Score Matching Enables Causal Discovery of Nonlinear Additive Noise Models

Paul Rolland [1] **Volkan Cevher** [1] **Matthäus Kleindessner** [2] **Chris Russel** [2] **Bernhard Schölkopf** [2]
Dominik Janzing [2] **Francesco Locatello** [2]

Abstract

This paper demonstrates how to recover causal graphs from the score of the data distribution in non-linear additive (Gaussian) noise models. Using score matching algorithms as a building block, we show how to design a new generation of scalable causal discovery methods. To showcase our approach, we also propose a new efficient method for approximating the score's Jacobian, enabling to recover the causal graph. Empirically, we find that the new algorithm, called SCORE, is competitive with state-of-the-art causal discovery methods while being significantly faster.

1. Introduction

In this work, we focus on causal discovery from purely observational data, i.e., finding a causal Directed Acyclic Graph (DAG) underlying a given data set. This problem is at the core of causality, since knowledge of the causal graph support the prediction of the effect of interventions (Peters et al., 2017; Schölkopf et al., 2021).

In general, the problem of causal discovery from observational data is ill-posed: there may be several generative models with various causal structures that can produce the same data distribution. Therefore, in order to make the problem well-posed, we need to rely on extra assumptions on the generative process. A popular solution is to assume that the noise injected during the generation of each variable is additive (see equation (1)). Under additional assumptions on the link functions, it has been shown that such model is identifiable from purely observational data (Peters et al., 2014).

Many causal discovery algorithms maximize a suitable loss function over the set of (DAGs). Unfortunately, solving such problem using classical loss functions is known to be NP-hard (Chickering, 1996). Therefore, recent methods focused on heuristic approximations, e.g., by using a greedy approach (PC, FCI (Spirtes et al., 2000; Zhang, 2008), GES (Chickering, 2002), CAM (Bühlmann et al., 2014) and others (Teyssier & Koller, 2012; Larranaga et al., 1996; Singh & Valtorta, 1993; Cooper & Herskovits, 1992; Bouckaert, 1992)), by expressing the problem as a continuous non-convex optimization problem and applying first-order optimization methods (GraNDAG (Lachapelle et al., 2019), NOTEARS (Zheng et al., 2018)), or by using Reinforcement Learning methods (RL-BIC (Zhu et al., 2019), CORL (Wang et al., 2021)).

There are two distinct aspects that make the search over DAGs difficult: the size of the set of DAGs, which grows super-exponentially with the number of nodes, and the acyclicity constraint. In order to reduce the impact of these two difficulties, approaches called order-based methods (Teyssier & Koller, 2012) tackle the problem in two phases. First, they find a certain *topological ordering* of the nodes, such that a node in the ordering can be a parent only of the nodes appearing after it in the same ordering. Second, the graph is constructed respecting the topological ordering and pruning spurious edges, e.g., using sparse regression (Bühlmann et al., 2014). While the first step still requires to solve a combinatorial problem, the set of permutations is much smaller than the set of DAGs. Moreover, once a topological order is fixed, the acyclicity constraint is naturally enforced, making the pruning step easier to solve.

The method that we propose is an order-based one, where the topological order is estimated based on an approximation of the *score* of the data distribution. The score of a distribution with a differentiable probability density $p(x)$ is defined as the map $\nabla \log p(x)$.[1] We show that for a nonlinear additive Gaussian noise model, it is possible to identify leaves of the causal graph by analysing its entailed observational score. By sequentially identifying the leaves

[1]École Polytechnique Fédéral de Lausanne, Lausanne, Switzerland [2]Amazon, Tuebingen, Germany. Correspondence to: Paul Rolland <paul.rolland@epfl.ch>, Francesco Locatello <locatelf@amazon.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

[1]The term *score* has been used in the causality literature with a different meaning. Classical works (Chickering, 2002) use this term referring to the objective of an optimization problem yielding the causal structure as solution. In this paper, the term score means $\nabla \log p(x)$ as in the statistics literature (Wilks, 1962).

of the causal graph, and removing the identified leaf variables, one can obtain a complete topological order with a time complexity *linear in the number of nodes*. Classical pruning techniques can then be used in order to obtain the final graph. While the proposed algorithm is designed for additive Gaussian noise models, we show that the main required ingredient for our method to work is the additive structure of the model, rather than the noise type. Hence, we expect similar methods to also be applicable to other types of noise (i.e, non-Gaussian). Closest to our work is LISTEN (Ghoshal & Honorio, 2018) which can be derived as a special case of our framework in the linear setting as described in the related work section.

In order to approximate the score of the data distribution from a sample, we exploit and extend recent work on score matching and density gradient estimation (Li & Turner, 2017). Score approximation methods from observational data have shown success in general machine learning tasks such as generative (Song & Ermon, 2019) and discriminative models (Zimmermann et al., 2021), leading to increased interest in developing scalable and efficient solutions. In particular, score-based generative models have shown state-of-the-art performance for image generation (Song & Ermon, 2019; Song et al., 2020b;a; Song & Ermon, 2020). As much of the prior work on causal discovery approaches has focused on leveraging machine/deep learning (Lachapelle et al., 2019; Zheng et al., 2018; Zhu et al., 2019; Wang et al., 2021) to provide a tractable approximation to an NP-hard problem, our work is especially relevant to bridge the gap between provably identifying the causal structure and leveraging advances in deep generative models to scale to large sample sizes and high dimensions.

Hereafter, we summarize our contributions:

- We start by showing that, in the case of non-linear additive Gaussian noise model, knowing the distribution's score function is sufficient to recover the full causal graph, and we provide a method for doing so. Our approach enjoys a linear complexity in the number of nodes to identify the topological order and introduces a new way of learning causal structure from observational data. To the best of our knowledge, the link between the score function and the causal graph structure established in Lemmata 1 and 2 is not only useful, but also novel.

- We propose a new method for estimating the score's Jacobian over a set of observations, exploiting and extending an existing method based on Stein's identity, which can be of independent interest. This method is then used to design a practical algorithm for estimating the causal topological order.

- We finally evaluate our proposed algorithm on both synthetic and real world data and show competitive results compared to state-of-the-art methods, while being significantly faster ($10\times$ faster than CAM (Bühlmann et al., 2014) on 20 nodes graphs and $5\times$ faster than GraN-DAG (Lachapelle et al., 2019) on 50 nodes). We also show that our method is robust to noise misspecification and works well when the additive noise is non-Gaussian.

2. Related Work

Causal discovery for non-linear additive models Many algorithms have been proposed in the past few years for the specific problem studied in this work. GraN-DAG (Lachapelle et al., 2019) aims to maximise the likelihood of the observed data under this model, and uses a continous contraint for the acyclicity of the causal graph, proposed in (Zheng et al., 2018), in order to use a continuous optimization method to find a first order stationary point of the problem. CAM (Bühlmann et al., 2014) further assumes that the link functions f_i in (1) also have an additive structure. They first estimate a topological order by greedily maximizing the data likelihood, and then prune the DAG using sparse regression techniques.

In the scope of linear additive models, (Ghoshal & Honorio, 2018) first proposed an approach to provably recover, under some hypothesis on the noise variances, the causal graph in polynomial time and sample complexity. Their approach can be seen as an order-based method, where the ordering is estimated by sequentially identifying leaves based on an estimation of the precision matrix. In spirit, their method is closely related to ours. For instance, if the link functions f_i in (1) are all linear, then the score of the joint distribution of X is given by $s(x) = -\Theta x$, where Θ is the precision matrix. Hence, the score's Jacobian, which is used in our algorithm to identify the causal graph, can be seen as a non-linear generalization of the precision matrix, which has shown success for identifying causal relations in linear settings (Loh & Bühlmann, 2014).

While our work focuses on the identifiable non-linear additive Gaussian noise model, other works target more general non-parametric model, but must then rely on different kinds of assumptions such as faithfulness, restricted faithfulness or sparsest Markov representation (Spirtes et al., 2000; Raskutti & Uhler, 2018; Solus et al., 2021). These works apply conditional independence tests, and learn a graph that matches the identified conditional independence relations (Spirtes et al., 2000; Zhang, 2008).

Score estimation In the scope of generative modelling (Song & Ermon, 2019), the score function is learned by fitting a neural network minimizing the empirical Fisher divergence (Hyvärinen & Dayan, 2005). While performing

well in practice, such method is quite computationally expensive and requires tuning of several training parameters.

For our purpose, we chose to instead minimize the kernelized Stein discrepancy, since this approach provides a close form solution, allowing fast estimation at all observations. In practice, such method performs similarly as score matching while being much faster to compute. Asymptotic consistency of the Stein gradient estimator, and its relation to score matching were analyzed in (Barp et al., 2019).

3. Preliminaries

3.1. Causal discovery for non-linear additive Gaussian noise models

Assume that a random variable $X \in \mathbb{R}^d$ is generated using the following model:

$$X_i = f_i(\text{pa}_i(X)) + \epsilon_i, \tag{1}$$

$i = 1, \dots, d$, where $\text{pa}_i(X)$ selects the coordinates of X which are parents of node i in some DAG. The noise variables $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$ are jointly independent. The functions f_j are assumed to be twice continuously differentiable and non-linear in every component. That is, if we denote the parents $\text{pa}_j(X)$ of X_j by $X_{k_1}, X_{k_2}, \dots, X_{k_l}$, then, for all $a = 1, \dots, l$, the function $f_j(x_{k_1}, \dots, x_{k_{a-1}}, \cdot, x_{k_{a+1}}, \dots, x_{k_l})$ is assumed to be non-linear for some $x_{k_1}, \dots, x_{k_{a-1}}, x_{k_{a+1}}, \dots, x_{k_l} \in \mathbb{R}^{l-1}$.

This model is known to be identifiable from observational data (Peters et al., 2014), meaning that it is possible to recover the DAG underlying the generative model (1) from the knowledge of the joint probability distribution of X. In the present work, we aim to identify the causal graph from the score function $\nabla \log p(x)$, which has a one-to-one correspondence with $p(x)$. Hence, any model identifiable from observational data will be identifiable from the knowledge of the data score function.

3.2. Score matching

The goal of score matching is to learn the score function $s(x) \equiv \nabla \log p(x)$ of a distribution with density $p(x)$ given an i.i.d. sample $\{x^k\}_{k=1,\dots,n}$. In this section, we present a method developed in (Li & Turner, 2017) for estimating the score at the sample points, i.e., approximating $\mathbf{G} \equiv (\nabla \log p(x^1), \dots, \nabla \log p(x^n))^T \in \mathbb{R}^{n \times d}$.

This estimator is based on the well known Stein identity (Stein, 1972), which states that for any test function $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$ such that $\lim_{\mathbf{x} \to \infty} \mathbf{h}(\mathbf{x}) p(\mathbf{x}) = 0$, we have

$$\mathbb{E}_p[\mathbf{h}(\mathbf{x}) \nabla \log p(\mathbf{x})^T + \nabla \mathbf{h}(\mathbf{x})] = 0, \tag{2}$$

where $\nabla \mathbf{h}(\mathbf{x}) \equiv (\nabla h_1(\mathbf{x}), \dots, \nabla h_{d'}(\mathbf{x}))^T \in \mathbb{R}^{d' \times d}$.

By approximating the expectation in (2) using Monte Carlo, we obtain

$$-\frac{1}{n} \sum_{k=1}^{n} \mathbf{h}(\mathbf{x}^k) \nabla \log p(\mathbf{x}^k)^T + \text{err} = \frac{1}{n} \sum_{k=1}^{n} \nabla \mathbf{h}(\mathbf{x}^k), \tag{3}$$

where err is a random error term with mean zero, and which vanishes as $n \to \infty$ almost surely. By denoting $\mathbf{H} = (\mathbf{h}(\mathbf{x}^1), \dots, \mathbf{h}(\mathbf{x}^n)) \in \mathbb{R}^{d' \times n}$ and $\overline{\nabla \mathbf{h}} = \frac{1}{n} \sum_{k=1}^{n} \nabla \mathbf{h}(\mathbf{x}^k)$, equation (3) reads $-\frac{1}{n} \mathbf{H} \mathbf{G} + \text{err} = \overline{\nabla \mathbf{h}}$. Hence, by using ridge regression, the Stein gradient estimator is defined as:

$$\hat{\mathbf{G}}^{\text{Stein}} \equiv \arg\min_{\hat{\mathbf{G}}} \|\overline{\nabla \mathbf{h}} + \frac{1}{n} \mathbf{H} \hat{\mathbf{G}}\|_F^2 + \frac{\eta}{n^2} \|\hat{\mathbf{G}}\|_F^2 \tag{4}$$

$$= -(\mathbf{K} + \eta \mathbf{I})^{-1} \langle \nabla, \mathbf{K} \rangle, \tag{5}$$

where $\mathbf{K} \equiv \mathbf{H}^T \mathbf{H}$, $\mathbf{K}_{ij} = \kappa(\mathbf{x}^i, \mathbf{x}^j) \equiv \mathbf{h}(x^i)^T \mathbf{h}(\mathbf{x}^j)$, $\langle \nabla, \mathbf{K} \rangle = n \mathbf{H}^T \overline{\nabla \mathbf{h}}$, $\langle \nabla, \mathbf{K} \rangle_{ij} = \sum_{k=1}^{n} \nabla_{x_j^k} \kappa(\mathbf{x}^i, \mathbf{x}^k)$ and $\eta \geq 0$ is a regularisation parameter. One can hence use the kernel trick, and use the estimator (5) using any kernel κ satisfying Stein's identity, such as the RBF kernel as shown in (Liu et al., 2016).

In the present work, we will exploit and extend this approach in order to obtain estimates of the score's Jacobian over the observations.

4. Causal Discovery via Score Matching

In this section, we will show how to recover the causal graph from the score function $\nabla \log p(x)$ for a non-linear additive model (1). We first design our proposed method in the case where the additive noise is Gaussian, and then discuss extensions to other types of noise.

4.1. Deduce the causal graph from the score of the data distribution

Suppose that we have access to enough observational data coming from an additive Gaussian noise model (1) so that we can accurately approximate the score function of the underlying data distribution. In order to extract information about the graph structure from the score function, let us write it in closed form for a model of the form (1). The associated probability distribution is given by

$$p(x) = \prod_{i=1}^{d} p(x_i | \text{pa}_i(x))$$

$$\log p(x) = \sum_{i=1}^{d} \log p(x_i | \text{pa}_i(x))$$

$$= -\frac{1}{2} \sum_{i=1}^{d} \left(\frac{x_i - f_i(\text{pa}_i(x))}{\sigma_i} \right)^2 - \frac{1}{2} \sum_{i=1}^{d} \log(2\pi\sigma_i^2).$$

Thus, the score function $s(\mathbf{x}) \equiv \nabla \log p(\mathbf{x})$ reads

$$\boxed{\begin{aligned} s_j(x) = &-\frac{x_j - f_j(\text{pa}_j(x))}{\sigma_j^2} \\ &+ \sum_{i \in \text{children}(j)} \frac{\partial f_i}{\partial x_j}(\text{pa}_i(x)) \frac{x_i - f_i(\text{pa}_i(x))}{\sigma_i^2}. \end{aligned}} \tag{6}$$

An immediate observation from equation (6) is that, if j is a leaf, then $s_j(x) = -\frac{x_j - f_j(\text{pa}_j(x))}{\sigma_j^2}$. Since $j \notin \text{pa}_j(x)$, we have that $\frac{\partial s_j(x)}{\partial x_j} = -\frac{1}{\sigma_j^2}$, and hence, it holds that $\text{Var}\left(\frac{\partial s_j(x)}{\partial x_j}\right) = 0$. The following Lemma shows that this condition is also sufficient for j to be a leaf, providing a way to *provably identify a leaf of the graph from the knowledge of the Jacobian of the score function*.

Lemma 1. *Let p be the probability density function of a random variable X defined via a non-linear additive Gaussian noise model (1), and let $s(x) = \nabla \log p(x)$ be the associated score function. Then, $\forall j \in \{1, \dots, d\}$, we have:*

(i) j is a leaf $\Leftrightarrow \forall x, \frac{\partial s_j(x)}{\partial x_j} = c$, with $c \in \mathbb{R}$ independent of x, i.e., $\text{Var}_X\left[\frac{\partial s_j(X)}{\partial x_j}\right] = 0$.

(ii) If j is a leaf, i is a parent of $j \Leftrightarrow s_j(x)$ depends on x_i, i.e., $\text{Var}_X\left[\frac{\partial s_j(X)}{\partial x_i}\right] \neq 0$.

Proof. (i) Equation (6) implies the "⇒" direction as described above.

We prove the other direction by contradiction. Suppose that j is not a leaf and that $\frac{\partial s_j(x)}{\partial x_j} = c \; \forall x$. We can thus write:

$$s_j(x) = cx_j + g(x_{-j}),$$

where $g(x_{-j})$ can depend on any variable but x_j. By plugging equation (6) in s_j, we get

$$\begin{aligned} &\frac{f_j(\text{pa}_j(x))}{\sigma_j^2} + \sum_{i \in \text{children}(j)} \frac{\partial f_i}{\partial x_j}(\text{pa}_i(x)) \frac{x_i - f_i(\text{pa}_i(x))}{\sigma_i^2} \\ &= \left(c + \frac{1}{\sigma_j^2}\right) x_j + g(x_{-j}). \end{aligned}$$

Let i_c be a child of node j such that $\forall i \in \text{children}(j)$, $i_c \notin \text{pa}_i$. Such a node always exist since j is not a leaf, and it suffices to pick a child of j appearing at last in some topological order. We then have

$$\begin{aligned} &\frac{\partial f_{i_c}}{\partial x_j}(\text{pa}_{i_c}(x)) \frac{x_{i_c} - f_{i_c}(\text{pa}_{i_c}(x))}{\sigma_{i_c}^2} - g(x_{-j}) \\ &= \left(c + \frac{1}{\sigma_j^2}\right) x_j - \frac{f_j(\text{pa}_j(x))}{\sigma_j^2} \\ &- \sum_{i \in \text{children}(j), i \neq i_c} \frac{\partial f_i}{\partial x_j}(\text{pa}_i(x)) \frac{x_i - f_i(\text{pa}_i(x))}{\sigma_i^2}. \end{aligned} \tag{7}$$

Now, due to the specific choice of i_c, we have that the RHS of (7) does not depend on x_{i_c} (note that we are here speaking about functional dependence on variables, not statistical dependence on a random variable). Hence, we have

$$\begin{aligned} &\frac{\partial}{\partial x_{i_c}} \left(\frac{\partial f_{i_c}}{\partial x_j}(\text{pa}_{i_c}(x)) \frac{x_{i_c} - f_{i_c}(\text{pa}_{i_c}(x))}{\sigma_{i_c}^2} - g(x_{-j}) \right) = 0 \\ &\Rightarrow \frac{\partial f_{i_c}}{\partial x_j} = \sigma_{i_c}^2 \frac{\partial g(x_{-j})}{\partial x_{i_c}}. \end{aligned}$$

Since g does not depend on x_j, this means that $\frac{\partial f_{i_c}}{\partial x_j}$ does not depend on x_j neither, implying that f_{i_c} is linear in x_j, contradicting the non-linearity assumption.

(ii) If j is a leaf, then, by equation (6), we have:

$$s_j(x) = -\frac{x_j - f_j(\text{pa}_j(x))}{\sigma_j^2}$$

If i is not a parent of j, then $\frac{\partial s_j}{\partial x_i} \equiv 0$, and hence we have $\text{Var}_X\left[\frac{\partial s_j(x)}{\partial x_i}\right] = 0$. On the other hand, if i is a parent of j, then we have $\frac{\partial s_j}{\partial x_i}(x) = \frac{1}{\sigma_j^2} \frac{\partial f_j}{\partial x_i}(\text{pa}_j(x))$. Moreover, since f_j cannot be linear in x_i, $\frac{\partial f_j}{\partial x_i}(\text{pa}_j(x))$ cannot be a constant, and hence $\text{Var}_X\left[\frac{\partial s_j(X)}{\partial x_i}\right] \neq 0$.

□

Discussion: Lemma 1 shows that, for non-linear additive Gaussian noise models, leaf nodes (and only leaf nodes) have the property that the associated diagonal element in the score's Jacobian is a constant. This hence provides a way to identify a leaf of the causal graph from the knowledge of the variance of the score's Jacobian diagonal elements. By repeating this method and always removing the identified leaves, we can estimate a full topological order. This procedure is summarized in Algorithm 1. In the following section, we present a new approach, exploiting Stein identities, to compute estimates of the score's Jacobian over a set of samples.

Note that the use of empirical variance to identify identically 0 function $\frac{\partial s_j}{\partial x_j}$ is not necessary. However, we did

not find any empirical benefit when using other deviation measures, such as the average distance to the median for example.

DAG pruning. Once a topological order is estimated, the DAG becomes constrained to be a sub-graph of a certain fully connected DAG. However, it is necessary to prune this fully connected DAG to remove spurious edges. In theory, it would be possible to make use the learnt score for this purpose, by using property (ii) of Lemma 1. However, more classical methods such as CAM appears to perform better in practice. The idea behind CAM is to assume that the link functions f_i in (1) have an additive structure. We then perform sparse regression on each component and use hypothesis testing for additive models (Marra & Wood, 2011) to decide upon existence of edges. For further details about this pruning technique, please refer to the original paper (Bühlmann et al., 2014).

Algorithm 1 SCORE-matching causal order search

Input: Data matrix $X \in \mathbb{R}^{n\times d}$.
Initialize $\pi = []$, nodes $= \{1, \dots, d\}$
for $k = 1, \dots, d$ **do**
  Estimate the score function $s_{nodes} = \nabla \log p_{nodes}$ (for example using Algorithm 1).
  Estimate $V_j = \mathrm{Var}_{X_{nodes}}\left[\frac{\partial s_j(X)}{\partial x_j}\right]$.
  $l \leftarrow$ nodes$[\arg\min_j V_j]$
  $\pi \leftarrow [l, \pi]$
  nodes $\leftarrow$ nodes $- \{l\}$
  Remove l-th column of X
end for
Get the final DAG by pruning the full DAG associated with the topological order π.

4.2. Approximation of the score's Jacobian

The Stein gradient estimator $\hat{\mathbf{G}}^{\text{Stein}}$ enables us to estimate the score function point-wise at each of our sample points. However, according to the previous section, what we need for identifying the graph is an estimate of the Jacobian of the score at all samples, in order to estimate its variance. Since we do not have a functional approximation of the score, we cannot use tricks such as auto-differentiation in order to obtain higher order derivative approximations. In this section, we extend the ideas of Stein based estimator to obtain estimates for the score's Jacobian.

For this purpose, we will use the second-order Stein identity (Diaconis et al., 2004; Zhu, 2021). Assuming that p is twice differentiable, for any $q : \mathbb{R}^d \to \mathbb{R}$ such that $\lim_{\mathbf{x}\to\infty} q(\mathbf{x})p(\mathbf{x}) = 0$ and such that $\mathbb{E}[\nabla^2 q(\mathbf{x})]$ exists, the second-order Stein identity states that

$$\mathbb{E}[q(\mathbf{x})p(\mathbf{x})^{-1}\nabla^2 p(\mathbf{x})] = \mathbb{E}[\nabla^2 q(\mathbf{x})], \tag{8}$$

which can be rewritten as

$$\begin{aligned}&\mathbb{E}[q(\mathbf{x})\nabla^2 \log p(\mathbf{x})]\\ &= \mathbb{E}[\nabla^2 q(\mathbf{x}) - q(\mathbf{x})\nabla \log p(\mathbf{x})\nabla \log p(\mathbf{x})^T].\end{aligned} \tag{9}$$

Recall that, in order to identify a leaf of the causal graph, we are only interested in estimating the diagonal elements of the score's Jacobian at the sample points, i.e., $J \equiv (\mathrm{diag}(\nabla^2 \log p(\mathbf{x}^1)), \dots, \mathrm{diag}(\nabla^2 \log p(\mathbf{x}^n)))^T \in \mathbb{R}^{n\times d}$. Using the diagonal part of the matrix equation (9) for various test functions gathered in $\mathbf{h} : \mathbb{R}^d \to \mathbb{R}^{d'}$, we can write

$$\begin{aligned}&\mathbb{E}[\mathbf{h}(\mathbf{x})\mathrm{diag}(\nabla^2 \log p(\mathbf{x}))^T]\\ &= \mathbb{E}[\nabla^2_{\text{diag}}\mathbf{h}(\mathbf{x}) - \mathbf{h}(\mathbf{x})\mathrm{diag}((\nabla \log p(\mathbf{x})\nabla \log p(\mathbf{x})^T))],\end{aligned} \tag{10}$$

where $(\nabla^2_{\text{diag}}\mathbf{h}(\mathbf{x}))_{ij} = \frac{\partial^2 h_i(\mathbf{x})}{\partial x_j^2}$. By approximating the expectations by an empirical average, we obtain, similarly as in (3),

$$\begin{aligned}&\frac{1}{n}\sum_{k=1}^{n}\mathbf{h}(\mathbf{x}^k)\mathrm{diag}(\nabla^2 \log p(\mathbf{x}^k))^T + \text{err}\\ &= \frac{1}{n}\sum_{k=1}^{n}\left(\nabla^2_{\text{diag}}\mathbf{h}(\mathbf{x}^k)\right.\\ &\qquad\left.-\mathbf{h}(\mathbf{x}^k)\mathrm{diag}\left(\nabla \log p(\mathbf{x}^k)\nabla \log p(\mathbf{x}^k)^T\right)\right).\end{aligned} \tag{11}$$

By denoting $\mathbf{H} = (\mathbf{h}(\mathbf{x}^1), \dots, \mathbf{h}(\mathbf{x}^n)) \in \mathbb{R}^{d'\times n}$ and $\overline{\nabla^2_{\text{diag}}\mathbf{h}} \equiv \frac{1}{n}\sum_{k=1}^{n}\nabla^2_{\text{diag}}\mathbf{h}(\mathbf{x}^k)$, equation (11) reads $\frac{1}{n}\mathbf{H}\mathbf{J} + \text{err} = \overline{\nabla^2_{\text{diag}}\mathbf{h}} - \frac{1}{n}\mathbf{H}\mathrm{diag}(\mathbf{G}\mathbf{G}^T)$. Hence, by using the Stein gradient estimator for $\mathbf{G}$, we define the Stein Hessian estimator as the ridge regression solution of the previous equation, i.e.,

$$\hat{\mathbf{J}}^{\text{Stein}} \equiv \tag{12}$$

$$\begin{aligned}&\arg\min_{\hat{\mathbf{J}}}\left\|\frac{1}{n}\mathbf{H}\hat{\mathbf{J}} + \frac{1}{n}\mathbf{H}\mathrm{diag}\left(\hat{\mathbf{G}}^{\text{Stein}}\left(\hat{\mathbf{G}}^{\text{Stein}}\right)^T\right) - \overline{\nabla^2_{\text{diag}}\mathbf{h}}\right\|_F^2\\ &\qquad + \frac{\eta}{n^2}\|\hat{\mathbf{J}}\|_F^2\\ &= -\mathrm{diag}\left(\hat{\mathbf{G}}^{\text{Stein}}\left(\hat{\mathbf{G}}^{\text{Stein}}\right)^T\right) + (\mathbf{K} + \eta\mathbf{I})^{-1}\langle\nabla^2_{\text{diag}}, \mathbf{K}\rangle,\end{aligned} \tag{13}$$

where $\mathbf{K}_{ij} = \kappa(\mathbf{x}^i, \mathbf{x}^j) \equiv \mathbf{h}(\mathbf{x}^i)^T\mathbf{h}(\mathbf{x}^j)$, $\langle\nabla^2_{\text{diag}}, \mathbf{K}\rangle = n\mathbf{H}^T\overline{\nabla^2_{\text{diag}}\mathbf{h}}$, $\langle\nabla^2_{\text{diag}}, \mathbf{K}\rangle_{ij} = \sum_{i=1}^{n}\frac{\partial^2\kappa(\mathbf{x}^i,\mathbf{x}^k)}{\partial\mathbf{x}_j^{k\,2}}$ and $\mathbf{G}^{\text{Stein}}$ is

defined in (5). The regularisation parameter η lifts the eigenvalues of the same matrix $\mathbf{K}$ as in the Stein gradient estimator $\hat{\mathbf{G}}^{\text{Stein}}$. We hence decide to use the same parameter for both ridge regression problems.

Choice of kernel Estimating the score's Jacobian with the method above requires a choice of kernel κ. A widely used kernel is the RBF kernel $\kappa_s(x,y) = e^{-\frac{\|x-y\|_2^2}{2s^2}}$, which has one parameter s called the bandwidth. This parameter can be estimated from the data to be fitted, using the commonly used median heuristic, i.e., choosing s to be the median of the pairwise distances between vectors in X. This estimation procedure even enjoys theoretical convergence properties (Garreau et al., 2017). Note that, when using Algorithm 2 for causal discovery in Algorithm 1, the kernel lengthscale is re-computed each time a node is removed from the data matrix X.

Algorithm 2 Estimating the Jacobian of the score

Input: Data matrix $X \in \mathbb{R}^{n\times d}$, regularisation parameter $\eta > 0$.
$s \leftarrow \text{median}(\{\|x_i - x_j\|_2 : i,j = 1,\dots,n, x_k = X[k,:]\})$.
Compute $\hat{\mathbf{J}}^{\text{Stein}}$ using RBF kernel κ_s, regularisation parameter η and data matrix X based on (13).

Algorithm complexity Estimating the topological order requires inverting d times an $n \times n$ kernel matrix, hence the complexity is $\mathcal{O}(dn^3)$ (and could be improved using, e.g., Strassen's algorithm (Strassen, 1969)). Including the pruning step, the final complexity is hence $\mathcal{O}(dn^3 + dr(n,d))$ where $r(n,d)$ is the complexity of fitting a generalized additive model using n data points in d dimensions. In comparison, the complexity of CAM is $\mathcal{O}(d^2 r(n,d))$. The total computational complexity of GraNDAG is not discussed in (Lachapelle et al., 2019); it is difficult to specify it since it depends on the number of iterations used in the Augmented Lagrangian method, which may depend on the dimension and number of samples. However, GraNDAG is particularly slow due the computation of the acyclicity constraint at each iteration, which requires computing the exponential of a $d \times d$ matrix, taking $\mathcal{O}(d^3)$ operations.

In practice, in our method, the time for estimating the topological order is much smaller than the time for pruning it (30% of the total time for $(d,n) = (20,1000)$ and 5% of the total time for $(d,n) = (50,1000)$). In comparison, CAM spends most of the time estimating the topological order (more than 95% of the total time in all tested scenarios). Hence, we expect the dominant term in our method's time complexity to be $dr(n,d)$, thus improving upon CAM's complexity. Moreover, in the case where n becomes very large, it is possible to use kernel approximation methods to reduce the time complexity of our method (Si et al., 2014).

4.3. Extension to non-Gaussian additive noise models

In the previous section, we exploited the structure of the additive Gaussian noise model to deduce the causal graph from the score function (1). Actually, the main ingredient required in our analysis is the additive structure. Indeed, for any additive noise model (including non-Gaussian noise), the score function has a similar structure as in (6).

Lemma 2. *Suppose that the random variable X is generated from (1) where the noise variables ϵ_i are i.i.d. with smooth probability distribution function p^ϵ. Then, the score function of X can be written as follows:*

$$s_j(\boldsymbol{x}) = \frac{d\log p^\epsilon}{dx}(x_j - f_j(pa_j(\boldsymbol{x}))) - \sum_{i\in children(j)} \frac{\partial f_i}{\partial x_j}(pa_i(\boldsymbol{x}))\frac{d\log p^\epsilon}{dx}(x_i - f_i(pa_i(\boldsymbol{x}))). \tag{14}$$

The decomposition of the score's components j into a common term $\frac{d\log p^\epsilon}{dx}(x_j - f_j(\text{pa}_j(\mathbf{x})))$ and a term involving only the parents of the node j is hence characteristic of general additive noise models. Recall that our method identifies leaves by identifying non-linearity in the components of the score. When the common term is linear in x_j, as it is the case with Gaussian noise, the second term is the only one carrying non-linearities, and the leaves can hence be perfectly identified with this method (see Lemma 1). However, intuitively speaking, even when the noise is non-Gaussian, i.e., when the common term carries non-linearities, the second term still carries non-linearities proportionally to the number of parents of node j. Hence, we may expect that the proposed algorithm can work in the case of general additive models, even when the noise is non-Gaussian. While this does not provide a formal identifiability statement, we will show in the experimental section that our proposed method outperforms other state-of-the-art algorithms on non-Gaussian additive models.

5. Numerical Experiments

We now apply Algorithm 1 with Algorithm 2 as score estimator to synthetic and real-world datasets and compare its performance to state-of-the-art methods, such as CAM (Bühlmann et al., 2014), GraNDAG (Lachapelle et al., 2019), SELF (Cai et al., 2018) and GES (Chickering, 2002). Some other methods such as NOTEARS, PC or FCI are omitted since they perform much worse (Bühlmann et al., 2014; Lachapelle et al., 2019). Recent work (Reisach et al., 2021) warned about the fact that sim-

ulated data sometimes lead to scenari where a topological order can simply be estimated by sorting the nodes variances. In order to defend ourselves against this, we randomly generate the noise variances in the generative model, and show that the estimated order when sorting the variance is much worse than the one estimated by Algorithm 1. The code can be found in `https://github.com/paulrolland1307/SCORE/`.

5.1. Synthetic data

We test our algorithm on synthetic data generated from a non-linear additive Gaussian noise model (1). Mimicking (Lachapelle et al., 2019; Zhu et al., 2019), we generate the link functions f_i by sampling Gaussian processes with a unit bandwidth RBF kernel. The noise variances σ_i^2 are independently sampled uniformly in $[0.4, 0.8]$. The causal graph is generated using the Erdös-Rényi model (Erdös & Rényi, 2011). For a fixed number of nodes d, we vary the sparsity of the sampled graph by setting the average number of edges to be either d (ER1) or $4d$ (ER4). Moreover, to test the robustness of the algorithm against noise type misspecification, we also generate data with Laplace noise instead of Gaussian noise. Additional experiments, using Gumbel noise and scale free graphs (Barabási & Albert, 1999) can be found in Appendix A.

For each method, we compute the structural Hamming distance (SHD) between the output and the true causal graph, which counts the number of missing, falsely detected or reversed edges, as well as the structural intervention distance (SID) (Peters & Bühlmann, 2015) which counts the number of interventional distribution which would be miscalculated using the chosen causal graph.

For all order-based causal discovery methods, we always apply the same pruning procedure, i.e., CAM with the same cutoff parameter of 0.001. Moreover, we compute a quantity measuring how well the topological order is estimated. For an ordering π, and a target adjacency matrix A, we define the topological order divergence $D_{top}(\pi, A)$ as

$$D_{top}(\pi, A) = \sum_{i=1}^{d} \sum_{j:\pi_i > \pi_j} A_{ij}.$$

If π is a correct topological order for A, then $D_{top}(\pi, A) = 0$. Otherwise, $D_{top}(\pi, A)$ counts the number of edges that cannot be recovered due to the choice of topological order. It hence provides a lower bound on the SHD of the final algorithm (irrespective of the pruning method).

The results of the synthetic experiments are shown in Tables 1 to 6. The computed quantities are averages over 10 independent runs. We can see that, for sparser graphs (ER1), our method performs similarly as the best method CAM. However, for denser graphs (ER4), our method performs better, and in particular seems to estimate a better topological order, since the D_{top} value is smaller. For 50 nodes graphs, the two best methods are CAM and ours, which both perform similarly. Note that, in order to run it within a reasonable time frame, we had to restrict the maximum number of neighbours, hence providing a sparsity prior to the algorithm, which fits the correct graph in this situation, since sparse Erdös-Renyi graphs usually do not contain high degree nodes. Since we restricted the number of neighbours in the graph, the order finding part of CAM does not yield a single topological order, hence we could not compute D_{top} in this setting. We also observe that the topological ordering resulting from sorting the variances (VarSort) is much worse in general than with all other methods, showing that finding a topological order for the generated datasets is not a trivial task. Finally, we observe that our method is quite robust to noise misspecification, since the accuracy remains very similar for Laplace noise.

In terms of running time (Table 7), we see that our method is significantly faster compared to the other competitive algorithms CAM and GraN-DAG. Actually, most of the time (95% for $d = 50$) is spent on pruning the final DAG using CAM.

5.2. Real data

We now compare the algorithms on a popular real-world dataset for causal discovery (Sachs et al., 2005) (11 nodes, 17 edges and 853 observations), as well as the pseudo-real dataset sampled from SynTReN generator (Van den Bulcke et al., 2006) (Table 8). We can see that on Sachs, our method matches the SHD of CAM while improving the SID. On the SynTReN datasets, GraN-DAG seems to perform best, but the confidence intervals highly overlap.

6. Conclusion

In this work, we demonstrated a new connection between score matching and causal discovery methods. We found that, in the case of non-linear additive Gaussian noise model, the causal graph can easily be recovered from the score function. In addition to generative models, this provides a new promising application for score estimation techniques. The proposed technique includes two modules: one that evaluates the (Jacobian of the) score, and one that prunes the final DAG given a topological order. Note that any score matching or pruning method can be plugged in to obtain a new practical algorithm.

Future work One focus of this work was to build a fast algorithm for estimating a topological order, while avoiding the combinatorial complexity of searching over permutations, and the use of any heuristic optimization approaches. For this reason, we avoided using popular score

Table 1: Synthetic experiment for $d = 10$ with Gaussian noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{1.1 \pm 0.9}$	$\mathbf{4.5 \pm 5.3}$	$\mathbf{0.4 \pm 0.6}$	$\mathbf{19.5 \pm 2.9}$	$\mathbf{35.0 \pm 9.1}$	$\mathbf{0.3 \pm 0.3}$
CAM	1.7 ± 1.0	6.4 ± 4.2	$\mathbf{0.4 \pm 0.5}$	24.4 ± 3.1	45.2 ± 10.2	4.4 ± 3.2
GraN-DAG	1.5 ± 1.4	6.5 ± 7.2	–	22.2 ± 2.6	42.0 ± 6.2	–
SELF	8.4 ± 1.6	32.5 ± 7.6	–	37.2 ± 2.1	83.0 ± 5.2	–
GES	7.8 ± 2.7	32.5 ± 13.6	–	34.3 ± 3.0	78.9 ± 6.0	–
VarSort	–	–	1.9 ± 1.1	–	–	9.7 ± 3.1

Table 2: Synthetic experiment for $d = 20$ with Gaussian noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{2.6 \pm 1.9}$	$\mathbf{9.9 \pm 8.5}$	1.2 ± 1.7	$\mathbf{47.5 \pm 4.5}$	$\mathbf{177.5 \pm 11.6}$	$\mathbf{3.1 \pm 1.5}$
CAM	3.5 ± 1.6	14.3 ± 9.8	$\mathbf{0.8 \pm 1.0}$	54.2 ± 5.4	201.9 ± 29.0	13.6 ± 6.9
GraN-DAG	7.6 ± 4.2	31.6 ± 22.7	–	49.3 ± 4.5	211.4 ± 36.6	–
SELF	16.6 ± 2.1	89.9 ± 31.2	–	75.5 ± 1.6	336.8 ± 31.2	–
GES	17.7 ± 3.8	77.3 ± 30.5	–	67.4 ± 6.1	322.9 ± 21.7	–
VarSort	–	–	3.7 ± 1.6	–	–	18.3 ± 6.7

Table 3: Synthetic experiment for $d = 50$ with Gaussian noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	10.4 ± 3.9	$\mathbf{50.9 \pm 32.9}$	3.9 ± 2.4	$\mathbf{131.5 \pm 7.5}$	$\mathbf{1262 \pm 110}$	16.3 ± 6.1
CAM	$\mathbf{8.3 \pm 2.9}$	53.7 ± 31.9	–	140.8 ± 5.5	1337 ± 94	–
GraN-DAG	20.2 ± 6.1	135.3 ± 45.9	–	140.8 ± 9.5	1432 ± 110	–
SELF	45.4 ± 3.5	326.6 ± 74.3	–	192.7 ± 3.2	2097 ± 103	–
GES	50.5 ± 4.2	233.5 ± 60.8	–	182.9 ± 7.3	2003 ± 105	–
VarSort	–	–	8.8 ± 3.0	–	–	43.3 ± 9.7

Table 4: Synthetic experiment for $d = 10$ with Laplace noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	1.4 ± 0.8	4.5 ± 4.7	0.8 ± 0.7	$\mathbf{19.6 \pm 2.5}$	$\mathbf{31.9 \pm 7.9}$	$\mathbf{0.2 \pm 0.4}$
CAM	1.5 ± 1.3	6.1 ± 6.5	$\mathbf{0.5 \pm 0.5}$	24.4 ± 1.5	44.4 ± 8.1	1.5 ± 1.6
GraN-DAG	$\mathbf{1.3 \pm 1.4}$	$\mathbf{4.4 \pm 4.9}$	–	20.3 ± 2.7	39.3 ± 13.0	–
SELF	9.7 ± 2.5	33.4 ± 10.8	–	38.2 ± 1.8	86.9 ± 4.3	–
GES	8.9 ± 2.2	28.3 ± 12.0	–	33.7 ± 2.3	78.9 ± 7.4	–
VarSort	–	–	1.6 ± 1.3	–	–	7.2 ± 2.3

Table 5: Synthetic experiment for $d = 20$ with Laplace noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{1.6 \pm 1.2}$	$\mathbf{6.8 \pm 11.4}$	0.5 ± 0.9	$\mathbf{48.0 \pm 4.0}$	$\mathbf{199.8 \pm 21.4}$	$\mathbf{4.9 \pm 1.8}$
CAM	2.3 ± 1.4	10.0 ± 7.0	$\mathbf{0.3 \pm 0.5}$	52.4 ± 3.9	208.7 ± 17.5	11.6 ± 7.9
GraN-DAG	4.9 ± 2.1	27.5 ± 13.2	–	48.2 ± 3.8	198.3 ± 42.8	–
SELF	16.4 ± 3.6	87.5 ± 32.3	–	77.4 ± 2.2	349.5 ± 19.0	–
GES	17.7 ± 6.8	72.6 ± 25.5	–	69.7 ± 7.1	325.5 ± 28.3	–
VarSort	–	–	3.4 ± 2.0	–	–	20.8 ± 4.5

Table 6: Synthetic experiment for $d = 50$ with Laplace noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	11.0 ± 4.5	71.8 ± 50.2	4.0 ± 2.5	$\mathbf{128.1 \pm 7.9}$	1384 ± 131	19.8 ± 3.5
CAM	$\mathbf{10.1 \pm 3.4}$	$\mathbf{66.1 \pm 47.9}$	–	134.6 ± 7.2	$\mathbf{1361 \pm 136}$	–
GraN-DAG	21.9 ± 3.9	165.7 ± 46.2	–	138.3 ± 8.8	1603 ± 166	–
SELF	42.4 ± 2.9	361.4 ± 112.5	–	191.4 ± 2.9	2053 ± 110	–
GES	52.4 ± 7.7	292.2 ± 105.9	–	182.5 ± 7.2	2028 ± 120	–
VarSort	–	–	8.1 ± 4.2	–	–	47.3 ± 8.7

Table 7: Run time (in seconds) comparison of the algorithms on ER1. The first row corresponds to the time spent for finding the topological order in our method. $(*)$ In order to run CAM on 50 nodes within a reasonable time, we had to use preliminary neighbour search while restricting the maximum number of neighbours to 20 (Bühlmann et al., 2014).

	$d = 10$	$d = 20$	$d = 50$
SCORE order	3.3 ± 0.1	8.5 ± 0.8	31 ± 2.9
SCORE	6.3 ± 0.2	32.7 ± 6.7	257 ± 17
CAM	30.1 ± 3.7	313 ± 80	$1143 \pm 79^{(*)}$
GraN-DAG	185 ± 26	357 ± 47	1410 ± 73

Table 8: Comparison of several algorithms on the real world dataset Sachs and 10 datasets sampled from SynTReN.

	Sachs		SynTReN	
	SHD	SID	SHD	SID
SCORE	12	45	36.2 ± 4.7	193.4 ± 60.2
CAM	12	55	40.5 ± 6.8	152.3 ± 48.0
GraN-DAG	13	47	34.0 ± 8.5	161.7 ± 53.4

matching algorithms developed for score-based generative modelling in high-dimensions (Song & Ermon, 2020), since re-training a neural network after each leaf removal would be quite expensive in practice. Amortization (Löwe et al., 2020) is a promising direction to alleviate this issue.

In addition, we would like to further study the application of score matching causal discovery methods to generative model other than additive (Gaussian) noise. Due to the one-to-one correspondence between the score of a distribution and its density function, it should be possible to recover the graph from the score for any identifiable model. The question is hence: How to read the graph from the score function for a given model, and is there a universal way to do it that encapsulates a large class of models?

7. Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n°725594 - time-data). This work was also supported by Hasler Foundation Program: Hasler Responsible AI (project number 21043).

References

Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Barp, A., Briol, F.-X., Duncan, A. B., Girolami, M., and Mackey, L. Minimum stein discrepancy estimators. *arXiv preprint arXiv:1906.08283*, 2019.

Bouckaert, R. R. Optimizing causal orderings for generating dags from data. In *Uncertainty in Artificial Intelligence*, pp. 9–16. Elsevier, 1992.

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

Cai, R., Qiao, J., Zhang, Z., and Hao, Z. Self: structural equational likelihood framework for causal discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Chickering, D. M. Learning bayesian networks is np-complete. In *Learning from data*, pp. 121–130. Springer, 1996.

Chickering, D. M. Optimal structure identification with greedy search. *Journal of machine learning research*, 3 (Nov):507–554, 2002.

Cooper, G. F. and Herskovits, E. A bayesian method for the induction of probabilistic networks from data. *Machine learning*, 9(4):309–347, 1992.

Diaconis, P., Stein, C., Holmes, S., and Reinert, G. Use of exchangeable pairs in the analysis of simulations.

In *Stein's Method*, pp. 1–25. Institute of Mathematical Statistics, 2004.

Erdös, P. and Rényi, A. *On the evolution of random graphs*, pp. 38–82. Princeton University Press, 2011. doi: doi:10.1515/9781400841356.38. URL https://doi.org/10.1515/9781400841356.38.

Garreau, D., Jitkrittum, W., and Kanagawa, M. Large sample analysis of the median heuristic. *arXiv preprint arXiv:1707.07269*, 2017.

Ghoshal, A. and Honorio, J. Learning linear structural equation models in polynomial time and sample complexity. In *International Conference on Artificial Intelligence and Statistics*, pp. 1466–1475. PMLR, 2018.

Hyvärinen, A. and Dayan, P. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-based neural dag learning. *arXiv preprint arXiv:1906.02226*, 2019.

Larranaga, P., Kuijpers, C. M., Murga, R. H., and Yurramendi, Y. Learning bayesian network structures by searching for the best ordering with genetic algorithms. *IEEE transactions on systems, man, and cybernetics-part A: systems and humans*, 26(4):487–493, 1996.

Li, Y. and Turner, R. E. Gradient estimators for implicit models. *arXiv preprint arXiv:1705.07107*, 2017.

Liu, Q., Lee, J., and Jordan, M. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pp. 276–284. PMLR, 2016.

Loh, P.-L. and Bühlmann, P. High-dimensional learning of linear causal networks via inverse covariance estimation. *The Journal of Machine Learning Research*, 15(1): 3065–3105, 2014.

Löwe, S., Madras, D., Zemel, R., and Welling, M. Amortized causal discovery: Learning to infer causal graphs from time-series data. *arXiv preprint arXiv:2006.10833*, 2020.

Marra, G. and Wood, S. N. Practical variable selection for generalized additive models. *Computational Statistics & Data Analysis*, 55(7):2372–2387, 2011.

Peters, J. and Bühlmann, P. Structural intervention distance for evaluating causal graphs. *Neural computation*, 27(3): 771–799, 2015.

Peters, J., Mooij, J. M., Janzing, D., and Schölkopf, B. Causal discovery with continuous additive noise models. 2014.

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Raskutti, G. and Uhler, C. Learning directed acyclic graph models based on sparsest permutations. *Stat*, 7(1):e183, 2018.

Reisach, A. G., Seiler, C., and Weichwald, S. Beware of the simulated dag! varsortability in additive noise models. *NeurIPS*, 2021.

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.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proceedings of the IEEE*, 109 (5):612–634, 2021.

Si, S., Hsieh, C.-J., and Dhillon, I. Memory efficient kernel approximation. In *International Conference on Machine Learning*, pp. 701–709. PMLR, 2014.

Singh, M. and Valtorta, M. An algorithm for the construction of bayesian network structures from data. In *Uncertainty in Artificial Intelligence*, pp. 259–265. Elsevier, 1993.

Solus, L., Wang, Y., and Uhler, C. Consistency guarantees for greedy permutation-based causal inference algorithms. *Biometrika*, 108(4):795–814, 2021.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pp. 11918–11930, 2019.

Song, Y. and Ermon, S. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Uncertainty in Artificial Intelligence*, pp. 574–584. PMLR, 2020a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, prediction, and search*. MIT press, 2000.

Stein, C. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, volume 6, pp. 583–603. University of California Press, 1972.

Strassen, V. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.

Teyssier, M. and Koller, D. Ordering-based search: A simple and effective algorithm for learning bayesian networks. *arXiv preprint arXiv:1207.1429*, 2012.

Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B., and Marchal, K. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics*, 7(1):1–12, 2006.

Wang, X., Du, Y., Zhu, S., Ke, L., Chen, Z., Hao, J., and Wang, J. Ordering-based causal discovery with reinforcement learning. *arXiv preprint arXiv:2105.06631*, 2021.

Wilks, S. S. Mathematical statistics, 1962.

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.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. Dags with no tears: Continuous optimization for structure learning. *arXiv preprint arXiv:1803.01422*, 2018.

Zhu, J. Hessian estimation via stein's identity in black-box problems. *arXiv preprint arXiv:2104.01317*, 2021.

Zhu, S., Ng, I., and Chen, Z. Causal discovery with reinforcement learning. *arXiv preprint arXiv:1906.04477*, 2019.

Zimmermann, R. S., Schott, L., Song, Y., Dunn, B. A., and Klindt, D. A. Score-based generative classifiers. *arXiv preprint arXiv:2110.00473*, 2021.

A. Additional Experiments

We show here additional synthetic experiments. Tables 9, 10 and 11 show the result for additive noise model with Gumbel noise on Erdös-Renyi graphs. Tables 12, 13 and 13 show the results for Gaussian noise with Scale-free graphs.

Table 9: Synthetic experiment for $d = 10$ with Gumbel noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{1.1 \pm 1.2}$	$\mathbf{4.5 \pm 5.0}$	$\mathbf{0.4 \pm 0.5}$	$\mathbf{21.7 \pm 2.9}$	$\mathbf{35.3 \pm 7.4}$	$\mathbf{0.3 \pm 0.4}$
CAM	2.0 ± 1.5	6.1 ± 5.8	1.6 ± 0.8	27.2 ± 1.8	48.9 ± 9.0	3.8 ± 2.5
GraN-DAG	2.1 ± 1.9	9.7 ± 10.4		22.9 ± 3.2	43.2 ± 11.7	
SELF	8.8 ± 2.7	37.0 ± 8.9	–	38.9 ± 1.2	85.9 ± 5.0	–
GES	7.6 ± 2.4	29.6 ± 11.5	–	34.9 ± 3.5	81.9 ± 5.3	–
VarSort			1.9 ± 0.8			8.2 ± 3.0

Table 10: Synthetic experiment for $d = 20$ with Gumbel noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{3.3 \pm 2.6}$	$\mathbf{12.0 \pm 11.5}$	$\mathbf{0.7 \pm 0.9}$	$\mathbf{52.9 \pm 4.4}$	$\mathbf{205.5 \pm 35.5}$	$\mathbf{5.1 \pm 1.6}$
CAM	5.8 ± 1.5	24.6 ± 13.0	3.0 ± 2.0	57.1 ± 4.2	230.0 ± 39.3	10.7 ± 5.8
GraN-DAG	7.4 ± 2.5	29.2 ± 11.3		54.9 ± 4.3	239.5 ± 43.6	
SELF	19.2 ± 2.1	96.2 ± 27.9	–	77.7 ± 1.4	342.9 ± 15.2	–
GES	19.0 ± 3.9	84.0 ± 32.7	–	72.7 ± 4.2	323.2 ± 28.5	–
VarSort			3.8 ± 1.7			20.8 ± 6.6

Table 11: Synthetic experiment for $d = 50$ with Gumbel noise

	ER1			ER4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{11.3 \pm 4.6}$	$\mathbf{68.2 \pm 45.1}$	4.1 ± 2.5	$\mathbf{132.6 \pm 8.0}$	1390 ± 132	$\mathbf{19.7 \pm 3.4}$
CAM	$\mathbf{11.0 \pm 3.7}$	69.7 ± 48.8	–	141.1 ± 6.7	$\mathbf{1350 \pm 137}$	–
GraN-DAG	22.5 ± 4.2	167.1 ± 47.3	–	139.9 ± 7.0	1552 ± 143	–
SELF	46.3 ± 3.7	306.5 ± 41.1	–	193.3 ± 3.1	2100 ± 102	–
GES	51.0 ± 5.1	273.0 ± 57.9	–	182.1 ± 3.2	2012 ± 105	–
VarSort	–	–	8.8 ± 1.6	–	–	45.5 ± 8.0

Table 12: Synthetic experiment for $d = 10$ with Gaussian noise on scale free graphs

	SF1			SF4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{0.3 \pm 0.6}$	$\mathbf{2.7 \pm 5.8}$	$\mathbf{0.1 \pm 0.3}$	$\mathbf{4.6 \pm 1.7}$	$\mathbf{21.5 \pm 9.6}$	$\mathbf{0.5 \pm 0.9}$
CAM	0.4 ± 0.5	2.8 ± 3.6	0.3 ± 0.3	9.6 ± 2.0	40.4 ± 11.4	4.1 ± 1.6
GraN-DAG	1.4 ± 1.0	12.5 ± 9.7	–	4.7 ± 1.8	23.0 ± 7.3	–
SELF	10.4 ± 2.7	60.2 ± 16.2	–	26.8 ± 1.4	84.6 ± 3.6	–
GES	12.5 ± 3.3	57.2 ± 15.2	–	22.7 ± 4.1	76.6 ± 7.2	–
VarSort	–	–	2.8 ± 1.7	–	–	7.0 ± 3.2

Table 13: Synthetic experiment for $d = 20$ with Gaussian noise on scale free graphs

	SF1			SF4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	$\mathbf{0.9 \pm 0.9}$	13.8 ± 12.6	0.7 ± 0.6	17.5 ± 3.5	179.2 ± 23.8	$\mathbf{3.6 \pm 1.4}$
CAM	$\mathbf{0.9 \pm 0.9}$	$\mathbf{12.9 \pm 14.0}$	$\mathbf{0.5 \pm 0.4}$	26.4 ± 3.9	253.7 ± 28.8	4.6 ± 3.2
GraN-DAG	3.2 ± 1.9	25.5 ± 15.6	–	$\mathbf{14.7 \pm 4.0}$	$\mathbf{168.0 \pm 39.2}$	–
SELF	18.9 ± 2.9	245.7 ± 28.2	–	65.9 ± 2.6	369.2 ± 7.8	–
GES	23.6 ± 3.6	166.4 ± 47.6	–	60.0 ± 4.0	345.7 ± 11.4	–
VarSort	–	–	7.4 ± 2.5	–	–	20.2 ± 7.2

Table 14: Synthetic experiment for $d = 50$ with Gaussian noise on scale free graphs

	SF1			SF4		
	SHD	SID	$D_{top}(\pi, A)$	SHD	SID	$D_{top}(\pi, A)$
SCORE (ours)	4.6 ± 2.4	132.6 ± 75.8	4.0 ± 1.0	68.3 ± 3.6	1724 ± 109	21.8 ± 5.0
CAM	$\mathbf{3.6 \pm 1.9}$	$\mathbf{115.4 \pm 72.6}$	–	85.3 ± 4.2	1935 ± 99	–
GraN-DAG	9.2 ± 3.3	281.8 ± 129.8	–	$\mathbf{63.8 \pm 9.7}$	$\mathbf{1677 \pm 118}$	–
SELF	57.6 ± 6.6	1780 ± 150	–	176.0 ± 4.0	2424 ± 16	–
GES	81.3 ± 8.8	1049 ± 174	–	167.6 ± 9.2	2289 ± 49	–
VarSort	–	–	21.0 ± 4.0	–	–	73.0 ± 10.6