Meta learning for Causal Direction

Jean-Francois Ton
University of Oxford
ton@stats.ox.ac.uk

Dino Sejdinovic
University of Oxford
dino.sejdinovic@stats.ox.ac.uk

Kenji Fukumizu
The Institute of Statistical Mathematics
fukumizu@ism.ac.jp

Abstract

The inaccessibility of controlled randomized trials due to inherent constraints in many fields of science has been a fundamental issue in causal inference. In this paper, we focus on distinguishing the cause from effect in the bivariate setting under limited observational data. Based on recent developments in meta learning as well as in causal inference, we introduce a novel generative model that allows distinguishing cause and effect in the small data setting. Using a learnt task variable that contains distributional information of each dataset, we propose an end-to-end algorithm that makes use of similar training datasets at test time. We demonstrate our method on various synthetic as well as real-world data and show that it is able to maintain high accuracy in detecting directions across varying dataset sizes.

1 Introduction

Discovering causal links between variables has been a long standing problem in many areas of science. Ideally all experiments are in a randomized controlled environment where each variable can be accounted for separately. However in most cases, this is impossible due to physical, financial or ethical reasons. The problem of determining causal direction becomes even more apparent, when trying to understand the true generating process of data. Although modern machine learning models are able to achieve impressive performances in capturing complex nonlinear relationships among variables, many of them do not take into account causal structure, which might lead to generalization errors when faced with different data than seen during training.

Hence, in recent years, researchers have focused on inferring causal relations from observational data and have developed many different algorithms. One of the major approaches is constraint-based methods, which analyze the conditional independence among variables to determine the causal graph up to a Markov equivalence class under certain assumptions [29]; in addition to early example of the PC algorithm [37], there are also nonlinear methods for capturing independence [41, 40, 47]. Another category is score-based methods which use search algorithms to find the best causal graph with respect to such a score as BIC [4]. These methods are, however, often unable to determine the correct structure and can be computationally very expensive. There are also hybrid methods which mitigate such difficulty [42].

A new line of research has taken specific interest in the bivariate case, i.e., the cause-effect inference, where one decides between causal hypotheses "$X \to Y$" and "$Y \to X$" [15, 12, 25, 43]. In this setting, methods that exploit the inherent asymmetries between cause and effect are the most prominent. The data is analysed under the *Functional Causal Model* (FCM, [30]) formalism, following respective model assumptions. Due to the intrinsic asymmetry of the problem, several statistics have been proposed to infer the direction of the cause-effect pairs [35, 15, 27, 25].

Preprint. Under review.

Most relevant prior work to this paper is the framework of Causal Generative Neural Networks (CGNN, [12]). When applied to cause-effect inference, CGNN learns a generative model using a neural network for each direction and compares their fitting to determine the directionality. Many advanced methods including CGNN, however, assume an access to a large dataset to make use of strong learning model such as neural networks. This in turn may lead significantly degraded performances in small data settings encountered in practical problems [22] (we shall show in Section 4 that CGNN does not perform well on small datasets).

In this paper, we revisit the problem of cause-effect inference from the viewpoint of empirical learning. Specifically, we are interested in learning from many examples of cause-effect datasets together with their true causal directions. The purpose of the current work is to develop a method for using effectively this empirical knowledge on causal directions, when we make cause-effect inference on a new unseen and purely observational dataset.

Learning-based cause-effect inference has been already explored in the literature. For instance, Randomized Causation Coefficient (RCC) [23] and Neural Causation Coefficient (NCC) [24] make a learning-based binary classifier for causal directions. RCC and NCC, however, require to synthesize vast amounts of problem specific data pairs up 10000 [24]. Other examples, such as NonSENS [26] and Causal Mosaic [43], aim to recover a FCM using nonlinear independent component analysis. An important assumption for these methods is availability of multiple datasets sharing the same causal mechanism and the same exponential family of latent variables. We then need to assume or select datasets to satisfy this.

Different from these works, for the purpose of alleviating the problem of small data, we consider the meta learning setting by introducing a dataset-feature extractor. The feature represents the distributional information of each dataset by employing the formalism of kernel mean embeddings [28] as well as DeepSets [45], aiming to encode similar causal mechanisms of datasets into similar features. We propose a neural network-based generative model that trains jointly on all the training datasets. The model has an encoder-decoder architecture, which has been employed by meta-learning frameworks successfully [10, 9]; the encoder gives the dataset-features, and the decoder realizes a generative model or FCM, which is accompanied with the Feature-wise Linear Modulation layers (FiLM [31]) to adapt the generator to the dataset at hand. With this meta-learning architecture, the proposed method is able to determine the cause-effect direction efficiently for new unseen, possibly small, and purely observational datasets.

The contributions of this work can be summarized as follows:

- We introduce a new meta learning algorithm that can leverage similar datasets for unseen causal pairs in causal direction discovery.
- We exploit structural asymmetries with an adaptive generative model, thus avoiding the need to retrain at test time.
- We propose an end-to-end algorithm that assumes no *a priori* assumptions on the causal mechanism between cause and effect.
- High performance on small dataset sizes can be achieved by virtue of meta learning.

2 Meta Learning for Detecting Causal Direction

We first give a brief summary of FCM and explain the proposed method including its building blocks.

2.1 Functional Causal Model

Functional Causal Models (FCM) have been widely used when conducting causal inference. Formally, a FCM on a random vector $X = (X_1, \ldots, X_d)$ is a triplet $C = (\mathcal{G}, f, \mathcal{E})$, representing the following equations:

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

where X_i are the observed variables, Z_i are the independent hidden variables and f_i being the mechanism linking the cause and the effect. Under this formulation, there is a clear asymmetry between cause and effect, given that one is used to infer the other, and hence numerous work has been done exploiting this fact.

For inferring causal direction $X \to Y$ for bivariate (X, Y), we can consider only the FCM $Y = f(X, Z)$, where Z and X are independent. Among other inference methods, CGNN [12] uses neural networks to train the mechanism f. More precisely, given dataset $\mathcal{D} = \{(x_j, y_j)\}_{j=1}^m$, we generate Z_j by the standard normal distribution $N(0, 1)$, and train neural network f so that the distribution of $\{(X_j, Y_j)\}_j$ be close to that of $\{(X_j, f(X_j, Z_j)\}_j$. The difference of the distributions is measured by the Maximum Mean Discrepancy (MMD, [14]). CGNN learns two models $\hat{Y} = f(X, Z)$ and $\hat{X} = g(X, Z)$, and chooses a better fit to determine the direction.

Unlike CGNN, which trains networks f and g for each dataset, our method considers a single neural network working for all the datasets in a cause-effect database $\{\mathcal{D}_i\}_{i=1}^N$ where $\mathcal{D}_i = \{(X_j^i, Y_j^i)\}_{j=1}^{m_i}$ is a dataset in the database. We assume that the causal direction is known for all $\mathcal{D}_i$. More specifically, given $X^i \to Y^i$ is the true causal direction, we wish to create a neural network $F(X, Z)$ so that the distribution of $\mathcal{D}_i$ is approximately the same as that of $\{(X_j^i, \hat{Y}_j^i)\}$ for *any* i, where $\hat{Y}_j^i = F(X_j^i, Z_j^i)$. This approach involves obvious difficulty, since a wide-variety of cause-effect relations must be learnt by a single network. To achieve successful training, we introduce two novel components to CGNN: (1) a dataset-feature to represent the causal mechanism of each dataset, and (2) the FiLM layers to adapt the base neural network to each dataset using the dataset-features.

2.2 Dataset features via Conditional Mean Embeddings (CME)

Kernel mean embeddings of distributions provide a powerful framework for representing probability distributions [36, 28]. Given sets $\mathcal{X}$ and $\mathcal{Y}$, with a distribution P over the random variables (X, Y) taking values in $\mathcal{X} \times \mathcal{Y}$, the conditional mean embedding (CME) of the conditional density $p(y|x)$, is defined as:

$$\mu_{Y|X=x} := \mathbb{E}_{Y|X=x}[\phi_y(Y)] = \int_{\mathcal{Y}} \phi_y(y) p(y|x) dy. \tag{2}$$

where ϕ_y is the feature map associated to the reproducing kernel Hilbert space (RKHS) of Y, $\mathcal{H}_Y$. Intuitively, the equation above allows us to represent a probability distribution $p(y|x)$ in the RKHS, by taking the expectation under $p(y|x)$ of the features $\phi_y(y) \in \mathcal{H}_Y$. Hence, for each value of the conditioning variable x, we obtain $\mu_{Y|X=x} \in \mathcal{H}_Y$.

Following [36], the CME can be associated with the operator $\mathcal{C}_{Y|X} : \mathcal{H}_X \to \mathcal{H}_Y$, which is known as the conditional mean embedding operator (CMEO) and satisfies

$$\mu_{Y|X=x} = \mathcal{C}_{Y|X} \phi_x(x). \tag{3}$$

such that $\mathcal{C}_{Y|X} := \mathcal{C}_{YX} \mathcal{C}_{XX}^{-1}$ where $\mathcal{C}_{YX} := \mathbb{E}_{Y,X}[\phi_y(Y) \otimes \phi_x(X)]$ and $\mathcal{C}_{XX} := \mathbb{E}_{X,X}[\phi_x(X) \otimes \phi_x(X)]$.

As a result, if we use finite-dimensional feature maps ϕ_x and ϕ_y, the finite sample estimator of $\mathcal{C}_{Y|X}$ based on the dataset $\{(x_j, y_j)\}_{j=1}^n$ can be written as

$$\widehat{\mathcal{C}}_{Y|X} = \Phi_y (K + \lambda I)^{-1} \Phi_x^T \tag{4}$$

where $\Phi_y := (\phi_y(y_1), \ldots, \phi_y(y_n))$ and $\Phi_x := (\phi_x(x_1), \ldots, \phi_x(x_n))$ are the feature matrices, $K := \Phi_x^T \Phi_x$ is the kernel matrix with entries $K_{i,j} = k_x(x_i, x_j) := \langle \phi_x(x_i), \phi_x(x_j) \rangle$, and $\lambda > 0$ is a regularization parameter.

CMEO is a canonical way to capture conditional densities and thus the mechanism in a functional causal model. We are able to encode the direction in the embedding using the CMEO through the conditional distribution. We give further motivation on why we use CMEO in Section 2.4.

2.3 A Feature-wise Linear Modulation (FiLM)

We propose an architecture that is able to adapt the network $f(X, Z)$, i.e. a FCM, with the dataset-feature by using Feature-wise Linear Modulation (FiLM) layers [31].

FiLM layers are known to allow network adaptation to new environments quickly and without adding significantly more model parameters. They have been shown to work effectively in various computer vision [31] and regression tasks [33]. In essence, FiLM layers work as follows: given a conditioning

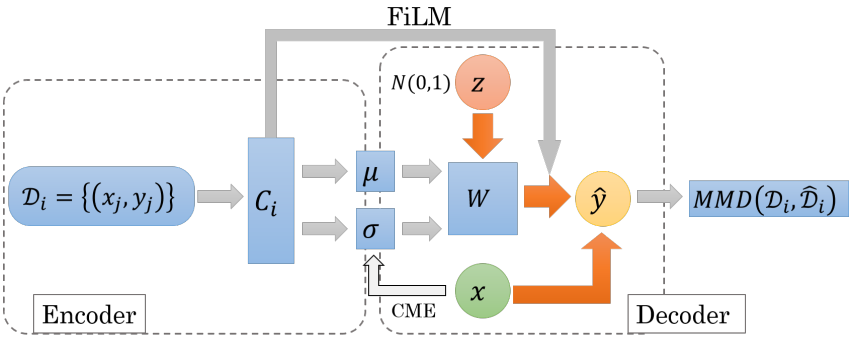

Figure 1: Proposed meta-CGNN Algorithm for only one dataset $\mathcal{D}_i$ in the mini-batch

variable C (this may be the label for image classification) and l_a being the a^{th} layer of a network, the FiLM layer FL_a, constructed by a neural network, adapts l_a to l_a^{FL} by

$$(\beta_a, \gamma_a) = FL_a(C), \qquad l_a^{FL} = \beta_a + \gamma_a \circ l_a, \tag{5}$$

where $\circ$ is the element-wise multiplication. Intuitively, the FiLM layer learns shift and scale parameters, conditioned on C, for any given layer l_a. We shall use the CMEO for C.

2.4 Proposed method

We propose a new meta learning algorithm, *meta-CGNN*, which works for cause-effect inference given cause-effect training database $\mathbb{D} = \{\mathcal{D}_i\}_{i=1}^N$ with known directionality, where $\mathcal{D}_i = \{(x_j^i, y_j^i)\}_{j=1}^{m_i}$. Without loss of generality, we assume $X^i \to Y^i$ for any dataset i. The FCM is trained on this database $\mathbb{D}$, and is used to infer the causal direction for an unseen dataset $\mathcal{D}_{test}$.

We use encoder-decoder based architecture for meta learning, which is similar to the Neural Process [10]. The encoder is realized by neural networks or CME, while the decoder is a neural network. The encoder first maps the dataset $\mathcal{D}_i$ into a dataset-feature C_i, which is given as an input to further neural networks, amortization network and FiLM. The amortization network outputs $(\mu(C), \sigma(C))$, with which we use to modulate the latent variable $Z_j^i \sim N(0,1)$ to $W_j^i := \mu(C_i) + \sigma(C_i) Z_j^i$. The decoder network then maps (X, W) to $\hat{Y}$ so that the distribution of $\{(X_j^i, \hat{Y}_j^i)\}_j$ is close to $\{(X_j^i, Y_j^i)\}_j$. By using an encoder network, which trains across datasets jointly, we are able to share distributional information between datasets (tasks) and apply it to a new unseen task. See Figure 1.

The overall functional causal model in the proposed meta-CGNN is thus

$$\hat{y}_j = F\left((x_j, z_j); C\right) \text{ , where } z_j \sim N(0,1), \quad z_j \perp\!\!\!\perp x_j. \tag{6}$$

Encoder: For representing dataset specific distributional information or mechanism for each task, in addition to CME, we also consider DeepSets.

DeepSets [45] has been used as task embedding in previous meta learning literature [9, 10, 44]. Using a neural network $\phi_{x,y}$, the task embedding is defined by

$$C_i = \frac{1}{m_i} \sum_{j=1}^{m} \phi_{x,y}([x_j, y_j]). \tag{7}$$

DeepSets is a simple flexible approach to encoding sets into vectors, which is also permutation invariant. The latter is important as we do not want the embeddings to change solely based on the order of the elements in the dataset.

The conditional mean embeddings is used more specifically as follows. We first compute the CMEO $\mathcal{C}_{Y|X}$ from $\mathcal{D}_i$ as described in (Eq 4), and use it to obtain the CME for each datapoint using (Eq 3). We then create $W_j^i = \mu(C_{i,j}) + \sigma(C_{i,j}) Z_j^i$, where $C_{i,j}$ is defined for data pair (x_j, y_j) by

$$C_{i,j} = \mathcal{C}_{Y|X} \phi_x(x_j). \tag{8}$$

ϕ_x is the feature vector such that $k(x_i, x_k) = \langle \phi_x(x_j), \phi_x(x_k) \rangle$, k being the RBF kernel. However, to obtain a finite dimensional representation efficiently, we will use Random Fourier Features [32] to approximate the CME. Throughout the paper we will use 100 features as we work on problems with at most 1500 datapoints. According to [21], we need approximately $\sqrt{N}$, where N is the number of datapoints.

As argued by [25], CME holds critical information for causal-effect inference by representing the mechanism in the FCM and the complexity of the mechanism tends to be larger in the anti-causal direction is larger than in the causal direction. Hence, we expect CME to capture relevant distributional information, and inform adapting our generative model to the task at hand.

Decoder: For the generative part of our model, the dataset-feature C_i or $C_{i,j}$ gives modulation through the FiLM and amortization network. Both FiLM and amortization networks take as input C_i (DeepSets) or $C_{i,j}$ (CME). The FiLM layer is able to adapt the weights of the decoder network depending on the distributional feature of a dataset. This is crucial for a single network to learn FCMs of all the datasets (see ablation study in Section 4.4). The amortization network works on $z \sim N(0, 1)$ similarly to FiLM. It can be interpreted as the adaption of latent Gaussian distribution; $p(w|C_i)$ with $W^i = \mu(C_i) + \sigma(C_i)Z$ regarded as a new latent variable reflecting the distribution of the dataset $\mathcal{D}_i$.

Training: The objective function of training is similar to [12]; by sampling $\{\hat{y}_j^i\}_{j=1}^{m_i}$ from (6) and estimating MMD [14] between the sampled data $\widehat{\mathcal{D}_i} = \{(x_j^i, \hat{y}_j^i)\}_{j=1}^{m_i}$ and the original data $\mathcal{D}_i = \{(x_j^i, y_j^i)\}_{j=1}^{m}$. The MMD estimator between $\mathcal{U} = \{u_i\}_{i=1}^m$ and $\mathcal{V} = \{v_i\}_{i=1}^n$ is, in general, given by

$$\widehat{\text{MMD}}^2(\mathcal{U}, \mathcal{V}) = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(u_i, u_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(v_i, v_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(u_i, v_j). \tag{9}$$

We use Gaussian kernel, which is characteristic [38]. With k Gaussian, the above expression is differentiable and thus can be optimized as already demonstrated in various works such as [12, 20]. A drawback of using MMD as a loss function however is that it scales quadratically in the sample size. We can again use Random Fourier Features [32, 23], which give us linear-time estimators of MMD.

Using stochastic mini-batches of q datasets $\{\mathcal{D}_i\}_{i=1}^q$, the objective function to minimize is

$$LOSS = \sum_{i=1}^{q} \widehat{\text{MMD}}^2(\widehat{\mathcal{D}_i}, \mathcal{D}_i) \tag{10}$$

This joint training, similar to that used in other encoder-decoder architectures for meta learning [9, 10, 44], allows us to utilize the information of all the available training datasets in a single generative model. This is in contrast to CGNN [12], which trains a separate generative model for each dataset.

Inference of Causal Direction: After training, when we wish to infer the causal direction for a new dataset $\mathcal{D}_{test} = \{(x_j, y_j)\}_{j=1}^m$, we feed both of $\mathcal{D}^{xy} = \{(x_j, y_j)\}_{j=1}^m$ and $\mathcal{D}^{yx} = \{(y_j, x_j)\}_{j=1}^m$ into the trained model and estimate the MMDs between the generated samples and the true ones, i.e., $\mathcal{M}_{xy} := \widehat{\text{MMD}}(\mathcal{D}^{xy}, \hat{\mathcal{D}}^{x\hat{y}})$ and $\mathcal{M}_{yx} := \widehat{\text{MMD}}(\mathcal{D}^{yx}, \hat{\mathcal{D}}^{y\hat{x}})$. If $\mathcal{M}_{xy} < \mathcal{M}_{yx}$, we deduce that $X \to Y$ better agrees with a postulated FCM than $Y \to X$, and similarly $Y \to X$ if $\mathcal{M}_{xy} > \mathcal{M}_{yx}$.

3 Related work

There have been already some works that exploit the asymmetry by taking a closer look at decomposing the joint distribution $P(X, Y)$ into either $P(Y|X)P(X)$ or $P(X|Y)P(Y)$. Relevant to this work is the approach using the asymmetry in terms of functional causal models (FCMs). Some of the previous methods make strong assumptions on the model; LiNGAM [35] considers linear non-Gaussian model for finding causal structure. For nonlinear relations, [15] discusses nonlinear additive noise models, and [46] invertible interactions between the covariates and the noise models. There are other methods to consider the nonlinear models such as Gaussian process regression [39].

Information theory gives an alternative view on asymmetry using Kolmogorov complexity, following the postulate that the mechanism in the causal direction should be less complex than the one in

the anti-causal direction. Several papers have proposed to approximate or use certain proxies for intractable Kolmogorov complexity [16, 19, 5, 25].

The method that comes closest to ours is CGNN [12]. However, our meta-CGNN method differs from CGNN in a multitude of aspects. First, our method employs meta learning, while CGNN only considers one dataset at a time. Hence CGNN is not able leverage similarity between datasets. A naïve way of training the CGNN jointly over datasets as in (Eq 10) was analysed in our ablation study and performed poorly.

Second, CGNN [12] averages over 32-64 separately trained generative networks per direction, which is computationally very expensive, training up to 128 models per dataset. In contrast, we merely train the model with four random initializations and average over the resulting MMDs for each dataset, thereby achieving similar results to CGNN for larger datasets and significantly better for smaller ones.

Third, CGNN needs to be trained separately for every new dataset, which in practice can be slow as well as computationally expensive. meta-CGNN does not need to be trained for a new dataset and can give a causal direction through a simple forward pass at the test time. This is a crucial difference which allows much faster inference once a new dataset is presented to the model.

Lastly, there have recently been other applications of meta learning to causal inference. MetaCI [34] uses a MAML-style[7] (optimization based) learner but mainly deals with counterfactual causality rather than causal directionality. Bengio et al. [1] also consider a meta learning method using FCM, based on the principle that the model assuming true causal direction can be "adapted faster" than the one assuming the anti-causal. However, their method is designed for a different settings. Firstly, the test distribution comes from an perturbation of the input distribution, which is modeled by a known parametric family. The choice of the model is not trivial for continuous domains in real-world data. Secondly, for training of neural netwroks, they assume to have access to a large training dataset around 3000 for a single mechanism, which is different from our setting of small data.

4 Experiments

4.1 Synthetic Datasets

For the synthetic experiments we use three different types of datasets taken from [12], each of which exhibits distinct cause-effect (CE) mechanism. The *CE-Net* contains 300 cause-effect pairs with random distributions for the causes and random Neural Networks as mechanisms to create the effects. The *CE-Gauss* also contains 300 data pairs, where the cause is sampled from a random mixture of Gaussians and the mechanism is drawn from a Gaussian process [27]. Lastly, *CE-Multi* datasets take the cause from Gaussian distributions and the mechanisms is built using random linear and polynomial functions. They also include multiplicative and additive noise before or after the causal mechanism, making the task harder. In addition, to confirm the advantage of meta learning, we use two different data size regimes: 1500 and 100 datapoints.

We measure the performance of distinguishing the causal direction using the Area Under the Precision Recall Curve (AUPRC), which is the same metric used in [12] (accuracy is presented in the Supplementary material). With AUPRC, we are able to take into account the confidence of an algorithm, thus allowing models not to commit to a prediction if not certain.

For training of meta-CGNN, we use 100 datasets for training and the remaining 200 for testing. We average the MMD of each dataset over 4 independent runs in order to get our final prediction. Note that CGNN [12] averages their model over 32-64 different runs, each of which takes 24 minutes. This would be infeasible without high-performance computing environments. Hence we have restricted ourselves to averaging CGNN over 12 runs in the comparisons.

Regarding architectures, we use a 2-hidden layer with ReLU activation function for the FiLM, amortization and encoder network. For the decoder we use a 1-hidden layer with ReLU activation function. As noted in [12], the number of hidden nodes in the decoder is very important; too small network is not able to realize the mapping properly, while too big network tends to overfit so that both directions have low MMD values. Hence for our experiments, for simplicity, we solely cross-validated over the number of decoder nodes $[5, 40]$ [12] by leaving out a few datasets at training aside for validation. Around 40 nodes for the 1-hidden layer was optimal for [12].

In addition, following [12], for our loss function, we use a sum over Gaussian kernel MMDs with bandwidth $\eta \in \{0.005, 0.05, 0.25, 0.5, 1, 5, 50\}$ together with an Adam optimizer [18] and a fixed learning rate 0.01. For the mini-batch size we fix q to 10.

We compare meta-CGNN against several competing ones that have open source codes; the methods are 1) Additive noise model (ANM) [27] with Gaussian process regression and HSIC test, 2) Information Geometric Causal Inference (IGCI) [5] with entropy estimator and Gaussian reference measure, 3) Conditional Distribution Similarity statistic (CDS) [8], which analyses the variance of the conditional distributions, 4) Regression Error based Causal Inference (RECI) [2], which analyses the residual of each direction using a polynomial fit, 5) Causal Generative Neural Networks (CGNN) [12]. We use the implementation by [17] which provides a GitHub repository toolbox for the above mentioned methods. We left out RCC [23] as they are known to be very data hungry [12, 43], see Supplement for results. In order to keep the comparisons fair, we use the same 200 data pairs as the one that our meta-CGNN is tested on. Finally, in order to demonstrate the effect of each building block in our model, we have also conducted an ablation study, highlighting the importance of the task embeddings and FiLM layer. Further experimental details can be found in the Supplementary Material.

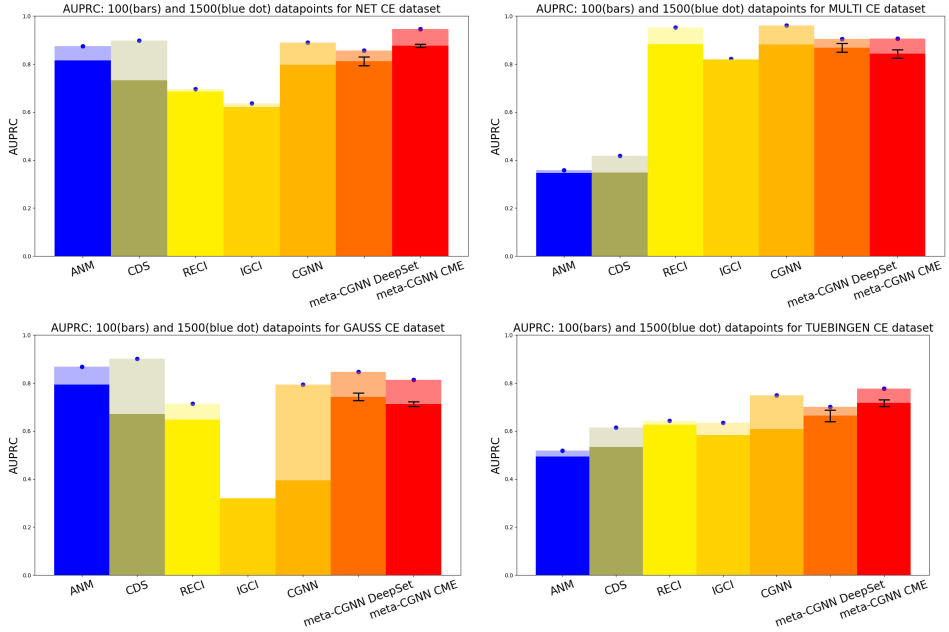

Figure 2: AUPRC for Net, Multi, Gauss and Tuebingen dataset. The blue dots represent the AUPRC with 1500 datapoints, whereas the barplots are with 100 datapoints. Note that the proposed meta-CGNN is in most cases significantly better that standard CGNN, which is trained individually on each dataset. In addition, meta-CGNN even though not the best on each dataset remains comparable across different datasets and sees the smallest drop in performance when reducing the dataset size.

4.2 Tuebingen Cause-Effect dataset

As a real-world example, we use the popular Tuebingen benchmark [27], from which we take 99 bivariate datasets. We use a similar setup as in the synthetic experiments, and the only difference is that we employ 5-fold cross-validation for training and testing. That way we obtain a prediction on each of the 99 datasets. We repeat this 3 times with different random splits and report the results in Figure 4, where we also check the performance on decreasing size of samples in a dataset.

4.3 Results

Figure 2 illustrates how both our meta-CGNN algorithms are the only ones that can retain high AUPRC across different datasets, as well as dataset sizes, i.e. 1500 and 100 datatpoints. ANM [27], for example, seems to do reasonably well on the *CE-Net* and *CE-Gauss* dataset, but completely fails in the *CE-Multi* and, more importantly, on the real-world Tuebingen dataset. This occurs mainly because of the strict assumptions which ANM imposes on the FCM. Similarly, RECI which performs

well on the *CE-Multi*, but not on the *CE-Net*, *CE-Gauss* and the Tuebingen dataset. Similar situation holds true for the remaining competing methods.

The only other method that also retains high performance across datasets is CGNN. However, in particular *CE-Gauss*, *CE-Net* and the Tuebingen dataset, we can see substantial decrease in AUPRC, as well as in accuracy (see Supplement), when only given little data. This illustrates the data dependence of CGNN and highlights the benefits of our approach.

Our proposed method, is amongst the top performing ones and is consistently doing well for both 1500 and 100 datapoints settings. On the *CE-Gauss* dataset, we can see that meta-CGNN with DeepSets is able to maintain its AUPRC, whereas standard CGNN is not. The same trend can been seen on the Tuebingen dataset. These results illustrate that meta-CGNN is able to retain high AUPRC, even when faced with little data, by leveraging the meta-learning setting.

Lastly, we want to emphasise that meta-CGNN does not need to be trained for each test dataset, but instead only needs a forward pass through the generative model to determine the causal direction, which makes it vastly more computationally efficient than CGNN.

4.4 Ablation study

In this section, we study the necessity of task embedding in our model and show that not every model can be trained in this joint dataset fashion. To this end, we used a CGNN [12] that has been trained exactly like our meta-CGNN, i.e. meta-CGNN where we removed the amortization network and the FiLM layer. In this case, only the decoder was trained jointly across the datasets and we see that this naïve version of extending CGNN does not perform well, with accuracy hovering around 55-60% for the Tuebingen dataset with at most 1500 datapoints per dataset. It behaves similarly on the synthetic data with a chance level accuracy for *CE-Net* dataset with 1500 datapoints.

Next, we investigate the importance of the FiLM layer and run the experiments from section 4 using DeepSets embeddings, with and without the FiLM layer while keeping the amortization network. We see from Table 1 a significant performance boost when using the FiLM as it allows the decoder at test time more flexibility to adapt, instead of relying solely on the amortization network. This shows the importance of each component in the proposed architecture and that adapting cause-effect methods into the meta learning setting is a non-trivial task. We see similar trends for meta-CGNN CME.

#datapoints	Gauss		Multi		Net	
	FiLM	noFiLM	FiLM	noFiLM	FiLM	noFiLM
1500	**0.78**	0.73	**0.75**	0.46	**0.70**	0.55
500	**0.80**	0.72	**0.73**	0.49	**0.70**	0.50
100	**0.73**	0.64	**0.72**	0.63	**0.72**	0.50

Table 1: Accuracy of the proposed method with DeepSet embedding with and without the FiLM layer

5 Conclusion

We introduced a novel meta learning algorithm for cause-effect inference which is able to perform well even in the small data regime, in sharp contrast to the existing methods. By leveraging a task feature extractor that can be learned during training, we are able to efficiently adapt our model at test time to new previously unseen datasets by using a FiLM layer. We extended the framework of Causal Generative Neural Network (CGNN) [12] by learning a single generative network, readily adaptable for new datasets, vastly alleviating the computation burden of CGNNs. In particular, instead of having to train multiple models on each dataset separately, our proposed methods are able to achieve similar or better performance with simple forward passes through our generative network at test time. Recently there has been an increase in interest in causality topics in reinforcement learning [48, 3, 6], where the proposed methods may also be applicable. Assuming that we have a skeleton graph of the relevant quantities in the RL model, meta-CGNN could be used to efficiently infer causal direction of unoriented edges.

Broader Impact

Understanding the underlying causal relationships between quantities of a system is of utmost importance in many disciplines, e.g. in social or medical sciences, or in AI safety. Another major concern, that has recently been raised in the machine learning community, is that most machine learning methods are focused on prediction and not on understanding the underlying causal structure. This could lead to a devastating generalization error when the data undergoes a covariate shift, due to e.g. interventions on the system.

However, inferring the causal structure from data remains challenging. In order to draw insightful conclusions, the gold standard is to conduct randomized trials, which in most cases are financially, physically or ethically impossible. For example, in studies trying to determine the causal relationship between lung cancer and smoking, it would be unethical to demand the test subjects to start smoking for the trials.

Hence, many methods have been proposed to infer the causal relationships between the variables solely from observational data. Many of which, even though able to find the skeleton of the graph representing such relationships, are not able to properly orient the edges (due to being able to determine the graph only up to its Markov equivalence class). This problem is amplified when only given little data. This is where meta learning approaches can be useful and our method is a step towards the vision of creating safe AI systems that are aware of the causal structure underlying the observational data and hence are able to generalize better.

A Additional details on experiments setup

In this section of the Appendix we will explain the experimental setup in more detail. For all our synthetic datasets, we have 300 datasets. We then use 100 pairs to train our meta-CGNN algorithm and test on the remaining 200. For the standard methods, we only consider the 200 testing datasets as they cannot incorporate information from different datasets. Each dataset then has either 1500 or 100 datapoints.

Note that for the Tuebingen dataset contains 99 bivariate datasets. We use the same setup as in [13], which means that datasets have at most 1500 or at most 100 datapoints. We use a similar setup as in the synthetic experiments, and the only difference is that we employ 5-fold cross-validation for training and testing. To be concrete, we train on 4 of the folds and test on the last one. We repeat this procedure such that each fold has been the test fold at one point, while the remaining were acting as training. That way we can obtain fair estimates of the directions. We repeat these experiments 3 times with different splits.

Next we provide the neural network architecture for meta-CGNN as well as hyper-parameters. For simplicity, just as in [13], we use a fixed learning rate of 0.01, Adam optimizer, and up to 500 epochs. We used a mini-batch size of 10, i.e. 10 datasets for each gradient update. In terms of the neural network architectures we used a 2-hidden layer network for the encoder, FiLM and amortization network. For the decoder we follow [13] and use a 1-hidden layer NN. All activations used are ReLU.

For the encoder network (meta-CGNN DeepSet), we fix $[40, 10]$ hidden nodes per layer. For the hyper-parameters λ of the Conditional Mean Embedding Operator (CMEO) as well as lengthscale l for the Random Fourier Features embedding we used $\lambda = 1$ and $l = 1$. These were the default settings and worked well. In the future we will cross-validate over these to see if there are any performance gains.

For the amortization network we have $[40, 20]$ hidden nodes layer and output μ_z and σ_z which are both $1D$. In our case the latent variable Z is always one dimensional. For the FiLM layer we have $[40, 40]$ hidden nodes and output 2x decoder hidden nodes. This is because we need a shift and a scale parameter for the hidden layer of the decoder. In terms of the decoder, we cross-validate and choose either 40 or 5 hidden nodes. The reason we chose values around 40 was because we took [13] as a starting point and it worked well, around 40.

B AUPRC and Accuracy of the methods

Here we present the AURPC (area under the precision recall curve) as well as accuracy of the proposed methods as well as competing methods. Note that we report the weighted accuracy for the Tuebingen dataset as in [27] to account for similarities in the datasets. We note that the accuracy follows a similar trend to the AUPRC, however we opted to show the AUPRC in the main text as it is more indicative of the confidence for each prediction. AUPRC also allows us to take into account the fact that a method might be uncertain about its prediction, i.e. it is better to give no answer than a wrong answer.

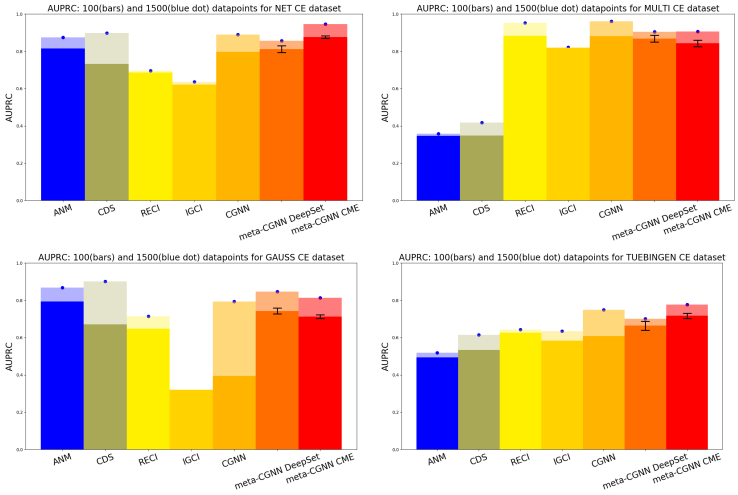

Figure 3: AUPRC for Net, Multi, Gauss and Tuebingen dataset. The blue dots represent the AUPRC with 1500 datapoints, whereas the barplots are with 100 datapoints. Note that the proposed meta-CGNN is in most cases significantly better that standard CGNN, which is trained individually on each dataset. In addition, meta-CGNN even though not the best on each dataset remains comparable across different datasets and sees the smallest drop in performance when reducing the dataset size.

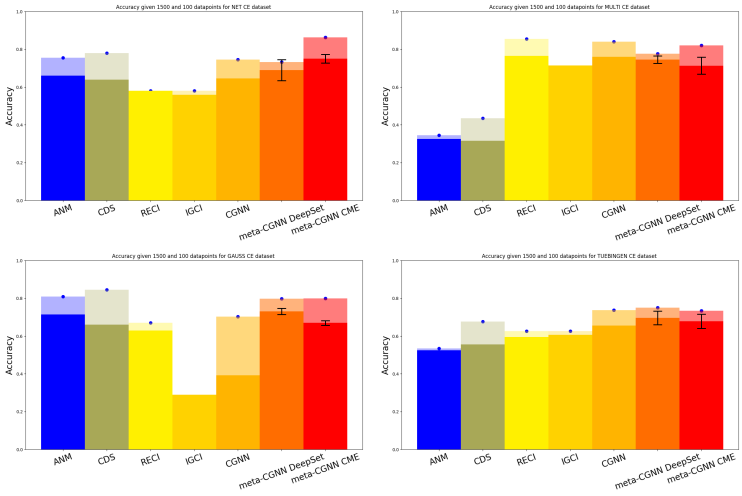

Figure 4: ACC for Net, Multi, Gauss and Tuebingen dataset. The blue dots represent the ACC with 1500 datapoints, whereas the barplots are with 100 datapoints. Note that the proposed meta-CGNN is in most cases significantly better that standard CGNN, which is trained individually on each dataset. In addition, meta-CGNN even though not the best on each dataset remains comparable across different datasets and sees the smallest drop in performance when reducing the dataset size.

ACC	Gauss	Multi	Net	Tueb
1500	0.50	0.92	0.67	0.39 ± 0.02
100	0.47	0.87	0.56	0.45 ± 0.03

AUPRC	Gauss	Multi	Net	Tueb
1500	0.51	0.90	0.65	0.56 ± 0.01
100	0.49	0.82	0.57	0.50 ± 0.03

Table 2: Results for RCC method. As we can see RCC also struggles with *CE-Gauss* dataset and seems to reasonable well on the *CE-Multi* dataset.

.

C Note on competing methods and RCC

For the competing methods, we used the excellent *Causal discovery toolbox* by [17], which is provided in an open source Github repository, with all the popular methods implemented. We have done minimal changes to the algorithms as one can easily streamline the experiments by just inserting the datasets into their framework. The only thing we changed was to use 12 instead of 32 or 64 different models for CGNN [13] and kept the hidden layer size as recommended in the paper around 40 [11]. The rest remained the same as described in the CGNN paper. The results we got on the 1500 datapoints are comparable to the one noted in the paper with a slight degradation of the performance due to less model averaging, i.e. 12 instead 32 or 64, which was not feasible due to our computational constraints.

In addition, we want to note that CGNN requires about 24 min per dataset at testing time, whereas our method can do inference in less than 1min. This is a big computational saving at test time while retaining similar or better performance.

Lastly, we also investigated the performance of RCC (randomized causation coefficient) method [23] in our setting. RCC also uses the formalism of kernel mean embeddings (approximated using random features) in order to build a classifier for causal direction. However, it is data hungry and typically requires a lot of generated synthetic data to train the classifier, as noted in [43]. Using again the toolbox by [17], we use the same procedure as in meta-CGNN, where for the synthetic data experiments datasets are used for training and remaining ones for testing. For the Tuebingen dataset, we again perform a 5-fold cross-validation, repeating the experiment three times, and reporting mean and standard deviation in Table 2. The performance of RCC is poor except for the *CE-Multi* dataset where most other methods also did reasonably well.

References

[1] Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. In *International Conference on Learning Representations*, 2020.

[2] Patrick Blöbaum, Dominik Janzing, Takashi Washio, Shohei Shimizu, and Bernhard Schölkopf. Analysis of cause-effect inference by comparing regression errors. *PeerJ Computer Science*, 5:e169, 2019.

[3] Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations*, 2019.

[4] David Maxwell Chickering. Optimal structure identification with greedy search. *Journal of machine learning research*, 3(Nov):507–554, 2002.

[5] Povilas Daniusis, Dominik Janzing, Joris Mooij, Jakob Zscheischler, Bastian Steudel, Kun Zhang, and Bernhard Schölkopf. Inferring deterministic causal relations. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 143–150, 2010.

[6] Ishita Dasgupta, Jane Wang, Silvia Chiappa, Jovana Mitrovic, Pedro Ortega, David Raposo, Edward Hughes, Peter Battaglia, Matthew Botvinick, and Zeb Kurth-Nelson. Causal reasoning from meta-reinforcement learning. *arXiv preprint arXiv:1901.08162*, 2019.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[8] José AR Fonollosa. Conditional distribution variability measures for causality detection. In *Cause Effect Pairs in Machine Learning*, pages 339–347. Springer, 2019.

[9] Marta Garnelo, Dan Rosenbaum, Chris J Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J Rezende, and SM Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1704–1713. PMLR, 2018.

[10] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018.

[11] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Causal generative neural networks. `https://openreview.net/pdf?id=ry5wc1bCW`.

[12] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Causal generative neural networks. *arXiv preprint arXiv:1711.08936*, 2017.

[13] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michele Sebag. Learning functional causal models with generative neural networks. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, pages 39–80. Springer, 2018.

[14] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[15] Patrik O Hoyer, Dominik Janzing, Joris M Mooij, Jonas Peters, and Bernhard Schölkopf. Nonlinear causal discovery with additive noise models. In *Advances in neural information processing systems*, pages 689–696, 2009.

[16] Dominik Janzing and Bernhard Scholkopf. Causal inference using the algorithmic markov condition. *IEEE Transactions on Information Theory*, 56(10):5168–5194, 2010.

[17] Diviyan Kalainathan and Olivier Goudet. Causal discovery toolbox: Uncover causal relationships in python. *Journal of Machine Learning Research*, 21(37):1–5, 2020.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[19] Jan Lemeire and Erik Dirkx. Causal models as minimal descriptions of multivariate systems.

[20] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. In *Advances in Neural Information Processing Systems*, pages 2203–2213, 2017.

[21] Zhu Li, Jean-Francois Ton, Dino Oglic, and Dino Sejdinovic. Towards A Unified Analysis of Random Fourier Features. In *International Conference on Machine Learning (ICML)*, pages PMLR 97:3905–3914, 2019.

[22] Andreas Look and Stefan Riedelbauch. Learning with little data: Evaluation of deep learning algorithms. 2018.

[23] David Lopez-Paz, Krikamol Muandet, and Benjamin Recht. The randomized causation coefficient. *The Journal of Machine Learning Research*, 16(1):2901–2907, 2015.

[24] David Lopez-Paz, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Léon Bottou. Discovering causal signals in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6979–6987, 2017.

[25] Jovana Mitrovic, Dino Sejdinovic, and Yee Whye Teh. Causal inference via kernel deviance measures. In *Advances in Neural Information Processing Systems*, pages 6986–6994, 2018.

[26] Ricardo Pio Monti, Kun Zhang, and Aapo Hyvarinen. Causal discovery with general non-linear relationships using non-linear ICA. In *35th Conference on Uncertainty in Artificial Intelligence (UAI 2019)*, 2019.

[27] Joris M Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *The Journal of Machine Learning Research*, 17(1):1103–1204, 2016.

[28] Krikamol Muandet, Kenji Fukumizu, Bharath Sriperumbudur, and Bernhard Schölkopf. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[29] Leland Gerson Neuberg. Causality: models, reasoning, and inference, by judea pearl, cambridge university press, 2000. *Econometric Theory*, 19(4):675–685, 2003.

[30] Judea Pearl. *Causality*. Cambridge university press, 2009.

[31] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[32] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[33] James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Advances in Neural Information Processing Systems*, pages 7957–7968, 2019.

[34] Ankit Sharma, Garima Gupta, Ranjitha Prasad, Arnab Chatterjee, Lovekesh Vig, and Gautam Shroff. Metaci: Meta-learning for causal inference in a heterogeneous population. *arXiv preprint arXiv:1912.03960*, 2019.

[35] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, and Antti Kerminen. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(Oct):2003–2030, 2006.

[36] Le Song, Kenji Fukumizu, and Arthur Gretton. Kernel embeddings of conditional distributions: A unified kernel framework for nonparametric inference in graphical models. *IEEE Signal Processing Magazine*, 30(4):98–111, 2013.

[37] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. 2000.

[38] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(70):2389–2410, 2011.

[39] Oliver Stegle, Dominik Janzing, Kun Zhang, Joris M Mooij, and Bernhard Schölkopf. Probabilistic latent variable models for distinguishing between cause and effect. In *Advances in neural information processing systems*, pages 1687–1695, 2010.

[40] Xiaohai Sun, Dominik Janzing, and Bernhard Schölkopf. Distinguishing between cause and effect via kernel-based complexity measures for conditional distributions. In *15th European Symposium on Artificial Neural Networks (ESANN 2007)*, pages 441–446. D-Side Publications, 2007.

[41] Xiaohai Sun, Dominik Janzing, Bernhard Schölkopf, and Kenji Fukumizu. A kernel-based causal learning algorithm. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 855–862, New York, NY, USA, 2007. Association for Computing Machinery.

[42] Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

[43] Pengzhou Wu and Kenji Fukumizu. Causal Mosaic: Cause-effect inference via nonlinear ICA and ensemble method. In *Proceedings of Artificial Intelligence and Statistics*, 2020, to appear.

[44] Jin Xu, Jean-Francois Ton, Hyunjik Kim, Adam R Kosiorek, and Yee Whye Teh. Metafun: Meta-learning with iterative functional updates. *arXiv preprint arXiv:1912.02738*, 2019.

[45] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in neural information processing systems*, pages 3391–3401, 2017.

[46] Kun Zhang and Aapo Hyvarinen. On the identifiability of the post-nonlinear causal model. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, page 647–655.

[47] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, UAI'11, page 804–813.

[48] Shengyu Zhu and Zhitang Chen. Causal discovery with reinforcement learning. In *International Conference on Learning Representations*, 2020.