Learning Functional Causal Models with Generative Neural Networks

Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag

Abstract We introduce a new approach to functional causal modeling from observational data, called *Causal Generative Neural Networks* (CGNN). 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. Firstly, CGNN is applied to cause-effect inference, where the task is to identify the best causal hypothesis out of "$X \rightarrow Y$" and "$Y \rightarrow X$". Secondly, CGNN is applied to the problem of identifying v-structures and conditional independences. Thirdly, CGNN is applied to multivariate functional causal modeling: given a skeleton describing the direct dependences in a set of random variables $\mathbf{X} = [X_1, \ldots, X_d]$, 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, CGNN is extended to handle the case of confounders, where latent variables are involved in the overall causal model.

O. Goudet (✉) · D. Kalainathan · P. Caillou · M. Sebag
Team TAU - CNRS, INRIA, Université Paris Sud, Université Paris Saclay, Paris, France
e-mail: olivier.goudet@inria.fr; Diviyan.kalainathan@lri.fr; Caillou@lri.fr; sebag@lri.fr

I. Guyon
INRIA, Université Paris Sud, Université Paris Saclay, Paris, France

ChaLearn, Berkeley, CA, USA
e-mail: guyon@chalearn.org

D. Lopez-Paz
Facebook AI Research, Menlo Park, CA, USA
e-mail: david@lopezpaz.org

H. J. Escalante et al. (eds.), *Explainable and Interpretable Models in Computer Vision and Machine Learning*, The Springer Series on Challenges in Machine Learning, https://doi.org/10.1007/978-3-319-98131-4_3

Keywords Generative neural networks · Causal structure discovery · Cause-effect pair problem · Functional causal models · Structural equation models

1 Introduction

Deep learning models have shown extraordinary predictive abilities, breaking records in image classification (Krizhevsky et al. 2012), speech recognition (Hinton et al. 2012), language translation (Cho et al. 2014), and reinforcement learning (Silver et al. 2016). However, the predictive focus of black-box deep learning models leaves little room for explanatory power. More generally, current machine learning paradigms offer no protection to avoid mistaking correlation by causation. For example, consider the prediction of target variable Y given features X and Z, assuming that the underlying generative process is described by the equations:

$$\begin{aligned} X, E_Y, E_Z &\sim \text{Uniform}(0, 1), \\ Y &\leftarrow 0.5X + E_Y, \\ Z &\leftarrow Y + E_Z, \end{aligned}$$

with (E_Y, E_Z) additive noise variables. The above model states that the values of Y are computed as a function of the values of X (we say that X causes Y), and that the values of Z are computed as a function of the values of Y (Y causes Z). The "assignment arrows" emphasize the asymmetric relations between all three random variables. However, as Z provides a stronger signal-to-noise ratio than X for the prediction of Y, the best regression solution in terms of least-square error is

$$\hat{Y} = 0.25X + 0.5Z$$

The above regression model, a typical case of inverse regression after Goldberger (1984), would wrongly explain some changes in Y as a function of changes in Z, although Z does not cause Y. In this simple case, there exists approaches overcoming the inverse regression mistake and uncovering all true cause-effect relations (Hoyer et al. 2009). In the general case however, mainstream machine learning approaches fail to understand the relationships between all three distributions, and might attribute some effects on Y to changes in Z.

Mistaking correlation for causation can be catastrophic for agents who must plan, reason, and decide based on observations. Thus, discovering causal structures is of crucial importance.

The gold standard to discover causal relations is to perform experiments (Pearl 2003). However, experiments are in many cases expensive, unethical, or impossible to realize. In these situations, there is a need for *observational causal discovery*, that is, the estimation of causal relations from observations alone (Spirtes et al. 2000; Peters et al. 2017).

In the considered setting, *observational* empirical data (drawn independent and identically distributed from an unknown distribution) is given as a set of n samples of real valued feature vectors of dimension d. We denote the corresponding random vector as $\mathbf{X} = [X_1, \ldots, X_d]$. We seek a Functional Causal Model (FCM), also known as Structural Equation Model (SEM), that best matches the underlying data-generating mechanism(s) in the following sense: under relevant manipulations/interventions/experiments the FCM would produce data distributed similarly to the real data obtained in similar conditions.

Let intervention *do(X=x)* be defined as the operation on distribution obtained by clamping variable X to value x, while the rest of the system remains unchanged (Pearl 2009). It is said that variable X_i is a **direct cause** of X_j with respect to $X_1, \ldots, X_d$ iff different interventions on variable X result in different marginal distributions on X_j, everything else being equal:

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

with $\mathbf{X}_{\setminus ij} := X_{\{1,\ldots,d\}\setminus i,j}$ the set of all variables except X_i and X_j, scalar values $x \neq x'$, and vector value $\mathbf{c}$. Distribution $P_{X_j|\text{do}(X_i=x,\mathbf{X}_{\setminus ij}=c)}$ is the resulting interventional distribution of the variable X_j when the variable X_i is clamped to value x, while keeping all other variables at a fixed value (Mooij et al. 2016).

As said, conducting such interventions to determine direct causes and effects raises some limitations. For this reason, this paper focuses on learning the causal structure from observational data only, where the goal and validation of the proposed approach is to match the known "ground truth" model structure.

A contribution of the paper is to unify several state-of-art methods into one single consistent and more powerful approach. On the one hand, leading researchers at UCLA, Carnegie Mellon, University of Crete and elsewhere have developed powerful algorithms exploiting Markov properties of directed acyclic graphs (DAGs) (Spirtes et al. 2000; Tsamardinos et al. 2006; Pearl 2009). On the other hand, the Tübingen School has proposed new and powerful functional causal models (FCM) 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).

In this paper, the learning of functional causal models is tackled in the search space of generative neural networks (Kingma and Welling 2013; Goodfellow et al. 2014), and aims at the functional causal model (structure and parameters), best fitting the underlying data generative process. The merits of the proposed approach, called Causal Generative Neural Network (CGNN) are extensively and empirically demonstrated compared to the state of the art on artificial and real-world benchmarks.

This paper is organized as follows: Sect. 2 introduces the problem of learning an FCM and the underlying assumptions. Section 3 briefly reviews and discusses the state of the art in causal modeling. The FCM modeling framework within the search space of generative neural networks is presented in Sect. 4. Section 5 reports on an extensive experimental validation of the approach comparatively to the state of the art for pairwise cause-effect inference and graph recovery. An extension of the proposed framework to deal with potential confounding variables is presented in Sect. 6. The paper concludes in Sect. 7 with some perspectives for future works.

2 Problem Setting

A Functional Causal Model (FCM) upon a random variable vector $\mathbf{X} = [X_1, \ldots, X_d]$ is a triplet $(\mathcal{G}, f, \mathcal{E})$, representing a set of equations:

$$X_i \leftarrow f_i(X_{\mathrm{Pa}(i;\mathcal{G})}, E_i), E_i \sim \mathcal{E}, \text{ for } i = 1, \ldots, d \tag{2}$$

Each equation characterizes the direct causal relation explaining variable X_i from the set of its causes $X_{\mathrm{Pa}(i;\mathcal{G})} \subset \{X_1, \ldots, X_d\}$, based on some *causal mechanism* f_i involving besides $X_{\mathrm{Pa}(i;\mathcal{G})}$ some random variable E_i drawn after distribution $\mathcal{E}$, meant to account for all unobserved variables.

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.

2.1 Notations

By abuse of notation and for simplicity, a variable X and the associated node in the causal graph, in one-to-one correspondence, 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 (1) a direct causal relationship ($X \rightarrow Y$ or $Y \rightarrow X$); (2) a causal relationship in either direction ($X - Y$); (3) a non-causal association ($X \leftrightarrow Y$) due to external common causes (Richardson and Spirtes 2002).

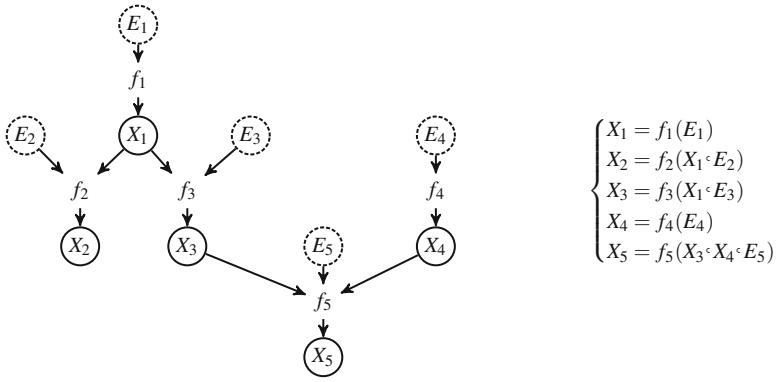


Fig. 1 Example of a functional causal model (FCM) on $\mathbf{X} = [X_1, \ldots, X_5]$. Left: causal graph $\mathcal{G}$; right: causal mechanisms

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, a.k.a. unshielded collider: Three variables $\{X, Y, Z\}$ form a v-structure iff their causal structure is: $X \rightarrow Z \leftarrow Y$.

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.

2.2 Assumptions and Properties

The state of the art in causal modeling most commonly involves four 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 an 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 an 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 $\mathscr{G}$ of an FCM iff every conditional independence relation that holds true in P is entailed by $\mathscr{G}$ (Spirtes and Zhang 2016). Therefore, if there exists an independence relation in $\mathbf{X}$ that is not a consequence of the Causal Markov assumption, 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: (1) $\{X, Y\}$ and $\{Y, Z\}$ are adjacent; (2) $\{X, Z\}$ are NOT adjacent; (3) $X \not\perp\!\!\!\perp Z|Y$, then their causal structure is a v-structure $(X \rightarrow Y \leftarrow Z)$.

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 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.

Considering the functional model on $\mathbf{X} = [X_1, \ldots, X_5]$ on Fig. 1, the associated DAG $\mathscr{G}$ and graph skeleton are respectively depicted on Fig. 2a, b. Causal modeling exploits observational data to recover the $\mathscr{G}$ structure from all CI (Markov proper-

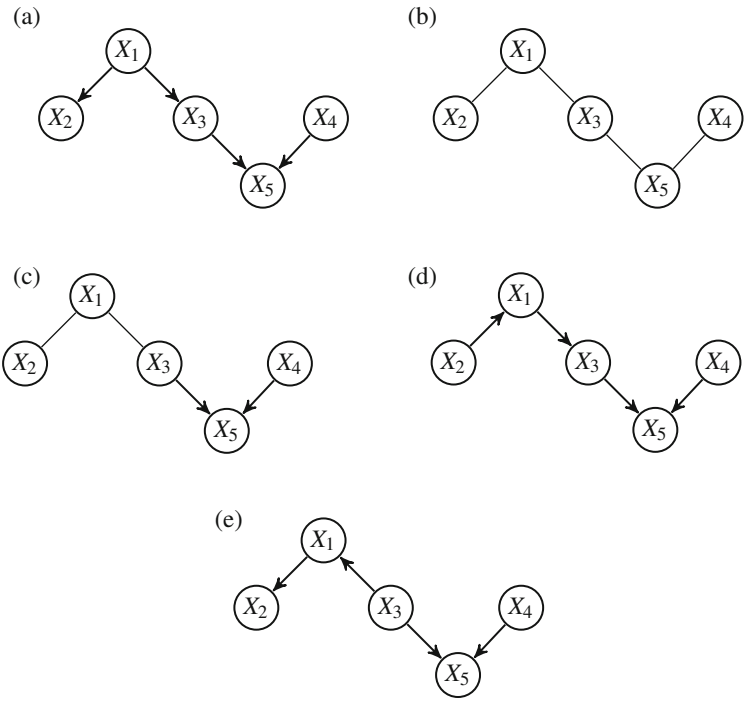


Fig. 2 Example of a Markov equivalent class. There exists three graphs (**a**, **d**, **e**) consistent with a given graph skeleton (**b**); the set of these consistent graphs defines the Markov equivalent class (**c**). (**a**) The exact DAG of $\mathscr{G}$. (**b**) The skeleton of $\mathscr{G}$. (**c**) The CPDAG of $\mathscr{G}$. (**d**) A Markov equivalent DAG of $\mathscr{G}$. (**e**) Another Markov equivalent DAG of $\mathscr{G}$

ties) between variables.[1] Under CSA, CMA and CFA, as $(X_3 \perp\!\!\!\perp X_4|X_5)$ does not hold, a v-structure $X_3 \rightarrow X_5 \leftarrow X_4$ is identified (Fig. 2c). 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. 2d, e encode the same conditional independences as the true DAG (Fig. 2a). 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. 2c, called *Completed Partially Directed Acyclic Graph* (CPDAG).

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

3.1.1 Constraint-Based Methods

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. 1993). PC first builds the DAG skeleton based on conditional independences among variables and subsets of variables. Secondly, it identifies v-structures (Fig. 2c). 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 χ-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 size with the number of variables. Additionally, the use of propagation rules to direct edges is prone to error propagation.

3.1.2 Score-Based Methods

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*,

[1]The so-called constraint-based methods base the recovery of graph structure on conditional independence tests. In general, 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.

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.[2]

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.

3.1.3 Hybrid Algorithms

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 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.

[2] After Ramsey (2015), in the linear model with Gaussian variable 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.

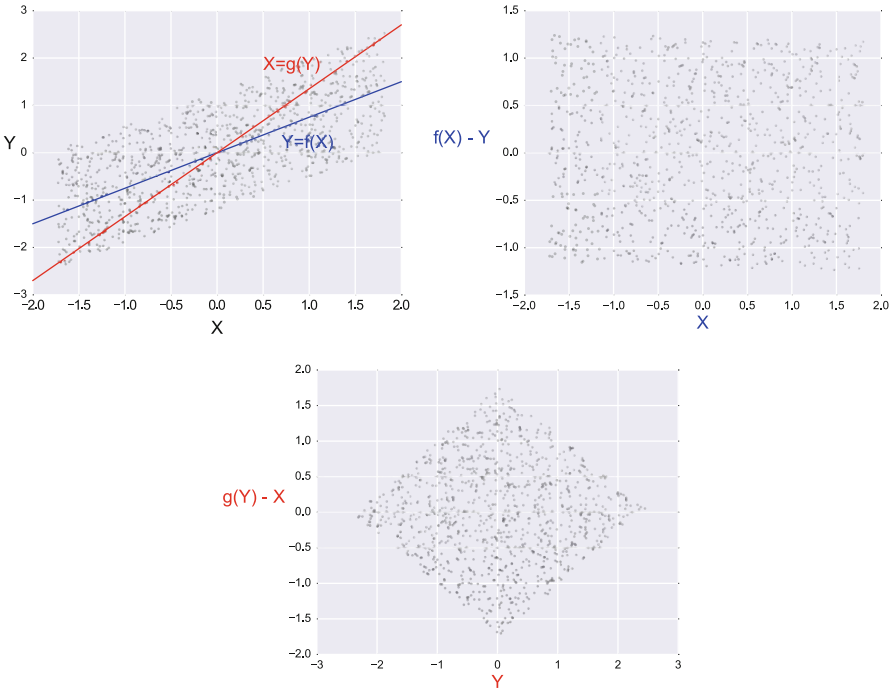


Fig. 3 Left: Joint distribution $P(X, Y)$ generated from DAG $X \to Y + E$, with E a uniform noise variable. The linear regression of Y on X (respectively of X on Y) is depicted as a blue (resp. red) curve. Middle: Error $f(X) - Y$ is independent of X. Right: Error $g(Y) - X$ is not independent of Y.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

3.2.1 The Intuition

Let us consider 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 $X = a'Y + b'$ (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).

3.2.2 Restriction on the Class of Causal Mechanisms Considered

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 noise 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 Y, 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 modeled as:

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

with $P_a^k(X_i)$ the kth 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).

3.2.3 Pairwise Methods

In the continuous, non-linear bivariate case, specific methods have been developed to orient the variable edge.[3] 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 (1) there exists no AMN $X = g(Y) + E'$ consistent with $P(X, Y)$; (2) 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

[3] These methods can be extended to the multivariate case and used for causal graph identification by orienting each edge in turn.

enforcing their smoothness (regularity). Other causal inference methods (Sgouritsa et al. 2015) are based on the idea that if $X \rightarrow Y$, 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 $\{X_i \rightarrow Y_i, Y_i \rightarrow X_i, X_i \perp\!\!\!\perp Y_i, X_i \leftrightarrow Y_i$ (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 $A \rightarrow B \leftarrow C$ (unshielded collider); still, a bivariate model based on cause-effect asymmetry would neither identify $A \rightarrow B$ 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.

To overcome these limitations, an extension of the bivariate post-nonlinear model (PNL) has been proposed (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.

This paper aims to propose a unified framework getting the best out of both worlds of CPDAG and bivariate approaches.

An inspiration of the approach is the CAM algorithm (Bühlmann et al. 2014), which is an extension to the graph setting of the pairwise additive model (ANM) (Hoyer et al. 2009). In CAM the FCM is modeled as:

$$X_i = \sum_{k \in \mathrm{Pa}(i;\mathcal{G})} f_k(X_k) + E_i, \text{ for } i = 1, \ldots, d \tag{4}$$

Our method can be seen an extension of CAM, as it allows non-additive noise terms and non-additive contributions of causes, in order to model flexible conditional distributions, and addresses the problem of learning FCMs (Sect. 2):

$$X_i = f_i(X_{\mathrm{Pa}(i;\mathcal{G})}, E_i), \text{ for } i = 1, \ldots, d \tag{5}$$

An other inspiration of our framework is the recent method of Lopez-Paz and Oquab (2016), where a conditional generative adversarial network is trained to model $X \to Y$ and $Y \to X$ in order to infer the causal direction based on the Occam's razor principle.

This approach, called **Causal Generative Neural Network (CGNN)**, features two original contributions. Firstly, multivariate causal mechanisms f_i are learned as **generative neural networks** (as opposed to, regression networks). The novelty is to use neural nets to model the joint distribution of the observed variables and learn a continuous FCM. This approach does not explicitly 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 methods searches a trade-off between data fitting and model complexity.

Secondly, the data generative models are trained using a non-parametric score, the Maximum Mean Discrepancy (Gretton et al. 2007). This criterion is used instead of likelihood based criteria, hardly suited to complex data structures, or mean square criteria, implicitly assuming an additive noise (e.g. as in CAM, Eq. (4)).

Starting from a known skeleton, Sect. 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 Sect. 5. In Sect. 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 Causal Generative Neural Networks

Let $\mathbf{X} = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, and further assume that the joint density function h of P is continuous and strictly positive on a compact subset of $\mathbb{R}^d$ and zero elsewhere.

This section first presents the modeling of continuous FCMs with generative neural networks with a given graph structure (Sect. 4.1), the evaluation of a candidate model (Sect. 4.2), and finally, the learning of a best candidate from observational data (Sect. 4.3).

4.1 Modeling Continuous FCMs with Generative Neural Networks

We first show that there exists a (non necessarily unique) *continuous* functional causal model $(\mathscr{G}, f, \mathscr{E})$ such that the associated data generative process fits the distribution P of the observational data.

Proposition 1 *Let $\boldsymbol{X} = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, and further assume that the joint density function h of P is continuous and strictly positive on a compact and convex subset of $\mathbb{R}^d$, and zero elsewhere. Letting $\mathscr{G}$ be a DAG such that P can be factorized along $\mathscr{G}$,*

$$P(X) = \prod_i P(X_i | X_{Pa(i;\mathscr{G})})$$

there exists $f = (f_1, \ldots, f_d)$ with f_i a continuous function with compact support in $\mathbb{R}^{|Pa(i;\mathscr{G})|} \times [0, 1]$ such that $P(X)$ equals the generative model defined from FCM $(\mathscr{G}, f, \mathscr{E})$, with $\mathscr{E} = \mathscr{U}[0, 1]$ the uniform distribution on $[0, 1]$.

Proof In section "Proofs" in Appendix.

In order to model such continuous FCM $(\mathscr{G}, f, \mathscr{E})$ on d random variables $\mathbf{X} = [X_1, \ldots, X_d]$, we introduce the CGNN (Causal Generative Neural Network) depicted on Fig. 4.

Definition 1 A CGNN over d variables $[\hat{X}_1, \ldots, \hat{X}_d]$ is a triplet $\mathscr{C}_{\widehat{\mathscr{G}}, \hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ where:

1. $\widehat{\mathscr{G}}$ is a Directed Acyclic Graph (DAG) associating to each variable $\hat{X}_i$ its set of parents noted $\hat{X}_{\mathrm{Pa}(i;\widehat{\mathscr{G}})}$ for $i \in [[1, d]]$

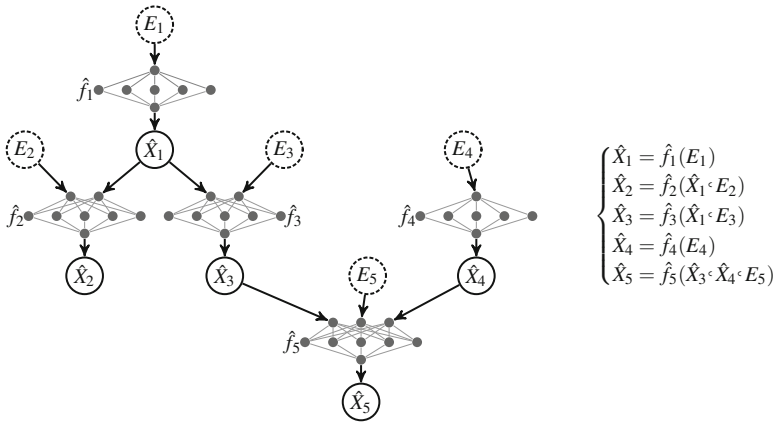


Fig. 4 Left: Causal generative neural network over variables $\hat{\mathbf{X}} = (\hat{X}_1, \ldots, \hat{X}_5)$. Right: Corresponding functional causal model equations

2. For $i \in [\![1, d]\!]$, causal mechanism $\hat{f}_i$ is a 1-hidden layer regression neural network with n_h hidden neurons:

$$\hat{X}_i = \hat{f}_i(\hat{X}_{\mathrm{Pa}(i;\hat{\mathscr{G}})}, E_i) = \sum_{k=1}^{n_h} \bar{w}_k^i \sigma \left(\sum_{j \in \mathrm{Pa}(i;\mathscr{G})} \hat{w}_{jk}^i \hat{X}_j + w_k^i E_i + b_k^i \right) + \bar{b}^i \tag{6}$$

   with $n_h \in \mathbb{N}*$ the number of hidden units, $\bar{w}_k^i, \hat{w}_{jk}^i, w_k^i, b_k^i, \bar{b}^i \in \mathbb{R}$ the parameters of the neural network, and σ a continuous activation function.
3. Each variable E_i is independent of the *cause* X_i. Furthermore, all noise variables are mutually independent and drawn after same distribution $\mathscr{E}$.

It is clear from its definition that a CGNN defines a continuous FCM.

4.1.1 Generative Model and Interventions

A CGNN $\mathscr{C}_{\widehat{\mathscr{G}}, \hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ is a **generative** model 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 $[\hat{x}_{1,j}, \ldots, \hat{x}_{d,j}]$ of the estimated distribution $\hat{P}(\hat{\mathbf{X}} = [\hat{X}_1, \ldots, \hat{X}_d])$ by proceeding as follow:

1. Draw $\{[e_{1,j}, \ldots, e_{d,j}]\}_{j=1}^n$, n samples independent identically distributed from the joint distribution of independent noise variables $\mathbf{E} = [E_1, \ldots, E_d]$.
2. Generate n samples $\{[\hat{x}_{1,j}, \ldots, \hat{x}_{d,j}]\}_{j=1}^n$, where each estimate sample $\hat{x}_{i,j}$ of variable $\hat{X}_i$ is computed in the topological order of $\widehat{\mathscr{G}}$ from $\hat{f}_i$ with the jth

estimate samples $\hat{x}_{Pa(i;\hat{\mathscr{G}}),j}$ of $\hat{X}_{\mathrm{Pa}(i;\hat{\mathscr{G}})}$ and the jth sample $e_{i,j}$ of the random noise variable E_i.

Notice that a CGNN generates a probability distribution $\hat{P}$ which is Markov with respect to $\widehat{\mathscr{G}}$, as the graph $\widehat{\mathscr{G}}$ is acyclic and the noise variables E_i are mutually independent.

Importantly, CGNN supports interventions, that is, freezing a variable X_i to some constant v_i. The resulting joint distribution noted $\hat{P}_{\mathrm{do}(\hat{X}_i=v_i)}(\hat{X})$, called *interventional distribution* (Pearl 2009), can be computed from CGNN by discarding all causal influences on $\hat{X}_i$ and clamping its value to v_i. It is emphasized that intervening is different from conditioning (*correlation does not imply causation*). The knowledge of interventional distributions is essential for e.g., public policy makers, wanting to estimate the overall effects of a decision on a given variable.

4.2 Model Evaluation

The goal is to associate to each candidate solution $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ a score reflecting how well this candidate solution describes the observational data. Firstly we define the model scoring function (Sect. 4.2), then we show that this model scoring function allows to build a CGNN generating a distribution $\hat{P}(\hat{X})$ that approximates $P(X)$ with arbitrary accuracy (Sect. 4.2.2).

4.2.1 Scoring Metric

The ideal score, to be minimized, is the distance between the joint distribution P associated with the ground truth FCM, and the joint distribution $\widehat{P}$ defined by the CGNN candidate $\mathscr{C}_{\hat{\mathscr{G}},\hat{f}} = (\hat{\mathscr{G}}, \hat{f}, \mathscr{E})$. A tractable approximation thereof is given by the Maximum Mean Discrepancy (MMD) (Gretton et al. 2007) between the n-sample observational data $\mathscr{D}$, and an n-sample $\widehat{\mathscr{D}}$ sampled after $\widehat{P}$. Overall, the CGNN $\mathscr{C}_{\hat{\mathscr{G}},\hat{f}}$ is trained by minimizing

$$S(\mathscr{C}_{\hat{\mathscr{G}},\hat{f}}, \mathscr{D}) = \widehat{\mathrm{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) + \lambda|\widehat{\mathscr{G}}|, \tag{7}$$

with $\widehat{\mathrm{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}})$ defined as:

$$\widehat{\mathrm{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) = \frac{1}{n^2}\sum_{i,j=1}^{n} k(x_i, x_j) + \frac{1}{n^2}\sum_{i,j=1}^{n} k(\hat{x}_i, \hat{x}_j) - \frac{2}{n^2}\sum_{i,j=1}^{n} k(x_i, \hat{x}_j) \tag{8}$$

where kernel k usually is taken as the Gaussian kernel ($k(x, x') = \exp(-\gamma\|x - x'\|_2^2)$). The MMD statistic, with quadratic complexity in the sample size, has the

good property that as n goes to infinity, it goes to zero iff $P = \hat{P}$ (Gretton et al. 2007). For scalability, a linear approximation of the MMD statistics based on $m = 100$ random features (Lopez-Paz 2016), called $\widehat{\text{MMD}}_k^m$, will also be used in the experiments (more in section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix).

Due to the Gaussian kernel being differentiable, $\widehat{\text{MMD}}_k$ and $\widehat{\text{MMD}}_k^m$ are differentiable, and backpropagation can be used to learn the CGNN made of networks $\hat{f}_i$ structured along $\hat{\mathscr{G}}$.

In order to compare candidate solutions with different structures in a fair manner, the evaluation score of Eq. (7) is augmented with a penalization term $\lambda|\widehat{\mathscr{G}}|$, with $|\widehat{\mathscr{G}}|$ the number of edges in $\hat{\mathscr{G}}$. Penalization weight λ is a hyper-parameter of the approach.

4.2.2 Representational Power of CGNN

We note $\mathscr{D} = \{[x_{1,j}, \ldots, x_{d,j}]\}_{j=1}^n$, the data samples independent identically distributed after the (unknown) joint distribution $P(\mathbf{X} = [X_1, \ldots, X_d])$, also referred to as observational data.

Under same conditions as in Proposition 1, ($P(X)$ being decomposable along graph $\mathscr{G}$, with continuous and strictly positive joint density function on a compact in $\mathbb{R}^d$ and zero elsewhere), there exists a CGNN $(\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$, that approximates $P(X)$ with arbitrary accuracy:

Proposition 2 *For $m \in [[1, d]]$, let Z_m denote the set of variables with topological order less than m and let d_m be its size. For any d_m-dimensional vector of noise values $e^{(m)}$, let $z_m(e^{(m)})$ (resp. $\widehat{z_m}(e^{(m)})$) be the vector of values computed in topological order from the FCM $(\mathscr{G}, f, \mathscr{E})$ (resp. the CGNN $(\mathscr{G}, \hat{f}, \mathscr{E})$). For any $\epsilon > 0$, there exists a set of networks $\hat{f}$ with architecture $\mathscr{G}$ such that*

$$\forall e^{(m)}, \|z_m(e^{(m)}) - \widehat{z_m}(e^{(m)})\| < \epsilon \tag{9}$$

Proof In section "Proofs" in Appendix.

Using this proposition and the $\widehat{\text{MMD}}_k$ scoring criterion presented in Eq. (8), it is shown that the distribution $\hat{P}$ of the CGNN can estimate the true observational distribution of the (unknown) FCM up to an arbitrary precision, under the assumption of an infinite observational sample:

Proposition 3 *Let $\mathscr{D}$ be an infinite observational sample generated from $(\mathscr{G}, f, \mathscr{E})$. With same notations as in Proposition 2, for every sequence ϵ_t, such that $\epsilon_t > 0$ and goes to zero when $t \to \infty$, there exists a set $\widehat{f}_t = (\hat{f}_1^t \ldots \hat{f}_d^t)$ such that $\widehat{MMD}_k$ between $\mathscr{D}$ and an infinite size sample $\widehat{\mathscr{D}}_t$ generated from the CGNN $(\mathscr{G}, \widehat{f}_t, \mathscr{E})$ is less than ε_t.*

Proof In section "Proofs" in Appendix.

Under these assumptions, as $\widehat{\text{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}_t) \to 0$, as $t \to \infty$, it implies that the sequence of generated $\hat{P}_t$ converges in distribution toward the distribution P of the observed sample (Gretton et al. 2007). This result highlights the generality of this approach as we can model any kind of continuous FCM from observational data (assuming access to infinite observational data). Our class of model is not restricted to simplistic assumptions on the data generative process such as the additivity of the noise or linear causal mechanisms. But this strength comes with a new challenge relative to identifiability of such CGNNs as the result of Proposition 3 holds for any DAG $\widehat{\mathscr{G}}$ such that P can be factorized along $\mathscr{G}$ and then for any DAG in the Markov equivalence class of $\mathscr{G}$ (under classical assumption of CMA, CFA and CSA). In particular in the pairwise setting, when only two variables X and Y are observed, the joint distribution $P(X, Y)$ can be factorized in two Markov equivalent DAGs $X \to Y$ or $Y \to X$ as $P(X, Y) = P(X)P(Y|X)$ and $P(X, Y) = P(Y)P(X|Y)$. Then the CGNN can reproduce equally well the observational distribution in both directions (under the assumption of Proposition 1). We refer the reader to Zhang and Hyvärinen (2009) for more details on this problem of identifiability in the bivariate case.

As shown in Sect. 4.3.3, the proposed approach enforces the discovery of causal models in the Markov equivalence class. Within this class, the non-identifiability issue is empirically mitigated by restricting the class of CGNNs considered, and specifically limiting the number n_h of hidden neurons in each causal mechanism (Eq. 6). Formally, we restrict ourselves to the sub-class of CGNNs, noted $\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}^{n_h}} = (\widehat{\mathscr{G}}, \hat{f}^{n_h}, \mathscr{E})$ with exactly n_h hidden neurons in each $\hat{f}_i$ mechanism. Accordingly, any candidate $\hat{\mathscr{G}}$ with number of edges $|\hat{\mathscr{G}}|$ involves the same number of parameters: $(2d + |\hat{\mathscr{G}}|) \times n_h$ weights and $d \times (n_h + 1)$ bias parameters. As shown experimentally in Sect. 5, this parameter n_h is crucial as it governs the CGNN ability to model the causal mechanisms: too small n_h, and data patterns may be missed; too large n_h, and overly complicated causal mechanisms may be retained.

4.3 Model Optimization

Model optimization consists at finding a (nearly) optimum solution $(\hat{\mathscr{G}}, \hat{f})$ in the sense of the score defined in the previous section. The so-called *parametric* optimization of the CGNN, where structure estimate $\hat{\mathscr{G}}$ is fixed and the goal is to find the best neural estimates $\hat{f}$ conditionally to $\hat{\mathscr{G}}$ is tackled in Sect. 4.3.1. The *non-parametric* optimization, aimed at finding the best structure estimate, is considered in Sect. 4.3.2. In Sect. 4.3.3, we present an identifiability result for CGNN up to Markov equivalence classes.

4.3.1 Parametric (Weight) Optimization

Given the acyclic structure estimate $\hat{\mathscr{G}}$, the neural networks $\hat{f}_1, \ldots, \hat{f}_d$ of the CGNN are learned end-to-end using backpropagation with Adam optimizer (Kingma and Ba 2014) by minimizing losses $\widehat{\text{MMD}}_k$ (Eq. (8), referred to as **CGNN** ($\widehat{\text{MMD}}_k$)) or $\widehat{\text{MMD}}_k^m$ (see section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix, **CGNN** ($\widehat{\text{MMD}}_k^m$)).

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, independent and identically distributed, are drawn in each epoch. In the $\widehat{\text{MMD}}_k^m$ variant, the parameters of the random kernel are resampled from their respective distributions in each training epoch (see section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix). After training, the score is computed and averaged over n_{eval} estimated samples of size n. Likewise, the noise samples are re-sampled anew for each evaluation sample. The overall process with training and evaluation is repeated nb_{run} times to reduce stochastic effects relative to random initialization of neural network weights and stochastic gradient descent.

4.3.2 Non-parametric (Structure) Optimization

The number of directed acyclic graphs $\hat{\mathscr{G}}$ over d nodes is super-exponential in d, making the non-parametric optimization of the CGNN structure an intractable computational and statistical problem. Taking inspiration from Tsamardinos et al. (2006); Nandy et al. (2015), we start from a graph skeleton recovered by other methods such as feature selection (Yamada et al. 2014). We focus on optimizing the edge orientations. Letting L denote the number of edges in the graph, it defines a combinatorial optimization problem of complexity $\mathscr{O}(2^L)$ (note however that not all orientations are admissible since the eventual oriented graph must be a DAG).

The motivation for this approach is to decouple the edge selection task and the causal modeling (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 Sect. 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 score $S(\mathscr{C}_{X_i \to X_j, \hat{f}}, \mathscr{D}_{ij})$ and $S(\mathscr{C}_{X_j \to X_i, \hat{f}}, \mathscr{D}_{ij})$ are computed, where $\mathscr{D}_{ij} = \{[x_{i,l}, x_{j,l}]\}_{l=1}^{n}$. The best orientation corresponding to a minimum 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 optimization, below.
- The optimization of the DAG structure is achieved using a hill-climbing algorithm aimed to optimize the global score $S(\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}}, \mathscr{D})$. Iteratively, (1) an edge $X_i - X_j$ is uniformly randomly selected in the current graph; (2) 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; (3) if this graph obtains a lower 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.

At the end of the process each causal edge $X_i \to X_j$ in $\mathscr{G}$ is associated with a score, measuring its contribution to the global score:

$$S_{X_i \to X_j} = S(\mathscr{C}_{\hat{\mathscr{G}} - \{X_i \to X_j\}, \hat{f}}, \mathscr{D}) - S(\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}}, \mathscr{D}) \tag{10}$$

During the structure (non-parametric) optimization, the graph skeleton is fixed; no edge is added or removed. The penalization term $\lambda|\widehat{\mathscr{G}}|$ entering in the score evaluation (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 Sect. 6 to compare structures with different skeletons, as the potential confounding factors will be dealt with by removing edges.

4.3.3 Identifiability of CGNN up to Markov Equivalence Classes

Assuming an infinite number of observational data, and assuming further that the generative distribution belongs to the CGNN class $\mathscr{C}_{\mathscr{G}, f}$, then there exists a DAG reaching an MMD score of 0 in the Markov equivalence class of $\mathscr{G}$:

Proposition 4 *Let $\boldsymbol{X} = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, generated by a CGNN $\mathscr{C}_{\mathscr{G}, f} = (\mathscr{G}, f, \mathscr{E})$ with $\mathscr{G}$ a directed acyclic graph. Let $\mathscr{D}$ be an infinite observational sample generated from this CGNN. We assume that P is Markov and faithful to the graph $\mathscr{G}$, and that every pair of*

variables (X_i, X_j) *that are d-connected in the graph are not independent. We note* $\widehat{\mathscr{D}}$ *an infinite sample generated by a candidate CGNN,* $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$*. Then,*

(i) If $\widehat{\mathscr{G}} = \mathscr{G}$ *and* $\hat{f} = f$*, then* $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) = 0$*.*
(ii) For any graph $\widehat{\mathscr{G}}$ *characterized by the same adjacencies but not belonging to the Markov equivalence class of* $\mathscr{G}$*, for all* $\hat{f}$*,* $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) \neq 0$*.*

Proof In section "Proofs" in Appendix.

This result does not establish the CGNN identifiability within the Markov class of equivalence, that is left for future work. As shown experimentally in Sect. 5.1, there is a need to control the model capacity in order to recover the directed graph in the Markov equivalence class.[4]

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 2.7 Ghz (CPU) or on Nvidia GTX 1080Ti graphics card (GPU).

5.1 Experimental Setting

The CGNN architecture is a 1-hidden layer network with ReLU activation function. The multi-scale Gaussian kernel used in the MMD scores has bandwidth 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). In this section the distribution $\mathscr{E}$ of the noise variables is set to $\mathscr{N}(0, 1)$. 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 1500 sample dataset is generated from the linear structural equation model with additive uniform noise $Y = X + \mathscr{U}(0, 0.5)$, $X \sim U([-2, 2])$ (Fig. 5). Both CGNNs associated to $X \to Y$ and $Y \to X$ are trained

[4] In some specific cases, such as in the bivariate linear FCM with Gaussian noise and Gaussian input, even by restricting the class of functions considered, the DAG cannot be identified from purely observational data (Mooij et al. 2016).

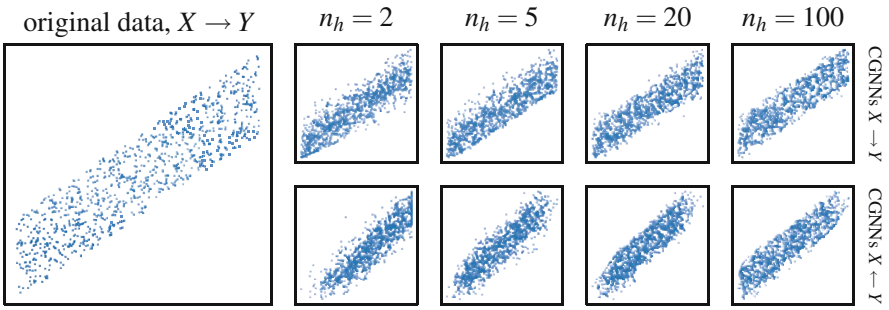
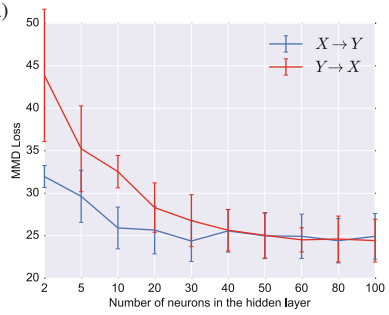


Fig. 5 Leftmost: Data samples. Columns 2–5: Estimate samples generated from CGNN with direction $X \to Y$ (top row) and $Y \to X$ (bottom row) for number of hidden neurons $n_h = 2, 5, 20, 100$

(a)



(b)

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

Fig. 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). (**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

until reaching convergence ($n_{epoch} = 1000$) 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 (cf. Sect. 4.3.3 above).

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* is based on comparing the model scores (Sect. 4.2) attached to both CGNNs.

5.2.1 Benchmarks

Five datasets with continuous variables are considered[5]:

- **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 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 1500 for the sake of an acceptable overall computational load.

5.2.2 Baseline Approaches

CGNN is assessed comparatively to the following algorithms[6]: (1) ANM (Mooij et al. 2016) with Gaussian process regression and HSIC independence test of the residual; (2) a pairwise version of LiNGAM (Shimizu et al. 2006) relying on Independent Component Analysis to identify the linear relations between variables; (3) IGCI (Daniusis et al. 2012) with entropy estimator and Gaussian reference measure; (4) the post-nonlinear model (PNL) with HSIC test (Zhang and Hyvärinen 2009); (5) GPI-MML (Stegle et al. 2010); where the Gaussian process regression with higher marginal likelihood is selected as causal direction; (6) CDS, retaining the causal orientation with lowest variance of the conditional probability distribution; (7) 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.

[5]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/.

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

5.2.3 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 $\{5, \ldots, 100\}$. The leave-one-dataset-out procedure sets this hyper-parameter n_h to values between 20 and 40 for the different datasets. For ANM and the bivariate fit, the kernel parameter for the Gaussian process regression ranges over $\{0.01, \ldots, 10\}$. For PNL, the threshold parameter alpha for the HSIC independence test ranges over $\{0.0005, \ldots, 0.5\}$. 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 4000 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.

5.2.4 Empirical Results

Figure 7 reports the area under the precision/recall curve for each benchmark and all algorithms.

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 real dataset **CE-Tueb**; this counter-performance is blamed on the differences between all five benchmark distributions and the lack of generalization/transfer learning.

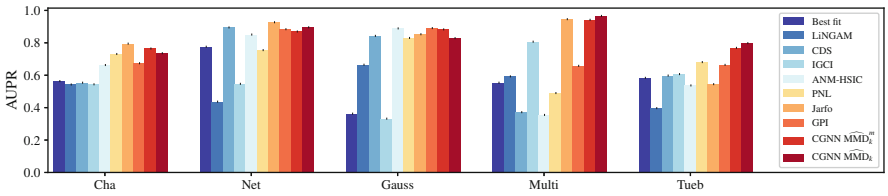


Fig. 7 Bivariate causal modelling: area under the precision/recall curve for the five datasets. A full table of the scores is given on Table 3 in section "Table of Scores for the Experiments on Cause-Effect Pairs" in Appendix

Lastly, generative methods GPI and **CGNN** ($\widehat{\text{MMD}}_k$) perform well on most datasets, including the real-world cause-effect pairs CE-Tüb, in counterpart for a higher computational cost (resp. 32 min on CPU for GPI and 24 min on GPU for CGNN). Using the linear MMD approximation Lopez-Paz (2016), **CGNN** ($\widehat{\text{MMD}}_k^m$) as explained in section "The Maximum Mean Discrepancy (MMD) Statistic" in Appendix reduces the cost by a factor of 5 without hindering the performance.

Overall, CGNN demonstrates competitive performance on the cause-effect inference problem, where it is necessary to discover distributional asymmetries.

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.

Given skeleton $A - B - C$, each dataset is used to model the possible four CGNN structures (Fig. 8, 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: $B = f_1(E_1)$, $A = f_2(B, E_2)$, $C = f_3(B, E_3)$

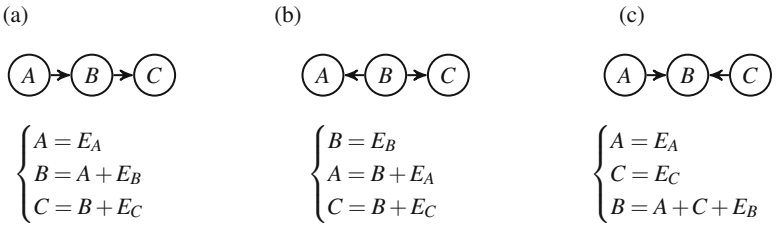


Fig. 8 Datasets generated from the three DAG configurations with skeleton $A - B - C$. (**a**) Chain structure. (**b**) Reversed v-structure. (**c**) V-structure

Table 1 CGNN-MMD scores for all models on all datasets

Score	Non v-structures		v-structure
	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)
$C_{Vstructure}$	0.202 (0.004)	0.180 (0.005)	**0.127** (0.005)

Smaller scores indicate a better match. CGNN correctly identifies v-structure vs. other structures. Bold value corresponds to best match for v-structure

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 1 (average over 64 runs; the standard deviation is indicated in parenthesis).

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 $\mathbf{X} = [X_1, \ldots, X_d]$ be a set of continuous variables, satisfying the Causal Markov, faithfulness and causal sufficiency assumptions. To that end, all experiments provide all algorithms *the true graph skeleton*, so their ability to orient edges is compared in a fair way. This allows us to separate the task of orienting the graph from that of uncovering the skeleton.

5.4.1 Results on Artificial Graphs with Additive and Multiplicative Noises

We draw 500 samples from 20 training artificial causal graphs and 20 test artificial causal graphs on 20 variables. Each variable has a number of parents uniformly drawn in $[[0, 5]]$; f_is are randomly generated polynomials involving additive/multiplicative noise.[7]

We compare CGNN to the PC algorithm (Spirtes et al. 1993), the score-based methods GES (Chickering 2002), LiNGAM (Shimizu et al. 2006), causal additive

[7]The data generator is available at https://github.com/GoudetOlivie/CGNN. The datasets considered are available at http://dx.doi.org/10.7910/DVN/UZMB69.

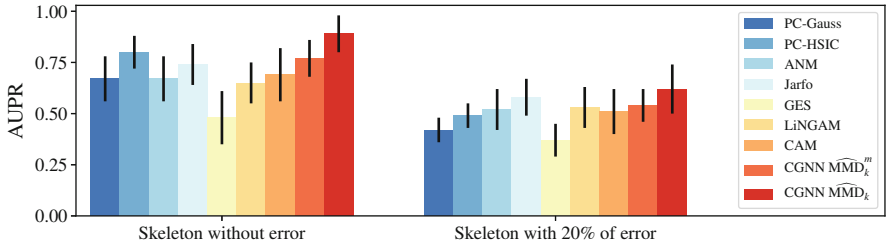


Fig. 9 Average (std. dev.) AUPR results for the orientation of 20 artificial graphs given true skeleton (left) and artificial graphs given skeleton with 20% error (right). A full table of the scores, including the metrics Structural Hamming Distance (SHD) and Structural Intervention (SID) (Peters and Bühlmann 2013) is shown on Table 4 in section "Table of Scores for the Experiments on Graphs" in Appendix

model (CAM) (Bühlmann et al. 2014) and with the pairwise methods ANM and Jarfo. For PC, we employ the better-performing, order-independent version of the PC algorithm proposed by Colombo and Maathuis (2014). PC needs the specification of a conditional independence test. We compare PC-Gaussian, which employs a Gaussian conditional independence test on Fisher z-transformations, and PC-HSIC, which uses the HSIC conditional independence test with the Gamma approximation (Gretton et al. 2005). PC and GES are implemented in the *pcalg* package (Kalisch et al. 2012).

All hyperparameters are set on the training graphs in order to maximize the Area Under the Precision/Recall score (AUPR). For the Gaussian conditional independence test and the HSIC conditional independence test, the significance level achieving best result on the training set are respectively 0.1 and 0.05 . For GES, the penalization parameter is set to 3 on the training set. For CGNN, n_h is set to 20 on the training set. For CAM, the cutoff value is set to 0.001.

Figure 9 (left) displays the performance of all algorithms obtained by starting from the exact skeleton on the test set of artificial graphs and measured from the AUPR (Area Under the Precision/Recall curve), the Structural Hamming Distance (SHD, the number of edge modifications to transform one graph into another) and the Structural Intervention Distance (SID, the number of equivalent two-variable interventions between two graphs) (Peters and Bühlmann 2013).

CGNN obtains significant better results with SHD and SID compared to the other algorithms when the task is to discover the causal from the true skeleton. One resulting graph is shown on Fig. 10. There are three mistakes on this graph (red edges) (in lines with an SHD on average of 2.5).

Constraints based method PC with powerful HSIC conditional independence test is the second best performing method. It highlights the fact that when the skeleton is known, exploiting the structure of the graph leads to good results compared to pairwise methods using only local information. Notably, as seen on Fig. 10, this type of DAG has a lot of v-structures, as many nodes have more than one parent in the graph, but this is not always the case as shown in the next subsection.

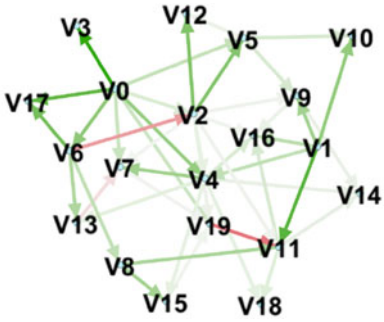


Fig. 10 Orientation by CGNN of artificial graph with 20 nodes. Green edges are good orientation and red arrows false orientation. Three edges are red and 42 are green. The strength of the line refers to the confidence of the algorithm

Overall CGNN and PC-HSIC are the most computationally expensive methods, taking an average of 4 h on GPU and 15 h on CPU, respectively.

The robustness of the approach is validated by randomly perturbing 20% edges in the graph skeletons provided to all algorithms (introducing about 10 false edges over 50 in each skeleton). As shown on Table 4 (right) in Appendix, and as could be expected, the scores of all algorithms are lower when spurious edges are introduced. Among the least robust methods are constraint-based methods; a tentative explanation is that they heavily rely on the graph structure to orient edges. By comparison pairwise methods are more robust because each edge is oriented separately. As CGNN leverages conditional independence but also distributional asymmetry like pairwise methods, it obtains overall more robust results when there are errors in the skeleton compared to PC-HSIC. However one can notice that a better SHD score is obtained by CAM, on the skeleton with 20% error. This is due to the exclusive last edge pruning step of CAM, which removes spurious links in the skeleton.

CGNN obtains overall good results on these artificial datasets. It offers the advantage to deliver a full generative model useful for simulation (while e.g., Jarfo and PC-HSIC only give the causality graph). To explore the scalability of the approach, five artificial graphs with 100 variables have been considered, achieving an AUPRC of 85.5 ± 4, in 30 h of computation on four NVIDIA 1080Ti GPUs.

5.4.2 Result on Biological Data

We now evaluate CGNN on biological networks. First we apply it on simulated gene expression data and then on real protein network.

Syntren Artificial Simulator

First we apply CGNN on SynTREN (Van den Bulcke et al. 2006) from sub-networks of E. coli (Shen-Orr et al. 2002). SynTREN creates synthetic transcriptional regulatory networks and produces simulated gene expression data that approximates

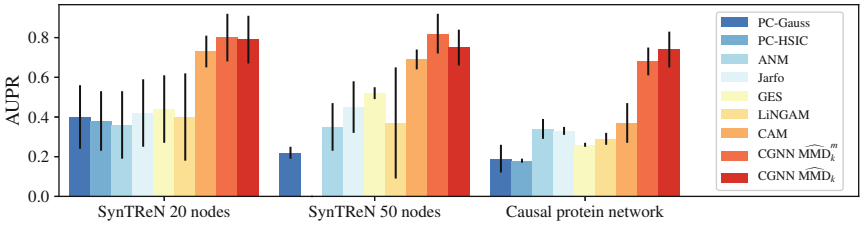


Fig. 11 Average (std. dev.) AUPR results for the orientation of 20 artificial graphs generated with the SynTReN simulator with 20 nodes (left), 50 nodes (middle), and real protein network given true skeleton (right). A full table of the scores, including the metrics Structural Hamming Distance (SHD) and Structural Intervention (SID) (Peters and Bühlmann 2013) is included in section "Table of Scores for the Experiments on Graphs" in Appendix

experimental data. Interaction kinetics are modeled by complex mechanisms based on Michaelis-Menten and Hill kinetics (Mendes et al. 2003).

With Syntren, we simulate 20 subnetworks of 20 nodes and 5 subnetworks with 50 nodes. For the sake of reproducibility, we use the random seeds of $0, 1 \ldots 19$ and $0, 1 \ldots 4$ for each graph generation with respectively 20 nodes and 50 nodes. The default Syntren parameters are used: a probability of 0.3 for complex 2-regulator interactions and a value of 0.1 for Biological noise, experimental noise and Noise on correlated inputs. For each graph, Syntren give us expression datasets with 500 samples.

Figure 11 (left and middle) and Table 5 in section "Table of Scores for the Experiments on Graphs" in Appendix display the performance of all algorithms obtained by starting from the exact skeleton of the causal graph with same hyper-parameters as in the previous subsection. As a note, we canceled the PC-HSIC algorithm after 50 h of running time.

Constraint based methods obtain low score on this type of graph dataset. It may be explained by the type of structure involved. Indeed as seen of Fig. 12, there are very few v-structures in this type of network, making impossible the orientation of an important number of edges by using only conditional independence tests. Overall the methods CAM and CGNN that take into account of both distributional asymmetry and multivariate interactions, get the best scores. CGNN obtain the best results in AUPR, SHD and SID for graph with 20 nodes and 50 nodes, showing that this method can be used to infer networks having complex distribution, complex causal mechanisms and interactions. The Fig. 12 shows the resulting graph obtain with CGNN. Edges with good orientation are displayed in green and edge with false orientation in red.

5.4.3 Results on Biological Real-World Data

CGNN is applied to the protein network problem Sachs et al. (2005), using the Anti-CD3/CD28 dataset with 853 observational data points corresponding to general

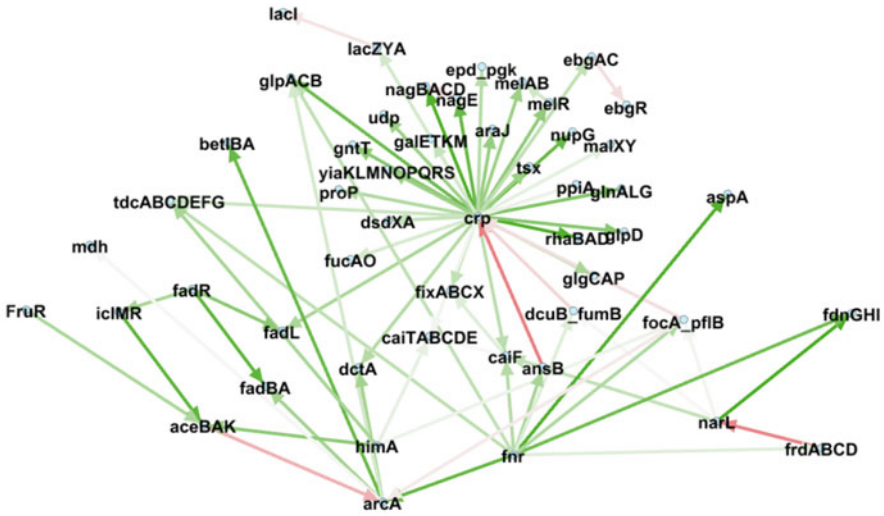


Fig. 12 Orientation by CGNN of E. coli subnetwork with 50 nodes and corresponding to Syntren simulation with random seed 0. Green edges are good orientation and red arrows false orientation. The strength of the line refers to the confidence of the algorithm

perturbations without specific interventions. All algorithms were given the skeleton of the causal graph (Sachs et al. 2005, Fig. 2) with same hyper-parameters as in the previous subsection. We run each algorithm on 10-fold cross-validation. Table 6 in Appendix reports average (std. dev.) results.

Constraint-based algorithms obtain surprisingly low scores, because they cannot identify many v-structures in this graph. We confirm this by evaluating conditional independence tests for the adjacent tuples of nodes *pip3-akt-pka*, *pka-pmek-pkc*, *pka-raf-pkc* and we do not find strong evidences for v-structure. Therefore methods based on distributional asymmetry between cause and effect seem better suited to this dataset. CGNN obtains good results compared to the other algorithms. Notably, Fig. 13 shows that CGNN is able to recover the strong signal transduction pathway *raf*$\rightarrow$*mek*$\rightarrow$*erk* reported in Sachs et al. (2005) and corresponding to clear direct enzyme-substrate causal effect. CGNN gives important scores for edges with good orientation (green line), and low scores (thinnest edges) to the wrong edges (red line), suggesting that false causal discoveries may be controlled by using the confidence scores defined in Eq. (10).

6 Towards Predicting Confounding Effects

In this subsection we propose an extension of our algorithm relaxing the causal sufficiency assumption. We are still assuming the Causal Markov and faithfulness assumptions, thus three options have to be considered for each edge (X_i, X_j) of the

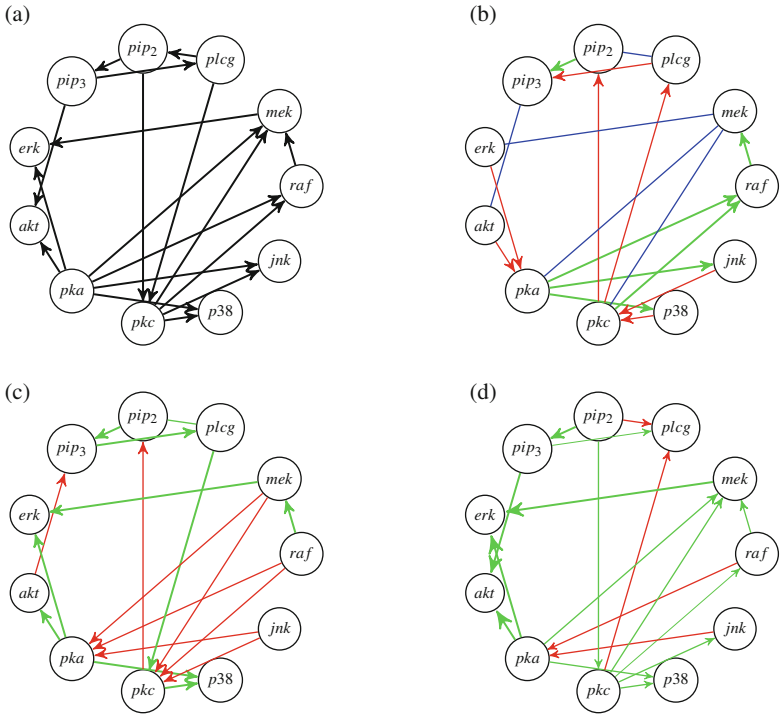


Fig. 13 Causal protein network. (**a**) Ground truth. (**b**) GES. (**c**) CAM. (**d**) CGNN

skeleton representing a direct dependency: $X_i \rightarrow X_j$, $X_j \rightarrow X_i$ and $X_i \leftrightarrow X_j$ (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.

We use such new models with correlated noise to study the robustness of our graph reconstruction algorithm to increasing violations of causal sufficiency, by occluding variables from our datasets. For example, consider the FCM on $\mathbf{X} = [X_1, \ldots, X_5]$ that was presented on Fig. 1. If variable X_1 would be missing from data, the correlated noise $E_{2,3}$ would be responsible for the existence of a double headed arrow connection $X_2 \leftrightarrow X_3$ in the skeleton of our new type of model. The resulting FCM is shown in Fig. 14. Notice that direct causal effects such as $X_3 \rightarrow X_5$ or $X_4 \rightarrow X_5$ may persist, even in presence of possible confounding effects.

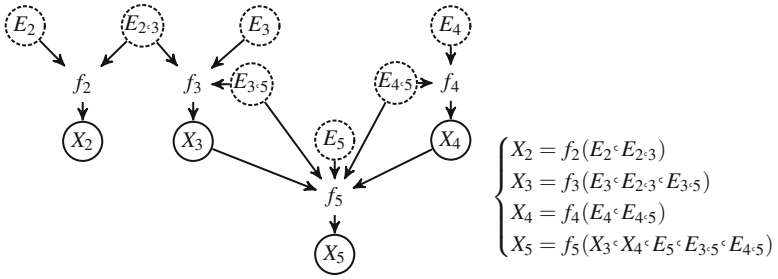


Fig. 14 The Functional Causal Model (FCM) on $\mathbf{X} = [X_1, \ldots, X_5]$ with the missing variable X_1

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

$$X_i \leftarrow f_i(X_{\mathrm{Pa}(i;\mathscr{G})}, E_i, E_{\mathrm{Ne}(i;\mathscr{S})}), \tag{11}$$

where $\mathrm{Ne}(i;\mathscr{S})$ is the set of indices of all the variables adjacent to variable X_i in the skeleton $\mathscr{S}$.

One can notice that this model corresponds to the most general formulation of the FCM with potential confounders for each pair of variables in a given skeleton (representing direct dependencies) where each random variable $E_{i,j}$ summarizes all the unknown influences of (possibly multiple) hidden variables influencing the two variables X_i and X_j.

Here we make a clear distinction between the directed acyclic graph denoted $\mathscr{G}$ and the skeleton $\mathscr{S}$. Indeed, due to the presence of confounding correlated noise, any variable in $\mathscr{G}$ can be removed without altering $\mathscr{S}$. We use the same generative neural network to model the new FCM presented in Eq. (11). 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 $\mathscr{G}$, the functions $\hat{f}_1, \ldots, \hat{f}_d$ 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.

All edges are evaluated with these correlated noises, the goal being to see whether introducing a correlated noise explains the dependence between the two variables X_i and X_j.

As mentioned before, the score used by CGNN is:

$$S(\mathscr{C}_{\hat{\mathscr{G}},\hat{f}}, \mathscr{D}) = \widehat{\mathrm{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) + \lambda|\widehat{\mathscr{G}}| \tag{12}$$

where $|\widehat{\mathscr{G}}|$ is the total number of edges in the DAG. In the graph search, for any given edge, we compare the score associated to the graph considered with and

without this edge. If the contribution of this edge is negligible compared to a given threshold lambda, the edge is considered as spurious.

The non-parametric optimization of the $\hat{\mathscr{G}}$ structure is also achieved using a Hill-Climbing algorithm; in each step an edge of $\mathscr{S}$ is randomly drawn and modified in $\hat{\mathscr{G}}$ using one out of the possible three operators: reverse the edge, add an edge and remove an edge. Other algorithmic details are as in Sect. 4.3.2: the greedy search optimizes the penalized loss function (Eq. 12). For CGNN, we set the hyperparameter $\lambda = 5 \times 10^{-5}$ fitted on the training graph dataset.

The algorithm stops when no improvement is obtained. Each causal edge $X_i \rightarrow X_j$ in $\mathscr{G}$ is associated with a score, measuring its contribution to the global score:

$$S_{X_i \rightarrow X_j} = S(\mathscr{C}_{\hat{\mathscr{G}}-\{X_i \rightarrow X_j\}, \hat{f}}, \mathscr{D}) - S(\mathscr{C}_{\hat{\mathscr{G}}, \hat{f}}, \mathscr{D}) \tag{13}$$

Missing edges are associated with a score 0.

6.2 Experimental Validation

6.2.1 Benchmarks

The empirical validation of this extension of CGNN is conducted on same benchmarks as in Sect. 5.4 ($\mathscr{G}_i$, $i \in [[2, 5]]$), where three variables (causes for at least two other variables in the graph) have been randomly removed.[8] 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.

6.2.2 Baselines

CGNN is compared with state of art methods: (1) 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. For CGNN, we set the hyperparameter $\lambda = 5 \times 10^{-5}$ fitted on the training graph dataset. Jarfo is trained on the 16,200 pairs of the cause-effect pair challenge (Guyon 2013, 2014) to detect for each pair of variable if $X_i \rightarrow Y_i$, $Y_i \rightarrow X_i$ or $X_i \leftrightarrow Y_i$.

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

Table 2 AUPR, SHD and SID on causal discovery with confounders

Method	AUPR	SHD	SID
RFCI-Gaussian	0.22 (0.08)	21.9 (7.5)	174.9 (58.2)
RFCI-HSIC	0.41 (0.09)	17.1 (6.2)	124.6 (52.3)
Jarfo	0.54 (0.21)	20.1 (14.8)	98.2 (49.6)
CGNN ($\widehat{\text{MMD}}_k$)	0.71[a] (0.13)	11.7[a] (5.5)	53.55[a] (48.1)

[a]Denotes significance at $p = 10^{-2}$

6.2.3 Results

Comparative performances are shown in Table 2, reporting the area under the precision/recall curve. Overall, these results confirm the robustness of the CGNN 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 or are removed.

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.

CGNN 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. We believe that our approach opens new avenues of research, both from the point of view of leveraging the power of deep learning in causal discovery and from the point of view of building deep networks with better structure *interpretability*. Once the model is learned, the CGNNs present the advantage to be fully parametrized and may be used to simulate interventions on one or more variables of the model and evaluate their impact on a set of target variables. This usage is relevant in a wide variety of domains, typically among medical and sociological domains.

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 could be tackled by using embedded framework for the structure optimization (inspired by lasso methods). Another perspective regards the extension of the approach to categorical variables.

Appendix

The Maximum Mean Discrepancy (MMD) Statistic

The Maximum Mean Discrepancy (MMD) statistic (Gretton et al. 2007) measures the distance between two probability distributions P and $\hat{P}$, defined over $\mathbb{R}^d$, as the real-valued quantity

$$\mathrm{MMD}_k(P, \hat{P}) = \left\| \mu_k(P) - \mu_k(\hat{P}) \right\|_{\mathscr{H}_k}.$$

Here, $\mu_k = \int k(x, \cdot) \mathrm{d}P(x)$ is the *kernel mean embedding* of the distribution P, according to the real-valued symmetric kernel function $k(x, x') = \langle k(x, \cdot), k(x', \cdot) \rangle_{\mathscr{H}_k}$ with associated reproducing kernel Hilbert space $\mathscr{H}_k$. Therefore, μ_k summarizes P as the expected value of the features computed by k over samples drawn from P.

In practical applications, we do not have access to the distributions P and $\hat{P}$, but to their respective sets of samples $\mathscr{D}$ and $\hat{\mathscr{D}}$, defined in Sect. 4.2.1. In this case, we approximate the kernel mean embedding $\mu_k(P)$ by the *empirical kernel mean embedding* $\mu_k(\mathscr{D}) = \frac{1}{|\mathscr{D}|} \sum_{x \in \mathscr{D}} k(x, \cdot)$, and respectively for $\hat{P}$. Then, the empirical MMD statistic is

$$\begin{aligned} \widehat{\mathrm{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}) &= \left\| \mu_k(\mathscr{D}) - \mu_k(\hat{\mathscr{D}}) \right\|_{\mathscr{H}_k} \\ &= \frac{1}{n^2} \sum_{i,j}^{n} k(x_i, x_j) + \frac{1}{n^2} \sum_{i,j}^{n} k(\hat{x}_i, \hat{x}_j) - \frac{2}{n^2} \sum_{i,j}^{n} k(x_i, \hat{x}_j). \end{aligned}$$

Importantly, the empirical MMD tends to zero as $n \to \infty$ if and only if $P = \hat{P}$, as long as k is a characteristic kernel (Gretton et al. 2007). This property makes the MMD an excellent choice to model how close the observational distribution P is to the estimated observational distribution $\hat{P}$. Throughout this paper, we will employ a particular characteristic kernel: the Gaussian kernel $k(x, x') = \exp(-\gamma \|x - x'\|_2^2)$, where $\gamma > 0$ is a hyperparameter controlling the smoothness of the features.

In terms of computation, the evaluation of $\mathrm{MMD}_k(\mathscr{D}, \hat{\mathscr{D}})$ takes $O(n^2)$ time, which is prohibitive for large n. When using a shift-invariant kernel, such as the Gaussian kernel, one can invoke Bochner's theorem (Edwards 1964) to obtain a linear-time approximation to the empirical MMD (Lopez-Paz et al. 2015), with form

$$\widehat{\mathrm{MMD}}_k^m(\mathscr{D}, \hat{\mathscr{D}}) = \left\| \hat{\mu}_k(\mathscr{D}) - \hat{\mu}_k(\hat{\mathscr{D}}) \right\|_{\mathbb{R}^m}$$

and $O(mn)$ evaluation time. Here, the *approximate empirical kernel mean embedding* has form

$$\hat{\mu}_k(\mathscr{D}) = \sqrt{\frac{2}{m}} \frac{1}{|\mathscr{D}|} \sum_{x \in \mathscr{D}} [\cos(\langle w_1, x\rangle + b_1), \ldots, \cos(\langle w_m, x\rangle + b_m)]\,,$$

where w_i is drawn from the normalized Fourier transform of k, and $b_i \sim U[0, 2\pi]$, for $i = 1, \ldots, m$. In our experiments, we compare the performance and computation times of both $\widehat{\mathrm{MMD}}_k$ and $\widehat{\mathrm{MMD}}_k^m$.

Proofs

Proposition 1 *Let $\boldsymbol{X} = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, and further assume that the joint density function h of P is continuous and strictly positive on a compact and convex subset of $\mathbb{R}^d$, and zero elsewhere. Letting $\mathscr{G}$ be a DAG such that P can be factorized along $\mathscr{G}$,*

$$P(X) = \prod_i P(X_i | X_{Pa(i;\mathscr{G})})$$

there exists $f = (f_1, \ldots, f_d)$ with f_i a continuous function with compact support in $\mathbb{R}^{|Pa(i;\mathscr{G})|} \times [0, 1]$ such that $P(X)$ equals the generative model defined from FCM $(\mathscr{G}, f, \mathscr{E})$, with $\mathscr{E} = \mathscr{U}[0, 1]$ the uniform distribution on $[0, 1]$.

Proof By induction on the topological order of $\mathscr{G}$. Let X_i be such that $|\mathrm{Pa}(i; \mathscr{G})| = 0$ and consider the cumulative distribution $F_i(x_i)$ defined over the domain of X_i $(F_i(x_i) = Pr(X_i < x_i))$. F_i is strictly monotonous as the joint density function is strictly positive therefore its inverse, the quantile function $Q_i : [0, 1] \mapsto dom(X_i)$ is defined and continuous. By construction, $Q_i(e_i) = F_i^{-1}(e_i)$ and setting $Q_i = f_i$ yields the result.

Assume f_i be defined for all variables X_i with topological order less than m. Let X_j with topological order m and Z the vector of its parent variables. For any noise vector $e = (e_i, i \in \mathrm{Pa}(j; \mathscr{G}))$ let $z = (x_i, i \in \mathrm{Pa}(j; \mathscr{G}))$ be the value vector of variables in Z defined from e. The conditional cumulative distribution $F_j(x_j | Z = z) = Pr(X_j < x_j | Z = z)$ is strictly continuous and monotonous wrt x_j, and can be inverted using the same argument as above. Then we can define $f_j(z, e_j) = F_j^{-1}(z, e_j)$.

Let $K_j = dom(X_j)$ and $K_{\mathrm{Pa}(j;\mathscr{G})} = dom(Z)$. We will show now that the function f_j is continuous on $K_{\mathrm{Pa}(j;\mathscr{G})} \times [0, 1]$, a compact subset of $\mathbb{R}^{|\mathrm{Pa}(j;\mathscr{G})|} \times [0, 1]$.

By assumption, there exist $a_j \in \mathscr{R}$ such that, for $(x_j, z) \in K_j \times K_{\mathrm{Pa}(j;\mathscr{G})}$, $F(x_j|z) = \int_{a_j}^{x_j} \frac{h_j(u,z)}{h_j(z)} \mathrm{d}u$, with h_j a continuous and strictly positive density function. For $(a, b) \in K_j \times K_{\mathrm{Pa}(j;\mathscr{G})}$, as the function $(u, z) \to \frac{h_j(u,z)}{h_j(z)}$ is continuous on the compact $K_j \times K_{\mathrm{Pa}(j;\mathscr{G})}$, $\lim_{x_j \to a} F(x_j|z) = \int_{a_j}^{a} \frac{h_j(u,z)}{h_j(z)} \mathrm{d}u$ uniformly on $K_{\mathrm{Pa}(j;\mathscr{G})}$ and

$\lim_{z \to b} F(x_j|z) = \int_{a_j}^{x_j} \frac{h_j(u,b)}{h_j(b)}$ on K_j, according to exchanging limits theorem, F is continuous on (a, b).

For any sequence $z_n \to z$, we have that $F(x_j|z_n) \to F(x_j|z)$ uniformly in x_j. Let define two sequences u_n and $x_{j,n}$, respectively on $[0, 1]$ and K_j, such that $u_n \to u$ and $x_{j,n} \to x_j$. As $F(x_j|z) = u$ has unique root $x_j = f_j(z, u)$, the root of $F(x_j|z_n) = u_n$, that is, $x_{j,n} = f_j(z_n, u_n)$ converge to x_j. Then the function $(z, u) \to f_j(z, u)$ is continuous on $K_{\mathrm{Pa}(i;\mathscr{G})} \times [0, 1]$.

Proposition 2 *For $m \in [[1, d]]$, let Z_m denote the set of variables with topological order less than m and let d_m be its size. For any d_m-dimensional vector of noise values $e^{(m)}$, let $z_m(e^{(m)})$ (resp. $\widehat{z_m}(e^{(m)})$) be the vector of values computed in topological order from the FCM $(\mathscr{G}, f, \mathscr{E})$ (resp. the CGNN $(\mathscr{G}, \hat{f}, \mathscr{E})$). For any $\epsilon > 0$, there exists a set of networks $\hat{f}$ with architecture $\mathscr{G}$ such that*

$$\forall e^{(m)}, \|z_m(e^{(m)}) - \widehat{z_m}(e^{(m)})\| < \epsilon \tag{14}$$

Proof By induction on the topological order of $\mathscr{G}$. Let X_i be such that $|\mathrm{Pa}(i; \mathscr{G})| = 0$. Following the universal approximation theorem Cybenko (1989), as f_i is a continuous function over a compact of $\mathbb{R}$, there exists a neural net $\hat{f}_i$ such that $\|f_i - \hat{f}_i\|_\infty < \epsilon/d_1$. Thus Eq. (14) holds for the set of networks $\hat{f}_i$ for i ranging over variables with topological order 0.

Let us assume that Proposition 2 holds up to m, and let us assume for brevity that there exists a single variable X_j with topological order $m + 1$. Letting $\hat{f}_j$ be such that $\|f_j - \hat{f}_j\|_\infty < \epsilon/3$ (based on the universal approximation property), letting δ be such that for all u $\|\hat{f}_j(u) - \hat{f}_j(u+\delta)\| < \epsilon/3$ (by absolute continuity) and letting $\hat{f}_i$ satisfying Eq. (14) for i with topological order less than m for $min(\epsilon/3, \delta)/d_m$, it comes: $\|(z_m, f_j(z_m, e_j)) - (\hat{z}_m, \hat{f}_j(\hat{z_m}, e_j))\| \leq \|z_m - \hat{z}_m\| + |f_j(z_m, e_j) - \hat{f}_j(z_m, e_j)| + |\hat{f}_j(z_m, e_j) - \hat{f}_j(\hat{z_m}, e_j)| < \epsilon/3 + \epsilon/3 + \epsilon/3$, which ends the proof.

Proposition 3 *Let $\mathscr{D}$ be an infinite observational sample generated from $(\mathscr{G}, f, \mathscr{E})$. With same notations as in Proposition 2, for every sequence ϵ_t such that $\epsilon_t > 0$ goes to zero when $t \to \infty$, there exists a set $\widehat{f}_t = (\hat{f}_1^t \dots \hat{f}_d^t)$ such that $\widehat{MMD}_k$ between $\mathscr{D}$ and an infinite size sample $\widehat{\mathscr{D}}_t$ generated from the CGNN $(\mathscr{G}, \widehat{f}_t, \mathscr{E})$ is less than ϵ_t.*

Proof According to Proposition 2 and with same notations, letting $\epsilon_t > 0$ go to 0 as t goes to infinity, consider $\hat{f}_t = (\hat{f}_1^t \dots \hat{f}_d^t)$ and $\hat{z}_t$ defined from $\hat{f}_t$ such that for all $e \in [0, 1]^d$, $\|z(e) - \widehat{z}_t(e)\| < \epsilon_t$.

Let $\{\hat{\mathscr{D}}_t\}$ denote the infinite sample generated after $\hat{f}_t$. The score of the CGNN $(\mathscr{G}, \hat{f}_t, \mathscr{E})$ is $\widehat{\mathrm{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}_t) = \mathbb{E}_{e,e'}[k(z(e), z(e')) - 2k(z(e), \widehat{z}_t(e')) + k(\widehat{z}_t(e), \widehat{z}_t(e'))]$.

As $\hat{f}_t$ converges towards f on the compact $[0, 1]^d$, using the bounded convergence theorem on a compact subset of $\mathbb{R}^d$, $\widehat{z}_t(e) \to z(e)$ uniformly for $t \to \infty$,

it follows from the Gaussian kernel function being bounded and continuous that $\widehat{\text{MMD}}_k(\mathscr{D}, \hat{\mathscr{D}}_t) \to 0$, when $t \to \infty$.

Proposition 4 *Let $\boldsymbol{X} = [X_1, \ldots, X_d]$ denote a set of continuous random variables with joint distribution P, generated by a CGNN $\mathscr{C}_{\mathscr{G},f} = (\mathscr{G}, f, \mathscr{E})$ with $\mathscr{G}$, a directed acyclic graph. And let $\mathscr{D}$ be an infinite observational sample generated from this CGNN. We assume that P is Markov and faithful to the graph $\mathscr{G}$, and that every pair of variables (X_i, X_j) that are d-connected in the graph are not independent. We note $\widehat{\mathscr{D}}$ an infinite sample generated by a candidate CGNN, $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$. Then,*

(i) If $\widehat{\mathscr{G}} = \mathscr{G}$ and $\hat{f} = f$, then $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) = 0$.
(ii) For any graph $\widehat{\mathscr{G}}$ characterized by the same adjacencies but not belonging to the Markov equivalence class of $\mathscr{G}$, for all $\hat{f}$, $\widehat{MMD}_k(\mathscr{D}, \widehat{\mathscr{D}}) \neq 0$.

Proof The proof of (i) is obvious, as with $\widehat{\mathscr{G}} = \mathscr{G}$ and $\hat{f} = f$, the joint distribution $\hat{P}$ generated by $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}} = (\widehat{\mathscr{G}}, \hat{f}, \mathscr{E})$ is equal to P, thus we have $\widehat{\text{MMD}}_k(\mathscr{D}, \widehat{\mathscr{D}}) = 0$.

(ii) Let consider $\widehat{\mathscr{G}}$ a DAG characterized by the same adjacencies but that do not belong to the Markov equivalence class of $\mathscr{G}$. According to Verma and Pearl (1991), as the DAG $\mathscr{G}$ and $\widehat{\mathscr{G}}$ have the same adjacencies but are not Markov equivalent, there are not characterized by the same v-structures.

a) First, we consider that a v-structure $\{X, Y, Z\}$ exists in $\mathscr{G}$, but not in $\widehat{\mathscr{G}}$. As the distribution P is faithful to $\mathscr{G}$ and X and Z are not d-separated by Y in $\mathscr{G}$, we have that $(X \not\perp\!\!\!\perp Z|Y)$ in P. Now we consider the graph $\widehat{\mathscr{G}}$. Let $\hat{f}$ be a set of neural networks. We note $\hat{P}$ the distribution generated by the CGNN $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}}$. As $\widehat{\mathscr{G}}$ is a directed acyclic graph and the variables E_i are mutually independent, $\hat{P}$ is Markov with respect to $\widehat{\mathscr{G}}$. As $\{X, Y, Z\}$ is not a v-structure in $\widehat{\mathscr{G}}$, X and Z are d-separated by Y. By using the causal Markov assumption, we obtain that $(X \perp\!\!\!\perp Z|Y)$ in $\hat{P}$.
b) Second, we consider that a v-structure $\{X, Y, Z\}$ exists in $\widehat{\mathscr{G}}$, but not in $\mathscr{G}$. As $\{X, Y, Z\}$ is not a v-structure in $\mathscr{G}$, there is an "unblocked path" between the variables X and Z, the variables X and Z are d-connected. By assumption, there do not exist a set D not containing Y such that $(X \perp\!\!\!\perp Z|D)$ in P. In $\widehat{\mathscr{G}}$, as $\{X, Y, Z\}$ is a v-structure, there exists a set D not containing Y that d-separates X and Z. As for all CGNN $\mathscr{C}_{\widehat{\mathscr{G}},\hat{f}}$ generating a distribution $\hat{P}$, $\hat{P}$ is Markov with respect to $\widehat{\mathscr{G}}$, we have that $X \perp\!\!\!\perp Z|D$ in $\hat{P}$.

In the two cases a) and b) considered above, P and $\hat{P}$ do not encode the same conditional independence relations, thus are not equal. We have then $\widehat{\text{MMD}}_k(\mathscr{D}, \mathscr{D}') \neq 0$.

Table of Scores for the Experiments on Cause-Effect Pairs

See Table 3.

Table 3 Cause-effect relations: area under the precision recall curve on five benchmarks for the cause-effect experiments (weighted accuracy in parenthesis for Tüb). Underline values correspond to best scores

Method	Cha	Net	Gauss	Multi	Tüb
Best fit	56.4	77.6	36.3	55.4	58.4 (44.9)
LiNGAM	54.3	43.7	66.5	59.3	39.7 (44.3)
CDS	55.4	89.5	84.3	37.2	59.8 (65.5)
IGCI	54.4	54.7	33.2	80.7	60.7 (62.6)
ANM	66.3	85.1	88.9	35.5	53.7 (59.5)
PNL	73.1	75.5	83.0	49.0	68.1 (66.2)
Jarfo	79.5	92.7	85.3	94.6	54.5 (59.5)
GPI	67.4	88.4	89.1	65.8	66.4 (62.6)
CGNN ($\widehat{\text{MMD}}_k$)	73.6	89.6	82.9	96.6	79.8 (74.4)
CGNN ($\widehat{\text{MMD}}_k^m$)	76.5	87.0	88.3	94.2	76.9 (72.7)

Table of Scores for the Experiments on Graphs

See Tables 4, 5 and 6.

Table 4 Average (std. dev.) results for the orientation of 20 artificial graphs given true skeleton (left), artificial graphs given skeleton with 20% error (middle). Underline values correspond to best scores

	Skeleton without error			Skeleton with 20% of error		
	AUPR	SHD	SID	AUPR	SHD	SID
Constraints						
PC-Gauss	0.67 (0.11)	9.0 (3.4)	131 (70)	0.42 (0.06)	21.8 (5.5)	191.3 (73)
PC-HSIC	0.80 (0.08)	6.7 (3.2)	80.1 (38)	0.49 (0.06)	19.8 (5.1)	165.1 (67)
Pairwise						
ANM	0.67 (0.11)	7.5 (3.0)	135.4 (63)	0.52 (0.10)	19.2 (5.5)	171.6 (66)
Jarfo	0.74 (0.10)	8.1 (4.7)	147.1 (94)	0.58 (0.09)	20.0 (6.8)	184.8 (88)
Score-based						
GES	0.48 (0.13)	14.1 (5.8)	186.4 (86)	0.37 (0.08)	20.9 (5.5)	209 (83)
LiNGAM	0.65 (0.10)	9.6 (3.8)	171 (86)	0.53 (0.10)	20.9 (6.8)	196 (83)
CAM	0.69 (0.13)	7.0 (4.3)	122 (76)	0.51 (0.11)	15.6 (5.7)	175 (80)
CGNN ($\widehat{\text{MMD}}_k^m$)	0.77 (0.09)	7.1 (2.7)	141 (59)	0.54 (0.08)	20 (10)	179 (102)
CGNN ($\widehat{\text{MMD}}_k$)	0.89[a] (0.09)	2.5[a] (2.0)	50.45[a] (45)	0.62 (0.12)	16.9 (4.5)	134.0[a] (55)

[a]Denotes statistical significance at $p = 10^{-2}$

Table 5 Average (std. dev.) results for the orientation of 20 and 50 artificial graphs coming from Syntren simulator given true skeleton. Underline values correspond to best scores

	Syntren network 20 nodes			Syntren network 50 nodes		
	AUPR	SHD	SID	AUPR	SHD	SID
Constraints						
PC-Gauss	0.40 (0.16)	16.3 (3.1)	198 (57)	0.22 (0.03)	61.5 (32)	993 (546)
PC-HSIC	0.38 (0.15)	23 (1.7)	175 (16)	–	–	–
Pairwise						
ANM	0.36 (0.17)	10.1 (4.2)	138 (56)	0.35 (0.12)	29.8 (13.5)	677 (313)
Jarfo	0.42 (0.17)	10.5 (2.6)	148 (64)	0.45 (0.13)	26.2 (14)	610 (355)
Score-based						
GES	0.44 (0.17)	9.8 (5.0)	116 (64)	0.52 (0.03)	21 (11)	462 (248)
LiNGAM	0.40 (0.22)	10.1 (4.4)	135 (57)	0.37 (0.28)	33.4 (19)	757 (433)
CAM	0.73 (0.08)	4.0 (2.5)	49 (24)	0.69 (0.05)	14.8 (7)	285 (136)
CGNN ($\widehat{\text{MMD}}_k^m$)	0.80[a] (0.12)	3.2 (1.6)	45 (25)	0.82[a] (0.1)	10.2[a] (5.3)	247 (134)
CGNN ($\widehat{\text{MMD}}_k$)	0.79 (0.12)	3.1[a] (2.2)	43 (26)	0.75 (0.09)	12.2 (5.5)	309 (140)

[a]Denotes statistical significance at $p = 10^{-2}$

Table 6 Average (std. dev.) results for the orientation of the real protein network given true skeleton. Underline values correspond to best scores

	Causal protein network		
	AUPR	SHD	SID
Constraints			
PC-Gauss	0.19 (0.07)	16.4 (1.3)	91.9 (12.3)
PC-HSIC	0.18 (0.01)	17.1 (1.1)	90.8 (2.6)
Pairwise			
ANM	0.34 (0.05)	8.6 (1.3)	85.9 (10.1)
Jarfo	0.33 (0.02)	10.2 (0.8)	92.2 (5.2)
Score-based			
GES	0.26 (0.01)	12.1 (0.3)	92.3 (5.4)
LiNGAM	0.29 (0.03)	10.5 (0.8)	83.1 (4.8)
CAM	0.37 (0.10)	8.5 (2.2)	78.1 (10.3)
CGNN ($\widehat{\text{MMD}}_k^m$)	0.68 (0.07)	5.7 (1.7)	56.6 (10.0)
CGNN ($\widehat{\text{MMD}}_k$)	0.74[a] (0.09)	4.3[a] (1.6)	46.6[a] (12.4)

[a]Denotes statistical significance at $p = 10^{-2}$

References

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

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

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv*.

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.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314.

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*.

Goldberger, A. S. (1984). Reverse regression and salary discrimination. *Journal of Human Resources*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Neural Information Processing Systems (NIPS)*, pages 2672–2680.

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.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*.

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*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *NIPS*.

Lopez-Paz, D. (2016). *From dependence to causation*. PhD thesis, University of Cambridge.

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

Mendes, P., Sha, W., and Ye, K. (2003). Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics*, 19(suppl_2):ii122–ii129.

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.

Peters, J. and Bühlmann, P. (2013). Structural intervention distance (sid) for evaluating causal graphs. *arXiv preprint arXiv:1306.1043*.

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

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.

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

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*.

Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. (2002). Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64.

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.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*.

Spirtes, P., Glymour, C., and Scheines, R. (1993). Causation, prediction and search. 1993. *Lecture Notes in Statistics*.

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.

Van den Bulcke, T., Van Leemput, K., Naudts, B., van Remortel, P., Ma, H., Verschoren, A., De Moor, B., and Marchal, K. (2006). Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC bioinformatics*, 7(1):43.

Verma, T. and Pearl, J. (1991). Equivalence and synthesis of causal models. In *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, UAI '90, pages 255–270, New York, NY, USA. Elsevier Science Inc.

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.