Research Article

Sergio Garrido*, Stanislav Borysov, Jeppe Rich, and Francisco Pereira

Estimating causal effects with the neural autoregressive density estimator

<https://doi.org/10.1515/jci-2020-0007>

received March 16, 2020; accepted July 12, 2021

Abstract: The estimation of causal effects is fundamental in situations where the underlying system will be subject to active interventions. Part of building a causal inference engine is defining how variables relate to each other, that is, defining the functional relationship between variables entailed by the graph conditional dependencies. In this article, we deviate from the common assumption of linear relationships in causal models by making use of neural autoregressive density estimators and use them to estimate causal effects within Pearl's do-calculus framework. Using synthetic data, we show that the approach can retrieve causal effects from non-linear systems without explicitly modeling the interactions between the variables and include confidence bands using the non-parametric bootstrap. We also explore scenarios that deviate from the ideal causal effect estimation setting such as poor data support or unobserved confounders.

Keywords: model specification, neural networks, generative models, do-calculus

MSC 2020: 62D20, 62G07, 62M45

1 Introduction

One way of thinking about causal models is to consider them as inference engines [1]. These engines take causal assumptions and queries as input and give an answer to those queries as output. According to refs. [2,3], there exists a hierarchy where the different levels of the hierarchy represent types of causal queries that can be answered. In particular, a researcher might ask questions about associations, interventions, or counterfactuals. This article focuses on the second level of this hierarchy, namely, intervention queries.

The most important assumption required to compute interventional queries is that the topology of the causal graph is known. This topology can be elicited either by expert knowledge or by “causal discovery” algorithms. Regardless of where the causal topology comes from, we will assume throughout the article that the causal graph is known. However, sensitivity with respect to this assumption will be investigated in the Experimental section.

Pearl's framework of causality was conceived in a non-parametric way, in the sense that he moved away from the assumption of cause-effect relations through simple parametric equations. These assumptions (e.g., a variable being a linear function of its causes) are often based on unrealistic simplifications and are not strictly required to estimate the models. While the assumption of linearity makes the estimation and, in some cases, the interpretation easier, such simplifications can have serious consequences for model parameters and predictions. In this article, we build on recent advancements in the deep generative modeling literature [4,5] to demonstrate how researchers can move away from these assumptions to

* **Corresponding author: Sergio Garrido**, Department of Transport, Technical University of Denmark, Lyngby, Denmark, e-mail: shgm@dtu.dk

Stanislav Borysov: Department of Transport, Technical University of Denmark, Lyngby, Denmark, e-mail: stabo@dtu.dk

Jeppe Rich: Department of Transport, Technical University of Denmark, Lyngby, Denmark, e-mail: rich@dtu.dk

Francisco Pereira: Department of Transport, Technical University of Denmark, Lyngby, Denmark, e-mail: pereira@dtu.dk

estimate these non-parametric relationships in practice. In particular, we propose the use of neural autoregressive density estimators (NADEs) [4,6,7] to model independent causal mechanisms [8]. The contributions of this article are the following:

- We propose a new way to estimate causal effects using Pearl's do-calculus. Using this framework, we can estimate arbitrary interventions as long as these can be identified. The "back-door" criterion and the "front-door" adjustment, explained in Section 4, are special cases hereof.
- The proposed estimation method scales *in principle* to high dimensions and complex relationships between the variables, as long as the structure representing these relationships is known.
- The resulting model preserves the properties of a generative model. It is therefore straightforward to create new data from the model or to impute missing values by doing ancestral sampling.
- Based on simulation we explore sensitivity with respect to different assumptions and demonstrate that the topology of the causal graph is indeed a sensitive assumption.

2 Related work

Estimating causal effects using linear functions is the common practice in the social sciences literature [9–11]. A common assumption in these studies is that the outcome variable is modelled as a linear function of the parameters and the covariates. An exception is when there is additional domain expertise as how the covariates relate to the outcome. In such cases, non-linear transformations of the covariates, such as polynomials, can be used. This way of modeling causal effects is restrictive and mainly applies to the estimation of single causal effects.

One of the main motivations of using neural networks in order to estimate causal mechanisms is that they can be viewed as universal approximators [12]. It is worth noting that while neural networks are a natural choice due to their flexibility with respect to both continuous and discrete problems, other universal approximators or powerful conditional density estimators, such as Gaussian processes [13], could have been used in the same way.

2.1 Non-linear causal effect estimation

The idea of estimating causal effects in non-linear settings is not new. For example, Hill [14] proposed a non-parametric Bayesian method – the Bayesian additive regression trees (BART) – to estimate average treatment effects (ATEs). More recently, the literature on non-linear causal effect estimation focused on estimating individual treatment effects, for example, by using representations of a feed-forward neural network [15,16], multi-task Gaussian processes [17], or a variant of a generative adversarial networks [18]. The approach proposed in this article can be viewed as a generalization of the idea in Hartford *et al.* [19], where deep neural networks are used to estimate causal effects using instrumental variables (deepIV), and further extended by Bennett *et al.* [20] to a generalized method of moments approach for instrumental variables. However, both papers are only concerned with the estimation of causal effects using instrumental variables, a special case of causal identification. All of this literature is mainly focused on the potential outcome framework literature, where the assumptions related to the causal graph are not stated as transparently as in Pearl's framework.

There has been a recent surge in the literature combining the causal graph and formalism with neural estimation approaches. In particular, ref. [21] used a neural network that takes as input the adjacency matrix of a causal graph in order to predict parameters of a sum-product network that estimates an interventional distribution. This work differs from ours in two important ways. First, their proposed method relies on heterogeneous data, that is, both observational and interventional data; and second, their method does not require to make explicit assumptions about the distribution of the variables that conform the system, while our method does. The work by Pawlowski *et al.* [22] is very close to the method proposed in

this article. There, the authors used deep generative models such as variational autoencoders and normalizing flows in order to estimate counterfactuals from data. Their approach requires assumptions similar to ours: both that the topology of the causal graph is known and some assumptions on the distribution of the variables of the system. They focus on the estimation of counterfactuals, which is the third level in Pearl's ladder of causation.

2.2 Neural autoregressive density estimators

The literature related to autoregressive density estimators is comprehensive. As mentioned in Section 1, we are interested in approaches that use neural networks to learn useful representations from data. Here, we can highlight Bengio and Bengio [6] who introduced the idea of using neural networks to estimate the parameters of distributions of any variable given previously observed variables. Larochelle and Murray [4], Uria et al. [7], and Germain et al. [23] extended this idea in order to, for example, make the neural network training more efficient or model a richer family of distributions. To the best of our knowledge, NADEs have not been used for causal inference.

3 Methods

We begin this section by stating clearly the assumptions we need in order to use the method we propose in this article. We continue the section by reviewing the basic terminology of graphical models in order to present the rules of do-calculus. We finish this section by presenting autoregressive density estimators, and the way they connect to causal modeling.

3.1 Assumptions

In order to use neural autoregressive density estimators to estimate causal effects we require the following assumptions:

- The underlying causal graph of the process is known. This graph is a directed acyclic graph (DAG) as presented in Section 3.2. Relaxations with respect to this assumption are explored in Section 4.6.
- The distribution family of every variable in the process is known. This is not so restrictive as we will see in Section 4.3.

These assumptions, either implicitly or explicitly, are in accordance with the common practice when estimating interventional distributions. However, it is worth stressing that in contrast to the majority of models in the literature, we do not make any assumptions about the functional form of the relationships between the variables and the corresponding parameters. These assumptions are relaxed exactly for the reason of being able to estimate more robust causal effect models for which these relationships are not known in advance.

3.2 Graphical models

Consider a set of J random variables $X_1, \dots, X_J$ representing a system of interest. Furthermore, let lowercase letter variables $x_1, \dots, x_J$ denote realizations of these random variables. We can represent such a system with

a graph $\mathcal{G}$ composed of vertices V that represent the variable, and edges E that represent the conditional dependencies between the variables. In this article, we work with DAG, meaning that (1) edges are directed from one variable to another (for example, $X_i \rightarrow X_j$), and (2) there are no cycles in the graph; in other words, there are no directed paths of a variable to itself (there is no j with $X_j \rightarrow \dots \rightarrow X_j$). The inclusion of both variables and conditional dependencies in a causal graph entails a joint probability distribution of the variables. The joint probability distribution is composed of the product of the distribution of each variable, conditioned on its parents on the graph $P(X) = \prod_j P(X_j | \text{PA}(X_j))$. This is also known as the causal Markov condition.

Working with graphs allows researchers to include causal semantics in their models, that is, it allows them to include a hypothesized *true* data generating process. A direct implication from such a hypothesized model is that researchers can simulate interventions to their system of interest. In addition, if the hypothesized model is correct and the mechanisms involved in the process are known, the outcome of such an intervention should correspond to an intervention in the real world. Mathematically, the interventional distribution of an outcome variable X_o under an intervention $X_j = x_j$ can be defined as $P(X_o | \text{do}(X_j = x_j))$ [2]. In this equation, the *do* operator, “ $\text{do}(X_j = x_j)$,” expresses that X_j is forced to be x_j , and any incoming edge into X_j is deleted.

3.3 Do-calculus

However, in many cases, the mechanisms are not known exactly, and performing an intervention in the real world is not possible. Therefore, it is required to estimate the effect of an intervention from observational data. In such cases, the interventional distributions can be derived as operations of observational distributions using the rules of do-calculus [2]. We define $\mathcal{G}_{\bar{X}_j}$ as the graph $\mathcal{G}$ after deleting all the incoming arrows to X_j . In a similar way, we define $\mathcal{G}_{\bar{X}_j}$ as the graph $\mathcal{G}$ after deleting all the outgoing arrows from X_j . Finally, we define the conditional independence relation of X_j, X_i conditioned on X_k as $X_j \perp\!\!\!\perp X_i | X_k$.

Theorem 3.4.1 in ref. [2], which outlines the rules of do-calculus, allowing us to identify interventional distributions from observational ones reads as follows:

Let $\mathcal{G}$ be the DAG associated with a causal model, and let $P(\cdot)$ be the probability distribution induced by that model. For any disjoint subsets of variables X, Y, Z , and W , we have the following rules:

Rule 1 (Insertion/deletion of observations):

$$P(y | \text{do}(X = x), z, w) = P(y | \text{do}(X = x), w) \quad \text{if} \quad Y \perp\!\!\!\perp_{\mathcal{G}_{\bar{X}}} Z | X, W.$$

Rule 2 (Action/observation exchange):

$$P(y | \text{do}(X = x), \text{do}(Z = z), w) = P(y | \text{do}(X = x), z, w) \quad \text{if} \quad Y \perp\!\!\!\perp_{\mathcal{G}_{\bar{X}Z}} Z | X, W.$$

Rule 3 (Insertion/deletion of actions):

$$P(y | \text{do}(X = x), \text{do}(Z = z), w) = P(y | \text{do}(X = x), w) \quad \text{if} \quad Y \perp\!\!\!\perp_{\mathcal{G}_{\bar{X}Z(W)}} Z | X, W,$$

where $Z(W)$ is the set of Z -nodes that are not ancestors of any W -node in $\mathcal{G}_{\bar{X}}$.

The interventional distribution allows us to estimate other quantities of interest, for example, the ATE of treatment x_j relative to treatment $\hat{x}_j$, on X_o as

$$\text{ATE} = E[X_o | \text{do}(X_j = x_j)] - E[X_o | \text{do}(X_j = \hat{x}_j)]. \quad (1)$$

3.4 Autoregressive density estimators

Before introducing the relationship with causal models, we briefly explain the concept of NADEs. These models are a family of generative models that attempt to estimate the joint distribution of the variables by

factorizing the joint distribution into a product of conditional distributions [4,6,23]. By using the chain rule of probability, any joint probability distribution $P(X)$ can be factorized as a conditional distribution where the probability of X_j is conditioned on a function f_j of its “parents:” $PA(X_j)$. Note, however, that in the machine learning literature these parents have no causal semantics as explained in the previous section. This applies to any arbitrary graph that contains all the variables of interest:

$$P(X) = \prod_j^J P(X_j | f_j(PA(X_j))). \quad (2)$$

Even though ref. [6] suggested that these models can be constructed on the basis of probabilistic causal models, most of the machine learning literature has moved away from this approach. Instead, the machine learning literature has focused on proving that the densities estimated by these models are robust with respect to changes in the joint probability factorization. There is a body of evidence from the deep generative model literature that suggests this is indeed possible [7,23,24]. Some of these models are valid as an approximation of associational inference. However, as we are interested in causal inference, and more specifically, interventional type of inference, further assumptions with respect to the factorization of the distribution are required. In other words, we cannot just impose an arbitrary factorization of conditional distributions as in the body of neural autoregressive estimator literature. Instead, we must apply a “causal factorization” as discussed in ref. [25]. The individual conditional distributions are also called independent causal mechanisms or Markov Kernels [26].

We propose to parametrize the functions f_j as independent fully connected feed-forward neural networks and use them as independent causal mechanisms in order to estimate the effects of interventions. For every variable X_j in our system, there is a neural network. Each of the neural networks takes the parents of the variable of interest $PA(X_j)$ and outputs parameters that define the distribution of X_j . For example, if X_j is a binary variable, the network outputs a probability, p , of $X_j = 1$. Likewise, if X_j is continuous and approximately Gaussian distributed, the neural network would output the mean μ and the standard deviation σ of X_j . These networks are trained jointly, using the negative log-likelihood in equation (2) as the loss function. Figure 1 illustrates an example of such a model for a causal graph of the “kidney stone” example [26,27]. This graph contains assumptions related to the conditional dependencies between variables, for example, that R (recovery) is dependent on both KS (size of kidney stones) and T (treatments). Each network takes parents of each variable as input and outputs the parameters of the respective variable. In this case, the output is a single neuron, since each random variable follows a Bernoulli distribution with parameter p .

Figure 1: Example of an NADE model. (Left) The causal graph we want to represent with neural networks. (Right) The NADE representing the causal graph. All the variables are parameterized as Bernoulli distributed random variables.

3.5 Connection to Pearl's do-calculus

The way to connect the model (described in the previous subsection) to the do-calculus framework is by realizing that equation (2) is closely related to the truncated factorization formula [2]. For an intervention where X_j takes a single value, $\text{do}(X_j = x'_j)$, this corresponds to

$$P(X_{o \neq j} | \text{do}(X_j = x'_j)) = \begin{cases} \prod_{o \neq j} P(X_o | \text{PA}(X_o)) & \text{if } X_j = x'_j \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

The neural networks will approximate each of these conditional distributions or independent causal mechanisms [26] that are fundamental to the estimation of the causal effects. In order to compute causal effects, it is required that effects can be identified and that we use the rules of the do-calculus as explained in more detail in refs. [2,3] and [28].

4 Simulation experiments

In order to present a use-case of the neural network approximation of the independent causal mechanisms, we present several simulation experiments based on the “kidney stones” example [26,27]. The case, which has been studied widely in the literature to show the difference between conditioning and intervening, is an excellent reference case for our approach.

In total, we run nine different experiments. In the first two, we show that our approach is able to recover linear functions both in binary and continuous variable settings. In the following seven experiments, we illustrate the ability of the approach to recover causal effects in non-linear settings where, as expected, a linear estimator is unable to recover the true causal effects. In addition, it is shown that the approach can be extended to the estimation of interventions in causal graphs where a “front-door adjustment” is required. This contrasts with the literature of using neural networks to estimate causal effects, restricted to specific scenarios such as instrumental variables [19]. A neural network with no activation functions is used as a benchmark. Such neural network is equivalent to a linear regression model. More details on the experimental setup can be found in Appendix A.

The kidney stone example contains three variables: stone size (KS), treatment type, (T), and Recovery (R). The assumed causal graph is presented in Figure 1, and we assume we observe all the variables, unless stated otherwise. The original observed values from Charig *et al.* [27] are reproduced in Table 1, while the data generation process (DGP) is presented in Algorithm 1.

Algorithm 1. DGP for the kidney stone example

```

1:  for each individual  $i$  do
2:    Draw  $KS \sim P(KS = \text{Small}) = 0.51$  (357/700)
3:    Draw  $T \sim P(T = A | KS = S) = 0.24$  (87/357) and
       $P(T = A | KS = L) = 0.77$  (263/343)
4:    Draw  $R$  with  $P(R | KS, T)$  as in Table 1.
5:  end for

```

The distribution of the recovery, R , under an intervention for a treatment T , is defined as $P(R | \text{do}(T = t))$, and is calculated analytically using a result known as the “backdoor adjustment formula” [2]. This result can be derived by using the rules of do-calculus in combination with the estimated probability distributions:

Table 1: Distribution of the original kidney stone example

Size	Treatment type	
	A	B
Small	93% (81/87)	87% (234/270)
Large	73% (192/263)	69% (55/80)
Total	78% (273/350)	83% (289/350)

$$\begin{aligned}
P(R = 1|\text{do}(T = A)) &= \sum_{ks} P(R = 1|T = A, KS = ks)P(KS = ks) \\
P(R = 1|\text{do}(T = B)) &= \sum_{ks} P(R = 1|T = B, KS = ks)P(KS = ks) \\
ATE &= E[R = 1|\text{do}(T = A)] - E[R = 1|\text{do}(T = B)].
\end{aligned} \tag{4}$$

In order to estimate the causal effects using the adjustment formula with the proposed neural estimator, one needs to propagate forward the specific neural network, which represents a specific causal mechanism, using the appropriate conditioning values. This is possible due to the modular nature of the proposed estimator, as explained in Section 3.4. Finally, the sum implies that the networks should be propagated several times, however, this can be easily done in parallel.

4.1 Experiment I: binary variables

To estimate causal effects with binary outcomes from Algorithm 1, we need to estimate the distributions in equation (4). In order to do this, we used neural networks as shown in Figure 1. There is one neural network for each variable in the system. Each neural network takes the parents of the corresponding variable as input and produces the parameters of a distribution as output. In this case, all the variables were modelled as random variables with Bernoulli distributions. This means that, for all neural networks in our system, the output of each one of them was a single parameter p .

Table 2 presents the estimated conditionals. The fact that we are able to recover the probabilities from the generated data is not surprising, since knowing the structure of the causal graph reduces the estimation problem to a conditional density estimation problem. For a linear model, the resulting ATE is 6.43%. This value is close to the analytical one since, in this model, the probability of the outcome does not include non-linear relationships between the covariates.

Table 2: Estimates of the probabilities estimated by the proposed model, and ATE in the kidney stone example with data generated only from binary variables, and the interventional distributions were estimated using equation (4)

Neural model results	
$p(\text{large})$	48.97%
$p(R = 1 \text{large}, A)$	75.58%
$p(R = 1 \text{large}, B)$	69.68%
$p(R = 1 \text{small}, A)$	95.68%
$p(R = 1 \text{small}, B)$	88.94%
Neural ATE	6.32%
Linear ATE	6.43%
True ATE	5.3%

Table 3: Estimates of the ATE under a continuous recovery model

Continuous case ATE (equation (4))	
Analytical ATE	4
Neural ATE	3.89
Linear ATE	3.73

4.2 Experiment II: continuous response variables

In the second simulation experiment, we generated the confounding variable KS , and the treatment variable T as before, but now changed the recovery variable R to be normally distributed with $R \sim \mathcal{N}(R|\mu = T * 4 + \exp(KS), \sigma = 2)$. That is, we changed Line 4 of Algorithm 1 to follow a normal distribution conditional on treatment and kidney size. The analytical causal effect of the treatment variable is 4. In this case, the output of the neural network that corresponds to the recovery variable has two units, one for the mean and one for the variance of a normal distribution. In that way, we are able to estimate the likelihood of the parameters, conditioned on our data, given the assumption of a normally distributed recovery variable. Table 3 shows the results of the proposed method. The treatment effect from the generated data is equivalent to the effect derived analytically. The difference between the analytical ATE and the neural approach is 0.11 points, while the difference with the linear model is 0.27 points. Neither of these differences is large enough to invalidate one of the approaches. However, this phenomenon does tell that the neural approach is competitive also in linear settings.

4.3 Experiment III: continuous confounding variables

In the third simulation experiment, the confounding variable KS was generated as a continuous variable. In this experiment, the aim was to test the impact of the second assumption in Section 3.1 that the distribution family of every variable in the process is known. To do this, two different data sets with different distributions of the confounding variable – gamma and lognormal – were generated. That is, we changed Line 2 in Algorithm 1 with a gamma distribution (left column of Table 4) or a lognormal distribution (right column of the same table). Due to the non-differentiable nature of its parameters, gamma distributions cannot be learnt in a straightforward way using neural networks. As a result, we used a lognormal parameterization for both generated data sets. That is, in the first neural network of Figure 1, the output is now a mean and a variance with which we can evaluate the likelihood using the kidney stone data.

The treatment variable was still a binary variable. A cutoff equal of 10 (the mean of the confounder) was applied to decide whether the kidney stone is small or large $T = 1\{KS > 10\}$. After deciding whether the kidney stone is large or small, we assigned the treatment using the same probabilities as in Table 1. Finally, the response variable was defined as a normal distribution $R \sim \mathcal{N}(R|\mu = T * 4 + KS, \sigma = 2)$. In the second data set, the confounding variable was drawn from a lognormal distribution with $\mu = 2.5$ and $\sigma = 0.25$. The rest of the DGP was the same.

Table 4: Estimates of the ATE with a continuous confounding variable generated by a gamma and a lognormal distribution

Continuous confounding ATE (equation (4))		
	Gamma confounding	Lognormal confounding
1 sample	3.77	4.15
5 samples	3.77	4.15
25 samples	3.77	4.15
50 samples	3.77	4.15
Total analytical ATE		4

The DGP for this experiment can be summarized as follows:

Algorithm 2. DGP for a continuous confounder

```

1:  for each individual  $i$  do
2:    Draw  $KS \sim \text{Gamma}(5, 2)$  or, in the second set  $KS \sim \text{Lognormal}(2.5, 0.25)$ 
3:    if  $KS > 10$  then
4:      Draw  $T$  with  $P(T = A) = 263/343$ 
5:    else if  $KS \leq 10$  then
6:      Draw  $T$  with  $P(T = A) = 87/357$ 
7:    end if
8:    Draw  $R \sim \mathcal{N}(R|\mu = T * 4 + KS, \sigma = 2)$ 
9:  end for
  
```

Since we have a continuous confounding variable, the sum in equation (4) turns into an integral. If the integral is mathematically intractable, numerical approximations can be used. It is proposed here to use Monte Carlo integration in order to obtain an approximation of the interventional distribution and the ATE. To do this, we sample from the inferred distribution and propagate forward through the neural network to get different estimates of the response variable.

The results of this experiment are shown in Table 4. Even though the analytical treatment effects are not recovered perfectly, they are approximated quite accurately in this way. The ATE is the same for any number of samples, as the functional relation between the mean of the outcome and the confounder is linear.

These results also reveal that the assumption of knowing the exact parameterization of the distribution is not critical. This is supported by the possibility of increasing the complexity of any modeled variable by using a mixture of densities [29] or normalizing flows [30–32]. In other words, we can increase the number of output units of any neural network to parametrize a variable with a highly complex distribution. Since the DGP of the outcome variable is linear, the linear benchmark is able to recover the treatment effects in the same way as the proposed approach.

4.4 Experiment IV: non-linear relation between all the variables in the system

In the fourth simulation study, data for both the confounding and the recovery value were generated as continuous variables. In this case, the effect of the confounding and treatment variable on the recovery is non-linear. In fact, this setting allows us to compute treatment effects conditioned on the confounding variable.

The confounding variable was simulated from a Log-normal distribution, $KS \sim \text{Lognormal}(\mu = 2.5, \sigma = 0.25)$. The treatment, as in the previous case, was modeled using a Bernoulli distribution with probability $p = \frac{1}{1 + \exp((KS - \mu)/10)}$. p was modeled using a transformation of KS in order to avoid extreme probabilities.

Finally, the response variable was simulated from a normal distribution, $R \sim \mathcal{N}\left(\frac{50T}{KS+3}, 1\right)$.

We summarize the DGP in the following algorithm:

Algorithm 3. DGP for the non-linear relation experiment

- 1: **for** each individual i **do**
 - 2: Draw $KS \sim \text{Lognormal}(2.5, 0.25)$, we subtract the mean of KS .
 - 3: Draw T with $P(T = A) = \frac{1}{1 + \exp(-KS/10)}$
 - 4: Draw $R \sim \mathcal{N}\left(\frac{50T}{KS + 3}, 1\right)$
 - 5: **end for**
-

These definitions make treatment effects non-linear, as illustrated in Figure 2, in which the means and 90% confidence bands using the non-parametric bootstrap [33] are plotted for both the neural approach (left) and the linear approach (right). The results are not surprising for the linear model, as the linear model is only able to retrieve a pooled estimate of the treatment effects. On the other hand, the neural network model is able to estimate the individual treatment precisely.

Nevertheless, Figure 2 indicates some weakness in our approach. As we move away from the support of the data, the estimated treatment effects become less precise. For values below $KS = 8$, the estimated treatment effect deviates drastically from the analytical one. The ability to accurately estimate causal effects out of the support of the observed data using neural networks depends on the predictive power of the neural network in those areas of the data. In other words, the estimates of the out of distribution causal effects depend on the generalizability of the neural networks.

In case of a linear model, the ATE is constant. That is, regardless of the value of KS , a unique value for the causal effect is found. The problem with this estimation is that the non-linear relation between the confounding variable and the treatment is aggregated and thereby give rise to aggregation bias [34]. This bias can be of any order of magnitude. For example, in our simulation study, with $R \sim \mathcal{N}(R|4 * T * \exp(-99I) + KS)$, the aggregated treatment effect would be close to 2 and underestimate the effect for small KS and overestimating it for large values.

Figure 2: Comparison between the linear and neural network models using the back-door adjustment formula. The solid blue line represents the mean of the bootstrap, while the bands around the mean represent the 90% confidence bands. A histogram of KS is represented in the background of the plots.

4.5 Experiment V: the front door adjustment

In the previous simulation experiments, all the variables in the causal graph were observed. In fact, because of the simplicity of the graph, the causal effects collapsed to a conditional estimation as noted in Section 4.4. In this simulation study, the aim is to highlight that our approach works in conditions where there are unobserved confounders and, as a result, the back-door criterion does not hold. In Figure 3, a causal graph with those conditions is presented. The white background of the KS circle denotes an unobserved variable and, hence, equation (4) cannot be used.

In this case, KS was simulated from a standard normal distribution, and T was simulated from a normal distribution $T \sim \mathcal{N}(\sin(KS), 0.1)$. Moreover, Mg , which is a mediator variable representing the amount of a certain chemical (in this case magnesium) in the treatment, was simulated from a normal distribution $Mg \sim \mathcal{N}(1 + T^2, 0.1)$, while R was simulated from a normal distribution $R \sim \mathcal{N}\left(\sin(KS^2) + \frac{5}{Mg}, 0.1\right)$. The graphical model and the generative process are shown in Figure 3.

Algorithm 4. Generative process for the front-door adjustment study

- 1: **for** each individual i **do**
 - 2: Draw $KS \sim \mathcal{N}(0, 1)$
 - 3: Draw $T \sim \mathcal{N}(\sin(KS), 0.1)$
 - 4: Draw $Mg \sim \mathcal{N}(1 + T^2, 0.1)$
 - 5: Draw $R \sim \mathcal{N}\left(\sin(KS^2) + \frac{5}{Mg}, 0.1\right)$
 - 6: **end for**
-

Given the conditions of the causal graph, the back-door criterion cannot be used, and we must rely on other methods to estimate the causal effect of T on R . The method to identify the causal effect in this graph is called the front-door adjustment [2]

$$P(R|\text{do}(T = \hat{t})) = \int_{mg} P(Mg = mg|T = \hat{t}) \int_{t'} P(R|Mg = mg, T = t') P(T = t'). \quad (5)$$

4.5.1 Auxiliary networks

As can be seen from equation (5), the neural autoregressive model does not provide all the conditional distributions necessary for the estimation of the treatment effects. In this case, we have to estimate an “auxiliary network,” which can be used to estimate the causal effects. For our particular causal graph, and the causal effect identified in equation (5), the neural network to be estimated takes Mg and T as input and predicts parameters associated with the distribution of R . With that auxiliary network, it is possible to estimate the causal effects.

Figure 3: Graphical model of an extended kidney stone example where the confounder KS is not observed.

Figure 4: Comparison of three different ways to estimate causal effects. (Left) Linear model. (Center) Our proposed approach. (Right) Simple conditioning. (Above) Comparing against the true $do(X = 0)$, and (Below) against the true $do(X = 0.5)$. The WD between the distributions is specified on each plot.

4.5.2 Results

Since the analytical solution of this causal graph is not readily available, the true causal effects were simulated from binned conditional distributions using the true generative process. We compared the distributions using two values of $T = 0$ and $T = 0.5$ (T is constrained between -1 and 1). Furthermore, the integrals in equation (5) are approximated using Monte Carlo integration as explained in Section 4.3.

In Figure 4, we compare the estimated distribution of R for three models with the respective true distribution. On the left, the linear model finds a single distribution for the different values of T , as expected. In the middle, our distribution is seen to match the true simulated distribution. The discrepancies between the true distribution and the real distribution arise from the different elements of stochasticity, e.g., the sampling of the true effects, the stochastic optimization procedure, and the Monte Carlo approximation of the integral. In the plot to the right of Figure 4 it is observed that pure conditioning with respect to T and integrating over Mg from the conditional network $P(R|X = x, Mg = mg)$ render a distribution that is very different from the true distribution. We also included the Wasserstein Distance (WD) in every plot as a metric of comparison between the different models and the true distribution. The proposed approach performs better for the interventional distributions when compared to the other approaches.

4.6 Experiment VI: unobserved confounder

In this section, we explore the robustness of our model when the assumption of knowing the causal graph is not satisfied. To examine this, three additional experiments were carried out. Two experiments in which the unobserved confounder has a linear relationship with the outcome variable with different “degrees” of confounding, and one experiment in which the relationship is non-linear. In all cases, it was assumed that there is a single unobserved confounder U , which affects both treatment and outcome variables. The factor multiplying this term in the “mild” case is 0.3 and 3 for the “strong” case. The true causal graph and DGP can be summarized as follows (Figure 5):

Algorithm 5. DGP for unobserved confounder experiment

```

1:      for each individual  $i$  do
2:          Draw  $KS \sim \text{Lognormal}(2.5, 0.25)$ , we subtract the mean of  $KS$ .
3:          Draw  $U \sim \mathcal{N}(0, 1)$ .
4:          Draw  $T$  with  $P(T = A) = \frac{1}{1 + \exp((-KS - U)/10)}$ .
5:          Draw  $R \sim \mathcal{N}\left(\frac{50T}{KS + 3} + 0.3TU, 1\right)$ .
6:      end for

```

The assumption of a known graph structure is violated by ignoring the existence of U . In other words, the causal effect is estimated using the same architecture as in Case IV, using the data from Algorithm 5. The results of the best performing models can be found in Figures 6 and 7.

Both figures show a deviation from the true treatment effect as expected. The results for both the neural and the linear model are relatively similar to those without unobserved confounding, in the sense that the neural model learns a non-linear treatment effect and the confidence bands are relatively tight. Likewise, for the mild confounding case, the linear model learns a linear treatment effect with big confidence bands. It is interesting, however, that in the strong confounding case, the confidence bands of the linear model are considerably tighter than the rest of the models. This, we believe, is due to the fact that the confounding captures most of the variance of the outcome variable and, consequently, the best fit linear model is the one closest to the value of the confounding.

The second type of model with which we tested the assumption of a known causal graph is a model where the relationship between the unobserved confounder and the outcome variable is non-linear. The DGP described in Algorithm 6 was used to simulate the data.

Algorithm 6. DGP for the unobserved confounder experiment, including non-linearity

```

1:  for each individual  $i$  do
2:      Draw  $KS \sim \text{Lognormal}(2.5, 0.25)$ , we subtract the mean of  $KS$ .
3:      Draw  $U \sim \mathcal{N}(0, 1)$ .
4:      Draw  $T$  with  $P(T = A) = \frac{1}{1 + \exp((-KS - U)/10)}$ .
5:      Draw  $R \sim \mathcal{N}\left(\frac{50T}{KS + 3} + TU^2, 1\right)$ .
6:  end for

```

The estimated causal effects using this simulation are presented in Figure 8.

As expected from all the simulations where the causal graph is not completely known, the estimated causal effects are far from the actual effects regardless of whether the estimator is linear or non-linear. The main lesson of this study cases is that, regardless of the estimator being used, if the true causal graph is not known, the estimators are not going to be particular strong predictors for the outcome.

Figure 5: Causal graph for the unobserved confounder experiment.

Figure 6: Comparison between the linear and neural network models using the back-door adjustment formula in the **mild** unobserved confounding scenario. The solid blue line represents the mean of the bootstrap, while the bands around the mean represent the 90% confidence bands. A histogram of KS is represented in the background of the plots.

Figure 7: Comparison between the linear and neural network models using the back-door adjustment formula in the **strong** unobserved confounding scenario. The solid blue line represents the mean of the bootstrap, while the bands around the mean represent the 90% confidence bands. A histogram of KS is represented in the background of the plots.

5 Conclusions and future work

In this article, we propose the NADE to estimate the causal conditionals. The motivation of this is to avoid having to make any unnecessary assumptions with respect to the functional form of the independent causal mechanisms. We show in several simulation examples, with different specifications, that the proposed method is able to recover both linear and non-linear relations, superseding any linear model.

Several conclusions are offered. It is shown that our estimator is effective in recovering non-linear functions when the causal graph is known and the model is supported by data. Based on simulation and bootstrapping, we explore the robustness of the estimator when we diverge from the assumption of knowing the causal graph and when the support of the data is weak. As expected, both of these assumptions affect the model results. However, even with limited data support, the model provides a reasonable

Figure 8: Comparison between the linear and neural network models using the back-door adjustment formula in the **non-linear** unobserved confounding scenario. The solid blue line represents the mean of the bootstrap, while the bands around the mean represent the 90% confidence bands. A histogram of KS is represented in the background of the plots.

good approximation, as long as the causal graph is known. Assumptions concerning the knowledge of the causal graph are found to be more critical for the model performance. Hence, not adjusting for unobserved confounders will reduce the quality of the estimated causal effects, regardless of the flexibility of the estimation method.

As a final contribution, the article can be seen as a bridge between theoretical methods from the causality literature and advanced methods from the statistical and machine learning literature.

The article points to several new research directions. Machine learning researchers who commonly apply neural networks as a tool to capture complex unknown relationships can introduce additional structure to their models by integrating causal models into their framework. Such causal models may go beyond the traditional domains in the causal literature and consider settings with data coming from images, audio, and speech. On the other hand, methods from the machine learning literature could also deem useful to social sciences in situations where the systems are well studied (in a causal way) but where functional relationships are not known exactly.

In future research, it would be interesting to explore the potential uses of the learned independent causal mechanisms (or single parent–child networks). For example, they could be used to transfer learning (also known as transportability in the causality literature [1]) or continual learning. In the same context, it would be interesting to study the learned representations in the independent causal mechanisms and whether these representations could be used for “downstream” tasks, for example, in reinforcement learning [35].

Acknowledgements: The authors would like to thank Miguel González Duque and the anonymous reviewers for their insightful comments on previous versions of this article.

Conflict of interest: Authors state no conflict of interest.

References

- [1] Pearl J, Bareinboim E. External validity: From do-calculus to transportability across populations. *Statist Sci.* 2014; 29:579–95.
- [2] Pearl J. *Causality: models, reasoning and inference*. 2nd edition. New York, NY, USA: Cambridge University Press; 2009. ISBN 978-0521895606.

- [3] Pearl J, Mackenzie D. The book of why: the new science of cause and effect. 1st edition. USA: Basic Books, Inc. 2018. ISBN 046509760X.
- [4] Larochelle H, Murray I. The neural autoregressive distribution estimator. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics; 2011. p. 29–37.
- [5] Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge, MA, USA: MIT Press, 2016. <http://www.deeplearningbook.org>.
- [6] Bengio Y, Bengio S. Modeling high-dimensional discrete data with multi-layer neural networks. In Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS'99. Cambridge, MA, USA: MIT Press; 1999. p. 400–6.
- [7] Uribe B, Murray M, Larochelle H. Rnade: The real-valued neural autoregressive density-estimator. *Adv Neural Informa Process Syst.* 2013;26:2175–83.
- [8] Schölkopf B. Causality for machine learning. 2019. arXiv:<http://arXiv.org/abs/arXiv:1911.10500>.
- [9] Angrist JD. Lifetime earnings and the vietnam era draft lottery: evidence from social security administrative records. *Am Econom Rev.* 1990;80(3):313–36. ISSN 00028282.
- [10] Miguel E, Kremer M. Worms: Identifying impacts on education and health in the presence of treatment externalities. *Econometrica.* 2004;72(1):159–217.
- [11] Angrist JD, Pischke J-S. Mostly harmless econometrics: an empiricist's companion. Princeton, NJ, USA: Princeton University Press; 2008.
- [12] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math Control Signal Syst.* December 1989;2(4):303–14. doi: 10.1007/bf02551274.
- [13] Williams CK, Rasmussen ER. Gaussian processes for machine learning. Vol. 2. Cambridge, MA: MIT Press; 2006.
- [14] Hill JL. Bayesian non-parametric modeling for causal inference. *J Comput Graph Stat.* January 2011;20(1):217–40. doi: 10.1198/jcgs.2010.08162.
- [15] Johansson F, Shalit U, Sontag D. Learning representations for counterfactual inference. In International Conference on Machine Learning; 2016. p. 3020–9.
- [16] Shi C, Blei DM, Veitch V. Adapting neural networks for the estimation of treatment effects. 2019. arXiv:<http://arXiv.org/abs/arXiv:1906.02120>.
- [17] Alaa AM, van der Schaar M. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In Advances in neural information processing systems. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 3424–32.
- [18] Yoon J, Jordon J, van der Schaar M. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In International Conference on Learning Representations. 2018. <https://openreview.net/forum?id=ByKWUeWA->.
- [19] Hartford J, Lewis G, Leyton-Brown K, Taddy M. Deep iv: A flexible approach for counterfactual prediction. In Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org; 2017. p. 1414–23.
- [20] Bennett A, Kallus N, Schnabel T. Deep generalized method of moments for instrumental variable analysis. In Advances in neural information processing systems. Red Hook, NY, USA: Curran Associates Inc.; 2019. p. 3559–69.
- [21] Zečević M, Singh Dhani D, Karanam A, Natarajan S, Kersting K. Interventional sum-product networks: Causal inference with tractable probabilistic models. 2001. arXiv preprint arXiv:2102.10440.
- [22] Pawlowski N, de Castro DC, Glocker B. Deep structural causal models for tractable counterfactual inference. *Adv Neural Inform Process Syst.* 2020;33:857–69.
- [23] Germain M, Gregor K, Murray I, Larochelle H. Made: Masked autoencoder for distribution estimation. In International Conference on Machine Learning; 2015. p. 881–9.
- [24] Papamakarios G, Pavlakou T, Murray I. Masked autoregressive flow for density estimation. In Advances in neural information processing systems. Red Hook, NY, USA: Curran Associates Inc.; 2017. p. 2338–47.
- [25] Parascandolo G, Kilbertus N, Rojas-Carulla M, Schölkopf B. Learning independent causal mechanisms. 2017. arXiv:<http://arXiv.org/abs/arXiv:1712.00961>.
- [26] Peters J, Janzing D, Schölkopf B. Elements of causal inference: foundations and learning algorithms. Cambridge, MA, USA: The MIT Press; 2017. ISBN 9780262037310.
- [27] Charig CR, Webb DR, Payne SR, Wickham JE. Comparison of treatment of renal calculi by open surgery, percutaneous nephrolithotomy, and extracorporeal shockwave lithotripsy. *Br Med J (Clin Res Ed).* 1986;292(6524):879–82.
- [28] Pearl J. The do-calculus revisited. 2012. arXiv:<http://arXiv.org/abs/arXiv:1210.4852>.
- [29] Bishop CM. Mixture density networks. 1994. <http://publications.aston.ac.uk/id/eprint/373/>.
- [30] Agnelli JP, Cadeiras M, Tabak EG, Turner CV, Vanden-Eijnden E. Clustering and classification through normalizing flows in feature space. *Multiscale Model Simulat.* 2010;8(5):1784–802.
- [31] Tabak EG, Turner CV. A family of non-parametric density estimation algorithms. *Commun Pure Appl Math.* 2013;66(2):145–64.
- [32] Rezende DJ, Mohamed S. Variational inference with normalizing flows. 2015. arXiv:<http://arXiv.org/abs/arXiv:1505.05770>.
- [33] Efron B, Hastie T. Computer age statistical inference. Vol. 5. Cambridge, UK: Cambridge University Press; 2016.
- [34] Stoker TM. Aggregation (Econometrics). UK, London: Palgrave MacMillan; 2016. p. 1–11. ISBN 978-1-349-95121-5. doi: 10.1057/978-1-349-95121-5_2620-1.

- [35] Zhang J. Designing optimal dynamic treatment regimes: a causal reinforcement learning approach. In *International Conference on Machine Learning*. PMLR; 2020. p. 11012–22.
- [36] Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, et al. PyTorch: An imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, d'Alche-Buc F, Fox E, Garnett R, editors. *Advances in neural information processing systems*. Curran Associates, Inc.; 2019:32. <https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf>.
- [37] Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with numpy. *Nature*. 2020;585(7825):357–62.

Appendix

A Detailed experimental setup

For every experiment, we selected the best performing hyperparameters for three types of neural network activations: linear (which collapses to a linear regression), (ii) Rectified Linear Units (ReLU), and (iii) hyperbolic tangent (Tanh). For each one of them, two different optimizers: RMSProp and Stochastic Gradient Descent were tested. For each of the optimizers four different learning rates were used: $1e^{-2}$, $5e^{-3}$, $1e^{-3}$, and $5e^{-4}$. Finally, five different architectures for the neural networks were tested: 1 Hidden Layer (HL) with 4 units, 1 HL with 8 units, 1 HL with 16 units, 2 HL with 4 units each, and 2 HL with 8 units each. For this article, a total of $9 \times 3 \times 4 \times 2 \times 5 = 1,080$ neural networks were trained. All the neural networks were trained using the Python programming language, specifically using the PyTorch [36] package for neural network estimation, and Numpy [37] for synthetic data generation. For each experiment, we generated 5,000 samples to train the neural network, however, in preliminary versions we also tried 2,500 and 1,000 observations without much difference in the results. The code to reproduce the results can be found in https://github.com/Chechgm/causal_effect_estimation_using_nade.

We selected a single neural network for each experiment and each activation function based on its performance with respect to the ground truth using the mean absolute error (MAE) metric. Alternative metrics such as the root mean squared error (RMSE) could also be used, however, the results would be similar. After the best architecture was selected, we ran 50 bootstrap samples in order to build the 90% confidence intervals shown in the plots.

Interesting observations were made during the experimental stage of the research. First, the ReLU activation function was inferior in all experiments to both the tanh and linear activation functions, in the sense that all ReLU architectures had greater error than any tanh or linear architecture. Second, the estimates of the interventional distribution under the front-door adjustment are highly variable. We believe this is the case, because only a single Monte-Carlo sample is used to approximate the second integral of equation (5). Third, no single combination of hyper-parameters seemed to be superior to others. This is a challenge as in real case scenarios as the ground truth causal effects are unobserved (otherwise there is no causal effect learning task). Nevertheless, it was found that better performing networks usually have higher likelihood with respect to the training data compared to those networks that performed poorly. This was not the case in all experiments but there is a correlation between high likelihood and the models that estimate the causal effects closer to the ground truth. This is not a trivial observation, as all the networks were trained to maximize the likelihood: why should the models with higher likelihood also estimate the causal effects best?