: Sam: Structural agnostic model, causal discovery and penalized adversarial learning (2018)
13. Lachapelle, S., Brouillard, P., Deleu, T., Lacoste-Julien, S.: Gradient-based neural dag learning. arXiv preprint arXiv:1906.02226 (2019)
14. Madigan, D., York, J., Allard, D.: Bayesian graphical models for discrete data. International Statistical Review/Revue Internationale de Statistique pp. 215–232 (1995)
15. Ng, I., Fang, Z., Zhu, S., Chen, Z., Wang, J.: Masked gradient-based causal structure learning. arXiv preprint arXiv:1910.08527 (2019)
16. Ng, I., Ghassami, A., Zhang, K.: On the role of sparsity and dag constraints for learning linear dags. Advances in Neural Information Processing Systems **33** (2020)
17. Nievergelt, J., Hinrichs, K.H.: Algorithms and Data Structures: With Applications to Graphics and Geometry. Prentice-Hall, Inc., USA (1993)
18. Niinimaki, T., Parviainen, P., Koivisto, M.: Partial order MCMC for structure discovery in Bayesian networks. arXiv preprint arXiv:1202.3753 (2012)
19. Ott, S., Imoto, S., Miyano, S.: Finding optimal models for small gene networks. In: Pacific symposium on biocomputing (2004)
20. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers, Inc., 2 edn. (1988)
21. Pearl, J.: Causality: models, reasoning, and inference. Econometric Theory **19**(46), 675–685 (2003)
22. Peters, J., Janzing, D., Scholkopf, B.: Causal inference on discrete data using additive noise models. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(12), 2436–2450 (2011)
23. Peters, J., Mooij, J.M., Janzing, D., Schölkopf, B.: Causal discovery with continuous additive noise models. The Journal of Machine Learning Research **15**(1), 2009–2053 (2014)
24. Sachs, K., Perez, O., Peer, D., Lauffenburger, D.A., Nolan, G.P.: Causal protein-signaling networks derived from multiparameter single-cell data. Science **308**(5721), 523–529 (2005)
25. Scanagatta, M., de Campos, C.P., Corani, G., Zaffalon, M.: Learning bayesian networks with thousands of variables. In: Advances in neural information processing systems. pp. 1864–1872 (2015)
26. Silander, T., Myllymaki, P.: A simple approach for finding the globally optimal Bayesian network structure. In: UAI (2006)

27. Spirtes, P., Glymour, C.N., Scheines, R., Heckerman, D., Meek, C., Cooper, G., Richardson, T.: Causation, prediction, and search. MIT press (2000)
28. Spirtes, P., Meek, C., Richardson, T.: Causal inference in the presence of latent variables and selection bias. In: UAI (1995)
29. Teyssier, M., Koller, D.: Ordering-based search: A simple and effective algorithm for learning bayesian networks. arXiv preprint arXiv:1207.1429 (2012)
30. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Machine learning **65**(1), 31–78 (2006)
31. Viinikka, J., Hyttinen, A., Pensar, J., Koivisto, M.: Towards scalable bayesian learning of causal dags. arXiv preprint arXiv:2010.00684 (2020)
32. Yu, Y., Chen, J., Gao, T., Yu, M.: Dag-gnn: Dag structure learning with graph neural networks. arXiv preprint arXiv:1904.10098 (2019)
33. Yu, Y., Gao, T.: Dags with no curl: Efficient dag structure learning. Advances in Neural Information Processing Systems (NeurIPS) Workshop on Causal Discovery and Causality-Inspired Machine Learning (2020)
34. Yu, Y., Gao, T., Yin, N., Ji, Q.: Dags with no curl: An efficient dag structure learning approach. In: International Conference on Machine Learning. pp. 12156–12166. Pmlr (2021)
35. Yuan, C., Malone, B.: Learning optimal Bayesian networks: A shortest path perspective **48**, 23–65 (2013)
36. Zhang, J.: On the completeness of orientation rules for causal discovery in the presence of latent confounders and selection bias. Artificial Intelligence **172**(16-17), 1873–1896 (2008)
37. Zheng, X., Aragam, B., Ravikumar, P.K., Xing, E.P.: Dags with no tears: Continuous optimization for structure learning. In: Advances in Neural Information Processing Systems. pp. 9472–9483 (2018)
38. Zheng, X., Dan, C., Aragam, B., Ravikumar, P., Xing, E.P.: Learning sparse nonparametric DAGs. In: International Conference on Artificial Intelligence and Statistics (2020)