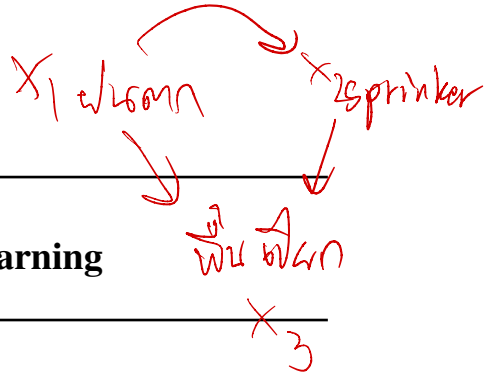


Masked Gradient-Based Causal Structure Learning

Ignavier Ng [*1] **Zhuangyan Fang** [*2] **Shengyu Zhu** [*3] **Zhitang Chen** [3] **Jun Wang** [4]

Abstract

This paper studies the problem of learning causal structures from observational data. We reformulate the Structural Equation Model (SEM) in an augmented form with a binary graph adjacency matrix and show that, if the original SEM is identifiable, then this augmented form can be identified up to super-graphs of the true causal graph under mild conditions. Three methods are further provided to remove spurious edges to recover the true graph. We next utilize the augmented form to develop a masked structure learning method that can be efficiently trained using gradient-based optimization methods, by leveraging a smooth characterization on acyclicity and the Gumbel-Softmax approach to approximate the binary adjacency matrix. It is found that the obtained entries are typically near zero or one, and can be easily thresholded to identify the edges. We conduct experiments on synthetic and real datasets to validate the effectiveness of the proposed method and show that the method can readily include different smooth functions to model causal relationships.

1. Introduction

A causal graphical model defined on a Directed Acyclic Graph (DAG) describes a causal system without latent variables or selection biases, and has found applications in many areas of empirical science including weather forecasting (Abramson et al., 1996), biomedicine and healthcare (Lucas et al., 2004), and biology (Sachs et al., 2005; Pearl, 2009). Although randomized controlled experiments can be used to find the causal structure effectively, they are generally expensive or even ethically impossible in practice. It is thus appealing to learn causal structures from passively observed data, which has been made possible under proper conditions (Spirtes et al., 2000; Pearl, 2009; Peters et al., 2017).

Existing approaches to learning causal structures from observational data roughly fall into two classes: constraint- and score-based methods. Constraint-based methods, such as the PC and fast causal inference (Spirtes et al., 2000), first use conditional independence tests to find causal skeleton and then determine edge directions according to certain orientation rules (Meek, 1995; Zhang, 2008). Under the Markov and faithfulness assumptions, this type of methods can identify causal graph up to the Markov equivalence class that may contain structurally diverse DAGs. Conditional independence testing can make errors with finite data and a small error in the skeleton may result in large errors in the inferred Markov equivalence class. Score-based methods evaluate the quality of candidate causal graphs with a predefined score function and then search for graphs with optimal score. Due to the combinatorial nature of the DAG constraint (Chickering, 1996; He et al., 2015), most score-based methods rely on local heuristics to perform the search. For example, Greedy Equivalence Search (GES) (Chickering, 2002b) attempts to search the space of graph structures by adding, deleting or reversing an edge following a series of rules and avoid cycles when an edge is added or reversed. Though GES finds global minimum with infinite data and proper model conditions, it is usually much less satisfactory in practice. We refer the reader to Glymour et al. (2019) for more details and an introduction to other types of methods.

More recently, Zheng et al. (2018) has proposed NOTEARS, a score-based method that formulates the combinatorial optimization problem as a continuous one using a novel smooth characterization on acyclicity. NOTEARS is specifically designed for linear Structural Equation Models (SEMs). Subsequent works DAG-GNN (Yu et al., 2019) and GraN-DAG (Lachapelle et al., 2020) have extended it to nonlinear cases where Neural Networks (NNs) are used to model causal relationships. With a proper score function, these methods can utilize continuous optimization methods to estimate the

[*]Equal contribution. Work was done when the first two authors were interns at Huawei Noah's Ark Lab. [1]University of Toronto [2]Peking University [3]Huawei Noah's Ark Lab [4]University College London. Correspondence to: Shengyu Zhu <zhushengyu@huawei.com>.

Preprint. Under review.

causal structure through the *weighted* adjacency matrices. However, to guarantee that such a matrix indeed indicates the causal structure, NOTEARS and DAG-GNN assume specific forms of SEMs, while GraN-DAG uses the NN weights to obtain an equivalent weighted adjacency matrix that may require much extra effort to construct with other model functions such as polynomial functions and convolutional NNs. Further, numeric optimization methods usually result in a matrix with a number of entries near zero but not exactly zero. It becomes key to finding a proper threshold to identify edges from the estimated entries, but no principled approach exists yet.

This work is devoted to developing a gradient-based structure learning method towards: 1) an adjacency matrix representing the causal structure exists for any SEM; 2) the estimate from numeric optimization methods can be easily thresholded for interpretation; and 3) the continuous formulation can readily include different functions to model causal relationships. Our contributions are summarized below:

- We reformulate the SEM in an augmented form with a *binary* adjacency matrix and characterize the identifiability that was somewhat omitted in recent gradient-based methods. It is shown that, if the original SEM is identifiable, then the augmented form can be identified up to super-graphs of the true causal graph. We further provide three methods to remove the spurious edges.
- We next develop a masked gradient-based causal structure learning method based on the augmented form. This method leverages the recent Gubmel-Softmax approach to approximate the binary adjacency matrix and the resulting entries are mostly near either zero or one.
- We conduct experiments to validate the effectiveness of our method on both synthetic and real datasets. We also show that the method can easily include smooth model functions for learning causal structures.

2. Background and Related Work

In this section, we briefly review SEMs and several recently developed gradient-based structure learning methods.

2.1. Structural Equation Model and Identifiability

Let $\mathcal{G}$ be a DAG with vertex set $V = \{X_1, X_2, \ldots, X_d\}$ where each node X_i respresents a random variable. For $X_i \in V$, we use $X_{\mathrm{pa}(i)}$ to denote the set of its parental nodes so that there is an edge from $X_j \in X_{\mathrm{pa}(i)}$ to X_i in $\mathcal{G}$. A commonly used model in causal structure learning is the SEM that contains two types of variables: substantive and noise variables (Spirtes, 2010). Each substantive variable is obtained from a function of some (or possibly none) other substantive variables and a unique noise variable. In this work, we focus on the following recursive SEM with additive noises w.r.t. a DAG $\mathcal{G}$:

$$X_i = f_i(X_{\mathrm{pa}(i)}) + \epsilon_i, \quad i = 1, 2, \ldots, d, \tag{1}$$

where ϵ_i's are jointly independent and f_i is a deterministic function with input argument $X_{\mathrm{pa}(i)}$. Such an SEM is also called Additive Noise Model (ANM) in the literature and we will use the two names interchangeably. We assume *causal minimality* for this SEM, which in this case is equivalent to that each f_i is not a constant function in any $X_j \in X_{\mathrm{pa}(i)}$ (Peters et al., 2014; 2017). We also assume that ϵ_i's have a strictly positive density (w.r.t. Lebesgue measure), and that each density function of ϵ_i and each causal relationship f_i are continuous, three times continuously differentiable.

For ease of presentation, we will consider that each variable X_i is scalar-valued; the proposed method can be directly extended to include vector-valued case by using vector-valued functions to model causal relationships, similar to Yu et al. (2019); Ng et al. (2019). Let X be the vector concatenating all the variables X_i and $P(X)$ the marginal distribution induced by the SEM defined on DAG $\mathcal{G}$. Then $P(X)$ is Markovian to $\mathcal{G}$, and $\mathcal{G}$ and $P(X)$ are said to form a causal Bayesian network (Spirtes et al., 2000; Pearl, 2009). The problem of causal structure learning is to use the observational data $\{x^{(k)}\}_{k=1}^n$, with $x^{(k)}$ being the k-th independent sample from $P(X)$, to infer the underlying causal graph.

An important issue of causal structure learning is the identifiability of the true causal graph, defined as follows.

Definition 1 (Identifiability)**.** *Consider an SEM defined on a DAG $\mathcal{G}$ with marginal distribution $P(X)$. Then $\mathcal{G}$ is said to be identifiable if no other SEMs can induce the same distribution $P(X)$ with a different DAG.*

In general, however, it is impossible to recover $\mathcal{G}$ using only the observational data from $P(X)$. To proceed, Peters et al. (2014) have shown that if we consider only a subclass of SEMs, the *restricted ANMs* where f_i's and the density functions of

ϵ_i's and X do not solve a system of three-order differential equations (see Peters et al. (2014, Cond. 19) or Appendix B), then the true DAG is identifiable. Other identifiable SEMs include linear non-Gaussian acyclic model (Shimizu et al., 2006; 2011), linear-Gaussian model with equal noise variances (Peters et al., 2014), and post-nonlinear causal model (Zhang & Hyvärinen, 2009). We will focus on the restricted ANMs on identifiability issues.

2.2. Gradient-Based Structure Learning Methods

NOTEARS (Zheng et al., 2018) is the first method that formulates the structure learning problem as a continuous optimization one. It is specifically developed for linear SEMs with $X_i = W_i^T X + \epsilon_i$, where $W_i \in \mathbb{R}^d$ is the coefficient vector and the indices of non-zero elements in W_i correspond to the parents of X_i. Defining $W = [W_1|W_2|\cdots|W_d]$ as the coefficient matrix, then W can be viewed as a weighted adjacency matrix of the causal DAG. Using least squares loss, NOTEARS solves the following problem to estimate W:

$$\begin{aligned} \min_{W \in \mathbb{R}^{d\times d}} \quad & \frac{1}{2n}\sum_{k=1}^{n}\left\|x^{(k)} - W^T x^{(k)}\right\|_2^2 + \lambda\|W\|_1 \\ \text{subject to} \quad & \operatorname{tr}\left(e^{W\circ W}\right) - d = 0, \end{aligned} \tag{2}$$

where $\circ$ denotes the element-wise product, the ℓ_1 penalty is used to induce sparsity w.r.t. edges, and $e^M = \sum_{k=0}^{\infty}\frac{M^k}{k!}$ is the matrix exponential of a square matrix M with M^k being the k-th power of M. The (j,i)-th entry of M^k is the sum of weight products of k-step paths from X_j to X_i. Eq. (2) is a smooth characterization of acyclicity and holds if and only if W is the weighted adjacency matrix corresponding to a DAG. This problem can be readily solved by the augmented Lagrangian method (Bertsekas, 1999), followed by thresholding on the estimated weights.

DAG-GNN (Yu et al., 2019) extends NOTEARS to handle nonlinear SEMs, based on that a linear SEM can be written as $X = (I - W^T)^{-1}\epsilon$ with $\epsilon = [\epsilon_1, \epsilon_2, \ldots, \epsilon_d]^T$. It considers data generating procedure as $X = f_2((I - W^T)^{-1} f_1(\epsilon))$, where $f_1 : \mathbb{R}^d \to \mathbb{R}^d$ and $f_2 : \mathbb{R}^d \to \mathbb{R}^d$ are possibly nonlinear functions and each variable admits a common model function, i.e., for $f_i, i = 1, 2$, there exists $f_i' : \mathbb{R} \to \mathbb{R}$ so that $f_i(X) = [f_i'(X_1), f_i'(X_2), \ldots, f_i'(X_d)]^T$. DAG-GNN then estimates W under the framework of variational autoencoders (Kingma & Welling, 2013), where the encoder and decoder are both graph neural networks. Along this direction, Ng et al. (2019) assumes the data generation as $X = f_2(W^T f_1(X)) + \epsilon$ and uses a graph autoencoder to find causal structures. These two methods are shown to perform better than NOTEARS on some nonlinear data models. However, the identifiability issue was not discussed and due to nonlinear transformations on W, the estimated weights are indeterminate and may lack causal interpretability.

Rather than assuming particular forms of SEMs, GraN-DAG models the conditional distribution of each variable given its parents with feed-forward NNs (Lachapelle et al., 2020). GraN-DAG defines the NN-path as a way of measuring how a variable X_j affects another variable $X_i, j \neq i$, and the sum of all NN-paths between X_j and X_i being zero implies that there is no edge from X_j to X_i. An equivalent weighted adjacency matrix is then constructed with the each entry being the sum of corresponding NN-paths. This approach is specifically designed for feed-forward NNs and may require extra effort to find an equivalent adjacency matrix for other model functions like polynomial regression models or convolutional NNs. Our approach, as shown in Sec. 5.1.2, is more generic as it can readily include different model functions.

In addition, Zhu et al. (2020) propose to use policy gradient to search for the DAG with optimal score. This approach does not require the score function be smooth, but dealing with large graphs is challenging. Other works using gradient-based methods include Goudet et al. (2018) and Kalainathan et al. (2018): the former proposes causal generative NNs for functional causal modeling assuming an initial skeleton of the causal graph, and the latter finds causal generative models in an adversarial way but does not ensure acyclicity.[1]

3. Augmented SEM and Identifiability

Both NOTEARS and DAG-GNN rely on a notion of weighted adjacency matrix, which however is not obvious for certain SEMs, e.g., when the functions f_i's are quadratic or sampled from Gaussian processes. Noticing that every directed graph

[1]The statement here was based on the first arXiv version of the paper Kalainathan et al. (2018). We just find that the authors have updated their paper with several major changes that were not in the previous version. We were unaware of these new changes when preparing this version. While there are certain similarities between our work and the latest version of Kalainathan et al. (2018), we believe that there are more significant differences and we will include these similarities and differences in the main content of our next version. Please see Appendix A for details.

corresponds to a *binary* adjacency matrix and vice versa, we therefore consider to estimate the binary adjacency matrix to learn the underlying causal structure.

Let $A = [A_1|A_2|\cdots|A_d]$ be the binary adjacency matrix associated with the true causal DAG $\mathcal{G}$, where $A_i \in \{0,1\}^d$ can be viewed as an indicator vector so that A_{ji}, the j-th entry of A_i, equals 1 if and only if X_j is a parent of X_i. To incorporate Eq. (1) with A, we define a new functions h_i as:

$$h_i : \mathbb{R}^d \longrightarrow \mathbb{R}, \quad x \longmapsto f_i(x_{\mathrm{pa}(i)}),$$

which maps an input $x \in \mathbb{R}^d$ to the value $f_i(x_{\mathrm{pa}(i)}) \in \mathbb{R}$. Note that the function value $h_i(x)$ is determined only by $x_{\mathrm{pa}(i)}$. We can then rewrite Eq. (1) in a form augmented by the binary adjacency matrix A:

$$X_i = h_i(A_i \circ X) + \epsilon_i, \quad i = 1, 2, \ldots, d, \tag{3}$$

where $\circ$ denotes the element-wise product. We refer to the model in the form of Eq. (3) as Augmented SEM (ASEM).

Based on Eq. (3), it appears that we could use a parametric or nonparametric model, together with a $d \times d$ binary matrix, to fit the observed data under the acyclicity constraint. Once we obtain the fitted model $\hat{h}_i$ and $\hat{A}_i$, we may output the graph indicated by $[\hat{A}_1|\hat{A}_2|\cdots|\hat{A}_d]$ as our estimate of the causal structure. Unfortunately, even if we can fit the data perfectly, it is not clear whether the estimated binary matrix indeed indicates the true causal graph: there may exist h_i, A_i and h_i', A_i' that induce the same distribution $P(X)$ but A_i and A_i' disagree for some i. In this section, we characterize the identifiability issue of the ASEM. A practical gradient-based algorithm for estimating the ASEM will be given in the next section.

3.1. Structure Identifiability

We describe formally the identifiability issue of the ASEM. Assume a marginal distribution $P(X)$ induced by an SEM in Eq. (1) with DAG $\mathcal{G}$. For an ASEM, let A be the associated binary adjacency matrix indicating a DAG $\mathcal{H}$ (i.e., $A_{ji} = 1$ implies an edge $X_j \to X_i$ in $\mathcal{H}$ and vice versa), $h_i : \mathbb{R}^d \to \mathbb{R}$ be some deterministic functions with input X, and $\tilde{\epsilon}_i$'s denote the jointly independent additive noises. If this ASEM induces the same distribution $P(X)$, then we would like to ask: *1) is the DAG $\mathcal{H}$ identifiable from $P(X)$? and 2) what is the relationship of $\mathcal{H}$ to $\mathcal{G}$ that we want to recover?*

Our answer is negative to the first question. A reason is that the function h_i can be degenerate w.r.t. some of its input arguments. That is, if h_i is a constant function w.r.t some $X_j, j \neq i$, then whether $A_{ji} = 0$ or 1 does not affect $P(X)$ but will affect the structure of $\mathcal{H}$. We may then restrict h_i to be non-degenerate w.r.t. all $X_j, j \neq i$, but this restriction is practically hard to place. Let $X_{\widetilde{\mathrm{pa}}(i)}$ be the set of variables X_j w.r.t. which $h_i(A_i \circ X)$ is non-degenerate, i.e., when $A_{ji} = 1$ and $h_i(X)$ is not a constant function in X_j. If $A_{ji} = 0$, we say that X_j is *masked* to X_i in the sense that the values of X_i do not depend on X_j. We next define $\tilde{f}_i$ as

$$\tilde{f}_i : \mathbb{R}^{|X_{\widetilde{\mathrm{pa}}(i)}|} \longrightarrow \mathbb{R}, \quad \tilde{x} \longmapsto h_i(A_i \circ x), \tag{4}$$

which maps $\tilde{x} \in \mathbb{R}^{|X_{\widetilde{\mathrm{pa}}(i)}|}$ to the value $h_i(A_i \circ x)$ with x being such that $x_{\widetilde{\mathrm{pa}}(i)} = \tilde{x}$. Such a mapping is possible because the function value $h_i(A_i \circ x)$ depends only on $x_{\widetilde{\mathrm{pa}}(i)}$. It can be verified that the functions $\tilde{f}_i$'s and the noises $\tilde{\epsilon}_i$'s in the ASEM form an SEM satisfying the causal minimality condition, and that the distribution of X induced by this SEM is identical to that induced by the ASEM, i.e., $P(X)$. Denote by $\widetilde{\mathcal{G}}$ the causal DAG of the reduced SEM with functions $\tilde{f}_i$'s and noise variables $\tilde{\epsilon}_i$'s. We have the following lemma, with a proof given in Appendix C.1.

Lemma 1. *$\mathcal{H}$ is a super-graph of $\widetilde{\mathcal{G}}$. i.e., all the edges in $\widetilde{\mathcal{G}}$ also exist in $\mathcal{H}$.*

We now proceed to the second question. Recall that the observed data are generated from a distribution induced by the SEM in Eq. (1). We may further assume a restricted ANM (see Peters et al. (2014, Cond. 19) or Appendix B) for the data generation procedure so that the causal graph $\mathcal{G}$ is identifiable. Notice that the reduced SEM with functions $\tilde{f}_i$'s and causal DAG $\widetilde{\mathcal{G}}$ is also an ANM in the form of Eq. (1) and induces the same distribution $P(X)$. It seems that we could conclude that $\mathcal{G} = \widetilde{\mathcal{G}}$ as the original SEM is identifiable. There is still a gap though: the reduced SEM may not fall into the class of restricted ANMs. Nevertheless, we can narrow down the class of functions h_i in the ASEM.

Definition 2. *An ASEM is called restricted if the reduced SEM according to Eq. (4) is a restricted ANM.*

Appendix B provides an explicit condition for a restricted ASEM. We have the following result, to the second question.

Theorem 1. *Assume a restricted ANM with graph $\mathcal{G}$ and distribution $P(X)$. If a restricted ASEM with causal graph $\mathcal{H}$ induces the same $P(X)$, then $\mathcal{H}$ is a super-graph of $\mathcal{G}$.*

Proof. By Def. 2, the reduced SEM defined in Eq. (4) is a restricted ANM with causal graph $\widetilde{\mathcal{G}}$ and has the same distribution $P(X)$. With the identifiability result of restricted ANMs (Peters et al., 2014, Thm. 28), we know that $\widetilde{\mathcal{G}}$ is identical to $\mathcal{G}$. Applying Lem. 1 completes the proof. □

Thm. 1 indicates that it is possible to apply a restricted ASEM to fitting the data and obtain a super-graph of the true causal graph $\mathcal{G}$ in the original SEM. In practice, we can follow Hoyer et al. (2009); Peters et al. (2014) to choose three times continuously differentiable functions for h_i's.

3.2. Removing Spurious Edges

Given Thm. 1, we next remove spurious edges to recover the causal graph $\mathcal{G}$. We provide three approaches below.

Computing absolute Jacobian matrix We may directly derive the reduced SEM as its directed graph $\widetilde{\mathcal{G}}$ is the same as $\mathcal{G}$. Since h_i's are three times continuously differentiable, we can find $X_{\widetilde{\mathrm{pa}}(i)}$ according to the derivative $A_{ji}\nabla_j h_i(A_i \circ X)$, where $\nabla_j h_i(A_i \circ X)$ means the gradient of h_i w.r.t. X_j evaluated at $A_i \circ X$. Similar to Lachapelle et al. (2020), we compute the expected absolute Jacobian matrix

$$\mathcal{J}_{ji} = \mathbb{E}_{X\sim P(X)}\left[|A_{ji}\nabla_j h_i(A_i \circ X)|\right],$$

where $\mathcal{J}_{ji} = 0$ implies that X_j is not a parent of X_i, as a result of a strict positive density of $P(X)$. In practice, $\mathcal{J}$ can be approximated with sample means.

Adding a regularization term Thm. 1 shows that the graph associated with a restricted ASEM inducing the same marginal distribution must be a super-graph of $\mathcal{G}$. If we pick the one with minimum edges among all such graphs, then $\mathcal{G}$ is recovered. In practice, we can add a sparsity regularization term to the score function.

Using conditional independence testing We may also use conditional independence tests, e.g., the nonparametric kernel based tests from Gretton et al. (2005); Zhang et al. (2012), to remove spurious edges. Denote by $X_{\mathrm{pa}(i,\mathcal{H})}$ and $X_{\mathrm{pa}(i,\mathcal{G})}$ the parental nodes of X_i in the graphs $\mathcal{H}$ and $\mathcal{G}$, respectively. For any X_i with non-empty $X_{\mathrm{pa}(i,\mathcal{H})}$, we search for $T \subseteq X_{\mathrm{pa}(i,\mathcal{H})}$ satisfying: 1) X_i is independent of any $X_j \in X_{\mathrm{pa}(i,\mathcal{H})} \setminus T$ conditional on T; 2) T is minimal, in the sense that any strict subset of T does not satisfy the first condition. The following proposition guarantees the correctness of finding such a subset, with a proof given in Appendix C.2.

Proposition 1. *$X_{\mathrm{pa}(i,\mathcal{G})}$ is the unique subset of $X_{\mathrm{pa}(i,\mathcal{H})}$ that satisfies the above two conditions.*

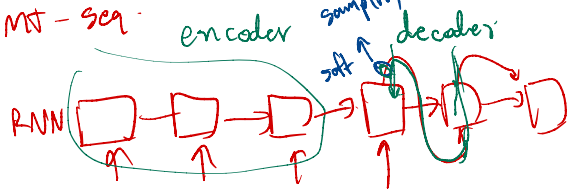
Notice that the faithfulness assumption is not needed here. In practice, heuristic methods such as Incremental Association Markov Blanket (IAMB) (Tsamardinos et al., 2003) can be used to search for the desired subset.

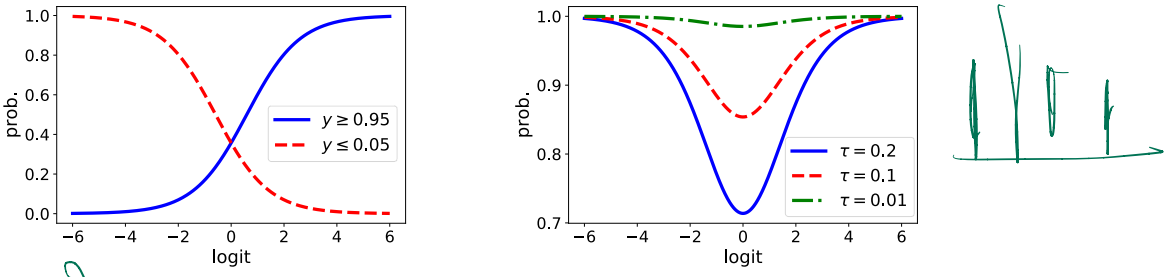
4. Masked Gradient-Based Causal Structure Learning with Gumbel-Sigmoid

The identifiability result in Sec. 3 guarantees the correctness of learning first an ASEM with binary adjacency matrix and then removing spurious edges under mild conditions. However, the discrete nature of binary valued entries prohibits direct use of first-order methods for efficient learning. To proceed, we relax each entry to take real values from $(0, 1)$. A naive approach is to apply sigmoid functions parameterized by real valued variables, but would result in estimated entries of the adjacency matrix lying in a very small range near 0, making it hard to identify what edges are indeed positives (see an example in Sec. 5.4). We would like each estimated entry to be either near 0 so that it almost *masks* a particular variable for causal interpretation, or close to 1 so that we can easily use thresholding to identify the edges.

4.1. Gumbel-Sigmoid to Approximate Binary Entries

We leverage the recent Gumbel-Softmax approach that relies on a continuous distribution over the simplex to approximate samples from a categorical distribution (Jang et al., 2017; Maddison et al., 2016). The approach is effective in learning discrete random variables, compared with the straight-through (Bengio et al., 2013) and several REINFORCE based methods (Mnih & Gregor, 2014; Gu et al., 2016). We use its two-dimensional version as we consider the binary graph adjacency matrix.





(a) $\tau = 0.2$ (b) $\Pr(y \leq 0.05 \text{ or } y \geq 0.95)$

Figure 1: Gumbel-Sigmoid with different logits.

For a random variable defined on $\{0, 1\}$ with class probabilities $\pi_0 \in (0, 1)$ and $\pi_1 = 1 - \pi_0$, its *binary* sample can be approximated by the differentiable softmax function:

$$y = \frac{1}{1 + \exp(-(\log(\pi_1/\pi_0) + (g_1 - g_0))/\tau)} = \sigma((u + g)/\tau), \tag{5}$$

where $\tau > 0$ is the temperature, $g_i, i = 0, 1$ are independent samples from $\text{Gumbel}(0, 1)$, $\sigma(\cdot)$ is the logistic sigmoid function, and we define $g = g_1 - g_0$ and $u = \log(\pi_1/\pi_0) \in \mathbb{R}$. A sample g_i can be obtained from $g_i = -\log(-\log(a_i))$ with $a_i \sim \text{Uniform}(0, 1)$. We call Eq. (5) Gumbel-Sigmoid with logit u and temperature τ, and write it as $\mathrm{g}_\tau(u)$.

We further characterize the distribution of g and show in Appendix D that $g \sim \text{Logistic}(0, 1)$. The following result can then be obtained using the cumulative distribution function of $\text{Logistic}(0, 1)$: for given u and τ,

$$\Pr(y \leq \delta) = \frac{1}{1 + (1/\delta - 1)^\tau e^u}, \quad \delta \in (0, 1).$$

Fig. 1a plots the probabilities $\Pr(y \leq \delta)$ and $\Pr(y \geq 1 - \delta)$ for $\tau = 0.2$ and $\delta = 0.05$. The sum $\Pr(y \leq \delta) + \Pr(y \geq 1 - \delta)$ characterizes the probability of a Gumbel-Sigmoid output lying within a neighborhood of the boundary of $(0, 1)$; see Fig. 1b for an illustration with $\delta = 0.05$. It can also be verified that $\Pr(y \leq \delta) + \Pr(y \geq 1 - \delta) \to 1$ as $\tau \to 0^+$ for any fixed u and $\delta \in (0, 0.5)$. In other words, a Sigmoid-Gumbel output can be arbitrarily close to 0 or 1 with high probability for a sufficiently small temperature.

We thus use Gubmel-Sigmoid to approximate binary entries in the adjacency matrix. Consider that an edge helps produce a better reconstruction and does not violate the acyclicity constraint. If we repeatedly apply the Gumbel-Sigmoid and estimate the score function, then a gradient-based algorithm, assuming that it is applicable for now, will push the logit so that the output is close to 1 in the expected sense. If the acyclicity constraint is violated, then some entries would be pushed towards 0 to meet the acyclicity constraint.

4.2. Acyclicity Constraint and Optimization Problem

We consider a real matrix $U \in \mathbb{R}^{d \times d}$ and use $\mathrm{g}_\tau(U) \in (0, 1)^{d \times d}$ to denote the output of applying Gumbel-Sigmoid defined in Eq. (5) to U_{ij}'s independently. To enforce acyclicity, we always set the (i, i)-th entry of $\mathrm{g}_\tau(U)$ to be 0 to avoid self-loops, and then consider the following acyclicity constraint from Eq. (2) (Zheng et al., 2018):

$$\mathbb{E}\left[\operatorname{tr}\left(e^{\mathrm{g}_\tau(U)}\right) - d\right] = 0,$$

where the expectation is taken w.r.t. $\text{Logistic}(0, 1)$ samples and $\operatorname{tr}\left(e^{\mathrm{g}_\tau(U)}\right)$ is differentiable w.r.t. $U_{ij}, i \neq j$. Notice that, since $\mathrm{g}_\tau(U_{ij})$ cannot be exactly 0, the l.h.s. cannot achieve exactly 0, either. Nevertheless, similar to Zheng et al. (2018); Lachapelle et al. (2020), it is sufficient to make $\mathbb{E}\left[\operatorname{tr}\left(e^{\mathrm{g}_\tau(U)}\right) - d\right] < \xi$ for some small tolerance ξ (e.g., 10^{-10}), followed by hard thresholding on the estimates.

We now present our score-based structure learning method based on the ASEM. Let $h_i : \mathbb{R}^d \to \mathbb{R}$ be a function used to model causal relationships, i.e., we use $h_i(\mathrm{g}_\tau(U_i) \circ X; \phi_i)$ to estimate the variable X_i where U_i is the i-th column of U

and ϕ_i is the function parameter. Notice that we have masked the i-th element of $\mathsf{g}_\tau(U_i)$ to be 0, so h_i is always a constant function w.r.t. X_i. Let $\phi = \{\phi_i\}_{i=1}^d$ and $h(\mathsf{g}_\tau(U), X; \phi) = \{h_i(\mathsf{g}_\tau(U_i) \circ X); \phi_i\}_{i=1}^d$. With a score function $\mathcal{L}(\cdot, \cdot)$ defined w.r.t. the observed and reconstructed samples, we have the following optimization problem:

$$\begin{aligned} \min_{U,\,\phi} \quad & \mathbb{E}\left[\frac{1}{2n}\sum_{k=1}^{n}\mathcal{L}\Big(x^{(k)},\, h\left(\mathsf{g}_\tau(U), x^{(k)}; \phi\right)\Big)\right] \\ \text{subject to} \quad & \mathbb{E}\left[\operatorname{tr}\left(e^{\mathsf{g}_\tau(U)}\right) - d\right] \le \xi, \end{aligned} \tag{6}$$

where expectations are taken w.r.t. $\mathrm{Logistic}(0, 1)$ samples. A sparsity-inducing term $\|\mathsf{g}_\tau(U)\|_1$ may be also added.

The above problem can be solved using existing continuous optimization methods when $\mathcal{L}(\cdot, \cdot)$ and h_i's are chosen properly. For example, we may use the least squares loss and the negative log-likelihood as score functions, which correspond respectively to the Bayesian Information Criterion (BIC) scores (excluding the model complexity penalty) (Chickering, 2002a) with equal and different variances under additive Gaussian noise assumption. The choices of h_i include polynomial functions and feed-forward NNs. In this work, we follow Zheng et al. (2018) to apply the augmented Lagrangian method, where at each step we approximately solve the subproblem utilizing first-order stochastic methods. Since the closed form of the expectation in Eq. (6) is not available, we use a sample estimate of the expectation and also $\operatorname{tr}(e^{\sigma(U/\tau)}) - d$ for our stopping criterion, i.e., the optimization stops once both of them are lower than ξ. Details regarding the optimization procedure and the stopping criterion are provided in Appendix E.

To ensure the entries close to either 0 or 1, we may use an annealing strategy for the temperature τ in Gumbel-Sigmoid. In practice, we find that a small fixed τ works well. Our stopping criterion is satisfied only when the entries $\sigma(U_{ji}/\tau)$, where i, j are such that the edge $X_j \to X_i \notin E$ for some DAG with E as edge set, are nearly zeros. On the other hand, if an edge indeed helps minimize the score function, then the logit will be pushed to a relatively large positive value so that the score function is minimized in the expected sense. For other edges that do not violate acyclicity nor help minimize the score function, the corresponding entries may have intermediate values. Nevertheless, these edges are treated as spurious edges and will be further processed. Thus, as a byproduct, we can readily use $\sigma(U/\tau)$ as our learned matrix which would indicate a DAG after hard thresholding at 0.5. Empirically, we find that all the entries are near 0 or 1 in most of our experiments where we set $\tau = 0.2$.

While we intend to develop the masked method for nonlinear SEMs, our method readily includes linear SEMs, with the masking being placed on the weights of linear models. Indeed, since the weights directly decide whether an edge exists, we can remove the masking and the rest is identical to NOTEARS with a slightly different optimization procedure.

4.3. Final Output

As discussed in Sec. 3.1, a learned matrix $\sigma(U/\tau)$, even after thresholding, is likely to contain spurious edges or false discoveries. The ℓ_1 penalty can be used to control false discoveries, yet picking a proper penalty weight is not easy. Thus, we stick to a small penalty weight (e.g., 10^{-3}) to slightly control false discoveries during training, followed by thresholding and checking the absolute Jacobian matrix (cf. Sec. 3.2). We do not use conditional independence testing as it relies heavily on data size and number of variables.

In practice, spurious edges may still exist after the above procedure due to finite samples and further pruning usually helps (as demonstrated in Appendix G.4 with an ablation study of its effect on our method). A useful approach is a Causal Additive Model (CAM) based method that applies significance testing of covariates using generalized additive models and declares significance if the p-values do not exceed a predefined value (Bühlmann et al., 2014). Though the true causal relationships may not follow causal additive assumption, the CAM pruning usually performs well.

5. Experiments

This section compares the proposed method against several traditional and recent structure learning methods, including PC (with Fisher-z test and p-value 0.01) (Spirtes & Glymour, 1991), GES (with BIC score) (Chickering, 2002b), CAM (Bühlmann et al., 2014), NOTEARS (Zheng et al., 2018), DAG-GNN (Yu et al., 2019), and GraN-DAG (Lachapelle et al., 2020). Their implementations are described in Appendix F. The proposed method uses 4-layer feed-forward NNs as the model functions and is denoted as MaskedNN. Detailed setting of NN model and hyperparameters is described in Appendix G.1. We use the least squares loss as our score function in all the experiments.

Table 1: Empirical results on nonlinear SEMs with Gaussian process.

	ER1 with 10 nodes		ER4 with 10 nodes		ER1 with 50 nodes		ER4 with 50 nodes	
	SHD	TPR	SHD	TPR	SHD	TPR	SHD	TPR
MaskedNN	**1.2 ± 1.6**	**0.87 ± 0.21**	**9.2 ± 3.3**	**0.78 ± 0.09**	**7.2 ± 4.0**	**0.90 ± 0.06**	**62.2 ± 16.4**	**0.73 ± 0.05**
GraN-DAG	**2.4 ± 2.2**	**0.86 ± 0.15**	**14.4 ± 4.8**	**0.66 ± 0.12**	**6.6 ± 3.5**	**0.92 ± 0.05**	**59.4 ± 12.9**	**0.75 ± 0.05**
CAM	5.0 ± 2.3	**0.92 ± 0.08**	**16.6 ± 3.4**	**0.63 ± 0.08**	**3.8 ± 1.9**	**0.96 ± 0.02**	**58.6 ± 6.6**	**0.76 ± 0.02**
DAG-GNN	6.6 ± 3.6	0.50 ± 0.24	37.0 ± 2.1	0.12 ± 0.03	32.2 ± 7.8	0.45 ± 0.10	186.2 ± 14.5	0.10 ± 0.04
NOTEARS	5.0 ± 2.9	0.62 ± 0.19	35.2 ± 2.5	0.15 ± 0.05	22.8 ± 7.0	0.67 ± 0.10	174.8 ± 13.5	0.16 ± 0.03
GES	**3.4 ± 1.7**	0.78 ± 0.13	29.4 ± 1.0	0.30 ± 0.30	19.0 ± 6.6	0.74 ± 0.09	147.4 ± 16.0	0.31 ± 0.05
PC	4.8 ± 1.9	0.69 ± 0.09	18.8 ± 3.54	0.55 ± 0.10	28.0 ± 6.5	0.67 ± 0.07	123.2 ± 11.4	0.43 ± 0.03

We report True Positive Rate (TPR) and Structural Hamming Distance (SHD) to evaluate the discrepancies between learned graphs and true graphs, averaged over five seeds. The SHD is the smallest number of edge additions, deletions, and reversals to convert the estimated graph into the true DAG. It takes into account both false positives and false negatives and a lower SHD indicates a better estimate.

5.1. Synthetic Data

We conduct experiments on different data models varying along graph size, level of edge sparsity, and causal relationships. We draw a random DAG $\mathcal{G}$ using the Erdös–Rényi (ER) graph model and then generate data in the causal order indicated by $\mathcal{G}$. We consider d-node ER graphs with on average d and $4d$ edges, denoted as ER1 and ER4, respectively. The causal relationships include functions sampled from Gaussian processes (cf. Sec. 5.1.1), quadratic functions (cf. Sec. 5.1.2), and causal additive functions (cf. Appendix G.3). We set unit variances for the noise variables and generate $n = 3,000$ samples in each experiment.

5.1.1. Gaussian Process

We first consider the data model previously used by Peters et al. (2014); Lachapelle et al. (2020): each function f_i is sampled from a Gaussian Process (GP) with RBF kernel of bandwidth one and the noises are zero-mean Gaussians. This setting is identifiable according to Peters et al. (2014).

The empirical results are reported in Table 1 with graph sizes $d \in \{10, 50\}$.[2] We observe that MaskedNN, GraN-DAG, and CAM outperform the other methods across almost all settings in terms of SHD and TPR. MaskedNN is observed to have a better performance with small graphs while CAM performs the best for 50-node graphs. Nevertheless, their differences are minor, compared with the performance of other methods. The performances of GES and PC degrade as the number of edges increases. Both DAG-GNN and NOTEARS perform poorly on this dataset, possibly because they may not be able to model this type of causal relationships, and moreover, they operate on the notion of weighted adjacency matrix which is not obvious here.

5.1.2. Quadratic Functions

We consider nonlinear causal relationships with quadratic functions, as in Zhu et al. (2020). This data model is also identifiable (Peters et al., 2014). With prior knowledge that the causal relationship follows a quadratic function, Zhu et al. (2020) used a similar idea from GraN-DAG to modify NOTEARS by constructing an equivalent weighted adjacency matrix. We refer to this method as NOTEARS-quad in this paper. A Quadratic Regression based pruning method (QR pruning) has also been shown to perform well for this data model, so we will use it for all the methods in this experiment. Details for experiment setup, NOTEARS-quad and QR-pruning can be found in Appendix G.2.

We may also utilize the prior knowledge to apply quadratic functions to modeling causal relationships and the resulting method is denoted as MaskedQR. Note that, for NOTEARS-quad, there is extra effort to derive explicitly the equivalent adjacency matrix and further to implement the method based on this matrix. By contrast, our method only requires replacing the feed-forward NNs with quadratic functions.

For better visualization, we report the average SHDs in Fig. 2 with detailed results including standard deviations and TPRs

[2]For GES and PC, we treat undireted edges as true positives if the true graph has a directed edge in place of the undirected ones.

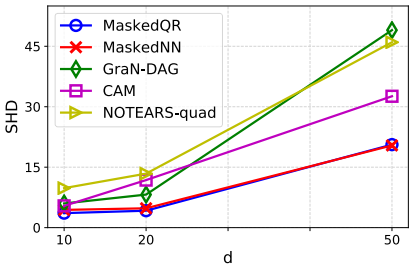


Figure 2: Quadratic functions.

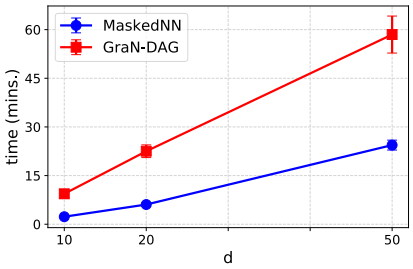


Figure 3: Training time.

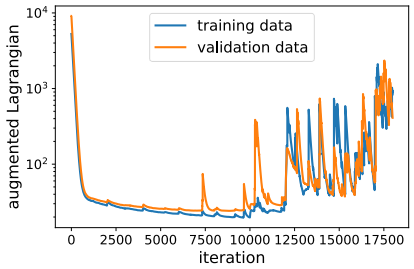


Figure 4: Training trajectories.

given in Appendix G.2. MaskedNN and MaskedQR have a similar performance and outperform the other methods. This also indicates that MaskedNN has a strong expressive power, as its performance is on par with MaskedQR that utilizes the knowledge of the form of causal relationships. CAM performs relatively well despite that the causal additive assumption does not hold. Although NOTEARS-quad is specifically designed for quadratic functions, it has a similar performance to GraN-DAG and is outperformed by CAM. We believe that it is because the equivalent adjacency matrix makes the optimization problem more complicated and harder to solve, as also observed in Zhu et al. (2020).

5.2. Computational Complexity and Training Time

Similar to NOTEARS and GraN-DAG, MaskedNN requires evaluations of the matrix exponential with $\mathcal{O}(d^3)$ cost per iteration. To reduce the number of $\mathcal{O}(d^3)$ iterations required to converge, NOTEARS adopts the proximal quasi-Newton algorithm. Although GraN-DAG uses a gradient-based method like ours, it has been observed that GraN-DAG performs fewer iterations than NOTEARS in practice and that the evaluation of matrix exponential does not dominate the toal cost at each iteration. To validate whether MaskedNN behaves similarly, we compare the training time of MaskedNN with GraN-DAG using the GP datasets from the first experiment. The experiments are run on a standard NC6 instance on Azure cloud, with 6-core Intel Xeon 2.6GHz CPU and one-half Nvidia Tesla K80 GPU. The pruning time is not included for both methods.

Fig. 3 reports the training time, with the plotted value being the average on both ER1 and ER4 graphs. The training time of MaskedNN and GraN-DAG seems to scale linearly when increasing the graph size up to 50. MaskedNN takes a much shorter time than GraN-DAG across all graph sizes.[3] Though the implementations of GraN-DAG and MaskedNN have many unnecessary I/O operations, this experiment shows that the $\mathcal{O}(d^3)$ costs are not a problem to MaskedNN, either.

5.3. Overfitting

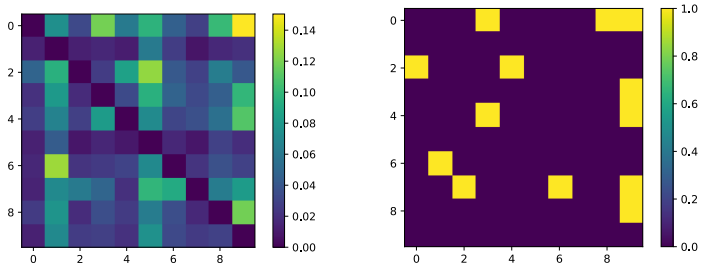
Unlike DAG-GNN and GraN-DAG that require a held-out dataset to avoid overfitting, we use the whole dataset during training. To investigate whether MaskedNN incurs overfitting, we generate a validation dataset of $3,000$ samples and monitor the augmented Lagrangian on both training and validation datasets. Note that the validation dataset is not used for learning causal structures. A typical example on a GP dataset with 50-node ER1 graph is plotted in Fig. 4.

We observe two phases: 1) the augmented Lagrangian is optimized to decrease on both datasets in the first $10,000$ iterations; and 2) with increased Lagrange multiplier and penalty parameter, the acyclicity term has a larger effect on the augmented Lagrangian, making the acyclicity term decrease towards the predefined tolerance and the least squares losses tend to increase. The augmented Lagrangian also oscillates in the second phase, due to the varying logits U_{ij}'s and the increasing Lagrange multiplier and penalty parameter. The similar behaviors in both phases indicate that MaskedNN can use all the observed data for training and does not need a held-out dataset for validation.

5.4. Sigmoid vs. Gumbel-Sigmoid

We provide an example to show the effectiveness of applying Gumbel-Sigmoid to approximating the binary adjacency matrix. We consider a dataset with 10-node ER1 graph from the first experiment in Sec. 5.1.1. Notice that this dataset is only for illustration purpose and similar results are found with other datasets as well. We apply logistic sigmoid and

[3]The training time here is longer than that reported in GraN-DAG paper (Lachapelle et al., 2020). We cannot conduct further verification as authors did not release the experiment environment.



(a) Sigmoid. (b) Gumbel-Sigmoid.

Figure 5: Visualization of the estimated matrices.

Gumbel-Sigmoid with MaskedNN and visualize the resulting adjacency matrices in Fig. 5.

Fig. 5a shows that the estimated entries from logistic sigmoid functions lie in a small range near 0. This is because the acyclicity term, the l.h.s. in Eq. (2), can be made very small if all the entries are close to 0. Consequently, it becomes hard to choose a proper threshold to identify the edges from the estimate. Gumbel-Sigmoid alleviates this problem by enforcing the estimated entries to be close to either 0 or 1, as shown in Fig. 5b. We can then identify the positive edges easily by a threshold at 0.5. For this example, we indeed recover the true causal graph after pruning.

5.5. Real Data

We consider the Sachs dataset to discover a protein signaling network based on expression levels of proteins and phospholipid (Sachs et al., 2005). This dataset is widely used for research on graphical models with experimental annotations well accepted by the biological research community. We use only the observational data with $n = 853$ samples. The true causal graph proposed by Sachs et al. (2005) has 11 nodes and 17 edges.

On this dataset, MaskedNN and CAM achieve the best SHD 12, and GraN-DAG has an SHD 13. DAG-GNN and NOTEARS are on par with MaskedNN and CAM in terms of true positives, but have more false discoveries. Particularly, DAG-GNN and NOTEARS result in SHDs 16 and 19, respectively, whereas an empty graph has an SHD 17. We also apply GES and PC: GES obtains 7 undirected edges while PC estimates 7 undirected and 1 directed edges. By contrast, the inferred graphs from MaskedNN, CAM, and GraN-DAG consist of only directed edges.

6. Concluding Remarks

In this work, we reformulate the SEM in the form of ASEM to incorporate the binary graph adjacency matrix and further investigate the identifiability issue of the ASEM. A practical structure learning method is then proposed, which can be efficiently trained using gradient-based methods by leveraging the recently developed Gumbel-Softmax approach. Extensive experiments validate its effectiveness as well as the flexibility of including different smooth model functions. A future direction is to use variable selection and/or Bayesian approach to improve both efficiency and efficacy.

Appendix

A. Relation to the latest version of Kalainathan et al. (2018)

In Sec. 2.2, we discussed the following work by Kalainathan et al. (2018):

- Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., and Sebag, M. Structural agnostic modeling: Adversarial learning of causal graphs. *arXiv preprint arXiv:1803.04929*, 2018.

We stated that this work *'finds causal generative models in an adversarial way but does not ensure acyclicity'*, which was based on the first arXiv version of Kalainathan et al. (2018). Right after the submission of the main content of our work, we found that the authors have updated their paper with major changes that were not included in the previous version:

- they also use the Gumbel-Softmax approach for estimating binary valued variables.
- they use the same acyclicity constraint from Zheng et al. (2018). The previous version could not guarantee acyclicity.
- they provide more details and discussions regarding the MDL (or BIC) score used in their paper.
- they further characterize the identifiability of Markov equivalence class with this score function.

We were unaware of these new changes when preparing our submission. While there are certain similarities between our work and the latest version of Kalainathan et al. (2018) (namely, masking, Gubmel-Softmax approach, and the acyclicity constraint from Zheng et al. (2018)), we believe that there are more significant differences:

- The motivation and theory are different. Kalainathan et al. (2018) considered the functional causal models and discussed the identification of the *Markov equivalence class* based on the property of their score function. Identification within the Markov equivalence class was also established, with the faithful assumption and a working hypothesis involving Kolmogorov complexity. We aim to learn the underlying causal structure, by formulating the SEM in the ASEM form and characterizing the identifibility of *the true causal graph* based on restricted ANMs (Peters et al., 2014), without the need of the faithful assumption and the working hypothesis.
- The focus and methodology are also different. They used *one* hidden layer NNs, with both structure gates and functional gates, for modeling causal relationships. Together with this structure, they spent a large content on the MDL score and derived the optimization objective which was trained in an adversarial way. Furthermore, it is difficult to extend to deeper NNs in their approach. On the other hand, we focus on a general framework for causal structure learning: we do not limit the form of score functions and the model functions for causal relationships. Our masking is only applied to causal structure indicated by a graph adjacency matrix (equivalent to their structural gates in their work). Consequently, the problem formulation and training procedure are much different.
- Other differences: 1) the use of the Gumbel-Softmax approach results in different forms of approximating the binary variables. We additionally validate the use and the benefit of the Gumbel-Softmax approach to our method. 2) we compare our method with recently developed gradient based methods including NOTEARS, GraN-DAG, and DAG-GNN. These comparisons are not included in their paper. 3) the experiment settings are also very different.

We will include these similarities and differences between our work and the latest version of Kalainathan et al. (2018) in the main content of our future revision.

B. Further Details on Restricted ANM and Restricted ASEM

We review the definition and the identifiability result of restricted ANMs, and establish a condition under which an ASEM is restricted. For ease of presentation, in this section we use X_S to denote the sub-vector of $X = [X_1, X_2, \ldots, X_d]^T$ corresponding to $S \subseteq V$ and x_S to denote a value of X_S. We also use $P(X_S)$ to denote the marginal distribution of $P(X)$ over S and $P(\epsilon_i)$ to denote the distribution of ϵ_i.

B.1. Restricted ANM

We first present the following condition.

Condition 1 (Peters et al. (2014, Cond. 19))**.** *The triple* $(f_j, P(X_i), P(\epsilon_j))$, $i \neq j$ *does not solve the following differential equation for all* x_i, x_j *with* $\nu''(x_j - f(x_i))f'(x_i) \neq 0$*:*

$$\eta''' = \eta''(-\frac{\nu''' f'}{\nu''} + \frac{f''}{f'}) - 2\nu'' f'' f' + \nu' f''' + \frac{\nu' \nu''' f'' f'}{\nu''} - \frac{\nu'(f'')^2}{f'}.$$

Here, $f := f_j$*, and* $x_i := \log p_{X_i}$ *and* $\eta := \log p_{\epsilon_i}$ *are the logarithm of the strictly positive densities of* X_i *and* ϵ_i*, respectively, and we have skipped the arguments* $x_j - f(x_i)$*,* x_i*, and* x_i *for* v*,* η *and* f *and their derivatives, respectively.*

The restricted ANM is defined based on Cond. 1.

Definition 3 (Restricted ANM (Peters et al., 2014, Def. 27)). *Consider an ANM with d variables defined in Eq. (1). We call this ANM restricted if for all $X_j \in V$, $X_i \in X_{\mathrm{pa}(j)}$, and all sets $S \subseteq V$ with $X_{\mathrm{pa}(j)\setminus\{i\}} \subseteq S \subseteq X_{\mathrm{pa}(j)\setminus\{i,j\}}$, there is an x_S with $P(x_S) > 0$, s.t. the triple*

$$\big(f_j(x_{\mathrm{pa}(j)\setminus\{i\}}, X_i), P(X_i \mid X_S = x_s), P(\epsilon_j)\big)$$

satisfies Cond. 1. Here, $f_j(x_{\mathrm{pa}(j)\setminus\{i\}}, X_i)$ is a function of X_i. In particular, we require the noise variables to have non-vanishing densities and the functions f_j to be continuous and three times continuously differentiable.

Peters et al. (2014) further characterizes the identifiability result, formally stated as follows.

Theorem 2 (Peters et al. (2014, Thm. 28)). *If the joint distribution $P(X)$ is generated by a restricted ANM with DAG $\mathcal{G}$ and $P(X)$ satisfies causal minimality w.r.t. $\mathcal{G}$ (given in Sec. 2.1), then $\mathcal{G}$ is identifiable from the joint distribution $P(X)$.*

B.2. Restricted ASEM

Before presenting a sufficient condition to make an ASEM restricted, we first define the set of non-descendants: we say that X_j is a descendant of X_i if there is a directed path from X_i to X_j in a DAG $\widetilde{\mathcal{G}}$ and the set of non-descendants of X_i is denoted by $X_{\widetilde{\mathrm{nd}}(i)}$. The following proposition is a direct consequence of Def. 3 and we omit its proof here.

Proposition 2. *Consider an ASEM defined in Eq. (3) and let $\widetilde{\mathcal{G}}$ be the DAG associated with the reduced SEM. Suppose that the density of $\tilde{\epsilon}_i$ is strictly positive, and that the density function of $\tilde{\epsilon}_i$ and the function h_i, $i = 1, 2, \ldots, d$ are three times continuously differentiable. If for all $X_j \in V$, $X_i \in X_{\widetilde{\mathrm{pa}}(j)}$, and all sets $S \subseteq V$ with $X_{\widetilde{\mathrm{pa}}(j)} \subseteq S \subseteq X_{\widetilde{\mathrm{nd}}(j)} \setminus \{X_i, X_j\}$, there is an $x \in \mathbb{R}^d$ with $P(x) > 0$ s.t.*

$$(h_j(A_{j,-i} \circ x_{-i}, X_i), P(X_i \mid X_S = x_S), P(\epsilon_j)))$$

satisfies Cond. 1, then this ASEM is restricted. Here, $h_j(A_{j,-i} \circ x_{-i}, X_i)$ is a function of X_i, $A_{j,-i}$ and x_{-i} are the sub-vectors of A_j and x without the i-th component, respectively.

C. Proofs

C.1. Proof of Lemma 1

Proof. It suffices to show that if X_j is not a parent of X_i in $\mathcal{H}$, then X_j is not a parent of X_i in $\widetilde{\mathcal{G}}$, either. By Eq. (3), that X_j is not a parent of X_i in $\mathcal{H}$ indicates $A_{ji} = 0$. Therefore, $h_i(A_i \circ X)$ is a constant function w.r.t. X_j. Then according to Eq. (4), we get $X_j \notin X_{\widetilde{\mathrm{pa}}(i)}$ and the input arguments of $\tilde{f}_i$ do not contain X_j. Thus, X_j cannot be a parent of X_i in $\widetilde{\mathcal{G}}$. □

C.2. Proof of Proposition 1

Proof. The Markovian condition implies that $X_{\mathrm{pa}(i,\mathcal{G})}$ satisfies the first condition. To show the second condition, let T' be a strict subset of $X_{\mathrm{pa}(i,\mathcal{G})}$. If X_i is independent of $X_{\mathrm{pa}(i,\mathcal{G})} \setminus T'$ given T', we have

$$P(X_i \mid T') = P(X_i \mid X_{\mathrm{pa}(i,\mathcal{G})}),$$

which violates the causal minimality of $P(X)$ w.r.t. $\mathcal{G}$. Thus, $X_{\mathrm{pa}(i,\mathcal{G})}$ satisfies the second condition.

We next show that $X_{\mathrm{pa}(i,\mathcal{G})}$ is the unique subset of $X_{\mathrm{pa}(i,\mathcal{H})}$ satisfying the two conditions. Let π be a topological order of $\mathcal{H}$. Clearly, π is also a topological order of $\mathcal{G}$. Suppose $X_i = \pi(k)$, i.e., X_i is the k-th variable in π, then $X_{\mathrm{pa}(i,\mathcal{G})} \subseteq X_{\mathrm{pa}(i,\mathcal{H})} \subseteq [\pi(k)]$, where $[\pi(k)] := \{\pi(1), \pi(2), ..., \pi(k-1)\}$. Since $P(X)$ has a strictly positive density, then there is a unique subset of $[\pi(k)]$ satisfying the two conditions, following Pearl (1988, Corollary 3). Therefore, any $T \subseteq X_{\mathrm{pa}(i,\mathcal{H})}$ satisfying the two conditions must be identical to $X_{\mathrm{pa}(i,\mathcal{G})}$. □

D. Derivation of Logistic distribution

We provide a derivation that if two independent variables $X, Y \sim \mathrm{Gumbel}(0, 1)$, then $Z = X - Y \sim \mathrm{Logistic}(0, 1)$. We will show that the CDF of Z matches the CDF of $\mathrm{Logistic}(0, 1)$.

The CDF of X is given by

$$\Pr(X \leq x) = e^{-e^{-x}}, \quad x \in \mathbb{R}.$$

We then get

$$\Pr\left(e^X \leq x'\right) = \Pr\left(X \leq \log x'\right) = e^{-x'}, \quad x' \in \mathbb{R}^+,$$

which indicates e^X follows the exponential distribution with the rate parameter being 1.

To find the distribution of Z, we notice that $Z = \log(e^{X-Y})$. Then for $z \in \mathbb{R}$, we have

$$\begin{aligned}
\Pr\left(Z \leq z\right) &= \Pr\left(\log\left(e^{X-Y}\right) \leq z\right) \\
&= \Pr\left(e^X / e^Y \leq e^z\right) \\
&= \int_{x'=0}^{\infty} \int_{y'=x'/e^z}^{\infty} e^{-x'} e^{-y'} dx' dy' \\
&= \frac{1}{1+e^{-z}},
\end{aligned}$$

which is exactly the CDF of $\mathrm{Logistic}(0, 1)$.

E. Optimization with Augmented Lagrangian

We restate the constrained optimization problem here:

$$\begin{aligned}
\min_{U,\,\phi} \quad & \mathbb{E}\left[\frac{1}{2n}\sum_{k=1}^{n} \mathcal{L}\Big(x^{(k)},\, h\left(\mathrm{g}_\tau(U), x^{(k)}; \phi\right)\Big) + \lambda \|\mathrm{g}_\tau(U)\|_1\right] \\
\text{subject to} \quad & \mathbb{E}\left[\mathrm{tr}\left(e^{\,\mathrm{g}_\tau(U)}\right) - d\right] \leq \xi,
\end{aligned}$$

where we have added the sparsity-inducing term with penalty weight $\lambda \geq 0$ to the score function and the expectations are taken w.r.t. independent $\mathrm{Logistic}(0, 1)$ samples. The augmented Lagrangian is

$$L_\rho(U, \phi, \alpha) = \mathbb{E}\left[\frac{1}{2n}\sum_{k=1}^{n} \mathcal{L}\Big(x^{(k)},\, h\left(\mathrm{g}_\tau(U), x^{(k)}; \phi\right)\Big) + \lambda \|\mathrm{g}_\tau(U)\|_1 + \alpha h(U)\right] + \frac{\rho}{2}\big(\mathbb{E}[h(U)]\big)^2,$$

where $h(U) \coloneqq \mathrm{tr}\left(e^{\,\mathrm{g}_\tau(U)}\right) - d \geq 0$, α is the Lagrange multiplier, and $\rho > 0$ is the penalty parameter. We also define $\mathbf{g} \in \mathbb{R}^{d \times d}$ as the independent $\mathrm{Logistic}(0, 1)$ samples associated with U in $\mathrm{g}_\tau(U)$, and write $L_\rho(U, \phi, \alpha; \mathbf{g})$ and $h(U; \mathbf{g})$ as a sample estimate of $L_\rho(U, \phi, \alpha)$ and $\mathbb{E}\left[h(U)\right]$ evaluated with a given sample $\mathbf{g}$, respectively.

We next write the following standard updating rules

$$U^{t+1},\, \phi^{t+1} = \arg\min_{U,\,\phi} L_{\rho^t}(U, \phi, \alpha^t), \tag{7}$$

$$\alpha^{t+1} = \alpha^t + \rho^t\, \mathbb{E}[h(U^{t+1})], \tag{8}$$

$$\rho^{t+1} = \begin{cases} \beta\rho^t, & \text{if } \mathbb{E}[h(U^{t+1})] \geq \gamma \mathbb{E}[h(U^t)], \\ \rho^t, & \text{otherwise}, \end{cases} \tag{9}$$

with $\beta > 1$ and $0 < \gamma < 1$ being tuning hyperparameters. The updating rules Eqs. (7), (8), and (9) all involve expectations w.r.t. $\mathrm{Logistic}(0, 1)$ samples and their closed forms are not easy to obtain. We therefore use sample estimates as approximations and details are given below:

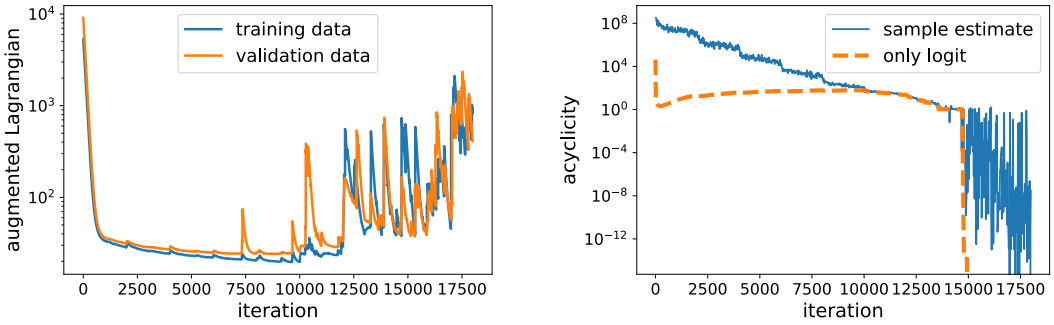
- The subproblem in Eq. (7) is approximately solved using first-order method Adam (Kingma & Ba, 2014), by running $1,000$ iterations with learning rate 3×10^{-2}. At each iteration, we draw independently $\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_B$ and approximate the gradient by $\nabla_{U,\phi}\left(\frac{1}{B}\sum_{i=1}^{B} L_\rho(U, \phi, \alpha; \mathbf{g}_i)\right)$. Notice that all the observed data $\{x^{(k)}\}_{k=1}^n$ are used for estimating the gradient. If the observed data cannot be loaded into memory, we may simply sample minibatches of the dataset to estimate the gradient.
- With an approximate solution U^{t+1} obtained from Eq. (7), we similarly use sample estimate to approximate $\mathbb{E}[h(U^{t+1})]$, which is $\frac{1}{B}\sum_{i=1}^{B} h(U^{t+1}; \mathbf{g}_i)$, in both Eqs. (8) and (9).

As shown by Jang et al. (2017), the Gumbel-Softmax approach is effective for single sample gradient estimation, so we simply pick $B = 1$ which is found to work well in our experiments. We pick $\gamma = 0.25$, same as in NOTEARS and DAG-GNN, and initialize the Lagrange multiplier α to 0, i.e., $\alpha^0 = 0$. We set the initial value of ρ by $\rho^0 = 10^{-\lceil 0.3d \rceil}$ for d-node graphs. This is because we initialize the logits U_{ij}'s to be 0 and consequently $h(U^0)$ would be very large with high probability for large graphs. Since both α and ρ do not decrease during training, a large ρ^0 may make the optimization somewhat 'omit' the score function and the resulting graph would be a DAG with high score. However, a small initial value ρ^0 would result in a long training time, so we choose larger β for larger graphs to accelerate training. In particular, we find that $\beta = 5$, 15, and 300 work well for graphs of 10, 20, and 50 nodes, respectively. These choices are found on synthetic datasets with causal additive model and known true causal graphs (see Sec. G.1 for further discussions). For graphs with other sizes, a linear interpolation on the logarithm scale of the graph sizes can be used as the choice for β, and one may also consider fine tuning using some synthetic datasets with known true graphs. Finally, the ℓ_1 penalty weight is chosen as $\lambda = 2 \times 10^{-3}$ to slightly control false discoveries.

Stopping criterion Simply using a single sample estimate $h(U^t)$ for stopping criterion may make the optimization stop too early. We consider further steps:

- we choose the tolerance ξ to be small to lower the probability of stopping the algorithm early. Particularly, we pick $\xi = 10^{-10}$ in our experiment which is much smaller than that used in NOTEARS and GraN-DAG.
- we set another stopping criterion $\operatorname{tr}\left(e^{\sigma(U^t/\tau)}\right) < \xi$ which uses only the logits. This criterion will be satisfied only when the entries $\sigma(U^t_{ij}/\tau)$, where i, j are such that the edge $X_j \to X_i \notin E$ for some DAG with E as edge set, are nearly zeros. If an edge indeed helps minimize the score function, then the logit will be pushed to a relatively large positive value so that the score function is minimized in the expected sense. Thus, as another benefit, we can then readily use $\sigma(U^t/\tau)$ as our learned matrix which indicates a DAG after a hard thresholding at threshold 0.5.
- notice that $h(U^t)$ and $\operatorname{tr}\left(e^{\sigma(U^t/\tau)}\right)$ are evaluated at the end of step t of the augmented Lagrangian, or the $t \times 10^3$-th iteration of the total optimization iterations. This also helps avoid terminating the optimization too early.

In Fig. 6, we provide an experimental result regarding $h(U)$ and $\operatorname{tr}\left(e^{\sigma(U/\tau)}\right)$, on the GP dataset with 50-node ER1 graph that was used in Sec. 5.3. We also re-plot the trajectories of the augmented Lagrangian here (notice that the validation dataset is not used to optimize U and ϕ). Fig. 6 and similar results on other datasets validate the effectiveness of the use of both stopping criteria. As discussed in Sec. 4.3, the matrix $\sigma(U^t/\tau)$ at the last step will be further processed to output our inferred causal graph.



(a) Augmented Lagrangian.

(b) $h(U)$ and $\operatorname{tr}(e^{\sigma(U/\tau)})$.

Figure 6: Training trajectories.

F. Implementations

Existing causal structure learning methods used in our experiments all have available implementations, as listed below:

- GES and PC: an implementation of both methods is available through the `py-causal` package at https://github.com/bd2kccd/py-causal.
- CAM (Peters et al., 2014): its codes are available through the CRAN R package repository at https://cran.r-project.org/web/packages/CAM.
- NOTEARS: codes are available at the first author's github repository https://github.com/xunzheng/notears.
- DAG-GNN: its Python codes are available at the first author's github repository https://github.com/fishmoon1234/DAG-GNN.
- GraN-DAG: implementation is available at the first author's github repository https://github.com/kurowasan/GraN-DAG. Note that for graphs of 50 nodes or more, GraN-DAG performs a preliminary neighborhood selection step to avoid overfitting.

In the experiments, we mostly use default hyperparameters for these algorithms unless otherwise stated.

G. Supplementary Experiment Details and Results

G.1. Hyperparameters

MaskedNN uses 4-layer feed-forward NNs with 16 Leaky ReLU units (Maas et al., 2013) at each hidden layer as the model functions. The NN weights are initialized using the Xavier uniform initialization (Glorot & Bengio, 2010) and each logit U_{ij} in $\mathrm{g}_\tau(U)$ is initialized to 0. We set a fixed temperature $\tau = 0.2$ for Gumbel-Sigmoid. The hyperparameters related to the augmented Lagrangian method have been discussed in Sec. E.

In practice, however, one could not perform hyperparameter tuning directly on the observed data as the true causal graph is not available. Similar to GraN-DAG, we conduct experiments on synthetic data models with known causal graphs to search for the hyperparameters, and then use these hyperparameters for all the experiments. In particular, we choose a causal additive data model (see Sec. G.3) and the CAM algorithm (Bühlmann et al., 2014), which is specifically designed for this type of data models, can serve as a good benchmark for our method.

G.2. Experiment Setup and Results for Nonlinear SEMs with Quadratic Functions

This section provides further experiment details for the nonlinear SEMs with quadratic functions in Section 5.1.2.

Setup For each variable X_i, we expand its set of parental nodes $X_{\mathrm{pa}(i)}$ to obtain both first- and second-order feature terms. That is, for $X_{\mathrm{pa}(i)} = \{X_{j_1}, X_{j_2}, \ldots, X_{j_{d'}}\}$ with d' being the cardinality of $X_{\mathrm{pa}(i)}$, the first-order feature terms correspond to $X_{j_1}, \ldots, X_{j_{d'}}$, and the second terms are $X_{j_1}^2, \ldots, X_{j_{d'}}^2, X_{j_1}X_{j_2}, \ldots, X_{j_1}X_{j_{d'}}, X_{j_2}X_{j_3}, \ldots, X_{j_{d'-1}}X_{j_{d'}}$. The coefficient for each term is either 0 or sampled from $\mathrm{Uniform}([-1, -0.5] \cup [0.5, 1])$, with equal probability. If a parent variable is not contained in any feature term with a non-zero coefficient, the corresponding edge is removed from the given DAG. We consider ER1 graphs with $d \in \{10, 20, 50\}$ nodes. Other settings such as noise distribution and number of samples are same as the previous experiments.

For this data model, there may exist very large variable values that would lead to numerical issues for the followed gradient-based structure learning methods including NOTEATS, GraN-DAG, and MaskedNN. We therefore limit the diameter of each underlying causal graph to 3 and normalize each variable's values by dividing the total number of corresponding first- and second-order terms. The normalized data are then used as observed data in the experiment. The true causal structure is identifiable according to Peters et al. (2014).

NOTEARS-quad from Zhu et al. (2020) NOTEARS-quad utilizes the prior knowledge that each causal function is quadratic and uses a similar idea from GraN-DAG to construct an equivalent weighted adjacency matrix for NOTEARS. It was fully described in Zhu et al. (2020) and we provide its details here for completeness. Note that the following content is

directly quoted from Zhu et al. (2020, Appendix E), with very slight modifications. Note also that one needs extra effort to derive explicitly the equivalent adjacency matrix and further to implement the method based on this matrix. In contrast, our proposed method in Sec. 4 only requires replacing the feed-forward NNs with quadratic functions.

To obtain NOTEARS-quad, we take the first variable X_1 for example. We first expand the rest variables $X_2, \ldots, X_d$ to contain both first- and second-order features: $X_2, \ldots, X_d, X_2X_3, \ldots, X_iX_j, \ldots, X_{d-1}X_d$ with $i, j = 2, \ldots, d$ and $i \leq j$. There are in total $d(d-1)/2$ terms and we use $\widetilde{\mathbf{X}}_1$ to denote the vector that concatenates these feature terms. Correspondingly, we use c_i and c_{ij} to denote the coefficients associated with these features and $\mathbf{c}_1$ to denote the concatenating vector of the coefficients. Notice that the variable X_l, $l \neq 1$ affects X_1 only through the terms X_l, X_iX_l with $i \leq l$, and X_lX_j with $j > l$. Therefore, an equivalent weighted adjacency matrix W lying in $\mathbb{R}^{d\times d}$ can be constructed with the $(l, 1)$-th entry $W_{l1} := |c_l| + \sum_{i=2}^{l} |c_{il}| + \sum_{j=l+1}^{d} |c_{lj}|$; in this way, $W_{l1} = 0$ implies that X_l has no effect on X_1. The least squares term, corresponding to variable X_1, in the loss function becomes $\sum_{k=1}^{n} \left(x_1^{(k)} - \mathbf{c}_1^T \widetilde{\mathbf{x}}_1^{(k)}\right)^2$ where n is the total number of samples. In summary, we have the following optimization problem

$$\begin{aligned} \min_{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_d} \quad & \sum_{i=1}^{d} \sum_{k=1}^{n} \left(x_1^{(k)} - \mathbf{c}_1^T \widetilde{\mathbf{x}}_1^{(k)}\right)^2 \\ \text{subject to} \quad & \operatorname{tr}\left(e^{W \circ W}\right) - d = 0, \end{aligned}$$

where $\circ$ denotes the element-wise product and the constraint enforces acyclicity w.r.t. a weighted adjacency matrix (cf. Eq. (2)). This problem can be solved by the augmented Lagrangian method where at each step the augmented Lagrangian is minimized by the first order method Adam (Kingma & Ba, 2014) with Tensorflow (Abadi et al., 2016). The Lagrange multiplier and penalty parameter are updated in the same fashion as in the original NOTEARS.

Quadratic regression based pruning method Zhu et al. (2020) also proposed a Quadratic Regression based Pruning method (QR pruning) to remove spurious edges. QR pruning applies quadratic regression to each variable X_i and its parents $X_{\mathrm{pa}(i)}$ indicated from the estimated DAG, followed by thresholding on the resulting coefficients of both first- and second-order terms. In this experiment, we pick a threshold 0.1 for QR-pruning. If the coefficient of an interaction term, e.g., $X_{i_1}X_{i_2}$, is non-zero after thresholding, then we have two directed edges which are $X_{i_1} \to X_i$ and $X_{i_2} \to X_i$.

Empirical results The empirical results are presented in Table 2.

Table 2: Empirical results on nonlinear SEMs with quadratic functions.

	10 nodes		20 nodes		50 nodes	
	SHD	TPR	SHD	TPR	SHD	TPR
MaskedNN	**4.4 ± 1.7**	**0.63 ± 0.16**	**4.8 ± 1.7**	**0.80 ± 0.16**	**20.4 ± 10.5**	**0.68 ± 0.12**
MaskedQR	**3.6 ± 3.0**	**0.72 ± 0.20**	**4.2 ± 2.9**	**0.80 ± 0.14**	**20.6 ± 10.7**	**0.65 ± 0.14**
GraN-DAG	6.0 ± 3.3	0.59 ± 0.21	8.2 ± 3.0	0.78 ± 0.19	49.0 ± 12.3	0.53 ± 0.11
CAM	5.4 ± 3.2	0.64 ± 0.20	11.8 ± 4.2	0.69 ± 0.18	32.6 ± 11.5	0.50 ± 0.13
NOTEARS-quad	9.8 ± 3.2	0.22 ± 0.08	13.4 ± 4.5	0.40 ± 0.12	46.0 ± 7.0	0.23 ± 0.05

G.3. Causal Additive Model

We consider a causal additive model in the following form:

$$X_i = \sum_{X_j \in X_{\mathrm{pa(i)}}} W_{ij} \cos(X_j + \alpha_{ij}) + \epsilon_i,$$

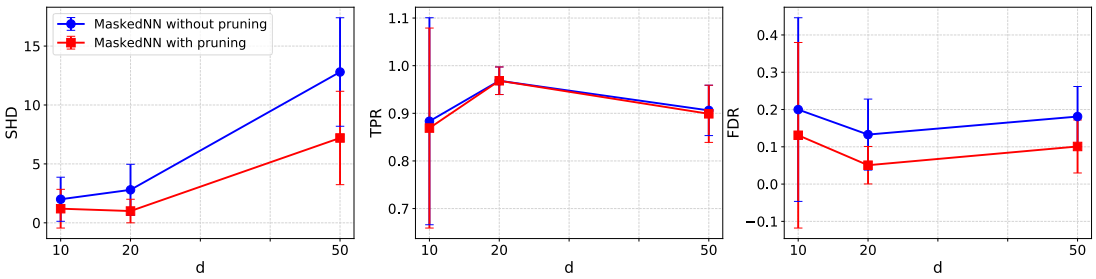
where W_{ij} and α_{ij} are independently sampled from $\mathrm{Uniform}\left([-2, 0.5] \cup [0.5, 2]\right)$. The graph setting and noise distribution are set to be identical to the first experiment in Sec. 5.1.1. This data model follows exactly the causal additive assumption and is known to be identifiable (Bühlmann et al., 2014).

As discussed in Secs. E and G.1, we first generate a few synthetic datasets to select hyperparameters. Once the hyperparameters are fixed, we use different seeds to generate datasets for experiments. The empirical results are reported in Table 3. CAM achieves the best SHD and TPR on average, while MaskedNN and GraN-DAG have slightly worse performance. This

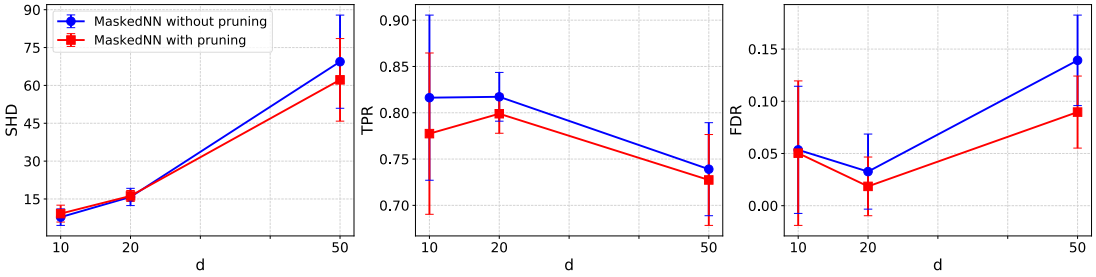
is not surprising, as this data model follows exactly the causal additive assumption (Bühlmann et al., 2014). Again the three methods outperform the rest methods by a large margin. Notice that this data model is similar to that used in the DAG-GNN paper. However, DAG-GNN is outperformed by the aforementioned methods and its performance is nearly the same as NOTEARS that is developed for linear SEMs.

Table 3: Empirical results on nonlinear SEMs with causal additive model.

	ER1 with 10 nodes		ER4 with 10 nodes		ER1 with 50 nodes		ER4 with 50 nodes	
	SHD	TPR	SHD	TPR	SHD	TPR	SHD	TPR
MaskedNN	**0.8 ± 1.3**	**0.92 ± 0.14**	**8.8 ± 2.6**	**0.80 ± 0.05**	**8.2 ± 2.4**	**0.91 ± 0.03**	**43.8 ± 9.9**	**0.83 ± 0.02**
GraN-DAG	**2.2 ± 1.7**	**0.85 ± 0.11**	**7.2 ± 4.0**	**0.84 ± 0.09**	**9.2 ± 2.8**	**0.89 ± 0.05**	**39.6 ± 4.0**	**0.86 ± 0.01**
CAM	**3.2 ± 1.2**	**1.00 ± 0.00**	**4.0 ± 1.8**	**0.92 ± 0.04**	**1.0 ± 0.9**	**1.00 ± 0.00**	**27.2 ± 4.4**	**0.88 ± 0.01**
DAG-GNN	3.6 ± 1.4	0.68 ± 0.09	26.8 ± 3.0	0.38 ± 0.08	25.6 ± 5.3	0.55 ± 0.09	154.4 ± 8.7	0.29 ± 0.09
NOTEARS	3.4 ± 2.3	0.77 ± 0.11	23.6 ± 2.0	0.45 ± 0.06	19.6 ± 3.6	0.70 ± 0.05	150.0 ± 9.6	0.38 ± 0.04
GES	5.2 ± 2.1	0.74 ± 0.06	33.4 ± 3.0	0.22 ± 0.09	22.0 ± 4.6	0.75 ± 0.06	183.6 ± 5.9	0.24 ± 0.01
PC	4.4 ± 4.0	0.71 ± 0.23	34.6 ± 1.6	0.19 ± 0.03	38.8±7.1	0.72 ± 0.07	212.4 ± 5.9	0.29 ± 0.02



(a) ER1 graphs.



(b) ER4 graphs.

Figure 7: The effect of pruning on MaskedNN.

G.4. Pruning

We investigate the effect of the additional pruning step on the proposed method. We use the same GP datasets from the first experiment in Sec. 5.1.1 and compare MaskedNN with or without pruning on ER1 and ER4 graphs with $d \in \{10, 20, 50\}$ nodes. The pruning method used here is the CAM pruning from Bühlmann et al. (2014). Notice that this pruning step is necessary for CAM and GraN-DAG. CAM first estimates a topological order of the variables and the graph has all the possible edges that does not violate the acyclicity constraint. It then uses pruning to remove spurious edges. GraN-DAG estimates an equivalent adjacency matrix and then removes an edge if the corresponding entry has the smallest value in the absolute Jacobian matrix until a DAG is obtained. Note that this DAG typically contains many spurious edges and pruning must be performed to reduce false discoveries.

Fig. 7 reports the empirical results in terms of SHD, TPR and also False Discovery Rate (FDR). We observe that the additional pruning step reduces the SHD and FDR much, and has little effect on the TPR on ER1 graphs. For ER4 graphs that are denser, CAM pruning reduces both FDR and TPR. Nevertheless, the overall metric SHD is increased by CAM pruning. This experiment demonstrates the practical importance of applying an additional pruning step in MaskedNN.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2016.

Abramson, B., Brown, J., Edwards, W., Murphy, A., and Winkler, R. L. Hailfinder: A bayesian system for forecasting severe weather. *International Journal of Forecasting*, 12(1):57 – 71, 1996.

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Bertsekas, D. P. *Nonlinear Programming*. Athena Scientific, 1999.

Bühlmann, P., Peters, J., Ernest, J., et al. CAM: Causal additive models, high-dimensional order search and penalized regression. *The Annals of Statistics*, 42(6):2526–2556, 2014.

Chickering, D. M. Learning Bayesian networks is NP-complete. In *Learning from Data: Artificial Intelligence and Statistics V*. Springer, 1996.

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

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

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Glymour, C., Zhang, K., and Spirtes, P. Review of causal discovery methods based on graphical models. *Frontiers in Genetics*, 10, 2019.

Goudet, O., Kalainathan, D., Caillou, P., Guyon, I., Lopez-Paz, D., and Sebag, M. Learning functional causal models with generative neural networks. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*. Springer, 2018.

Gretton, A., Bousquet, O., Smola, A., and Schölkopf, B. Measuring statistical dependence with Hilbert-Schmidt norms. In *International Conference on Algorithmic Learning Theory*, 2005.

Gu, S., Levine, S., Sutskever, I., and Mnih, A. Muprop: Unbiased backpropagation for stochastic neural networks. In *ICLR*, 2016.

He, Y., Jia, J., and Yu, B. Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 16:2589–2609, 2015.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems 21*, 2009.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations (ICLR)*, 2017.

Kalainathan, D., Goudet, O., Guyon, I., Lopez-Paz, D., and Sebag, M. Structural agnostic modeling: Adversarial learning of causal graphs. *arXiv preprint arXiv:1803.04929*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *ICLR*, 2014.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2013.

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-based neural dag learning. In *ICLR*, 2020.

Lucas, P. J. F., van der Gaag, L. C., and Abu-Hanna, A. Bayesian networks in biomedicine and healthcare. *Artificial Intelligence in Medicine*, 30(3):201–214, March 2004.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables, 2016.

Meek, C. Causal inference and causal explanation with background knowledge. In *UAI*, 1995.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. In *ICML*, 2014.

Ng, I., Zhu, S., Chen, Z., and Fang, Z. A graph autoencoder approach to causal structure learning. *arXiv preprint arXiv:1911.07420*, 2019.

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

Pearl, J. *Causality*. Cambridge University Press, 2009.

Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. Causal discovery with continuous additive noise models. *The Journal of Machine Learning Research*, 15(1):2009–2053, 2014.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

Sachs, K., Perez, O., Pe'er, D., Lauffenburger, D. A., and Nolan, G. P. Causal protein-signaling networks derived from multiparameter single-cell data. *Science*, 308(5721):523–529, 2005.

Shimizu, S., Hoyer, P. O., Hyvärinen, A., and Kerminen, A. A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

Shimizu, S., Inazumi, T., Sogawa, Y., Hyvärinen, A., Kawahara, Y., Washio, T., Hoyer, P. O., and Bollen, K. Directlingam: A direct method for learning a linear non-Gaussian structural equation model. *Journal of Machine Learning Research*, 12 (Apr):1225–1248, 2011.

Spirtes, P. Introduction to causal inference. *Journal of Machine Learning Research*, 11(May):1643–1662, 2010.

Spirtes, P. and Glymour, C. An algorithm for fast recovery of sparse causal graphs. *Social Science Computer Review*, 9(1): 62–72, 1991.

Spirtes, P., Glymour, C. N., and Scheines, R. *Causation, Prediction, and Search*. MIT Press, second edition, 2000.

Tsamardinos, I., Aliferis, C. F., and Statnikov, A. R. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference*, 2003.

Yu, Y., Chen, J., Gao, T., and Yu, M. DAG-GNN: DAG structure learning with graph neural networks. In *ICML*, 2019.

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

Zhang, K. and Hyvärinen, A. On the identifiability of the post-nonlinear causal model. In *Conference on Uncertainty in Artificial Intelligence*, 2009.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. Kernel-based conditional independence test and application in causal discovery. In *Conference on Uncertainty in Artificial Intelligence*, 2012.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. DAGs with NO TEARS: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems*, 2018.

Zhu, S., Ng, I., and Chen, Z. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020.