LEARNING FUNCTIONAL CAUSAL MODELS WITH GENERATIVE NEURAL NETWORKS

Olivier Goudet[*†], **Diviyan Kalainathan**[*†],
Philippe Caillou[†], **David Lopez-Paz**[‡], **Isabelle Guyon**[†],
Michèle Sebag[†], **Aris Tritas**[†], **Paola Tubaro**[†]

ABSTRACT

We introduce a new approach to functional causal modeling from observational data. The approach, called *Causal Generative Neural Networks* (CGNN), leverages the power of neural networks to learn a generative model of the joint distribution of the observed variables, by minimizing the Maximum Mean Discrepancy between generated and observed data. An approximate learning criterion is proposed to scale the computational cost of the approach to linear complexity in the number of observations.

The performance of CGNN is studied throughout three experiments. First, we apply CGNN to the problem of cause-effect inference, where two CGNNs model $P(Y|X, \text{noise})$ and $P(X|Y, \text{noise})$ identify the best causal hypothesis out of "$X \to Y$" and "$Y \to X$". Second, CGNN is applied to the problem of identifying v-structures and conditional independences. Third, we apply CGNN to problem of multivariate functional causal modeling: given a skeleton describing the dependences in a set of random variables $\{X_1, \ldots, X_d\}$, CGNN orients the edges in the skeleton to uncover the directed acyclic causal graph describing the causal structure of the random variables.

On all three tasks, CGNN is extensively assessed on both artificial and real-world data, comparing favorably to the state-of-the-art. Finally, we extend CGNN to handle the case of confounders, where latent variables are involved in the overall causal model.

Keywords generative neural networks · causal structure discovery · cause-effect pair problem · functional causal models · structural equation models · structural causal models · causal model search · maximum mean discrepancy

1 INTRODUCTION

Functional Causal Models (FCMs), also known as Structural Equation Models (SEMs) or Structural Causal Models (SCMs) have recently gained a lot of attention in the machine learning literature (Pearl, 2009; Spirtes and Zhang, 2016). In parallel, much excitement has been surrounding the giant steps made by the deep learning community in predictive modeling. Yet, the causal modeling community has only made few steps towards embracing such emerging techniques (Lopez-Paz et al., 2016; Rojas-Carulla et al., 2017). Reciprocally, deep learning publications focus for the most part on predictive models, having little explanatory power of the data generating mechanism. The present work is a step towards merging both approaches. By combining **generative** neural networks, we build (eventually deep) neural network structures modeling the joint distributions of (eventually large) systems of variables, inferring a plausible causal structure.

[*]Equal contribution
[†]TAU, CNRS − INRIA − LRI, Univ. Paris-Sud, Université Paris-Saclay, France · {firstname.name}@lri.fr
[‡]Facebook AI Research (FAIR), Paris, France · dlp@fb.com

In the considered setting, *observational* data, drawn *i.i.d.* from an unknown joint distribution, are given as a set of n samples of a random vector $\mathbf{X} = [X_1, ..., X_d]$ on $\mathcal{R}^d$. The task is to find a functional causal model that represents well the underlying data-generating mechanism, meant as: under relevant manipulations/interventions/experiments the FCM-generated data follows the same distribution as the observational data would under same manipulations/interventions/experiments. More precisely, let us consider an intervention denoted as *do(X=x)* that forces variable X to take value x, while the rest of the system remains unchanged (Pearl, 2009). Variable X_i is considered to be **a direct cause** of X_j with respect to $X_1, ..., X_d$ if:

$$P_{X_j|\text{do}(X_i=x,\mathbf{X}_{\setminus ij}=\mathbf{c})} \neq P_{X_j|\text{do}(X_i=x',\mathbf{X}_{\setminus ij}=\mathbf{c})} \tag{1}$$

for some value pairs x, x', some vector $\mathbf{c}$ in $\mathcal{R}^{d-2}$, where $\mathbf{X}_{\setminus ij}$ denotes the set of all variables except X_i and X_j. With same notations, $P_{X_j|\text{do}(X_i=x,\mathbf{X}_{\setminus ij}=c)}$ denotes in the following the X_j distribution under interventions *do(X_i=x)* (X_i is forced to take value x) and *do($X_{\setminus ij} = c$)* (all other variables are kept to a fixed vector value c) (Mooij et al., 2016).

As opposed to mainstream causal modelling approaches that rely on (often impossible, expensive, or unethical) interventions, the proposed approach aims to identify the causal structure based on observational data only, restricted to non-temporal causal modeling (data series are excluded). The validation of the approach follows the empirical supervised learning methodology, learning causal models from observational data and measuring the model accuracy by comparison with the ground truth model structures.

The main contribution of the paper is the *Causal Generative Neural Network* framework[1] (CGNN), that unifies several state-of-art approaches with complementary strengths: i) algorithms exploiting Markov properties of (conditional) variable independence (Spirtes et al., 2000; Tsamardinos et al., 2006; Pearl, 2009); and ii) algorithms exploiting the asymmetries in the joint distribution of cause-effect pairs (Hoyer et al., 2009; Stegle et al., 2010; Daniusis et al., 2012; Mooij et al., 2016). CGNN seeks the functional causal model (FCM) as a generative neural net (neural architecture and weights) that leverages the representational power of generative neural networks to reproduce the observational data in a sense made precise in the following. The contribution of the paper includes: i) an effective loss criterion and algorithm for CGNN learning in the 2-variable and 3-variable cases; ii) its extension to the full multiple variable case, considering a graph skeleton ; iii) a fast approximation of the algorithm, with linear complexity in the dataset size; iv) the approach extension to handle (unknown) confounders. All contributions are extensively empirically validated on thoroughly studied benchmarks and successfully compared with the state of the art.

The paper is organized as follows: Section 2 defines the FCM learning problem and discusses the underlying assumptions. Section 3 reviews the state of the art. A comprehensive overview of CGNN is given in Section 4. Section 5 reports extensive experimentations and comparisons with state-of-art method, conducted to assess the proposed approach, regarding pairwise cause-effect inference and graph recovery. Section 6 addresses causal modelling with confounders, and the paper concludes with some perspectives for further research.

2 PROBLEM SETTING

Formally, FCMs of $\mathbf{X} = [X_1, ..., X_d]$ are described by a set of equations of the form:

$$X_i = f_i(X_{\text{Pa}(i;\mathcal{G})}, E_i), \forall i \in [\![1, n]\!] \tag{2}$$

where function f_i represents a deterministic causal mechanism and consequence X_i is expressed as a function of its direct causes $X_{\text{Pa}(i;\mathcal{G})}$ (set of variables excluding X_i, $X_{\text{Pa}(i;\mathcal{G})} \subset \mathbf{X}_{\setminus i}$) and a stochastic noise variable E_i modeling all external influences on X_i.

[1]For the sake of reproducibility, the CGNN code is available at `https://github.com/GoudetOlivier/CGNN`.

Letting $\mathcal{G}$ denote the causal graph obtained by drawing arrows from causes $X_{\mathrm{Pa}(i;\mathcal{G})}$ towards their effects X_i, we restrict ourselves to directed acyclic graphs (DAG), where the propagation of interventions to end nodes is assumed to be instantaneous. This assumption suitably represents causal phenomena in cross-sectional studies. An example of functional causal model with five variables is illustrated on Fig. 1.

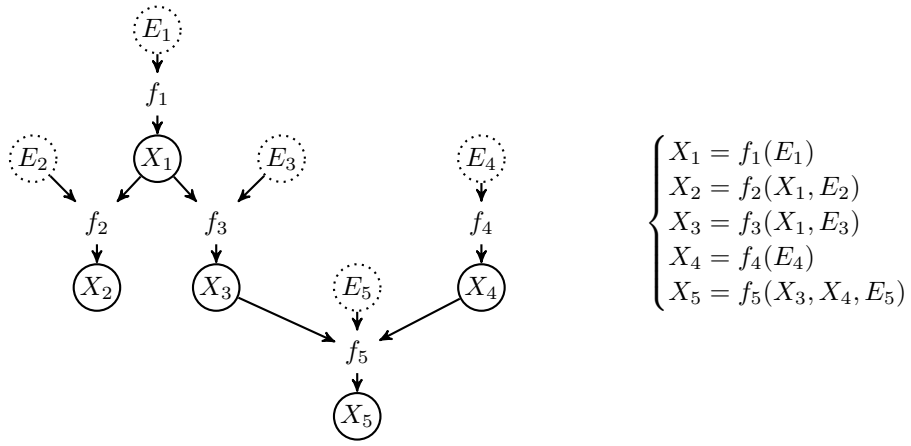


Figure 1: The graph $\mathcal{G}$ of a Functional Causal Model (FCM) on $\mathbf{X} = [X_1, \ldots, X_5]$.

Notations. By abuse of notation and for simplicity, a variable X and the associated node in the causal graph, in one-to-one correspondance, are noted in the same way. Variables X and Y are adjacent iff there exists an edge between both nodes in the graph. This edge can model i) a direct causal relationship ($X \to Y$ or $Y \to X$); ii) a causal relationship in either direction ($X - Y$); iii) a non-causal association ($X \leftrightarrow Y$) due to external common causes (Richardson and Spirtes, 2002).
Conditional independence: $(X \perp\!\!\!\perp Y|Z)$ is meant as variables X and Y are independent conditionally to Z, i.e. $P(X, Y|Z) = P(X|Z)P(Y|Z)$.
V-structure or unshielded collider: Three variables $\{X, Y, Z\}$ form a v-structure iff their causal structure is: $X \to Z \leftarrow Y$.
Skeleton of the DAG: the skeleton of the DAG is the undirected graph obtained by replacing all edges by undirected edges.
Markov equivalent DAG: two DAGs with same skeleton and same v-structures are said to be *Markov equivalent* (Pearl and Verma, 1991). A *Markov equivalence class* is represented by a *Completed Partially Directed Acyclic Graph* (CPDAG) having both directed and undirected edges.

Assumptions. The state of the art in causal modeling most commonly involves the following assumptions:
Causal sufficiency assumption (CSA): $\mathbf{X}$ is said to be *causally sufficient* if no pair of variables $\{X_i, X_j\}$ in $\mathbf{X}$ has a common cause external to $\mathbf{X}_{\setminus i,j}$.
Causal Markov assumption (CMA): all variables are independent of their non-effects (non descendants in the causal graph) conditionally to their direct causes (parents) (Spirtes et al., 2000). For a FCM, this assumption holds if the graph is a DAG and error terms E_i in the FCM are independent on each other (Pearl, 2009).
Conditional independence relations in a FCM: if CMA applies, the data generated by the FCM satisfy all conditional independence (CI) relations among variables in $\mathbf{X}$ via the notion of d-separation (Pearl, 2009). CIs are called Markov properties. Note that there may be more CIs in data than present in the graph (see the Faithfulness assumption below). The joint distribution of the variables is expressed as the product of the distribution of each variable conditionally on its parents in the graph.
Causal Faithfulness Assumption (CFA): the joint distribution $P(\mathbf{X})$ is *faithful* to the graph $\mathcal{G}$ of a FCM

iff every conditional independence relation that holds true in P is entailed by $\mathcal{G}$ (Spirtes and Zhang, 2016). Therefore, if there exists an independence relation in $\mathbf{X}$ that is not a consequence of the Causal Markov condition, then $\mathbf{X}$ is *unfaithful* (Scheines, 1997). It follows from CMA and CFA that every causal path in the graph corresponds to a dependency between variables, and vice versa.

V-structure property. Under CSA, CMA and CFA, if variables $\{X, Y, Z\}$ satisfy: i) $\{X, Y\}$ and $\{Y, Z\}$ are adjacent; ii) $\{X, Z\}$ are NOT adjacent; iii) $X \not\perp\!\!\!\perp Z|Y$, then their causal structure is a v-structure ($X \to Y \leftarrow Z$).

3 State of the art

This section reviews methods to infer causal relationships, based on either the Markov properties of a DAG such as v-structures or colliders, or asymmetries in the joint distributions of pairs of variables.

3.1 Learning the CPDAG

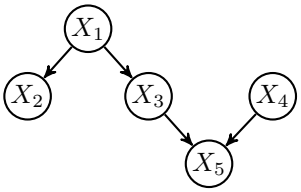
Structure learning methods classically use conditional independence (CI) relations in order to identify the Markov equivalence class of the sought Directed Acyclic Graph, referred to as CPDAG, under CSA, CMA and CFA.

Let us consider for instance the functional model on $\mathbf{X} = [X_1, \ldots, X_5]$ depicted on Fig. 1. The associated DAG $\mathcal{G}$ and graph skeleton are respectively depicted on Fig. 2(a) and (b). Causal modeling exploits observational data to recover the $\mathcal{G}$ structure from all CI (Markov properties) between variables[2]. Under CSA, CMA and CFA, as $(X_3 \perp\!\!\!\perp X_4|X_5)$ does not hold, a v-structure $X_3 \to X_5 \leftarrow X_4$ is identified (Fig. 2(c)). However, one also has $(X_1 \perp\!\!\!\perp X_5|X_3)$ and $(X_2 \perp\!\!\!\perp X_3|X_1)$. Thus the DAGs on Figs. 2(d) and (e) encode the same conditional independences as the true DAG (Fig. 2(a)). Therefore the true DAG cannot be fully identified based only on independence tests, and the edges between the pairs of nodes $\{X_1, X_2\}$ and $\{X_1, X_3\}$ are left undirected. The identification process thus yields the partially undirected graph depicted on Fig. 2(c), called *Completed Partially Directed Acyclic Graph* (CPDAG).

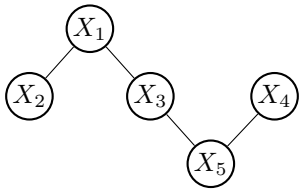
The main three families of methods used to recover the CPDAG of a FCM with continuous data are constraint-based methods, score-based methods, and hybrid methods (Drton and Maathuis, 2016).

Constraint-based methods exploit conditional independences between variables to identify all v-structures. One of the most well-known constraint-based algorithms is the PC algorithm (Spirtes et al., 2000). PC first builds the DAG skeleton based on conditional independences among variables and subsets of variables. Secondly, it identifies v-structures (Fig. 2(c)). Finally, it uses propagation rules to orient remaining edges, avoiding the creation of directed cycles or new v-structures. Under CSA, CMA and CFA, and assuming an oracle indicating all conditional independences, PC returns the CPDAG of the functional causal model. In practice, PC uses statistical tests to accept or reject conditional independence at a given confidence level. Besides mainstream tests (e.g., s Z-test or T-Test for continuous Gaussian variables, and chi-squared or G-test for categorical variables), non-parametric independence tests based on machine learning are becoming increasingly popular, such as kernel-based conditional independence tests (Zhang et al., 2012). The FCI algorithm (Spirtes et al., 1999) extends PC; it relaxes the *causal sufficiency* assumption and deals with latent variables. The RFCI algorithm (Colombo et al., 2012) is faster than FCI and handles high-dimensional DAGs with latent variables. Achilles' heel of constraint-based algorithms is their reliance on conditional independence tests. The CI accuracy depends on the amount of available data, with exponentially increasing

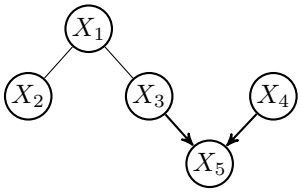
[2]The so-called constraint-based methods base the recovery of graph structure on conditional independence tests. All proofs of model identifiability assume the existence of an "oracle" providing perfect knowledge of the CIs, i.e. *de facto* assuming an infinite amount of training data.



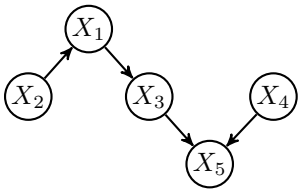
(a) The exact DAG of $\mathcal{G}$.



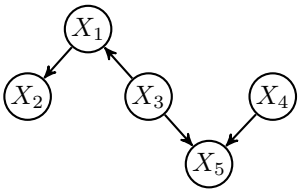
(b) The skeleton of $\mathcal{G}$.



(c) The CPDAG of $\mathcal{G}$.



(d) A Markov equivalent DAG of $\mathcal{G}$.



(e) Another Markov equivalent DAG of $\mathcal{G}$.

Figure 2: Example of a Markov equivalent class. There exists three graphs (a, d, e) consistent with a given graph skeleton (b); the set of these consistent graphs defines the Markov equivalent class (c).

size with the number of variables. Additionally, the use of propagation rules to direct edges is prone to error propagation.

Score-based methods explore the space of CPDAGs and minimize a global score. For example, the space of graph structures is explored using operators (*add edge*, *remove edge*, and *reverse edge*) by the Greedy Equivalent Search (GES) algorithm (Chickering, 2002), returning the optimal structure in the sense of the Bayesian Information Criterion[3].

In order to find the optimal CPDAG corresponding to the minimum score, the GES algorithm starts with an empty graph. A first forward phase is performed, iteratively adding edges to the model in order to improve the global score. A second backward phase iteratively removes edges to improve the score. Under CSA, CMA and CFA, GES identifies the true CPDAG in the large sample limit, if the score used is decomposable, score-equivalent and consistent (Chickering, 2002). More recently, Ramsey (2015) proposed a GES extension called Fast Greedy Equivalence Search (FGES) algorithm. FGES uses the same scores and search algorithm with different data structures; it greatly speeds up GES by caching information about scores during each phase of the process.

Hybrid algorithms combine ideas from constraint-based and score-based algorithms. According to Nandy et al. (2015), such methods often use a greedy search like the GES method on a restricted search space for the sake of computational efficiency. This restricted space is defined using conditional independence tests. For instance the Max-Min Hill climbing (MMHC) algorithm (Tsamardinos et al., 2006) firstly builds the skeleton of a Bayesian network using conditional independence tests and then performs a Bayesian-scoring greedy hill-climbing search to orient the edges. The Greedy Fast Causal Inference (GFCI) algorithm proceeds in

[3]After Ramsey (2015), in the linear Gaussian case the individual BIC score to minimize for a variable X given its parents is up to a constant $n \ln(s) + c\, k \ln(n)$, where $n \ln(s)$ is the likelihood term, with s the residual variance after regressing X onto its parents, and n the number of data samples. $c\, k \ln(n)$ is a penalty term for the complexity of the graph (here the number of edges). $k = 2p + 1$, with p the total number of parents of the variable X in the graph. $c = 2$ by default, chosen empirically. The global score minimized by the algorithm is the sum over all variables of the individual BIC score given the parent variables in the graph.

the other way around, using FGES to get rapidly a first sketch of the graph (shown to be more accurate than those obtained with constraint-based methods), then using the FCI constraint-based rules to orient the edges in presence of potential confounders (Ogarrio et al., 2016).

3.2 Exploiting asymmetry between cause and effect

The abovementioned score-based and constraint-based methods do not take into account the full information from the observational data (Spirtes and Zhang, 2016), such as data asymmetries induced by the causal directions.

The intuition. Let us consider a FCM $Y = X + E$, with E a random noise independent of X by construction. Graph constraints cannot orient the $X - Y$ edge as both graphs $X \to Y$ and $Y \to X$ are Markov equivalent. However, the implicit v-structure $X \to Y \leftarrow E$ can be exploited provided that either X or E does not follow a **Gaussian distribution**. Consider the linear regression $Y = aX + b$ (blue curve in Fig. 3); the residual is independent of X. Quite the contrary, the residual of the linear regression $X = a'X + b'$ (red curve in Fig. 3) is *not* independent of Y as far as the independence of the error term holds true (Shimizu et al., 2006). In this toy example, the asymmetries in the joint distribution of X and Y can be exploited to recover the causal direction $X \to Y$ (Spirtes and Zhang, 2016).

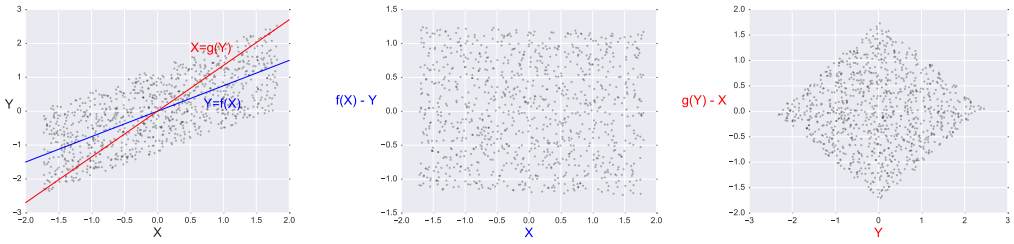


Figure 3: Left: Joint distribution $P(X, Y)$ generated from DAG $X \to Y$. The linear regression of Y on X (respectively of X on Y) is depicted as a blue (resp. red) curve. Middle: Error $f(X) - Y$ is independent of X. Right: Error $f(Y) - X$ is not independent of Y. The asymmetry establishes that the true causal model is $X \to Y$. Better seen in color.

Causal inference is bound to rely on assumptions such as non-Gaussianity or additive noise. In the absence of any such assumption, Zhang et al. (2016) show that, even in the bivariate case, for any function f and latent variable E independent of X such that $Y = f(X, E)$, it is always feasible to construct some $\tilde{f}$ and $\tilde{E}$, with $\tilde{E}$ independent of X, such that $X = \tilde{f}(Y, \tilde{E})$. An alternative, supporting asymmetry detection and hinting at a causal direction, is based on restricting the class of functions f (e.g. only considering regular functions). According to Quinn et al. (2011), the first approach in this direction is LiNGAM Shimizu et al. (2006). LiNGAM handles linear structural equation models, where each variable is continuous and is modeled by:

$$X_i = \sum_k \alpha_k P_a^k(X_i) + E_i, i \in [\![1, n]\!] \tag{3}$$

with $P_a^k(X_i)$ the k-th parent of X_i and α_k a real value. Assuming further that all probability distributions of source nodes in the causal graph are non-Gaussian, Shimizu et al. (2006) show that the causal structure is fully identifiable (all edges can be oriented).

Pairwise methods. In the continuous, non-linear bivariate case, specific methods have been developed to orient the variable edge[4]. A well known example of bivariate model is the additive noise model (ANM) (Hoyer et al., 2009), with data generative model $Y = f(X) + E$, f a (possibly non-linear) function and E a noise independent of X. The authors prove the identifiability of the ANM in the following sense: if $P(X, Y)$ is consistent with ANM $Y = f(X) + E$, then i) there exists no AMN $X = g(Y) + E'$ consistent with $P(X, Y)$; ii) the true causal direction is $X \to Y$. Under the independence assumption between E and X, the ANM admits a single non-identifiable case, the linear model with Gaussian input and Gaussian noise (Mooij et al., 2016).

A more general model is the post-nonlinear model (PNL) (Zhang and Hyvärinen, 2009), involving an additional nonlinear function on the top of an additive noise: $Y = g(f(X) + E)$, with g an invertible function. The price to pay for this higher generality is an increase in the number of non identifiable cases.

The Gaussian Process Inference model (GPI) (Stegle et al., 2010) infers the causal direction without explicitly restricting the class of possible causal mechanisms. The authors build two Bayesian generative models, one for $X \to Y$ and one for $Y \to X$, where the distribution of the cause is modeled with a Gaussian mixture model, and the causal mechanism f is a Gaussian process. The causal direction is determined from the generative model best fitting the data (maximizing the data likelihood). Identifiability here follows from restricting the underlying class of functions and enforcing their smoothness (regularity). Other causal inference methods (Sgouritsa et al., 2015) are based on the idea that if $X \to Y$, the marginal probability distribution of the cause $P(X)$ is independent of the causal mechanism $P(Y|X)$, hence estimating $P(Y|X)$ from $P(X)$ should hardly be possible, while estimating $P(X|Y)$ based on $P(Y)$ may be possible. The reader is referred to Statnikov et al. (2012) and Mooij et al. (2016) for a thorough review and benchmark of the pairwise methods in the bivariate case.

A new ML-based approach tackles causal inference as a pattern recognition problem. This setting was introduced in the Causality challenges (Guyon, 2013; 2014), which released 16,200 pairs of variables $\{X_i, Y_i\}$, each pair being described by a sample of their joint distribution, and labeled with the true ℓ_i value of their causal relationship, with ℓ_i ranging in $\{$ $X_i \to Y_i$, $Y_i \to X_i$, $X_i \perp\!\!\!\perp Y_i$, $X_i \leftrightarrow Y_i$ (presence of a confounder) $\}$. The causality classifiers trained from the challenge pairs yield encouraging results on test pairs. The limitation of this ML-based causal modeling approach is that causality classifiers intrinsically depend on the representativity of the training pairs, assumed to be drawn from a same "Mother distribution" (Lopez-Paz et al., 2015).

Note that bivariate methods can be used to uncover the full DAG, and independently orient each edge, with the advantage that an error on one edge does not propagate to the rest of the graph (as opposed to constraint and score-based methods). However, bivariate methods do not leverage the full information available in the dependence relations. For example in the linear Gaussian case (linear model and Gaussian distributed inputs and noises), if a triplet of variables $\{A, B, C\}$ is such that A, B (respectively B, C) are dependent on each other but $A \perp\!\!\!\perp C$), a constraint-based method would identify the v-structure $A \to B \leftarrow C$ (unshielded collider); still, a bivariate model based on cause-effect asymmetry would not identify $A \to B$ nor $B \leftarrow C$.

3.3 Discussion

This brief survey has shown the complementarity of CPDAG and pairwise methods. The former ones can at best return partially directed graphs; the latter ones do not optimally exploit the interactions between all variables. This paper aims to propose a unified framework getting the best out of both worlds of CPDAG and bivariate approaches.

[4]These methods can be extended to the multi-variate case and used for causal graph identification by orienting each edge in its turn.

An inspiration for this framework is the extension of the bivariate post-nonlinear model (PNL) (Zhang and Hyvärinen, 2009), where an FCM is trained for any plausible causal structure, and each model is tested *a posteriori* for the required independence between errors and causes. The main PNL limitation is its super-exponential cost with the number of variables (Zhang and Hyvärinen, 2009). Another hybrid approach uses a constraint based algorithm to identify a Markov equivalence class, and thereafter uses bivariate modelling to orient the remaining edges (Zhang and Hyvärinen, 2009). For example, the constraint-based PC algorithm can identify the v-structure $X_3 \rightarrow X_5 \leftarrow X_4$ in an FCM (Fig. 2), enabling the bivariate PNL method to further infer the remaining arrows $X_1 \rightarrow X_2$ and $X_1 \rightarrow X_3$. Note that an effective combination of constraint-based and bivariate approaches requires a final verification phase to test the consistency between the v-structures and the edge orientations.

The proposed approach follows a 3-step integrated methodology: i) a graph skeleton is identified based on mainstream Markov properties; ii) edges are oriented using bivariate methods; iii) the full graph optimization (examining local multivariate relationships) is conducted in a score-based fashion.

This approach, called **Causal Generative Neural Network (CGNN)**, features two original contributions. Firstly, local multivariate causal models are learned as **generative neural networks** (as opposed to, regression network). The novelty is to use neural nets to model the distribution $P(X|X_{\mathrm{Pa}(i;\mathcal{G})}, noise)$ of a node conditionally to its parent nodes (causes) and a random variable. This approach does not restrict the class of functions used to represent the causal models (see also (Stegle et al., 2010)), since neural networks are universal approximators. Instead, a regularity argument is used to enforce identifiability, in the spirit of supervised learning: the selected model achieves an optimal trade-off between data fitting and model complexity.
Secondly, the data generative models are trained using a stable criterion, the Maximum Mean Discrepancy (Gretton et al., 2007). This criterion overcomes the limitation of (Lopez-Paz and Oquab, 2016), using conditional generative conditional adversarial networks (Mirza and Osindero, 2014) for bivariate causal modeling.[5]

The CGNN approach features the following distinctive properties:

- Every edge in the *Directed Acyclic Graph* is associated with a confidence score, enabling to leave some edges undirected (non identifiable).
- Multivariate dependencies between inputs and output are explicitly identified, thus addressing some cases violating the faithfulness assumption (e.g., a XOR-based functional model).
- The approach is non-parametric in the sense that it makes assumptions neither on the underlying distribution of observed data X nor on the data-generative model.

Section 4 presents a version of the proposed approach under the usual Markov, faithfulness, and causal sufficiency assumptions. The empirical validation of the approach is detailed in Section 5. In Section 6, the causal sufficiency assumption is relaxed and the model is extended to handle possible hidden confounding factors. Section 7 concludes the paper with some perspectives for future work.

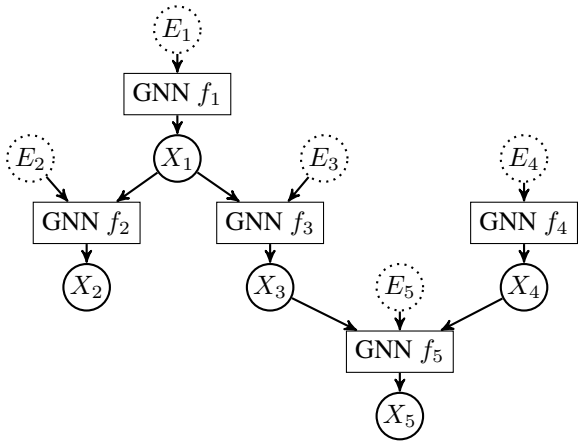
4 Neural Functional Causal Models

The *Causal Generative Neural Network* (CGNN) represents a Functional Causal Model (FCM, Section 2):

$$X_i = f_i(X_{\mathrm{Pa}(i;\mathcal{G})}, E_i),$$

[5] The known instability and high variance of generative adversarial network training makes causal model selection exponentially less reliable as the number of variables increases.

as a global generative neural network (Fig. 4), where each function f_i is modeled by a 1-hidden layer generative neural network (legend "GNN f_i" in Fig. 4). The f_i input includes: i) a real valued vector, sampling the random variable vector $X_{\text{Pa}(i;\mathcal{G})}$; ii) a scalar value, sampling the random noise variable E_i. The f_i output is an X_i sample. f_i thus defines a generative model, implementing the distribution of X_i conditionally to the value of its parent random variables $X_{\text{Pa}(i;\mathcal{G})}$ by varying the instantiation of E_i.



$$\begin{cases} X_1 = f_1(E_1) \\ X_2 = f_2(X_1, E_2) \\ X_3 = f_3(X_1, E_3) \\ X_4 = f_4(E_4) \\ X_5 = f_5(X_3, X_4, E_5) \end{cases}$$

Figure 4: Left: Causal Generative Neural Network over variables $\mathbf{X} = (X_1, \ldots, X_5)$. Right: Corresponding Functional Causal Model equations.

Notations.
• ***Data samples***: $\mathbf{z} = \{[x_{1,j}, \ldots, x_{d,j}]\}_{j=1}^n$, are n samples independent identically distributed after the (unknown) joint distribution $P(\mathbf{X} = [X_1, \ldots, X_d])$, also referred to as observational data.
• ***Noise samples***: $\{[e_{1,j}, \ldots, e_{d,j}]\}_{j=1}^n$, are n samples independent identically distributed from the (mutually independant after the Causal Markov assumption) joint distribution of noise variables $\mathbf{E} = [E_1, \ldots, E_d]$. Each noise variable is assumed to be a centered Gaussian variable[6]; hence $P(\mathbf{E}) = \mathcal{N}(0,1)^d$.
• ***Structure estimate*** $\hat{\mathcal{G}}$: graph associating to each variable X_i its set of parents noted $\text{Pa}(i; \hat{\mathcal{G}})$ for $i \in [[1, d]]$.
• ***Neural network estimates*** $\hat{f} = \{\hat{f}_1, \ldots, \hat{f}_d\}$: set of neural nets modeling a structure estimate.
• ***Estimate samples w.r.t.*** $\hat{f}$: $\hat{\mathbf{z}}_{\hat{f}} = \{[\hat{x}_{1,j}, \ldots, \hat{x}_{d,j}]\}_{j=1}^{n'}$, are n' independent samples iidentically distributed, where each estimate sample $\hat{x}_{i,j}$ of variable X_i is computed from $\hat{f}_i$ with the j-th estimate samples $\hat{x}_{Pa(i;\hat{\mathcal{G}}),j}$ of $X_{\text{Pa}(i;\hat{\mathcal{G}})}$ and the j-th sample $e_{i,j}$ of the random noise variable E_i. For simplicity, subscript $\hat{f}$ is omitted when no confusion is to fear from the context.

The model is "generative" in the sense that any sample $[e_{1,j}, \ldots, e_{d,j}]$ of the "noise" random vector $\mathbf{E} = [E_1, \ldots, E_d]$ can be used as "input" to the network to generate a data sample $[\hat{x}_{1,j}, \ldots, \hat{x}_{d,j}]$.

After these notations, Functional Causal Modeling consists of estimating the graph $\hat{\mathcal{G}}$ and neural networks $\hat{f}$ that best describe the data samples. This task involves two milestones:

[6]With no loss of generality according to Stegle et al. (2010), since the general case of a random variable E_i can be viewed as a function of a centered Gaussian variable $\tilde{E}_i$:

$$E_i = g_i(\tilde{E}_i) \text{ for some function } g_i, \text{ with } \tilde{E}_i \sim \mathcal{N}(0,1)$$

1. **Model evaluation**, associating a score to each candidate solution $(\hat{\mathcal{G}}, \hat{f})$.

2. **Model optimization**, finding a (nearly) optimum solution $(\hat{\mathcal{G}}, \hat{f})$ in the sense of the above score.

Let us define a model scoring function (Section 4.1). The so-called *parametric* optimization of the CGNN, where structure estimate $\hat{\mathcal{G}}$ is fixed and the goal is to find the best neural estimates $\hat{f}$ conditionally to $\hat{\mathcal{G}}$ is tackled in Section 4.2). The *non-parametric* optimization, aimed at finding the best structure estimate, is considered in Section 4.3.

4.1 Model evaluation

The goal is to associate to each candidate solution $(\hat{\mathcal{G}},\ \hat{\mathbf{f}})$ a score reflecting how well this candidate solution describes the observational data. It comes naturally to compare the sample estimate $\hat{\mathbf{z}}$ to the observational data $\mathbf{z}$, using the empirical Maximum Mean Discrepancy (MMD) statistic (Gretton et al., 2007).

The MMD computes the distance between two sample sets by embedding each of them into a reproducing kernel Hilbert space ($\mathcal{RKHS}$) based on some kernel k defined on $\mathcal{R}^d$. Formally, to $\mathbf{z} = (z_1, \ldots, z_n)$ is associated the $\mu_k(\mathbf{z})$ function defined on $\mathcal{R}^d$ with $\mu_k(\mathbf{z})(x) = \frac{1}{n}\sum_{i=1}^{n} k(z_i, x)$. The MMD measures the squared distance between the functions associated to samples $\mathbf{z}$ and $\hat{\mathbf{z}}$ as:

$$\text{MMD}_k(\mathbf{z}, \hat{\mathbf{z}}) = \frac{1}{n^2}\sum_{i,j}^{n} k(z_i, z_j) + \frac{1}{n'^2}\sum_{i,j}^{n'} k(\hat{z}_i, \hat{z}_j) - \frac{2}{n \times n'}\sum_{i=1}^{n}\sum_{j=1}^{n'} k(z_i, \hat{z}_j), \tag{4}$$

In the limit of infinite samples ($n, n' \to \infty$), the MMD statistic goes to 0 iff z and $\hat{z}$ are drawn from the same distribution (Gretton et al., 2007). The sought score function, measuring the quality of the candidate solution $(\hat{\mathcal{G}},\ \hat{\mathbf{f}})$ from the MMD distance between the observational and the estimated samples is noted $\mathcal{L}^{MMD}_{\mathcal{G}}$, where kernel k is the sum of Gaussian kernels ($k(z, \hat{z}) = \exp(-\gamma\|z - \hat{z}\|_2^2)$) with different γ bandwidths. The motivation for this multi-scale kernel is to alleviate the dependency on the sample size (Li et al., 2015). In the experiments, γ varies in $\{0.005, 0.05, 0.25, 0.5, 1, 5, 50\}$.

As $\mathcal{L}^{MMD}_{\mathcal{G}}$ is computed with complexity $\mathcal{O}(nn')$ with n and n' respectively the size of the observational and estimated samples, and for the sake of computational tractability, a linear approximation of the MMD (Lopez-Paz et al., 2015) is also considered. This approximation is based on a Fourier transform of the Gaussian kernel, exploiting Bochner's theorem (Edwards, 1964). Formally, the empirical kernel mean embeddings $\mu_k(\mathbf{z})$ is approximated using m vectors from $span(\{cos(\langle\theta_j, \cdot\rangle + b_j)\}_i^m)$, with $\theta_j \sim \mathcal{N}(0, 2\gamma I_d)$ and I_d the $d \times d$ identity matrix, and $b_j \sim \mathcal{U}[0, 2\pi]$:

$$\mu_k(\mathbf{z})(x) \sim \frac{1}{n}\sum_{i=1}^{n}\frac{1}{m}\sum_{j=1}^{m} 2\cos(\langle\theta_j, z_i\rangle + b_j)\cos(\langle\theta_j, x\rangle + b_j) \tag{5}$$

In the experiments, we choose an approximation with $m = 100$ random vectors. Setting $n = n'$ for simplicity, the approximate score function $\mathcal{L}^{F}_{\mathcal{G}}$ is finally defined as:

$$\mathcal{L}^{\text{F}}_{\hat{\mathcal{G}}}(\mathbf{z}, \hat{\mathbf{z}}) = \frac{4}{m^2 n^2}\sum_{j=1}^{m}\left(\sum_{i=1}^{n}\frac{1}{n_\gamma}\sum_{l=1}^{n_\gamma}\cos(\langle\theta_j^l, z_i\rangle + b_j) - \sum_{i=1}^{n}\frac{1}{n_\gamma}\sum_{l=1}^{n_\gamma}\cos(\langle\theta_j^l, \hat{z}_i\rangle + b_j)\right)^2 \tag{6}$$

In order to compare candidate solutions with different structures in a fair manner, the evaluation score is augmented with a penalization term accounting for the number of edges in the graph. The overall score associated to solution $(\hat{\mathcal{G}},\ \hat{\mathbf{f}})$ is finally defined as:

$$\mathcal{C}_{\hat{\mathcal{G}},\hat{\mathbf{f}}} = \mathcal{L}^{\mathrm{C}}_{\hat{\mathcal{G}}}(\mathbf{z}, \hat{\mathbf{z}}_{\hat{f}}) + \lambda \cdot n(\hat{\mathcal{G}}) \qquad \text{with } C \ = \ \text{MMD or F} \tag{7}$$

with $n(\hat{\mathcal{G}})$ the total number of edges in $\hat{\mathcal{G}}$. Penalization weight λ is a hyper-parameter of the approach. Assuming each $\hat{f}_i$ GNN has n_h hidden neurons, the overall model includes $(2d + n(\hat{\mathcal{G}})) \times n_h$ weights and $d \times (n_h + 1)$ bias parameters.

4.2 Parametric (weight) optimization

Given an acyclic structure estimate $\hat{\mathcal{G}}$, the neural networks $\hat{f}_1, \ldots, \hat{f}_d$ are learned end-to-end using backpropagation with Adam optimizer (Kingma and Ba, 2014) by minimizing losses $\mathcal{L}^{MMD}_{\mathcal{G}}$ (Eq. (4), referred to as **CGNN-MMD**) or $\mathcal{L}^{F}_{\mathcal{G}}$ (Eq. (6), referred to as **CGNN-Fourier**).

The procedure closely follows that of supervised continuous learning (regression), except for the fact that the loss to be minimized is the MMD loss instead of the mean squared error. Neural nets $\hat{f}_i$, $i \in [[1, d]]$ are trained during n_{train} epochs, where the noise samples are iid drawn in each epoch (in the CGNN-Fourier variant, θ^l_j and b_j are likewise resampled from their respective distributions in each training epoch). After training, the score is computed and averaged over n_{eval} estimated samples of size n. Likewise, the noise samples are re-sampled anew (θ^l_j and b_j are likewise resampled anew from their respective distributions) for each evaluation sample. The overall process with training and evaluation is repeated nb_{run} times to reduce stochastic effects.

4.3 Non-parametric (structure) optimization

The number of directed acyclic graphs $\hat{\mathcal{G}}$ over d nodes is super-exponential in d, making the non-parametric optimization of the CGNN structure an intractable computational and statistical problem. An approximate two-phase approach is thus retained, taking inspiration from Tsamardinos et al. (2006); Nandy et al. (2015). The first phase builds an approximate graph skeleton using feature selection methods (Yamada et al., 2014). The second phase focuses on optimizing the edge orientations. Letting L denote the number of edges in the graph, this second phase defines a combinatorial optimization problem of complexity $\mathcal{O}(2^L)$ (note however that not all orientations are admissible since the eventual oriented graph must be a DAG).

The motivation for this 2-phase approach is to decouple the edge selection task and the causal modelling (edge orientation) tasks, and enable their independent assessment.

Any $X_i - X_j$ edge in the graph skeleton stands for a direct dependency between variables X_i and X_j. Given Causal Markov and Faithfulness assumptions, such a direct dependency either reflects a direct causal relationship between the two variables ($X_i \to X_j$ or $X_i \leftarrow X_j$), or is due to the fact that X_i and X_j admit a latent (unknown) common cause ($X_i \leftrightarrow X_j$). Under the assumption of *causal sufficiency*, the latter does not hold. Therefore the $X_i - X_j$ link is associated with a causal relationship in one or the other direction. The causal sufficiency assumption will be relaxed in Section 6.

The edge orientation phase proceeds as follows:

- Each $X_i - X_j$ edge is first considered in isolation, and its orientation is evaluated using CGNN: Both models $Y = f(X, E)$ and $X = f(Y, E)$ are optimized and the associated score is computed (Eq. 6). The best orientation in the sense of the score is retained. After this step, an initial graph is built with complexity $2L$ with L the number of edges in the skeleton graph.

- The initial graph is revised to remove all cycles. Starting from a set of random nodes, all paths are followed iteratively until all nodes are reached; an edge pointing toward an already visited node and forming a cycle is reversed. The resulting DAG is used as initial DAG for the structured optimisation, below.

- The optimization of the DAG structure is achieved using a hill-climbing algorithm aimed to optimize the global score $C_{\hat{\mathcal{G}}}$. Iteratively, i) an edge $X_i - X_j$ is uniformly randomly selected in the current graph; ii) the graph obtained by reversing this edge is considered (if it is still a DAG and has not been considered before) and the associated global CGNN is retrained; iii) if this graph obtains a better global score than the former one, it becomes the current graph and the process is iterated until reaching a (local) optimum. More sophisticated combinatorial optimization approaches, e.g. Tabu search, will be considered in further work. In this paper, hill-climbing is used for a proof of concept of the proposed approach, achieving a decent trade-off between computational time and accuracy.

During the structured optimization, the graph skeleton is fixed; no edge is added or removed. The penalization term (Eq. 7) can thus be neglected at this stage and only the MMD-losses are used to compare two graphs. The penalization term will be used in Section 6 to compare structures with different skeletons, as the potential confounding factors will be dealt with by removing edges.

5 EXPERIMENTS

This section reports on the empirical validation of CGNN compared to the state of the art under the no confounding assumption. The experimental setting is first discussed. Thereafter, the results obtained in the bivariate case, where only asymmetries in the joint distribution can be used to infer the causal relationship, are discussed. The variable triplet case, where conditional independence can be used to uncover causal orientations, and the general case of $d > 2$ variables are finally considered. All computational times are measured on Intel Xeon 2.7Ghz (CPU) or on Tesla K40m graphics card (GPU).

5.1 EXPERIMENTAL SETTINGS

The CGNN architecture is a 1-hidden layer network with ReLU activation function. The multi-scale Gaussian kernel used in the MMD scores has bandwith γ ranging in $\{0.005, 0.05, 0.25, 0.5, 1, 5, 50\}$. The number nb_{run} used to average the score is set to 32 for CGNN-MMD (respectively 64 for CGNN-Fourier). The number n_h of neurons in the hidden layer, controlling the identifiability of the model, is the most sensitive hyper-parameter of the presented approach. Preliminary experiments are conducted to adjust its range, as follows. A 1,500 sample dataset is generated from the linear structural equation model with additive uniform noise $Y = X + \mathcal{U}(0, 0.5), X \sim U([-2, 2])$ (Fig. 5). Both CGNNs associated to $X \to Y$ and $Y \to X$ are trained until reaching convergence ($n_{epoch} = 1,000$) using Adam (Kingma and Ba, 2014) with a learning rate of 0.01 and evaluated over $n_{eval} = 500$ generated samples. The distributions generated from both generative models are displayed on Fig. 5 for $n_h = 2, 5, 20, 100$. The associated scores (averaged on 32 runs) are displayed on Fig. 6a, confirming that the model space must be restricted for the sake of identifiability (Zhang and Hyvärinen, 2009).

Preliminary experiments show that the split of the data between training and test set does not yield a better robustness; model overfitting is limited as the noise random variable is sampled anew in each epoch. The model precision is improved by using the full dataset as training set, with full batch size.

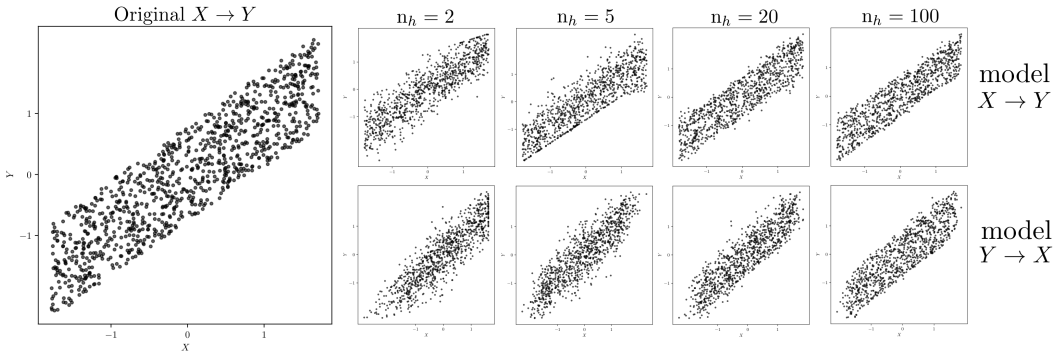
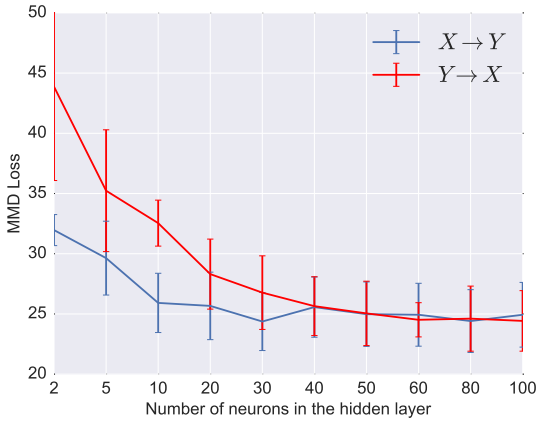


Figure 5: Leftmost: Data samples. Columns 2 to 5: Estimate samples generated from CGNN with direction $X \to Y$ (top row) and $Y \to X$ (bottom row) for number of hidden neurons $n_h = 2, 5, 20, 100$.



(a) $C_{X \to Y}$, $C_{Y \to X}$ with various n_h values.

(b) Scores $C_{X \to Y}$ and $C_{Y \to X}$ with their difference. *** denotes the significance at the 0.001 threshold with the t-test.

n_h	$C_{X \to Y}$	$C_{Y \to X}$	Diff.
2	32.0	43.9	11.9^{***}
5	29.6	35.2	5.6^{***}
10	25.9	32.5	6.6^{***}
20	25.7	28.3	2.6^{***}
30	24.4	26.8	2.4^{***}
40	25.6	25.6	0.7
50	25.0	25.0	0.6
100	24.9	24.4	-0.5

Figure 6: CGNN sensitivity w.r.t. the number of hidden neurons n_h: Scores associated to both causal models (average and standard deviation over 32 runs).

5.2 Learning bivariate causal structures

As said, under the no-confounder assumption a dependency between variables X and Y exists iff either X causes Y ($Y = f(X, E)$) or Y causes X ($X = f(Y, E)$). The identification of a *Bivariate Structural Causal Model* (Pearl, 2003) is based on comparing the model scores (Section 4.1) attached to both CGNNs.

Benchmarks. Five datasets with continuous variables are considered[7]:
• **CE-Cha**: 300 continuous variable pairs from the cause effect pair challenge (Guyon, 2013), restricted to pairs with label $+1$ ($X \to Y$) and -1 ($Y \to X$).
• **CE-Net**: 300 artificial pairs generated with a neural network initialized with random weights and random

[7]The first four datasets are available at `http://dx.doi.org/10.7910/DVN/3757KX`. The *Tuebingen cause-effect pairs* dataset is available at `https://webdav.tuebingen.mpg.de/cause-effect/`

distribution for the cause (exponential, gamma, lognormal, laplace...).
• **CE-Gauss**: 300 artificial pairs without confounder sampled with the generator of Mooij et al. (2016): $Y = f_Y(X, E_Y)$ and $X = f_X(E_X)$ with $E_X \sim p_{E_X}$ and $E_Y \sim p_{E_Y}$. p_{E_X} and p_{E_Y} are randomly generated Gaussian mixture distributions. Causal mechanism f_X and f_Y are randomly generated Gaussian processes.
• **CE-Multi**: 300 artificial pairs generated with linear and polynomial mechanisms. The effect variables are built with post additive noise setting ($Y = f(X) + E$), post multiplicative noise ($Y = f(X) \times E$), pre-additive noise ($Y = f(X + E)$) or pre-multiplicative noise ($Y = f(X \times E)$).
• **CE-Tueb**: 99 real-world cause-effect pairs from the *Tuebingen cause-effect pairs* dataset, version August 2016 (Mooij et al., 2016). This version of this dataset is taken from 37 different data sets coming from various domain: climate, census, medicine data.

For all variable pairs, the size n of the data sample is set to 1,500 for the sake of an acceptable overall computational load.

Baseline approaches. CGNN is assessed comparatively to the following algorithms [8]: i) ANM (Mooij et al., 2016) with Gaussian process regression and HSIC independence test of the residual; ii) a pairwise version of LiNGAM (Shimizu et al., 2006) relying on Independent Component Analysis to identify the linear relations between variables; iii) IGCI (Daniusis et al., 2012) with entropy estimator and Gaussian reference measure; iv) the post-nonlinear model (PNL) with HSIC test (Zhang and Hyvärinen, 2009); v) GPI-MML (Stegle et al., 2010); where Gaussian regressions are used to model both causal directions and the causal orientation is selected based on the best mean square error; vi) CDS, retaining the causal orientation with lowest variance of the conditional probability distribution; vii) Jarfo (Fonollosa, 2016), using a random forest causal classifier trained from the ChaLearn Cause-effect pairs on top of 150 features including ANM, IGCI, CDS, LiNGAM, regressions, HSIC tests.

Hyper-parameter selection. For a fair comparison, a leave-one-dataset-out procedure is used to select the key best hyper-parameter for each algorithm. To avoid computational explosion, a single hyper-parameter per algorithm is adjusted in this way; other hyper-parameters are set to their default value. For CGNN, n_h ranges over $\{5, 10, 15, 20, 25, 30, 35, 40, 50, 100\}$. For ANM and the bivariate fit, the kernel parameter for the Gaussian process regression ranges over $\{0.01, 0.1, 0.2, 0.5, 0.8, 1, 1.2, 1.5, 2, 5, 10\}$. For PNL, the threshold parameter alpha for the HSIC independence test ranges over $\{0.0005, 0.005, 0.01, 0.025, 0.04, 0.05, 0.06, 0.075, 0.1, 0.25, 0.5\}$. For CDS, the $ffactor$ involved in the discretization step ranges over $[[1, 10]]$. For GPI-MML, its many parameters are set to their default value as none of them appears to be more critical than others. Jarfo is trained from 4,000 variable pairs datasets with same generator used for **CE-Cha-train**, **CE-Net-train**, **CE-Gauss-train** and **CE-Multi-train**; the causal classifier is trained on all datasets except the test set.

Empirical results. Table 1 reports the area under the precision/recall curve for each benchmark and all algorithms, together with the computational time (CPU and GPU) and the computational complexity of the algorithm.

Methods based on simple regression like the bivariate fit and Lingam are outperformed as they underfit the data generative process. CDS and IGCI obtain very good results on few datasets. Typically, IGCI takes advantage of some specific features of the dataset, (e.g. the cause entropy being lower than the effect entropy in **CE-Multi**), but remains at chance level otherwise. ANM-HSIC yields good results when the additive assumption holds (e.g. on **CE-Gauss**), but fails otherwise. PNL, less restrictive than ANM, yields overall good results compared to the former methods. Jarfo, a voting procedure, can in principle yield the best of the above methods and does obtain good results on artificial data. However, it does not perform well on the

[8] Using the R program available at `https://github.com/ssamot/causality` for ANM, IGCI, PNL, GPI and LiNGAM.

Table 1: Bivariate Causal Modelling: Area under the precision/recall curve for the five benchmarks. For **Tueb**, the specific accuracy evaluation taking into account the cause effect pair weights as in (Mooij et al., 2016) is also given in parenthesis. The computational time is reported in both CPU and GPU environments.

Algorithm	**Cha**	**Net**	**Gauss**	**Multi**	**Tueb**	computational time*	scaling
Regression based							
Bivariate fit	56.4	77.6	36.3	55.4	58.4 (44.9)	$<$ 1s	$O(N^3)$
LiNGAM	54.3	43.7	66.5	59.3	39.7 (44.3)	$<$ 1s CPU	$O(N)$
Cond. distribution							
CDS	55.4	89.5	84.3	37.2	59.8 (65.5)	$<$ 1s CPU	$O(N)$
Entropy evaluation							
IGCI	54.4	54.7	33.2	80.7	60.7 (62.6)	$<$ 1s CPU	$O(N)$
Independence tests							
ANM-HSIC	66.3	85.1	88.9	35.5	53.7 (59.5)	$<$ 1s	$O(N^3)$
PNL	73.1	75.5	83.0	49.0	68.1 (66.2)	23 min CPU	$O(N^2)$
Classifier							
Jarfo	**79.5**	**92.7**	85.3	94.6	54.5 (59.5)	$<$ 1s	$O(N^3)$
Non-parametric							
GPI-MML	67.4	88.4	**89.1**	65.8	66.4 (62.6)	32min CPU	$O(N^3)$
CGNN-MMD	73.6	89.6	82.9	**96.6**	**79.8** (74.4)	11h CPU (24 min GPU)	$O(N^2)$
CGNN-Fourier	76.5	87.0	88.3	94.2	76.9 (72.7)	1h CPU (5 min GPU)	$O(N)$

real dataset **CE-Tueb**; this counter-performance is blamed on the differences between all five benchmark distributions and the lack of generalization / transfer learning. Generative models GPI and CGNN, although they do not offer strong identifiability guaranties such as ANM and PNL, yield good results on all datasets including the real-world one, showing their robustness w.r.t. various distributions. Similar results are obtained for CGNN-MMD and CGNN-Fourier. The leave-one-dataset-out procedure sets hyper-parameter n_h to values between 20 and 40 for the different datasets. Since the method is relatively insensitive to n_h in that range, n_h is set to 20 for all generative neural networks in the sequel.

Generative methods are the most computationally expensive ones. CGNN cost is decent on GPUs, especially so as the Fourier approximation scales linearly with the size of the data sample (Eq. 6); this linear complexity is a significant advantage compared to other generative causal modeling methods.

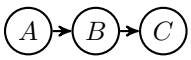
This first series of experiments thus shows that CGNN can effectively exploit the joint distribution asymmetries for bivariate causal modelling.

5.3 Identifying v-structures

A second series of experiments is conducted to investigate the method performances on variable triplets, where multivariate effects and conditional variable independence must be taken into account to identify the Markov equivalence class of a DAG. The considered setting is that of variable triplets (A, B, C) in the linear Gaussian case, where asymmetries between cause and effect cannot be exploited (Shimizu et al., 2006) and conditional independence tests are required. In particular strict pairwise methods can hardly be used due to un-identifiability (as each pair involves a linear mechanism with Gaussian input and additive Gaussian noise) (Hoyer et al., 2009).

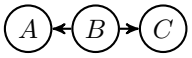
With no loss of generality, the graph skeleton involving variables (A, B, C) is $A - B - C$. All three causal models (up to variable renaming) based on this skeleton are used to generate 500-sample datasets, where the random noise variables are independent centered Gaussian variables.

(a) *Chain structure*



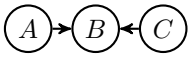
$$\begin{cases} A = E_A \\ B = A + E_B \\ C = B + E_C \end{cases}$$

(c) *reversed-V structure*



$$\begin{cases} B = E_B \\ A = B + E_A \\ C = B + E_C \end{cases}$$

(d) *V-structure*



$$\begin{cases} A = E_A \\ C = E_C \\ B = A + C + E_B \end{cases}$$

Figure 7: Datasets generated from the three DAG configurations with skeleton $A - B - C$

Given skeleton $A - B - C$, each dataset is used to model the possible four CGNN structures (Fig. 7, with generative SEMs):

- Chain structures ABC ($A = f_1(E_1)$, $B = f_2(A, E_2)$, $C = f_3(B, E_3)$ and CBA ($C = f_1(E_1)$, $B = f_2(C, E_2)$, $A = f_3(B, E_3)$)
- V structure: $A = f_1(E_1)$, $C = f_2(E_2)$, $B = f_3(A, C, E_3)$
- reversed V structure: $B = f_1(E_1)$, $A = f_2(B, E_2)$, $C = f_3(B, E_3)$

Let C_{ABC}, C_{CBA}, $C_{V-structure}$ and $C_{reversedV}$ denote the scores of the CGNN models respectively attached to these structures. The scores computed on all three datasets are displayed in Table 2 (average over 64 runs; the standard deviation is indicated in parenthesis).

	non V-structures		V structure
Score	Chain str.	Reversed-V str.	V-structure
C_{ABC}	*0.122 (0.009)*	*0.124 (0.007)*	0.172 (0.005)
C_{CBA}	*0.121 (0.006)*	*0.127 (0.008)*	0.171 (0.004)
$C_{reversedV}$	*0.122 (0.007)*	*0.125 (0.006)*	0.172 (0.004)
$C_{V\,structure}$	0.202 (0.004)	0.180 (0.005)	**0.127** (0.005)

Table 2: CGNN-MMD scores for all models on all datasets. Smaller scores indicate a better match. CGNN correctly identifies V-structure vs. other structures.

CGNN scores support a clear and significant discrimination between the V-structure and all other structures (noting that the other structures are Markov equivalent and thus can hardly be distinguished).

This second series of experiments thus shows that CGNN can effectively detect, and take advantage of, conditional independence between variables.

5.4 Multivariate causal modeling under Causal Sufficiency Assumption

Let $\mathbf{X} = [X_1, ..., X_d]$ be a set of continuous variables, satisfying the Causal Markov, faithfulness and causal sufficiency assumptions. It is further assumed that the skeleton of the causal graph is known, in order to independently assess the performance in terms of CPDAG identification, and causal modeling *per se*.

Benchmarks. Four continuous artificial benchmarks[9] $\mathcal{G}_i$ with 20 variables are considered, for $i \in [[2, 5]]$: i) Primary causes (variables with no parents in the graph) and random noise variables E are drawn according to a uniform or Gaussian distribution; ii) Causal mechanisms f_i are polynomial ; iii) Effect variables Y are drawn from $f(Pa(Y), E)$, where the number of parents $Pa(Y)$ is uniformly drawn between 1 and i for graph $\mathcal{G}_i$; iv) additive or multiplicative noise is added. The dataset size is 500.

Baselines. All algorithms are provided with the true skeleton of the data generating graph for a fair comparison. CGNN is compared with: i) the constraint-based PC algorithm (Spirtes et al., 2000), where edges are directed based on colliding structures; ii) the order-independent constraint-based algorithm with majority-based orientation (Colombo and Maathuis, 2014). This method is assessed using the Gaussian conditional independence test with Fisher z-transformation (PC-Gaussian) and with the gamma HSIC conditional independence test (Gretton et al., 2005) (PC-HSIC); iii) the score-based GES algorithm (Chickering, 2002), greedily optimizing the Bayesian Information Criterion (BIC). PC and GES are run with the *pcalg* package (Kalisch et al., 2012); iv) the pairwise ANM, LiNGAM and Jarfo algorithms (Section 5.2) are also considered.

Hyper-parameters. Preliminary experiments are used to adjust: i) the α significance threshold for the independence test in PC-Gaussian and PC-HSIC ($\alpha = 0.1$); ii) the penalization parameter λ for GES ($\lambda = 3.11$).

Experimental setting. **CGNN-MMD** and **CGNN-Fourier** start from an initial orientation of all skeleton edges (Section 5.2) and apply a hill-climbing optimization method subject to the DAG constraint. In this step, each candidate DAG is optimized with a Tensorflow computation graph on GPUs. The training and test computational time of the full DAG (with respectively 1,000 and 500 epochs) is 35 seconds on a Tesla K40m graphics card.

The comparative assessment of **CGNN-MMD** and **CGNN-Fourier** is displayed in Table 3, reporting the average area under the precision-recall curve for edge orientation over all graph edges and all graphs. The ratio of edges for which the algorithm provides an orientation is given in brackets.

The best performing method overall is the constraint-based method PC-HSIC, particularly so when the number of connections increases. These good performances are explained from the vote-based edge orientation: when the graph skeleton is known, the majority rule is robust as one edge belongs to many v-structures. The second best method is CGNN-MMD, dominating CGNN-Fourier. Complementary experiments are launched to assess the scalability: on $\mathcal{G}_3$ with $d = 100$, the area under the precision/recall curve (averaged over 5 graphs) is 85.5% ± 4, with computational cost 30 hours on a GPU server with 4 GeForce GTX 1080Ti.

[9] The data generator is available at `https://github.com/GoudetOlivier/CGNN`. The datasets considered are available at `http://dx.doi.org/10.7910/DVN/UZMB69`. Graph $\mathcal{G}_1$ is not considered for causal modeling here is restricted to the pairwise setting (each node involves at most one parent).

Table 3: Multivariate causal modeling: Area under the precision/recall curve for the edge orientation test. The ratio of skeleton edges for which the algorithm provides an orientation is given in brackets. The rightmost column indicates the computational time in a CPU (Intel Xeon 2.7Ghz) or GPU (Tesla K40m graphics card) environment.

Algorithm	$\mathcal{G}_2$	$\mathcal{G}_3$	$\mathcal{G}_4$	$\mathcal{G}_5$	Time
Constraint-based					
PC-Gaussian	82.3 $\pm$4 (87.8)	80.0 $\pm$7 (89.2)	88.1 $\pm$10 (95.7)	92.5$\pm$4 (96.5)	1 to 10s CPU
PC-HSIC	93.4 $\pm$3 (78.5)	93.0 $\pm$4 (77.9)	**98.9** $\pm$2 (88.0)	**96.7** $\pm$2 (91.9)	2 to 15h CPU
Score-based					
GES	75.3 $\pm$7 (81.2)	73.6 $\pm$7 (77.7)	69.3$\pm$11 (78.6)	75.6 $\pm$5 (78.5)	1 to 2s CPU
Pairwise orientation					
LiNGAM	64.4 $\pm$4 (100)	71.1 $\pm$1 (100)	71.6 $\pm$7 (100)	72.1 $\pm$1 (100)	1 to 2s CPU
ANM	72.9 $\pm$9 (100)	72.5 $\pm$4 (100)	79.9 $\pm$5 (100)	71.8 $\pm$2 (100)	5 to 20s CPU
Jarfo	69.9 $\pm$9 (100)	87.3 $\pm$3 (100)	88.5 $\pm$5 (100)	70.2 $\pm$7 (100)	10 to 30s CPU
CGNN-Fourier	94.5 $\pm$2 (100)	84.9 $\pm$9 (100)	88.4 $\pm$4 (100)	87.0 $\pm$6 (100)	2 to 3 h GPU
CGNN-MMD	**96.9** $\pm$1 (100)	**96.5** $\pm$3 (100)	97.2 $\pm$3 (100)	88.2 $\pm$6 (100)	3 to 4 h GPU

6 Relaxing the sufficiency assumption

This section tackles multi-variate causal modeling under Causal Markov and Faithfulness assumptions, relaxing the sufficiency assumption, i.e. addressing potential confounders. Each edge $X - Y$ in the graph skeleton can thus be associated with three orientations: $X \to Y$, $X \leftarrow Y$ and $X \leftrightarrow Y$ (both variables are consequences of common hidden variables).

6.1 Principle

Hidden common causes are modeled through correlated random noise. Formally, an additional noise variable $E_{i,j}$ is associated to each $X_i - X_j$ edge in the graph skeleton. The performance of the CGNN extension w.r.t. confounders, called CCGNN (*Confounder-robust CGNN*), is investigated by considering the same benchmarks (Section 5.4) and removing some variables in $\mathbf{X}$. For instance, in the FCM on $\mathbf{X} = [X_1, \ldots, X_5]$ considered in Fig. 1, hiding variable X_1 would be handled through correlated noise $E_{2,3}$. The resulting FCM (Fig. 8) includes both bi-directed edges (e.g. $X_2 \leftrightarrow X_3$) and direct causal effects (e.g., $X_3 \to X_5$ or $X_4 \to X_5$).

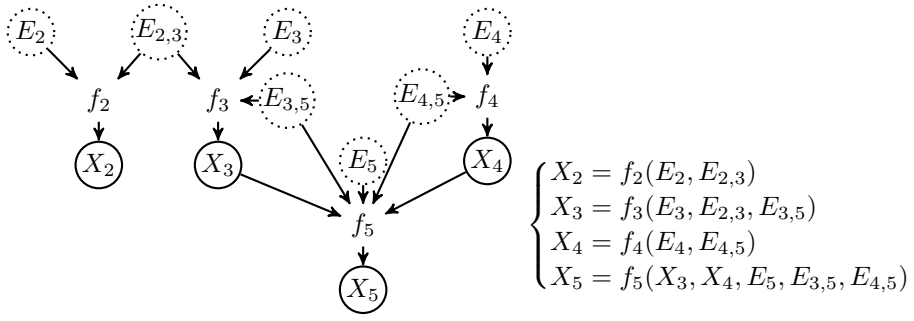


Figure 8: The Functional Causal Model (FCM) on $\mathbf{X} = [X_1, \ldots, X_5]$ with the missing variable X_1

Formally, given a graph skeleton $\mathcal{S}$, the FCM with correlated noise is defined as:

$$X_i \leftarrow f_i(X_{\mathrm{Pa}(i;\mathcal{G})}, E_i, E_{\mathrm{Ne}(i;\mathcal{S})}), \tag{8}$$

where $\mathrm{Ne}(i;\mathcal{S})$ is the set of indices of all the variables adjacent to variable X_i in the skeleton $\mathcal{S}$. Here we make a clear distinction between the directed acyclic graph denoted $\mathcal{G}$ and the skeleton $\mathcal{S}$. Indeed, due to the presence of confounding correlated noise, any variable in $\mathcal{G}$ can be removed without altering $\mathcal{S}$. We use the same hierarchical generative neural network to model the new FCM presented in equation 8. The difference is the new noise variables having effect on pairs of variables simultaneously. However, since the correlated noise FCM is still defined over a directed acyclic graph $\mathcal{G}$, the functions $f_1, \ldots, f_d$ of the model, which we implement as neural networks, the model can still be learned end-to-end using backpropagation based on the CGNN loss.

The non-parametric optimization of the $\hat{\mathcal{G}}$ structure is also achieved using a Hill-Climbing algorithm; in each step an edge of $\mathcal{S}$ is randomly drawn and modified in $\hat{\mathcal{G}}$ using one out of the possible three operators: reverse the edge, add an edge and remove an edge. Other algorithmic details are as in Section 4.3: the greedy search optimizes the penalized loss function (Eq. 7) with penalization weight (set to 5×10^{-5} in the experiments). The algorithm stops when no improvement is obtained. Each causal edge $X_i \to X_j$ in $\mathcal{G}$ is associated with a score, measuring its contribution to the global score: $C_{X_i \to X_j} = C_{\mathcal{G}} - C_{\mathcal{G}-\{X_i \to X_j\}}$. Missing edges are associated with a score 0.

6.2 Experimental validation

Benchmarks. The empirical validation of CCGNN is conducted on same benchmarks as in Section 5.4 ($\mathcal{G}_i$, $i \in [[2, 5]]$), where 3 variables (causes for at least two other variables in the graph) have been randomly removed[10]. The true graph skeleton is augmented with edges $X - Y$ for all X, Y that are consequences of a same removed cause. All algorithms are provided with the same graph skeleton for a fair comparison. The task is to both orient the edges in the skeleton, and remove the spurious direct dependencies created by latent causal variables.

Baselines. CCGNN is compared with state of art methods: i) constraint-based RFCI (Colombo et al., 2012), extending the PC method equipped with Gaussian conditional independence test (RFCI-Gaussian) and the gamma HSIC conditional independence test (Gretton et al., 2005) (RFCI-HSIC). We use the order-independent constraint-based version proposed by Colombo and Maathuis (2014) and the majority rules for the orientation of the edges. The significance level α for the conditional independence tests is set to $\alpha = 0.1$; ii) Jarfo trained on the 16,200 pairs of the cause-effect pair challenge (Guyon, 2013; 2014) to detect for each pair of variable if $X_i \to Y_i$, $Y_i \to X_i$ or $X_i \leftrightarrow Y_i$.

The comparative performances are shown in Table 4, reporting the area under the precision/recall curve. Overall, these results confirm the robustness of the CCGNN proposed approach w.r.t. confounders, and its competitiveness w.r.t. RFCI with powerful conditional independence test (RFCI-HSIC). Interestingly, the effective causal relations between the visible variables are associated with a high score; spurious links due to hidden latent variables get a low score.

7 Discussion and Perspectives

This paper introduces CGNN, a new framework and methodology for functional causal model learning, leveraging the power and non-parametric flexibility of Generative Neural Networks. On the one hand, CGNN

[10] The datasets considered are available at `http://dx.doi.org/10.7910/DVN/UZMB69`

Table 4: Causal modeling with confounders: Average area under the precision/recall curve and computational time.

Algorithm	$\mathcal{G}_2$	$\mathcal{G}_3$	$\mathcal{G}_4$	$\mathcal{G}_5$	Time
RFCI-Gaussian	53.4 $\pm$11	49.0 $\pm$10	51.1 $\pm$13	63.6 $\pm$11	10 to 30s CPU
RFCI-HSIC	76.2 $\pm$11	65.2 $\pm$7	73.1 $\pm$6	**72.9** $\pm$6	3 to 18h CPU
Jarfo	64.1 $\pm$11	62.1 $\pm$7	72.2 $\pm$7	64.9 $\pm$4	10 to 30s CPU
CGNN-MMD	**79.8** $\pm$12	**76.1** $\pm$11	**84.4** $\pm$7	70.9 $\pm$4	4 to 6 h GPU

does not come with identifiability guarantees; on the other hand, it seamlessly accommodates causal modeling in presence of confounders, and its extensive empirical validation demonstrates its merits compared to the state of the art on medium-size problems. It is believed that CGNNs can pave the way toward intelligible deep networks, yielding interpretable network architectures. In the meanwhile, CGNNs can be used as simulators, characterizing the data distribution under *interventions* on one or several causal variables.

The main limitation of CGNN is its computational cost, due to the quadratic complexity of the CGNN learning criterion w.r.t. the data size, based on the Maximum Mean Discrepancy between the generated and the observed data. A linear approximation thereof has been proposed, with comparable empirical performances.

The main perspective for further research aims at a better scalability of the approach from medium to large problems. On the one hand, the computational scalability will be sought through modular design, gradually building the network architecture from subsets of variables. On the other hand, the thorough inspection of the residuals will be used to guide the identification of the links between the variable subsets. Another perspective regards the extension of the approach to categorical variables.

REFERENCES

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

Colombo, D. and Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research*, 15(1):3741–3782.

Colombo, D., Maathuis, M. H., Kalisch, M., and Richardson, T. S. (2012). Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321.

Daniusis, P., Janzing, D., Mooij, J., Zscheischler, J., Steudel, B., Zhang, K., and Schölkopf, B. (2012). Inferring deterministic causal relations. *arXiv preprint arXiv:1203.3475*.

Drton, M. and Maathuis, M. H. (2016). Structure learning in graphical modeling. *Annual Review of Statistics and Its Application*, (0).

Edwards, R. (1964). Fourier analysis on groups.

Fonollosa, J. A. (2016). Conditional distribution variability measures for causality detection. *arXiv preprint arXiv:1601.06680*.

Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B., Smola, A. J., et al. (2007). A kernel method for the two-sample-problem. 19:513.

Gretton, A., Herbrich, R., Smola, A., Bousquet, O., and Schölkopf, B. (2005). Kernel methods for measuring independence. *Journal of Machine Learning Research*, 6(Dec):2075–2129.

Guyon, I. (2013). Chalearn cause effect pairs challenge.

Guyon, I. (2014). Chalearn fast causation coefficient challenge.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., and Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. In *Neural Information Processing Systems (NIPS)*, pages 689–696.

Kalisch, M., Mächler, M., Colombo, D., Maathuis, M. H., Bühlmann, P., et al. (2012). Causal inference using graphical models with the r package pcalg. *Journal of Statistical Software*, 47(11):1–26.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *ArXiv e-prints*.

Li, Y., Swersky, K., and Zemel, R. S. (2015). Generative moment matching networks. In *ICML*, pages 1718–1727.

Lopez-Paz, D., Muandet, K., Schölkopf, B., and Tolstikhin, I. O. (2015). Towards a learning theory of cause-effect inference. In *ICML*, pages 1452–1461.

Lopez-Paz, D., Nishihara, R., Chintala, S., Schölkopf, B., and Bottou, L. (2016). Discovering causal signals in images. *arXiv preprint arXiv:1605.08179*.

Lopez-Paz, D. and Oquab, M. (2016). Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., and Schölkopf, B. (2016). Distinguishing cause from effect using observational data: methods and benchmarks. *Journal of Machine Learning Research*, 17(32):1–102.

Nandy, P., Hauser, A., and Maathuis, M. H. (2015). High-dimensional consistency in score-based and hybrid structure learning. *arXiv preprint arXiv:1507.02608*.

Ogarrio, J. M., Spirtes, P., and Ramsey, J. (2016). A hybrid causal search algorithm for latent variable models. In *Conference on Probabilistic Graphical Models*, pages 368–379.

Pearl, J. (2003). Causality: models, reasoning and inference. *Econometric Theory*, 19(675-685):46.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pearl, J. and Verma, T. (1991). *A formal theory of inductive causation*. University of California (Los Angeles). Computer Science Department.

Quinn, J. A., Mooij, J. M., Heskes, T., and Biehl, M. (2011). Learning of causal relations. In *ESANN*.

Ramsey, J. D. (2015). Scaling up greedy causal search for continuous variables. *arXiv preprint arXiv:1507.07749*.

Richardson, T. and Spirtes, P. (2002). Ancestral graph markov models. *The Annals of Statistics*, 30(4):962–1030.

Rojas-Carulla, M., Baroni, M., and Lopez-Paz, D. (2017). Causal discovery using proxy variables. *arXiv preprint arXiv:1702.07306*.

Scheines, R. (1997). An introduction to causal inference.

Sgouritsa, E., Janzing, D., Hennig, P., and Schölkopf, B. (2015). Inference of cause and effect with unsupervised inverse regression. In *AISTATS*.

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

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

Spirtes, P., Meek, C., Richardson, T., and Meek, C. (1999). An algorithm for causal inference in the presence of latent variables and selection bias.

Spirtes, P. and Zhang, K. (2016). Causal discovery and inference: concepts and recent methodological advances. In *Applied informatics*, volume 3, page 3. Springer Berlin Heidelberg.

Statnikov, A., Henaff, M., Lytkin, N. I., and Aliferis, C. F. (2012). New methods for separating causes from effects in genomics data. *BMC genomics*, 13(8):S22.

Stegle, O., Janzing, D., Zhang, K., Mooij, J. M., and Schölkopf, B. (2010). Probabilistic latent variable models for distinguishing between cause and effect. In *Neural Information Processing Systems (NIPS)*, pages 1687–1695.

Tsamardinos, I., Brown, L. E., and Aliferis, C. F. (2006). The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78.

Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., and Sugiyama, M. (2014). High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207.

Zhang, K. and Hyvärinen, A. (2009). On the identifiability of the post-nonlinear causal model. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*, pages 647–655. AUAI Press.

Zhang, K., Peters, J., Janzing, D., and Schölkopf, B. (2012). Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*.

Zhang, K., Wang, Z., Zhang, J., and Schölkopf, B. (2016). On estimation of functional causal models: general results and application to the post-nonlinear causal model. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(2):13.