SUB CRUCE LUMEN

THE UNIVERSITY
*of* ADELAIDE

# Connecting Machine Learning to Causal Structure Learning with the Jacobian Matrix

Xiongren Chen

Supervised by Prof. Javen Shi and Co-supervised by Dr. Guansong Pang

A thesis submitted for the degree of
MASTER OF PHILOSOPHY
The University of Adelaide

May 8, 2021

# Contents

# List of Figures

# List of Tables

University of Adelaide

# *Abstract*

**Connecting Machine Learning to Causal Structure Learning with the Jacobian Matrix**

by Xiongren Chen

In this thesis, a novel approach is proposed to connect machine learning to causal structure learning with the Jacobian matrix of neural networks w.r.t. input variables. Causal learning distinguishing causes and effects is the way human understanding and modeling the world. In the machine learning era, it also ensures that the model is more interpretable and sufficiently robust. Due to the enormous cost of the traditional intervention and randomized experimental methods, studies of causal learning have focused on passive observational data which can generally be divided into static data and time-series data. For different data types and different levels of causal modeling, different machine learning techniques are applied to do causal modeling and the causal structure can be read out by the Jacobian matrix. We focus on three aspects in this thesis. Firstly, a novel framework of neural networks to causal structure learning on static data under structural causal models assumptions is proposed and the results of various experiments show our method has achieved state-of-the-art performance. Secondly, we extend static data causal modeling to the highest level as the physical system which is usually in terms of ordinary differential equations. Lastly, our Jacobian-based causal modeling framework is applied to time series data with the ORE-RNN technique and the results show that the success of temporal causal structure learning in time series cases.

# Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Xiongren Chen

October 2020

# *Acknowledgements*

I would like to thank my principal supervisor, Professor Javen Shi, one of the best researcher I have ever met, for bringing me to a world leading machine learning institute, the patient guidance, encouragement and advice he has provided throughout my time as his student. I will not forget his massive advice on my thesis, which are extremely important for paper writings. I would like to thank my co-supervisor Dr. Guansong Pang, for teaching me to do research not with the aim of improving technology but by solving specific problems and then verifying the ideas through experiments. Last but not least, I would thank my beloved wife Ruby and daughter, for amazing support throughout my life.

# Chapter 1

# Introduction

In this chapter, we briefly introduce the motivation and contributions of this research and give the organization of this thesis at the end of this chapter.

## 1.1 Motivation of Research

Machine learning has achieved great success in CV, NLP and other fields, especially in the accuracy of prediction and some may have exceeded the human capabilities [1]. However, it is difficult for these models to answer questions about causes and effects. In financial markets, for example, what is at work in the current U.S. stock booming during the pandemic of Covid-19? Is it a quantitative easing policy by the Federal Reserve? So where do the markets go if the Fed doesn't implement monetary easing? In terms of time, how long will the impact of this policy be, a year or a month? Most machine learning algorithms including deep learning are based on correlation, which is encoded and learned to improve accuracy in prediction [2]. However, correlation only shows that there is a relationship between the variables and does not give information about the dependencies in directions. For example, two variables may have a common causal variable, then the two variables are correlated but do not have a causal dependency. Such non-causal relationship models are less explainable and weak robust. When the value of variables interference by other factors outside of the environment, such as human intervention, the model will not get the expected results and the model will break down. Most scientific research requires learning causality rather than correlation between variables. The natural sciences, for example, we need to know the direct causes of global warming, the interactions of cells and viruses, and the effects of policy on climate change. While in finance, we need to know the direct effects of policy on markets. Causality ensures that the model is more interpretable and sufficiently robust. At the same time, a sufficiently robust model based on causality can also answer and solve the problem of external intervention. Since most of the data in machine learning algorithms learned comes from a closed environment, if the closed environment is cracked and the variation in the variables is likely to be due to external interventions, the predictive models encoding correlations between variables in the closed environment data will collapse. Causal models can predict the effect of an externally intervening variable on other variables because they understand the mechanism by which the variable takes its value (usually represented by the Structure Causal Models [3]).

Probability theory(please see the detail at Section 3.2.1) allows us to learn from data to get the probability space(**probabilistic learning**), through which we know the possible distributions of the data, and the distribution obtained by learning can naturally give us the probability of different results of the next experiment(**probabilistic reasoning**). Causality, with the structure of directed edges between variables, not only allows us to predict the results of the next experiment but also can make inferences about interventions, counterfactuals(**causal reasoning**). And the leaning process of the causal structure from random experiments and observed data is **causal structure learning**.

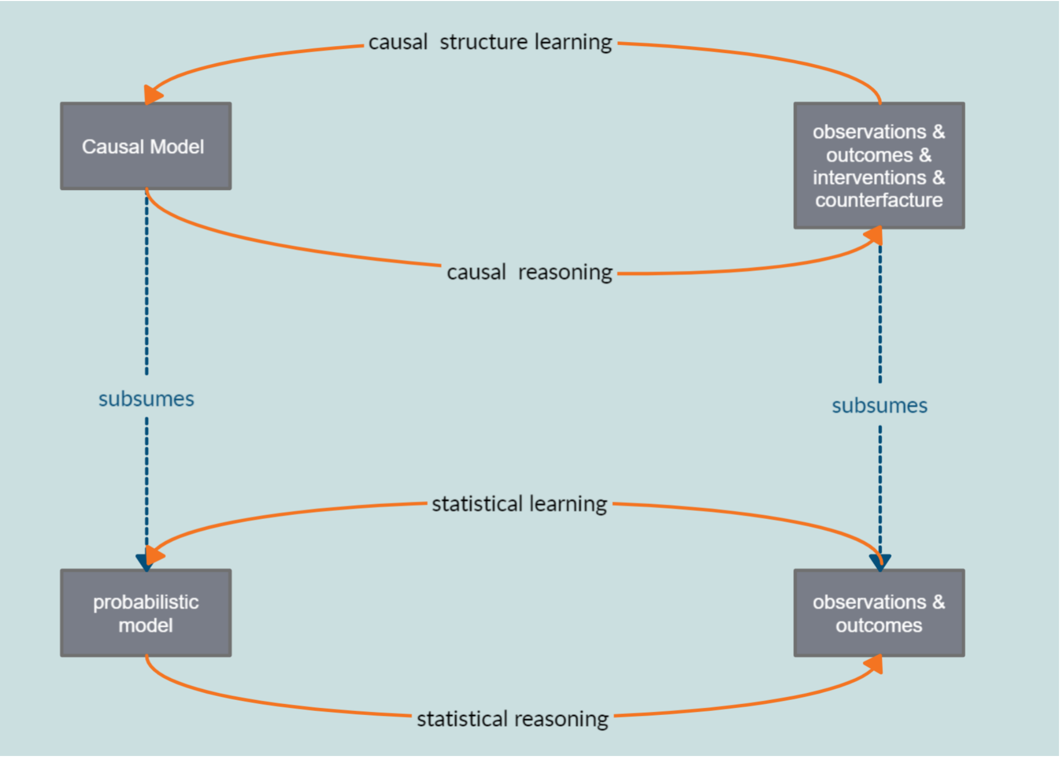
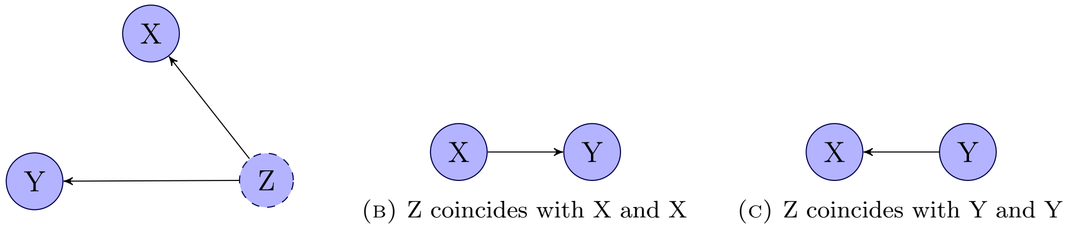


FIGURE 1.1. Adapted from the paper [3]. We will focus on causal structure learning in this thesis.

Reichenbach's common cause principle gives a clear explanation of the connection between statistical and causality [3]: if two random variables $X$ and $Y$ are statistically dependent, then there exists a third variable $Z$ that affects both $X$ and $Y$. In other words, $Z$ screens $X$ and $Y$ from each other in the sense of that $X$ and $Y$ are independent of each other given $Z$. In the form of a graph, there are three nodes $X$, $Y$, and $Z$ and two arrows pointing from $Z$ to $X$ and $Z$ to $Y$. $Z$ may coincide with either $X$ or $Y$, then there are only two points $X$ and $Y$ and one arrow in the graph. If $Z$ and $X$ coincide, then the arrow points from $X$ to $Y$. If $Z$ and $Y$ coincide, then the arrow points from $Y$ to $X$. For example, we have two random variables A={rain, no rain} and B={floor wet, floor not wet}, the corresponding causality is that A causes B, and B cannot cause A. If we show causal relationships in the form of a graph, the nodes are A and B and the direction of the arrow is from A to B.



(A) Z affects both X and Y

(B) Z coincides with X and X causes Y

(C) Z coincides with Y and Y causes X

FIGURE 1.2. Reichenbach's common cause principle gives a clear explanation of three random variables X, Y and Z. [3]

The causal structure can be obtained by the intervention that changes the values of variables and sees their impacts on other variables in experiments or by randomized experiments. However, due to the limitations of the experimental setting, we can not do human interventions or the cost of the interventions is huge. For example, we cannot allow patients to use drugs that have not been clinically validated, which is illegal and unethical. We also can't arbitrarily change financial policy to see how it affects the market, because the costs of doing so are enormous. So most experiments can only get passive observations and we just learn causal relationships between variables relying on observed data. There are numerous non-machine learning methods and machine learning methods are proposed to learn causal structures on observed data. However, non-machine learning methods rely heavily on conditional independent tests and they are inefficient as the number of variables grows. For machine learning methods, the connection between causality and machine learning is unclear so that only simple Multiple Layer Perception(MLP) architecture can be used to handle it. In this thesis, we try to connect machine learning to causal structure learning with the Jacobian matrix of output $Y$ w.r.t input $X$ to link the two clear and use advanced machine learning techniques to do causal modeling.

## 1.2 List of Common Symbols

The common symbols are listed in Table 1.1.

| Symbol | Description |
|---|---|
| $X, Y, Z$ | random variables; $N$ for the noise variable |
| x | value of a random variable |
| $P_X$ | probability distribution of variable $X$ |
| $P(x)$ | probability density of variable $X$ |
| $G$ | graph |
| $\mathbb{E}(X)$ | expectation of $X$ |
| $\pi_j^G$ | the set of parents of node $j$ in $G$ |
| $x_{\pi_j^G}$ | the vector containing the variables corresponding to the parents of $j$ in $G$ |
| $f$ | function |
| $P(A \mid B)$ | the probability of event $A$ given event $B$ |

Table 1.1. List of the common symbols

## 1.3 Contributions

As more information becomes available, there are generally 4 levels of causal modeling which are summarized in the table and we will present these 4 levels in more detail in Section 3.2. Observed data can generally be divided into two categories: static data that is non-time sequence and time-series data. For different data types and for different levels of causal modeling, we use different machine learning methods to do causal modeling, which is summarized in the Table 1.3.

The main contributions of this thesis are summarized as follows:

- A novel approach to bridging causal structure learning and machine learning is proposed for both static and time-series data, which is effective and computationally easy to apply to complex deep learning techniques and opens the possibility of designing better frameworks for causal learning in the future.

| Model | Predict in i.i.d. setting | Predict under changing distr. or intervention | Answer counterfactual questions | Obtain physical insight | Learn from data |
|---|---|---|---|---|---|
| Physical System | yes | yes | yes | yes | ? |
| Structural causal model | yes | yes | yes | ? | ? |
| Causal graphical model | yes | yes | no | ? | ? |
| Statistical | yes | no | no | no | yes |

TABLE 1.2. Adapted from paper [2]: a summarization of different level of Causal Modeling.

| Data Type | Level of Causal modeling | Machine learning techniques we used | Chapter | Contributions |
|---|---|---|---|---|
| Static data | Structural causal model | MAF [34] | 3 | 2.1 |
| Static data | Physical System | Neural ODE [51] | 4 | 3.1 |
| Time Series | Physical System | ODE-RNN [51] | 5 | 4.1 |

TABLE 1.3. Thesis Overview(Main Body)

- Each Chapter's contributions can be viewed at the end of each Chapter's Introduction Section and we summarized references in the Table 1.3.

## 1.4 Organization

The rest of this thesis is organized as follows:

- In Chapter 2, a novel framework of neural networks to causal structure learning on non-sequence(static) data under structural causal models assumptions is proposed and the results of various experiments show our method has achieved state-of-art performance;
- In Chapter 3, we extend non-sequence data causal modeling to "the most detailed model" [2] as the physical system which is usually in terms of ordinary differential equations;
- In Chapter 4, our Jacobian-based causal modeling framework is applied to time series data which widely exists in our world.
- In Chapter 5, we summarize the thesis and point the directions in the possible future works.

# Chapter 2

# Causal Structure Learning for Static Data

## 2.1 Introduction

How to find meaningful relationships, especially causal relationships, from massive amounts of non-sequence observed data is one of the research areas most likely to create business value and make scientific discoveries in data science, and it is receiving widespread attention from international peers. Causality strictly distinguishes between the cause and effect variables and has an important role that cannot be replaced by the relationship in revealing the mechanism of things and guiding the intervention behavior.

Causal learning has been widely studied in many applications. For example, [27] applied causal structure into the operational risk model to learn which human factor attributing to the operational risk in finance, while in the medical field, [28] learned the causal structure of clinical conditions and outcomes from static observation data, and the causal network of protein interaction published in Science [29] has been commonly accepted by researchers in this field. There are also studies of causal inference in epidemiology [30], education [31], and environmental health [32].

In general, performing random experiments is an effective method for obtaining causal relationships [4], but random experimentation is mostly impossible to intervene or the cost of interventions is enormous. Thus, the existing methods of causal learning are mostly based on observational data. In particular, the recent NOTEARS [22] reformulates the combinatorial optimization problem into a continuous problem with acyclicity constraint, which significantly reduces the size of the search space and enables efficient learning of linear structural equation models. Following this line, many methods extend NOTEARS to learn nonlinear causal models by leveraging neural networks. DAG-GNN [23] extends NOTEAR's continuous linear causal model to a non-linear model with VAE and Graph Convolutional Networks (GNN [83]). However, DAG-GNN use an adjacency matrix and neural networks to represent the function $f_j$ but only use the weighted adjacency matrix as causal structure, which makes the method biased. To solve this issue, gradient-based methods, e.g., GraN-DAG [24], determine the causal relationship between two variables through neural network connectivity. However, neither DAG-GNN nor GradN-DAG are suitable for counterfactual inference. Can we have a model that can do causal discovery, interventional effect estimation, and counterfactual reasoning? Causal autoregressive flows (CAREFL) [85] provides a promising way for counterfactual reasoning by using normalizing flows model to reconstruct the structural causal models. However, due to the use of fixed causal order as input, the search space of graphs grows exponentially with increasing size of the input variables, which is not suitable for multivariate causal discovery.

In this chapter, we propose a Jacobian matrix scored-based method called Causal Normalizing Flows (CNF) with the assumption of Additive Noise Models (ANMs) [20] to learn causal dependencies among the input variables. We use the Jacobian matrix of fitted functions w.r.t. input variables as causal relationships and a reverse-order flow which is effective to fit the structural causal models. Our method also extends NOTEARS [22] that enforces the important acyclicity constraint on the continuous adjacency matrix of graph nodes and significantly reduces the computational complexity of the search space of graphs. Furthermore, the flow-based model can do transformations between input variables and noise, so our model naturally supports counterfactual reasoning and interventions.

In summary, this work makes the following five main contributions.

- To the best of our knowledge, we are the first to propose a single flow model that is able to represent all possible causal orderings without enumerating nor even sampling from the huge factorial permutation space. This is based on our key observation: for an arbitrary order, say $(2, 1, 3)$, of the variables fed into a masked autoregressive density estimation (MADE) block, if we feed the MADE block's output to a second MADE block in the reverse order $(3, 1, 2)$, the combined two-block unit is able to represent all possible orderings. Stacking multiple reverse-order units forms a flow that captures richer non-linear causal relations.
- Formulating causal discovery in our reverse-order flow allow us to utilize the Jacobian matrix, which naturally represents contributions of the variables to each SCM function (one function per target variable), hence the final causal graph can be easily inferred.
- The invertible nature of the flows naturally empower counterfactual reasoning, as the exogenous (noise) variables can be easily estimated.
- Extensive experiments show that the our method outperforms the previous methods in a range of tasks including causal discovery, intervention, and counterfactual reasoning.

## 2.2 Background

### 2.2.1 Causal Models

Following [4], we use the Directed Acyclic Graph (DAG) with arrows pointing from the parent (direct cause) node to the child (direct effect) node as a formalism to represent causal relationships. Based on the DAG, there are two major ways to represent the causal mechanism underlying the data distribution, including causal Bayesian Networks (CBNs) and Structural Causal Models (SCMs).

**Causal Bayesian Networks** Let $G = (V, E)$ be a DAG over a set of variables $X = (X_1, X_2, \cdots, X_d)$ and $P$ be a joint distribution over $X$. The pair $\langle G, P \rangle$ is a causal Bayesian network if the Causal Markov condition and modularity condition hold [4]. More specifically, the causal Markov condition implies that the joint distribution $p(\mathbf{x})$ enjoys the the following factorization:

$$p(\mathbf{x}) = \prod_{i=1}^{d} p(x_i | x_{PA_i^G}), \tag{2.1}$$

where $x_{PA_j^G}$ stand for the parent nodes in $G$. This factorization implies a variable $X_i$ is conditionally independent given its parent nodes (direct causes). Modularity means that the causal process of a variable $X_i$, defined by $p(x_i|x_{PA_i^G})$, is invariant when other variables are intervened on, which enables the calculation of intervention distribution using do-calculus [4]. For example, the interventional distribution after intervening on $X_k$ can be written as

$$p(\mathbf{x}|do(X_k = x_k)) = \prod_{i \neq k} p(x_i|x_{PA_i^G})\delta_{X_k, x_k}. \tag{2.2}$$

Causal Bayesian Network provides a principled way to represent causality, but it has two practical drawbacks. First, it is hard to make stronger restrictions to ensure identifiability of causal model learned from data. Second, it is inconvenient to perform counterfactual inference.

**Structural Causal Models** In its general form, a SCM is a tuple $\langle S, P \rangle$ consisting of a set of equations $S = (S_1, \ldots, S_d)$:

$$S_i : X_i := f_i(X_{PA_i^G}, N_i) \qquad i = 1, ..., d, \tag{2.3}$$

and a probability distribution $P$ over $X = (X_1, \ldots, X_d)$. In Eq 2.3, $X_{PA_i}$ denotes the direct causes of $X_i$ and $N_i$ represent disturbances or errors. $N_i$s are required to be jointly independent, i.e., $q(\mathbf{n}) = \prod_{i=1}^{d} q(n_i)$. The causal relations between variables in a SCM can also be represented as a graph $G$ derived from the structural equations.

In an SCM, it is convenient to enforce additional assumptions on $f_i$ and $N_i$ that make the causal structure identifiable, i.e., uniquely recovered, from observational data. For example, if $f_i$ are linear functions and $N_i$ are non-Gaussian, the causal model is identifiable from observational data [17]. In the bivariate case, identifiability can also be guaranteed when $f_i$ are nonlinear functions, for example, the additive noise models (ANMs) [44]. In addition, SCM enables counterfactual inference, in which we infer the values of $N_i$ based on observed $X$ data and manipulate the target variables to calculate counterfactual outcomes [4].

### 2.2.2 The SCM and the Rule of Change of Variables

For Equation 2.3, we try to model $P_X$ by a set of equations and place restrictions on functions $f_i$ so that identifiability can be achieved. If the functions $f_i$ are non-invertible, it is hard to estimate these functions because the corresponding conditional distributions $p(x_i|x_{PA_i^G})$ may not have a simple form. As a consequence, the likelihood might be intractable. Fortunately, if the functions $f_i$ are invertible, we can infer the error terms $N_i$ from equation (2.3) as

$$N_i := f_i^{-1}(X_{PA_i^G}, X_i), \; i = 1, \ldots, d. \tag{2.4}$$

Equivalently, we can write the inverse function as $\mathbf{n} = f^{-1}(\mathbf{x})$. According to the rule of change of variables, we have

$$p(\mathbf{x}) = q(\mathbf{n}) \left| \det \frac{\partial \mathbf{n}}{\partial \mathbf{x}} \right| = q(f^{-1}(\mathbf{x})) \left| \det(\frac{\partial f^{-1}}{\partial \mathbf{x}}) \right|, \tag{2.5}$$

where $\det(\frac{\partial f^{-1}}{\partial \mathbf{x}})$ is the determinant of the Jacobian matrix of $f^{-1}$ over $\mathbf{x}$. Assume the error terms $N_i$ follow a simple distribution, for example, Gaussian distribution, the likelihood can be expressed in terms of the $q$ distribution, provided that the computation of the Jacobian matrix is tractable. Then standard maximum likelihood estimation can be used to estimate the parameters in the functions $f_i$. In the following section, we will review a special type of density estimation model called Masked Autoregressive Density Estimation (MADE) [48], which enjoys invertibility and easy computation of the Jacobian matrix. Our approach will be built upon the MADE model.

### 2.2.3 Masked Autoregressive Density Estimation (MADE) and the Order of Variables

In MADE [48], given any variable order, the joint distribution $p(\mathbf{x})$ can be decomposed into a product of one-dimensional conditionals as $p(\mathbf{x}) = \prod_{i=1}^{d} p(x_i|x_{1:i-1})$, which is called the autoregressive property. MADE uses a masked matrix in each layer of neural networks to ensure the output $x_i$ to depend only on the preceding inputs $x_{1:i-1}$. To simplify the calculation, $p(x_i|x_{1:i-1})$ is chosen to be a simple known distribution such as Gaussian parameterized by mean and variance as

$$p(x_i \mid x_{1:i-1}) = \mathcal{N}(x_i \mid \mu_i, (\exp \alpha_i)^2), \tag{2.6}$$

where $\mu_i = f_{\mu_i}(x_{1:i-1})$ and $\alpha_i = f_{\alpha_i}(x_{1:i-1})$ are nonlinear functions which can be fitted by neural networks. Therefore, the form of $f_j$ and $f_j^{-1}$ in MADE are given by

$$\begin{aligned} f_i &\implies X_i = N_i \exp \alpha_i + \mu_i, \\ f_i^{-1} &\implies N_i = (X_i - \mu_i) \exp(-\alpha_i), \end{aligned} \tag{2.7}$$

where $N_i$ follows a normal distribution. For example, when $d = 3$, we can use a neural network with three nodes to represent $p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)$, as shown in Figure 2.1. This network approximates the following equations:

$$\begin{aligned} X_1 &:= N_1 \exp \alpha_1 + \mu_1, \\ X_2 &:= N_2 \exp \alpha_2(X_1) + \mu_2(X_1), \\ X_3 &:= N_3 \exp \alpha_3(X_1, X_2) + \mu_3(X_1, X_2). \end{aligned} \tag{2.8}$$

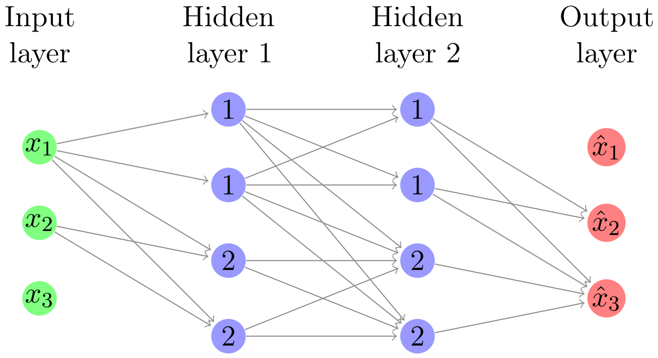


FIGURE 2.1. Autoregressive property of MADE naturally supports causal modeling with known causal order $x_1$, $x_2$, $x_3$.

The connection between MADE and SCM was recently established in [85]. In specific, if the causal order among the variables is unknown, one can consider MADE as a specific SCM in which a variable $x_i$ depends on the variables before it in the form of Eq. 2.8. Then, by fitting stacked MADE in the same order, i.e., causal autoregressive flow [85], to the data, one can obtain the sparse causal graph with additional constraints.

However, in practice, the causal order among variables is usually unknown. [85] proposes to shuffle the variables to obtain all the possible orders, then fit a MADE to the shuffle data, and finally compare the goodness of fit. This method works well in the two variable case, but enumerating or even sampling from the entire permutation space is computationally expensive when $d$ is large. Next, we will show how to extend MADE to represent all possible orderings without enumerating or sampling from the permutation space.

## 2.3 Reverse Order Flow, Causality, and Jacobian Analysis

In this section, we first present the details of our reverse-order flow model which models the distribution input data in both original and reverse order. Then, we demonstrate how the proposed model can be used for causal discovery by analysis of the Jacobian matrix.

### 2.3.1 Reverse-Order Flow

As described in Section 2.3, MADE requires a fixed order to present autoregressive property and is suitable to do causal modeling with known causal order. For example, causal autoregressive flow [85] stacks a series of MADE with fixed order as a causal model. However, for causal discovery, the causal order is what we should learned from the data and thus we introduce reverse-order flow to represent all possible causal orders. We first show that by reversing the order of the output of the first MADE block, and then feeding the order reversed output into the second MADE block, this combined two-block unit, which we call Reverse-Order Unit (ROU), can indeed represent all possible orderings. Here we give a three variable example illustrated in Figure 2.2. We can pick an arbitrary order (the order is not important as you shall see later) say $(2, 1, 3)$. The input layer and two hidden layers and the first output layer form a the first block of MADE, which is followed by the second MADE. Note that in the middle of the unit, the output of the first MADE block is fed into the second MADE in the reverse order $(3, 1, 2)$. This crucial operation enables the final output variables to take all the other variables as inputs (i.e., parents in terms of causality). This removes the limitation of traditional MADE, where a variable can only be represented by the preceding variables. By stacking multiple these units together, we can get an reverse-order Flow that is also able to represent any ordering and and stacked architecture provides richer representations.

### 2.3.2 Reverse-Order Flow for Structural Causal Models

The ability of representing any ordering of the variables makes our reverse-order flow ideal for learning structural causal models from data. Figure 2.2 shows a neural network of ROU with three variables of unknown causal order. The first block of the ROU can infer the relations of $X_i$ on the preceding variables $X_{1:i-1}$ which is $g_1(X_{1:i-1}, X_i)$ while another one inferring relations of $X_i$ on the following variables $X_{i+1:d}$ as $g_2(X_{i+1:d}, g_1(X_{1:i-1}, X_i))$. We can rewrite equation (2.4) as

$$\begin{aligned} N_i &:= g_{2_i}(g_{1_i}(X_{1:i-1}, X_i), X_{i+1:d}). \\ X_i &:= g_{1_i}^{-1}(g_{2_i}^{-1}(X_{i+1:d}, N_i), X_{1:i-1}). \end{aligned} \tag{2.9}$$

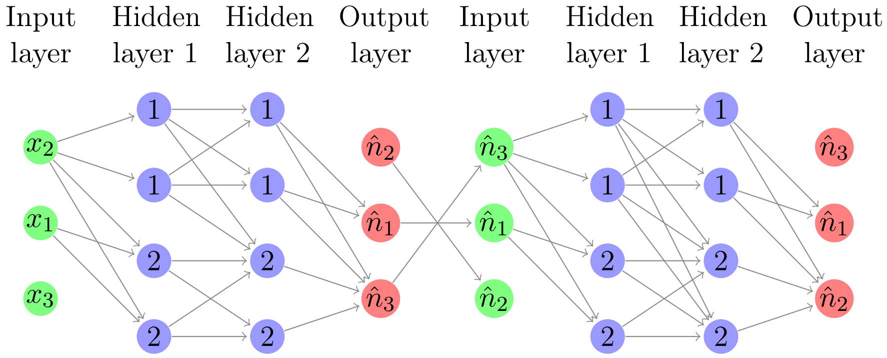


FIGURE 2.2. ROU with three variables of unknown order, where $\hat{n}_i$ is the output $n_i$ for each block.

Therefore, our ROU can model the relation between a variable $X_i$ and all its potential parents and the functions (denoted as $g$) in all the involved functions in Eq. 2.9 can be estimated by maximum likelihood:

$$\max_g -\text{Loss} = \mathbb{E}_{X\sim P_X}\left[\sum_{j=1}^{d} \log q(g_2^{-1}(g_1^{-1}(X)) + \sum_{i=1}^{2} \log \left|\det(\frac{\partial g_i}{\partial X})\right|\right]. \tag{2.10}$$

There are some examples that extend the framework of ANMs to obtain the DAG identifiability. For example, the linear Gaussian case with equal error variances, the linear non-Gaussian ANMs, and nonlinear Gaussian ANMs [3]. In this paper, we assume the nonlinear Gaussian cases and use neural networks to fit the function $f_j$ to make sure the requirement of nonlinear satisfied [24]. Therefore, the true graph should be the one with the minimum loss and if the neural networks with the minimum loss, we can have the Jacobian Matrix from the neural networks as the causal dependencies to form the true graph.

### 2.3.3 Jacobian Matrix as Causal Dependencies

Suppose $x_j := f_j(x_k)$ is a nonlinear function with first-order partial derivatives exist $\frac{\partial f_j}{\partial x_k}$ on $\mathbb{R}^d$, we can define a Jacobian matrix of neural networks over random variables $X$ as

$$J = \left[\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_d}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_d}{\partial x_1} & \cdots & \frac{\partial f_d}{\partial x_d} \end{bmatrix}. \tag{2.11}$$

With the transformation $f_i$ as $n_i = (x_i - \mu_i)\exp(-\alpha_i)$ in MADE, where $n_i$ is the noise variable, each element in the Jacobian matrix is given by

$$\frac{\partial f_i}{\partial x_j} = \begin{cases} \exp(-\alpha_i), & \text{if } i = j \\ -\frac{\partial \mu_i}{\partial x_j}\exp(-\alpha_i) + (x_i - \mu_i)(-\exp(-\alpha_i))\frac{\partial \alpha_i}{\partial x_j}, & \text{if } i > j \\ 0, & \text{otherwise} \end{cases}, \tag{2.12}$$

where $\mu_i = f_{\mu_i}(x_{1:i-1})$ and $\alpha_i = f_{\alpha_i}(x_{1:i-1})$ are nonlinear functions. With 2 blocks of MADE in ROU, we can multiply two Jacobian matrix of MADE as the new Jacobian matrix of ROU. Sparce-DAG [26] proposed if $||\frac{\partial f_j}{\partial x_k}||_{L^2} = 0$ then there is no dependency of $x_j$ on $x_k$, where $||\cdot||_{L^2}$ is the usual $L^2$ norm. In the experiments we found that the case of $||\frac{\partial f_j}{\partial x_k}||_{L^2} = 0$ is relatively rare, so generally we need to set a threshold to do eliminate the edges with small norms.

For example, the relationships of input data $X$ in SCM language is

$$\begin{aligned} X_1 &:= N_1, \\ X_2 &:= f_2(X_1) + N_2, \\ X_3 &:= f_3(X_1, X_2) + N_3. \end{aligned} \tag{2.13}$$

The order of input X is unknown. We assume the input order (not the causal order) of input X as ($X_2$,$X_1$,$X_3$), and the networks of ROU would be trained from the neural networks shown in Figure 2.2.

When we trained the neural networks to the maximum log-likelihood and the Jacobian matrix should be satisfied with that if $||\frac{\partial f_j}{\partial x_k}||_{L^2} = 0$ then there is no dependency of $x_j$ on $x_k$ as the true graph. The neural networks of ROU shown in Figure 2.3 satisfies the causal structure by checking the connectivity between output variables and input ones.

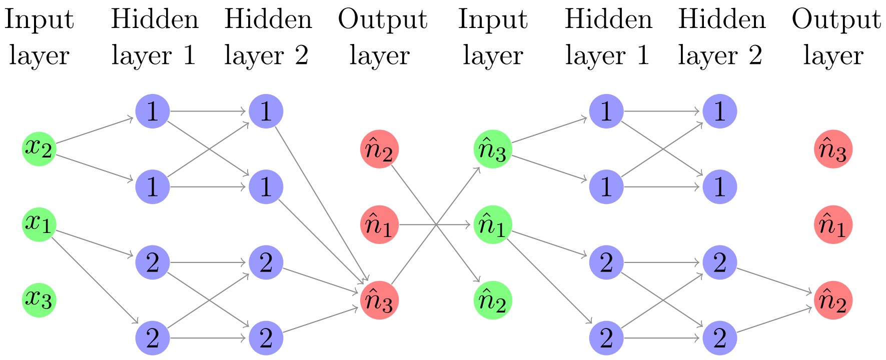


FIGURE 2.3. The neural networks of ROU with the maximum log-likelihood.

### 2.3.4 Adding Acylicity Constraint to Loss function with Augmented Lagrangian

The weighted adjacency matrix of nonlinear extension can be defined as $W(f) = ||J||_{L^2}$, the optimization problem is given by [23]

$$\begin{gathered} \min_f \text{Loss}, \\ \text{s.t. } h(W(f)) = 0, \end{gathered} \tag{2.14}$$

where Loss is the objective function in Eq. 2.10 with $g$ replaced by its inverse $f$ and $h(W(f))$ is the constraint and we can add the constraint to the objective function to form a Lagrangian function

$$L_c(\theta, \lambda) = f(\theta) + \lambda h(W(f)), \tag{2.15}$$

where $\lambda$ is the Lagrangian multiplier. The Lagrangian function is an unconstrained optimization problem and its solution is an optimal solution to the constrained problem (2.14). However, the Lagrangian function can not guarantee an optimal solution so we should add an augmented term to ensure the feasibility and optimal solution of the method, having the form of

$$L_c(\theta, \lambda) = f(\theta) + \lambda h(W(f)) + \frac{\rho}{2} \mid h(W(f)) \mid^2, \tag{2.16}$$

where $\rho$ is the penalty parameter. We follow the strategy of [23] to do optimization for the above Augmented Lagrangian function.

### 2.3.5 Interventions

We use $do(X = x)$ [4] to denote the intervention that causes the event x to occur. The following example shows how SCMs implements the intervention. The relationship between the variables $X_1$,$X_2$,$X_3$ can be represented by the following equations:

$$\begin{aligned} X_1 &:= N_1, \\ X_2 &:= f_2(X_1, N_2), \\ X_3 &:= f_3(X_2, N_3). \end{aligned} \tag{2.17}$$

When we preform the intervention $do(X_2 = x_2)$, we cut off all edges pointing to $X_2$ and assign $X_2$ to $x_2$. Thus, the new SCMs have a new collection of structural equations:

$$\begin{aligned} X_1 &:= N_1, \\ X_2 &:= x_2, \\ X_3 &:= f_3(X_2, N_3). \end{aligned} \tag{2.18}$$

In summary, a SCM estimates $P(X_j|do(X_i = x_i))$ by completing the intervention $do(X_i = x_i)$ on the original model and obtaining a new model. Subsequently, $P(X_j)$ can be estimated by the new model. In causal normalizing models, it is easy to do interventions and answer the questions. Take the above SCM for example, we can draw samples from the mutual independent distribution $P(N_1)$ and $P(N_3)$ and feed the samples into the flows, which help us to calculate the intervention distribution of $P(X_j|do(X_i = x_i))$.

### 2.3.6 Counterfactual Reasoning

Counterfactual inference tries to answer a question like: “What is the value of $Y$ if $X = x_1$ instead of $X = x_0$ in reality?". That is, we have a set of realistic observations $(x_0)$ to consider in the model and when all else being unchanged and we intervene to make an event happen $(X = x_1)$, what we will obtain? The requirement here is to update the noise distribution with the existing observations and to obtain the counterfactual distribution through the intervention. Consider the following simple example to explain [3]:

$$\begin{aligned} X &:= N_X, \\ Y &:= X^2 + N_Y, \\ Z &:= 2Y + X + N_Z. \end{aligned} \tag{2.19}$$

If the realistic observations is $(X, Y, Z) = (1, 2, 4)$ and the noise variable should be updated to $(N_X, N_Y, N_Z) = (1, 1, -1)$. Therefore, we can infer that $Z = 11$ when $X = 2$, which is a counterfactual statement. We can easily to do counterfactuals in causal normalizing models by transforming the observations to obtain SCMs with new noise distribution and doing interventions with the new SCMs.

## 2.4 Experiments

In this section, we evaluate the performance of our method CNF on both synthetic and real-world datasets.

### 2.4.1 Competing Methods and Evaluation Metrics

CNF is compared with five recent state-of-the-art methods, including Gradient-based Neural DAG Learning (GraN-DAG [24], Learning Sparse non-parametric DAGs (sparse-DAG [26]), NOTEARS [22], NOTEARS's nonlinear extension DAG-GNN [23], and causal additive models (CAM [46]).

In the implementation of our method, we combine 3 ROU with 1 hidden layer of 100 neurons and use ReLU as the activation function. The implementation of all contenders are taken from their authors, with their hyperparameters set as recommended in original papers.

Two widely-used performance metrics, true positive rate (TPR) and structural Hamming distance (SHD), are used to evaluate the effectiveness of the methods. Larger TPR (smaller SHD) indicates better performance.

### 2.4.2 Causal Discovery on Synthetic Data

Following [24, 25], we use Erdos-Renyi (ER) as the generation scheme of graph and ERx for x$d$ edges, and then generate synthetic data sets from SCMs $X_j = f_j(X_{pa(X_j)}) + z_j$ for all $j$ in topological order on the given graph. Function $f_j$ can be Gaussian Process with a unit bandwidth RBF kernel and independently sampled $\sigma_j^2$, MLP networks with mutually independent noise $\sigma_j^2$, or additive models with Gaussian Processes. In our experiments, we use Gaussian Processes with unit independent Gaussian noises.

#### Effectiveness

We evaluate our model on three synthetic datasets, including ER1 and ER4 for 10 nodes, and ER1 for 50 nodes. The SHD of ER4 for 50 nodes is too large to have meaningful comparison, so this dataset is omitted in our experiments. The comparison results are shown in Table 2.1. We can see that our model CNF performs the best at 10 nodes, while CAM leads ER1 in 50 nodes but CAM performs poorly in ER4 with 10 nodes. This indicates that CAM is not suitable for datasets with dense edges. We also tested 10 nodes with 45 edges, on which CAM performed even worse (the averaged SHD is 31.2 in the case of 5 samples), far worse than CNF (SHD averaged 23.5 in 5 samples) and GraN-DAG (SHD averaged 25.1 in 5 samples). The main reason may be that CNF uses an advanced normalizing flow model MAF to fit the data, which can stack batch normalizing layers to largely enhance the fitting ability of our model. NOTEARS and its nonlinear extension DAG-GNN fail to work effectively, because NOTEARS can only deal with linear causality. Although DAG-GNN improves NOTEARS by multiplying linear adjacency matrix with some nonlinear functions to gain better fitting capability, its causality modeling is largely limited by the use of the linear adjacency matrix. Sparse-DAG has a similar issue as DAG-GNN.

TABLE 2.1. Comparison of different methods on non-linear SCMs generated from Gaussian processes (GPs) with unit independent Gaussian noise.

| | ER1 with 10 nodes | | ER4 with 10 nodes | | ER1 with 50 nodes | |
|---|---|---|---|---|---|---|
| | SHD | TPR | SHD | TPR | SHD | TPR |
| **CNF** | **1.3±2.3** | **0.92±0.26** | **16.4±4.9** | **0.77±0.12** | 18.6±6.2 | 0.77±0.10 |
| GraN-DAG | 2.4±2.2 | 0.85±0.13 | 18.6±4.1 | 0.66±0.11 | 15.1±7.7 | 0.79±0.05 |
| Sparse-DAG | 3.6±2.7 | 0.82±0.22 | 20.1±6.7 | 0.63±0.10 | 20.9±5.9 | 0.73±0.06 |
| CAM | 5.1±2.1 | 0.90±0.06 | 20.8±1.6 | 0.61±0.08 | **5.3±1.8** | **0.95±0.01** |
| NOTEARS | 4.8±3.0 | 0.62±0.18 | 35.2±2.7 | 0.16±0.04 | 22.8±7.1 | 0.66±0.12 |
| DAG-GNN | 7.0±3.5 | 0.51±0.26 | 37.0±2.2 | 0.12±0.09 | 33.4±7.4 | 0.44±0.10 |

### 2.4.3 Causal Discovery on Real-world Data

Our model is evaluated a widely-used real-world benchmark dataset [29] with causal relations verified and accepted by the biological community. The dataset consists of 11 continuous variables corresponding to different proteins and phospholipids in cells of the human immune system of 7466 observations, each of which indicates the measured level of each biological molecule in a single cell under different experimental interventions.

While the ground truth of the consensus network is 17 edges, our model obtains a SHD of 14, with 9 edges predicted, in which there are 6 expected edges and 3 reversed edges. Particularly, the 6 true positives include Raf → Mek, Plcg → PIP2, PIP3 → PIP2, Erk → Akt, PKC → Mek, and PKC → P38. The 3 reversed edges include PKA → Raf, PKA → Erk, and PKA → Akt. There are 8 missing edges, which are Mek → Erk, Plcg → PIP3, PKA → Mek, PKA → P38, PKA → Jnk, PKC → Raf, PKC → PKA, and PKC → Jnk. By comparison, DAG-GNN obtains a SHD of 19, with 18 edges predicted, GraN-DAG gains a SHD of 13, with 16 edges predicted, Sparse-DAG obtains a SHD of 16, with 13 edges predicted.

### 2.4.4 Intervention and Counterfactual Reasoning

We generate a four-variable synthetic data for the intervention experiments, following the experimental design in CAREFL [85]. Specifically, the data is generated by using the following equation:

$$\begin{aligned} X_1 &:= N_1, \\ X_2 &:= N_2, \\ X_3 &:= X_1 + c_1 X_2^3 + N_3, \\ X_4 &:= c_2 X_1^2 - X_2 + N_4, \end{aligned} \tag{2.20}$$

where $N_i$ is sampled from standard Gaussian distributions, and $(c_1, c_2)$ are coefficients that is also drawn from standard Gaussian distributions. The experiment is to estimate expectations of $E(X_3|do(X_1 = \alpha))$ and $E(X_4|do(X_1 = \alpha))$ by doing the intervention $do(X_1 = \alpha)$ on the original structural equation models. For counterfactual reasoning experiments, we consider a set of realistic observations $(x_1, x_2, x_3, x_4) = (0.2, 0.15, 0.14, -0.1)$ and $(x_1, x_2, x_3, x_4) = (2, 1.5, 1.4, -1)$ and perform interventions with $x_1 = \alpha$ for $X_3$ and $X_4$, respectively.

Our model is compared with CAREFL and ANM models with respective linear regression (ANM-linear) and Gaussian Process (ANM-GP). The empirical result are shown in Figure 2.4. It is clear that our model CNF largely outperforms the competing methods in mean square error for intervention experiments on SCMs of $X_3$ and/or $X_4$, and achieves better intervention distributions of $X_3$ and $X_4$. As shown in Figure 2.5, our model also achieves comparably good counterfactual reasoning to the very recent state-of-the-art model CAREFL.

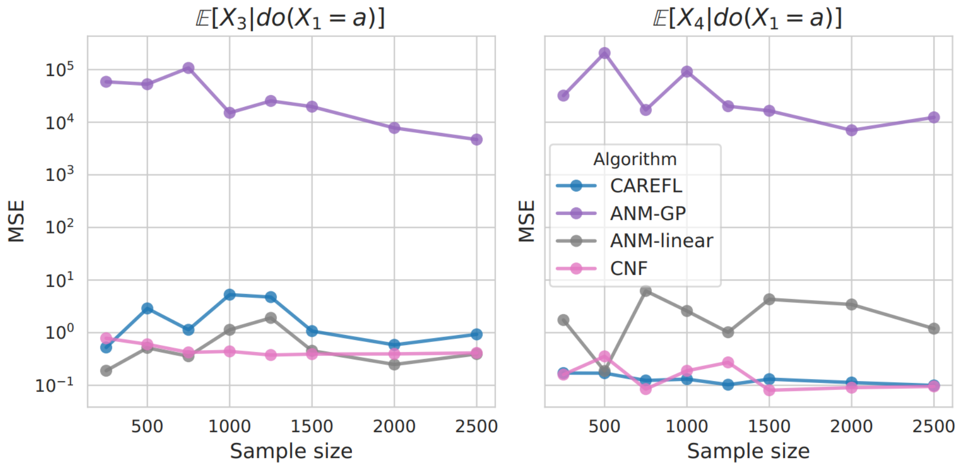


FIGURE 2.4. Mean square error for intervention experiments on SCMs of $X_3$ and $X_4$.

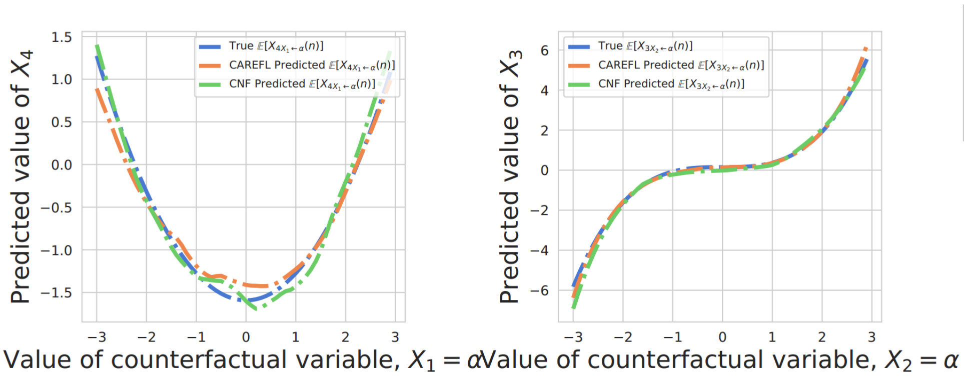


FIGURE 2.5. Predictions on SCMs of $X_3$ and $X_4$ under counterfactuals.

## 2.5 Related Work

Most existing methods of causal inference are constraint-based, score-based, and structure causal model-based methods. Score-based methods that we use in this paper define a score function such as Bayesian information criterion (BIC) scores [35] and Bayesian Dirichlet (BD) score [36] and try to optimal one with the highest score from a set of DAGs. Due to the huge super-exponential search space with a growing number of variables, a greedy search algorithm is introduced to solve this intractable problem. GES [38] uses BIC as a score function and try to find the local optimal graph from adding edges and removing edges phases.

For structure causal model-based methods, representative algorithms include Linear Non-Gaussian Acyclic Model (LiNGAM [17]), Post-NonLinear (PNL [41]), ANMs [43] and their extensions [44] and Information Geometric Causal Inference (IGCI [45]). LiNGAM [17] assumes the function $f_i$ is linear with non-Gaussian noises and acyclic dependency paths, which is based on Independent Component Analysis (ICA) and rely heavily on the initial solutions. In PNL [41], there are two non-linear functions in an assignment as the form of $X_j := f_{j2}(f_{j1}(x_{\pi_j^G}) + N_j)$. PNL has broad and general applicability but the two non-linear functions increased computational complexity. ANMs [43] describe a method for implementing the discovery of causality between two variables under nonlinear conditions and their successors [20] extend the ANM model to the case of multidimensional variables with the method of regression with subsequent independence test (RESIT), which is applicable to cases of the same variance error data or discrete data. IGCI [45] assumes that the causal influence process is noiseless and the derivatives of nonlinear functions between two variables are statistically uncorrelated. Therefore, ICGI-like methods focus primarily on no-noise or low-noise and complex functions cases. Causal Additive Model (CAM [46]) is a example of nonlinear Gaussian ANM, which satisfies the requirement of identifiability, having a form of $X_j := \sum_{k \in PA_j} f_{j,k}(X_k) + N_j$, where $PA_j$ denotes the parent nodes of $X_j$ in $G$. However, such assumptions of functions are too strong to generalize.

NOTEAR [22] reformulates the combinatorial optimization problem into a continuous problem with acyclicity constraint and significantly reduces the size of the search space of linear structural equation models. DAG-GNN [23] extends NOTEAR's continuous linear Structural Causal Model (SCM) to a non-linear model with VAE and Graph Convolutional Networks (GNN [83]), learning a neural network by maximizing an evidence lower band. However, DAG-GNN use an adjacency matrix and neural networks to represent the function $f_j$ but only use the weighted adjacency matrix as causal structure, which makes the method biased. To solve this issue, gradient-based methods are proposed by GraN-DAG [24], Masked Gradient-Based Causal Structure Learning (Masked-Grad [25]) Learning Sparse Non-parametric DAGs (Sparse-DAG [26]), determining the causal relationship between two variables through neural network connectivity. Gradient-based methods outperform DAG-GNN at all aspects of benchmarks in empirical comparisons, which are proven to be a good way to learn causal structure. However, GraN-DAG uses weights in neural networks to ensure connectivity with less generalizability compared to the Jacobian matrix which we use in this paper and the loss function does not include the divergence of distribution of input data and target data. Masked-Grad tries to learn a binary matrix instead of a continuous weighted matrix with the same framework of GraN-DAG. Sparse-DAG has the same issue with DAG-GNN, only ensuring the connectivity of the input layer and first hidden layer, however, its performance is competitive with fewer samples of data. Causal Autoregressive Flows (CAREFL) [85] use autoregressive flow to do causal modeling and interventions and counterfactuals.

## 2.6 Conclusion

In this chapter, we propose a score-based method with normalizing flows called CNF to learn causal dependencies of input observational data with the Jacobian analysis and do interventions and counterfactuals with the invertible nature of our model. We use the Jacobian matrix of output w.r.t. input as causal relationships and this method can be generalized to any neural networks especially flow-based generative neural networks such as Masked Autoregressive Flow (MAF) which compute the log-likelihood loss and divergence of distribution of input data and target distribution. This method also enforce the important acyclicity constraint of NOTEARS on the continuous adjacency matrix of graph nodes to reduce the computational complexity of the search space of the graph. We did massive experiments and the results show that our method outperforms the original NOTEARS, its nonlinear extension DAG-GNN, and other machine learning-based methods such as GraN-DAG. We argue that the Jacobian matrix is the key to causal structure learning and we will extend this approach to the physical system in form of the ordinary differential equation on static data and discrete time-series data in following works.

# Chapter 3

# Learning to Model Physical System for Static Data

## 3.1 Introduction

As a financial quantitative for years, I always have to make predictions that generally have an assumed premise. For example, in the financial markets, if the U.S. dollar depreciates, how it affects the price of crude oil, whether it goes up or down, and then we make investments based on that prediction. It is a common practice to obtain historical price observations of the dollar and crude oil to calculate a correlation coefficient. The correlation coefficient and the change in the dollar are then used to calculate the change in the price of crude oil. From a statistical point of view, the correlation coefficient is a statistical indicator of how closely two variables are correlated and generally reflects the degree of linear correlation. A change in one variable can be obtained through the correlation coefficient for another variable. In the field of machine learning, the technique of learning the relationship between variables from data, and then making predictions is very well established. However, we still need to be very cautious about using this technique in the financial field, as predictions based solely on correlations between data are not widely used stress tests in the financial market. For example, a typical stress test case would be that if the central bank adjusts the interest rate, how does it affect a stock market index. This is where we have to make calculations using human expertise in the financial market, rather than simply using correlations. Human expertise in a particular field is generally presented in the form of differential equations, which in the physical world can also be called physical systems. And the causal relationships between variables can be easily read out from the differential equations. In this work, we try to build differential equations by observational data given certain assumptions and constraints and then read out the causal relationships between variables from the physical systems [2].

Differential equations are widely used in various areas of modern science, such as the Black-Scholes option pricing model for the financial system, population development models, and traffic flow models for the social sciences, and especially in physics, where they are used extensively in electromagnetic fluid dynamics, chemical fluid dynamics, power meteorology, ocean dynamics, and groundwater dynamics. As an example, R.M.Anderson gives an ordinary differential equation model of infectious disease dynamics as [49],

$$\begin{aligned}\frac{dX(t)}{dt} &:= A - dX - \beta XY + \sigma Z,\\ \frac{dY(t)}{dt} &:= \beta XY - (\gamma + \alpha + d)Y,\\ \frac{dZ(t)}{dt} &:= \gamma Y - (\sigma + d)Z,\end{aligned} \tag{3.1}$$

Where $X(t), Y(t)$ and $Z(t)$ denote the number of susceptible, infections, and removed individuals respectively. And $A$ denotes constant immigration rate, $d$ is for constant natural death rate, $\beta$ represents transmission coefficient, $\alpha$ denotes disease-related death rate, $\gamma$ is for the recovery rate and $\sigma$ represents a loss of immunity rate. With differential equations, we can know the values of arbitrary variables in history, as well as predict future trends and changes in the system, or we can intervene with the system to get the desired results. At the same time, we can make inferences and give intuitive explanations, which is not possible with today's machine learning techniques. And, of course, we can easily read out causal relationships between variables.

Usually, an Ordinary Differential Equation(ODE) has the form of

$$\frac{d\mathbf{h}(t)}{dt} := f(t, \mathbf{h}(t)), \tag{3.2}$$

with some known initial value, for example, $\mathbf{h}(t = t_0) = \mathbf{h}_0$. If $f$ is Lipschitz, we can have a unique solution $\mathbf{h}(t)$ according to The Picard–Lindelöf theorem [50]. Equation 3.2 can also have the form as

$$\mathbf{h}(t + dt) := \mathbf{h}(t) + f(t, \mathbf{h}(t))dt, \tag{3.3}$$

where $dt$ is the terms of infinitesimal differentials of time $t$. If we can get the solution to the ordinary differential equation, we can know which variables in the system affect the state at the next point of time. These variables can then have any direct causal influence on the result in the future. However, it is almost impossible to obtain ordinary differential equations and their solutions from large amounts of observations and existing human expertise in a particular field, and random experiments and systematic interventions are generally required.

In the era of machine learning, Neural ODE [51] takes inspiration from the following iterative process of ResNet [52],

$$\mathbf{h}(t + 1) := \mathbf{h}(t) + f(\mathbf{h}(t)), \tag{3.4}$$

which is equivalent to the Euler iterative solution of a differential equation [53]. If we use more layers and smaller steps, it can be optimized to Equation 3.3. That is the basic idea of Neural ODEs and function $f$ can be trainable neural networks. we can easily read out causal relationships between variables as a form of the Jacobian matrix

$$J = \left[\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_d}\right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_d}{\partial x_1} & \cdots & \frac{\partial f_d}{\partial x_d} \end{bmatrix}. \tag{3.5}$$

**Contributions** The main contributions of this work can be summarized as follows,

- We extend causal modeling to the physical system which is "the most detailed model" [2] in causal modeling. This is the first research work trying to read out the causal dependencies from physical system using neural networks.

- By comparing extensive experiments with current state-of-art methods for learning causal structures, the method in this paper wins in datasets with denser causal relationships. It is shown that the method in this work is better suited to handle more complex causal relationships between nodes.

This work is ready to submit to IJCAI 2021.

## 3.2 From Statistical to Physical System

### 3.2.1 The Great Success of Statistical

Probability theory relies on a probability space$(\Omega, F, P)$ totaling a measure of 1 (P($\Omega$)=1). The first term $\Omega$ is a non-empty set, sometimes called the sample space. And the second term $F$ is a subset of the sample space $\Omega$ and $(\Omega, F)$ together is called the probability measure space. The third term $P$ is called the probability, or probability measure [3]. It is a function from the set $F$ to the real domain $R$. Each event is assigned a probability value between 0 and 1 by this function. For example, for the toss of a coin the sample space $\Omega$ is {head, tail}, $F$ is obtained from a random coin toss experiment, which may be $A = \{head\}$ or $B = \{tail\}$, and the corresponding probabilities P(A) = 0.5 and P(B) = 0.5. Probability theory allows us to infer the probability of the possible outcome of the next experiment from the data obtained from random experiments. In general, we need to learn from historical random experiments to get the probability space, through which we know the possible distributions of the data, and the distribution obtained by learning from historical data can naturally give us the probability of different results of the next experiment. For example, an independent random experiment has a set of observations, $(x_1, y_1) \cdots (x_n, y_n)$, where $x_n$ is the input data and $y_n$ is the output data. We assume that $(x_n, y_n)$ are from variables $X$ and $Y$ which are independent and identically distributed(i.i.d.) with the unknown joint distribution $P_{XY}$. Generally, existing machine learning and statistical methods follow the assumption that the data is i.i.d.. In machine learning, supervised learning is that we need to know $Y$ given a value of $X$ as the function $Y = f(x)$, or the probability of $Y$ given $X$ as $P(Y|X)$. Learning the decision function $Y = f(x)$ or the conditional probability distribution $P(Y|X)$ directly from the data is typically used as a model for prediction, which we consider to be discriminative models. Typical discriminative models include K-Nearest Neighbors(KNN [54]), MultiLayer Perceptron(MLP), Decision Tree, Logistic Regression, Maximum Entropy Models, Support Vector Machine(SVM), Conditional Random Fields(CRFs [55]), etc. Another method is learning a join distribution $P(XY)$ through observed data, and then finding $P(Y|X)$ with $P(Y|X) = P(XY)/P(X)$, which we called generative models. Typical generative models include the Hidden Markov Models(HMM), Mixed Gaussian models(MGMs), Averaged One-Dependence Estimators (AODE [56]), Latent Dirichlet Allocation(LDA [57]), and the Restricted Boltzmann Machine [58].

The great success of deep neural networks in statistical methods is generally considered to be due to [2]: (1) large amounts of data, especially precisely labeled data; (2) very powerful computational power, especially with the development of GPUs; (3) very complex and large computational systems with a large number of trainable parameters and (4) a closed static environment in which all data is assumed to be independent and identically distributed and the data distribution is constant.

And since the existing deep learning models rely on i.i.d. data obtained in a closed environment, the models are working for some tasks but not for others. For example, if we add some noise to an image, the model may not be able to accurately identify and classify it. The same is true in the field of reinforcement learning, where a model trained in one game is difficult to transfer to another game because the model relies on a closed training environment and the i.i.d. data generated from the environment. If the environment changes or if human intervention occurs, the model will fail. For example, a set of i.i.d. data A={rain, not rain} and B={floor is wet, floor is not wet}. The model can learn from this set of data and go on to predict B from A, or predict A from B. However, if human intervention occurs and someone pours water on the floor causing the floor to be wet, then the previous model must fail. The open environment and systems interventions are not the realm of statistical but they are the realm of causal inference.

### 3.2.2 Causal Graphical Models

We use the Directed Acyclic Graph with arrows pointing from the parent(direct cause) node to the child(direct effect) node as a formalism to represent causal relationships. These models are causal graphical models or graphical causal models that contain the observed data distribution and graph structure with nodes and arrows. We give its definition as follows.

**Definition of Causal Graphical Model** A Causal Graphical Model contains a Directed Acyclic Graph $G(V, E)$ where $V$ is for nodes or vertices representing variables $X = (X_1, X_2, \cdots, X_n)$ and $E$ is for edges between nodes and a set of probability density function $P(X_j|X_{PA_j^G})$ , such that the joint distribution $P(X)$ over $X$ equals the recursive product decomposition as follows [3],

$$P(X) = \prod_{d} P(X_j|X_{PA_j^G}), \tag{3.6}$$

where $X_{PA_j^G}$ is for the parent nodes in a DAG. This equation implies that variables $X_i$ is conditionally independent given the parent nodes of $X_i$. Causal Graphical Models can use do-calculus [84] to intervene in the system and have a new distribution but they can not answer counterfactual questions. Since this paper does not deal with interventions and counterfactuals, we skipped this part and if interested you can check out Peter's paper [3]. The problem with Causal Graphical Models is that it is hard to make stronger restrictions on Causal Graphical Models to ensure identifiability. For example, decomposing $P(AB)$ can get $P(AB) = P(A)P(B|A)$ or $P(AB) = P(A)P(B|A)$ and we can't make other assumptions here to get the correct DAG. Therefore, we have to introduce Structural Causal Models(SCMs), which can guarantee the identifiability after adding some restrictions on the functions.

### 3.2.3 Structural Causal Models(SCMs)

We give the definition of SCM as follows.

**Definition of Structural Causal Model** In a structural causal model over variables $X_1, X_2, \cdots, X_d$, there is a collection of $d$ equations(assignments):

$$X_j := f_j(X_{pa_j}, N_j) \qquad j = 1, ...d, \tag{3.7}$$

Where $X_{pa_j}$ denotes the set of parent nodes of $X_j$ and $N_j$ represents mutually independent noise usually are Gaussian noise with zero mean. For example, we can get a

SCM of rain and damp floors case we mentioned above as

$$\begin{aligned} A &:= N_1, \\ B &:= f_2(A) + N_2. \end{aligned} \tag{3.8}$$

An SCM is based on data generative assumptions, which allows the addition of rich assumptions about how the data are generated, and thus the causal structure of the data can be obtained based on function assumptions. SCMs are also the model basis for most current causal discovery methods. In a Causal Graphical Model, the decomposition of jointly distributed probabilities is difficult to distinguish between directions, such as $P(AB) = P(A)P(B|A)$or $P(AB) = P(B)P(A|B)$. It is also difficult to make assumptions over probabilities to ensure causal direction since conditional probability and some simple continuous probability distributions are invertible. It is also difficult to distinguish directions in structural learning if noisy variables are not introduced in SCMs. For example, two random variables $X$ and $Y$ with relationship as $Y = 2X + 1$ can be algebraically transformed to $X = (Y - 1)/2$ . This symmetry is unintuitive in a causal relationship since we cannot assume that it must be raining if the floor is wet, and we cannot assume that the air temperature has also changed by artificially adjusting the thermometer readings. In an SCM, we can also think of $X_{pa_j}$ as an endogenous variable, the noise variable $N_j$ as an exogenous variable for unconsidered environmental factors, and there is only one exogenous variable. Endogenous variables are dependent on other variables and there is at least one edge pointing to the node; exogenous variables are independent of other variables and there are no edges pointing to the node. At the same time, assignment function $f_j$ can be linear or nonlinear. In the era of deep learning, it is easy to fit complex nonlinear functions with neural networks. Therefore, as a broadly used modeling framework, SCMs can generate a wide variety of powerful models to simulate complex data.

However, given a distribution $P_X$ on $X(X_1, X_2, \cdots, X_d)$, we can get different SEMs to entail this distribution. In the previous example of two variables, $P(AB) = P(A)P(B|A)$ can get a SCM or $P(AB) = P(A)P(B|A)$ can get an another SCM but both point to $P(AB)$ at the same time. Therefore, we need additional information to help us get the right SCM, and this additional information would be the assumption of the data generation method $f_j$. We outline below several assumptions of $f_j$ to ensure identifiability results.

### Linear Non-Gaussian Acyclic Models

Linear Non-Gaussian Acyclic Models(LiNGAM [17]) require that the function $f_j$ in the assignment satisfy three conditions to ensure identifiability. First condition is that graph is a directed acyclic graph, in which the variables $X_1, X_2, \cdots, X_d$ have a sequential causal order and the preceding variables do not affect the following variables. Secondly, the model is linear which requiring the variables to be linear summations of the parent node variables in graph. The last condition is that the noise variables are non-Gaussian or there is only noise variable with Gaussian distribution. Further more, Noise variables are independent of other variables including noise variables. LiNGAM has the form of

$$X_j := \sum_{k \in Pa_j} \beta_{jk} X_k + N_j \qquad j = 1, ...d, \tag{3.9}$$

where all $N_j$ follow non-Gaussian distribution or only a $N_j$ is Gaussian distributed and all $\beta_{jk}$ are non-zero for all $k \in Pa_j$. Therefore, the SEM is identifiable from the joint distribution $P_X$.

### Linear Gaussian Models with Equal Error Variances

Linear Gaussian Models with Equal Error Variances(LGMEER [59]) require that the function $f_j$ in the assignment satisfies two conditions to ensure identifiability from the joint distribution over $X(X_1, X_2, \cdots, X_d)$: (1) the noise variables are Gaussian with variance $\sigma_2$ independent on $j$;(2) The model is linear which requiring the variables to be linear summations of the parent node variables in graph. LGMEER has the form of

$$X_j := \sum_{k \in Pa_j} \beta_{jk} X_k + N_j \qquad j = 1, ...d, \tag{3.10}$$

where all $\beta_{jk}$ are non-zero for all $k \in Pa_j$ and LGMEER is identifiable from the joint distribution $P_X$.

### Additive Noise Models(ANMs [3])

LiNGAM and LGMEER only solve the problem where the function is linear; in the nonlinear case, we generally assume an ANM which has the form of

$$X_j := f_j(X_{pa_j}) + N_j \qquad j = 1, ...d, \tag{3.11}$$

where $X_{pa_j}$ denotes the set of parent nodes of $X_j$ and $N_j$ represents mutually independent noise. An ANM with nonlinear assignments can ensure identifiable from the joint distribution $P_X$. If the assumption of Gaussian Noise $N_j$, then we have Nonlinear Gaussian Additive Noise Models which is also identifiable. If we have a stronger restriction on assignments $f_j$ with the form of

$$X_j := \sum_{k \in Pa_j} f_{jk}(X_k) + N_j \qquad j = 1, ...d, \tag{3.12}$$

where all $f_{jk}$ are three times identifiable and nonlinear, then the model is a Causal Additive Model(CAM [46]).

## 3.2.4 Physical systems or Ordinary Differential Equations

SCMs can also be formed as differential equations. Let us first consider the case of discrete time in linear mode. There is an SCM over variables $X(X_1, X_2, \cdots, X_d)$ having the following form,

$$X := WX + N,$$

where $W$ is $d \times d$ adjacency matrix and $N$ represents noise vector. If $X$ is a sequence of variables $X^t$ with a value at time $t$, then we have iteration assignment,

$$X^t := WX^{t-1} + N^{(t-1)}.$$

As the linearity of the assignment, we have the form of the case of continuous time as

$$\frac{dX(t)}{dt} := C$$

where $C$ is constant matrix and we can certainly read out the causal relationships from $C$. For a nonlinear case, a SCM can be replaced by differential equations as

$$\frac{dX(t)}{dt} := f(X),$$

or

$$X(t + \Delta t) := X_t + \Delta t \cdot f(X).$$

If we can get the solution to the ordinary differential equation, we can know which variables in the system affect the state at the next point of time. These variables can then have any direct causal influence on the result in the future and the causal relationships can be read out by Jacobian matrix of $f(X)$ on variable $X$. The various levels of causal modeling are summarized in Table 1.2 from Peter's paper. It is clear that the physical system is at top level and contains the most information. This chapter aims to do causal modeling at the highest level and determine the validity of modeling by reading out the causal structure via the Jacobian Matrix.

## 3.3 Neural ODEs for Causal Structure Learning

### 3.3.1 From ResNet to Neural ODE

Neural ODE [51] takes inspiration from the following iterative process of ResNet [52],

$$\mathbf{h}(t + 1) := \mathbf{h}(t) + f(\mathbf{h}(t)),$$

which is equivalent to the Euler iterative solution of a differential equation. If we use more layers and smaller steps, it can be optimized to Equation 3.3. That is the basic idea of Neural ODEs and function $f$ can be trainable neural networks. We need to solve the equation and obtain the function $h(t)$ and its arguments $\theta$, so we use the conventional methods of solving ordinary differential equations, which starts solving the problem from the initial state $h_0$. This problem is generally called the initial value problem(IVP). Conventional methods for obtaining numerical solutions to differential equations by integrating the time variable include simple Euler methods and higher-order variants of the Runge-Kutta method, such as RK2 and RK4. However, these methods require very small post-integration slices of the time variable, which is equivalent to having many layers of ResNet, which can lead to high Memory costs. That's not what the introduction of differential equations was about. For example, when using the Euler method to solve Equation 3.3, after K-step iterations we get

$$\begin{aligned} \mathbf{h}_1 &:= \mathbf{h}_0 + f(\mathbf{h}_0), \\ &\cdots \\ \mathbf{h}_k &:= \mathbf{h}_{k-1} + f(\mathbf{h}_{k-1}), \end{aligned} \tag{3.13}$$

which is similar to having $k$ blocks of ResNet. If $k$ is 1M, it would be ResNet with 1M layers and will cause memory issues. Neural ODE introduced the Adjoint method to solve the issues. The Adjoint method is the introduced second time backward ODE that keeps track of the gradient at time $t$ and then backpropagates with the gradient at time $t$. Since the gradient at any time can be obtained from the integral, the memory issues can be solved. For example, we have the following loss function evaluating from time $t_0$ to $t_1$ with parameters $\theta_t$,

$$L(h(t_1)) = L(\int_{t_0}^{t_1} f(h(t), t, \theta)dt) = L(ODESolve(h(t_0), f, t_0, t_1, \theta)). \tag{3.14}$$

We can compute the gradient of $L$ w.r.t. hidden state with infinitesimal change and define it as Adjoint state

$$a(t) = -\frac{\partial L}{\partial \mathrm{h}(t)}. \tag{3.15}$$

It's derivative on time $t$, which describes the dynamics of Adjoint state is given by,

$$\frac{da(t)}{dt} = -a(t)^T \frac{\partial f(t, \mathrm{h}(t), \theta_t)}{\partial \mathrm{h}(t)}. \tag{3.16}$$

It is also an ODE and its solution can also be written in integral form as follows,

$$a(t) = \int a(t)^T \frac{\partial f(t, \mathrm{h}(t), \theta_t)}{\partial \mathrm{h}(t)} dt. \tag{3.17}$$

Numerical solutions at different time $t$ can be obtained by an ODE solver. The gradient at any time $t$ can be obtained by invoking the ODE solver backward in time from the initial point which is the gradient at time $t_1$(the gradient of the loss function on the output layer and it is easy to compute), e.g. the gradient at time $t_0$ can be solved as follows,

$$a(t_0) = \int_{t_1}^{t_0} -a(t)^T \frac{\partial f(t, \mathrm{h}(t), \theta_t)}{\partial \mathrm{h}(t)} dt. \tag{3.18}$$

Similarly, we can compute the gradient of loss function w.r.t. parameters $\theta$,

$$\frac{dL}{d\theta} = \int_{t_1}^{t_0} -a(t)^T \frac{\partial f(t, \mathrm{h}(t), \theta_t)}{\partial \theta} dt. \tag{3.19}$$

It can also be solved by an ODE solver and all three integrals can be solved with an ODE solver by vectorizing the problem.

### 3.3.2 Continuous Normalizing Flow and SEMs

We assume the assignments of SCMs are ANMs. Therefore, we can train a model which transform $N_j$ from simple distribution to input data $X$ as

$$\begin{aligned} Z_j^{(0)} &:= N_j, \\ Z_j^{(t)} &:= X_j^{(t)} - f_j(X_{\pi_j^G}^{(t)}), \\ Z_j^{(1)} &:= X_j, \end{aligned} \tag{3.20}$$

where $t$ is state variable in model, which can be $t$ hidden layer in neural networks or $t$ block in normalizing flows. We also can have residual form of equation (3.20) as follows,

$$\begin{aligned} Z_j^{(0)} &:= N_j, \\ Z_j^{(t+1)} &:= Z_j^{(t)} + g_j(Z_j^{(t)}), \quad \text{where } g_j(Z_j^{(t)}) = \Delta \left[ X_j^{(t)} - f_j(X_{\pi_j^G}^{(t)}) \right], \\ Z_j^{(1)} &:= X_j. \end{aligned} \tag{3.21}$$

If we continuously add more blocks or layers to a limit and we can have the continuous dynamics of $Z_j^{(t)}$ with an ordinary differential equation(ODE) [51] parameterized by $\theta$,

$$\frac{dZ(t)}{dt} = f(Z(t), t, \theta). \tag{3.22}$$

The equation (3.22) can be solved by a black box of ODE solver and this continuous dynamics models called Continuous Normalizing Flows(CNF [51]). The change of log density is also a differential equation named Instantaneous Change of Variables [51],

$$\frac{\partial \log p(Z(t))}{\partial t} = -\mathrm{Tr}\left(\frac{df}{dZ(t)}\right). \tag{3.23}$$

Therefore, the change from $Z(0)$ to $Z(1)$ can be computed by,

$$\log p(Z(t_1)) = \log p(Z(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{df}{dZ(t)}\right) dt, \tag{3.24}$$

which is the log function we try to maximize. We can solve the integral with an ODE solver and backpropagate the solution with the Adjoint Method( [60])

## 3.4 Acyclicity Constraint and Jacobian Matrix

### 3.4.1 Linear Case: NOTEAR's Acyclicity Constraint

We consider a linear case of SCM in NOTEAR [22], which has the form of $f_j(X) = W_j^T(X)$. We define $W = [W_1|W_2|\cdots|W_d] \in \mathbb{R}^{d\times d}$ as the coefficient matrix which encodes a graph. When $W_{ij} = 0$ then there is no edges from node $i$ to node $j$, when $W_{ij} \neq 0$ there exists an edge pointing from node $i$ to node $j$ in the graph. NOTEAR proposed that if the graph is directed acyclic, then the following condition should to be satisfied,

$$h(W) = \mathrm{Tr}(e^{W\circ W}) - d = 0, \tag{3.25}$$

where $\circ$ is the Hadamard product, Tr is the trace function of matrix and $e^M = \sum_{k=0}^{\mathrm{inf}} \frac{M^k}{k!}$. Let us see why this constraint can express the condition of a directed acyclicity. If the element $(i, j)$ in the $k$-th power of a non-negative adjacency matrix A $(A^k)_{ij} > 0$ , then there exists a path of length $k$ between node $i$ and node $j$. If the element (i, i) in the $k$-th power is greater than 0, then there exists a cycle in the graph. The zero power has a value of 1, then the exponential power of matrix A must be $d$ which is the dimension of data to ensure that the graph is a DAG. Also to ensure non-negativity, the Hadamard product can be used. And it is easy to calculate the gradient of $h(W)$ by the following equation,

$$\nabla h(W) = (e^{W\circ W})^T \circ 2W. \tag{3.26}$$

Meanwhile, we can use the equation as follows to simplify the calculation,

$$h(W) = \mathrm{Tr}[(I + \alpha W \circ W)^d] - d = 0, \tag{3.27}$$

where $\alpha$ can be any value greater than 0 and the gradient computation can be done by deep learning framworks such as Pytorch's Autograd rather than being written manually in code implementation.

### 3.4.2 Non-Linear Case: the Jacobian Matrix and Acyclicity Constraint

However, In nonlinear SCM cases, we cannot find a linear $W$ but we can use partial derivatives to represent the causal dependency of $f_j$ on the $k$-th variable. We define the partial derivatives of of $f_j$ on the $k$th variable by $\partial_k f_j$ and there exits an edge from the node $j$ to the node $k$ if and only if $\partial_k f_j \neq 0$. Therefore, the Jacobian matrix $J$ represents causal dependencies between input variables $X_1, X_2, \cdots, X_n$ and $h(W)$ in nonlinear SCM cases is,

$$h(J) = \mathrm{Tr}(e^{J \circ J}) - d = 0 \tag{3.28}$$

It's also easy to get that $J$ equals $W$ in linear cases, so it can also be argued that $W$ is only a special case of $J$.

### 3.4.3 Augmented Lagrangian Optimization

And now, the maximum likelihood optimization problems we need to solve is

$$\log p(Z(t_1)) = \log p(Z(t_0)) - \int_{t_0}^{t_1} \mathrm{Tr}\left(\frac{df}{dZ(t)}\right) dt \;\; \text{s.t.} \;\; h(J) = 0. \tag{3.29}$$

We can use the Augmented Lagrangian method to solve this optimization problem. The Augmented Lagrangian method adds a quadratic penalty to the Lagrangian method so that the converted problem can be solved more easily. Therefore, the maximum likelihood optimization problem can be transformed with the Augmented Lagrangian method as [23],

$$L(J, \theta, \lambda) = \log p(Z(t_1) \mid \theta, J) - \frac{\rho}{2}|h(J)|^2 - \lambda h(J), \tag{3.30}$$

where $\rho$ and $\lambda$ are quadratic penalty coefficient and Lagrangian multiplier respectively. When $\rho$ is sufficiently large, $J_*$ and $\theta_*$ are the minimum points of the loss function, and the parameters obtained must satisfy $h(J) = 0$. Therefore, we incrementally increase the value of $\rho$ and then optimize the entire neural network under this condition, while updating the Lagrange multiplier $\lambda$ accordingly to make it converge to the optimal points.

## 3.5 Related Work

Traditionally, there are three main families of methods for causal structure learning, namely, constraint-based methods, score-based methods, and structural causal function model-based methods. Constraint-based methods use a conditional independence test between variables to determine a particular structure and then determine the direction based on a particular V-structure [3]. The score-based approach uses a score function to search for the optimal network structure and is the basis of the methodology of this paper. The structural causal model-based approach is based on the structural causal model of the data generating mechanism and extends the structural causal model to increase the expressive power to discover the causal relationship between variables.

### 3.5.1 Constraint-based Methods

Constraint-based methods are used to learn a set of causal networks that satisfy the conditional independence between variables in data. We use statistical test methods to verify that candidate causal networks satisfy the Causal Faithfulness Assumption.

**Definition of Causal Faithfulness Assumption [3]** If variables $X_i$ and $X_j$ are independent of each other or conditionally independent given a set of variables $Z$, then all paths between variables $X_i$ and $X_j$ are $d$-separated by the set of variables $Z$ in the causal graph $G$ that defines the process by which data $X$ is generated. Then the joint distribution $P_X$ over random variables $X$ is Causal Faithfulness to the graph $G$.

There are three steps in this family of algorithms, the skeleton learning stage, direction learning stag, and possible orientation stage. In the learning phase of the skeleton graph, a skeleton graph without orientations is obtained by the independence of the variables with independence tests or conditional independence tests technologies. Commonly used tests for conditional independence are the statistical analysis-based chi-square test or the information theory-based mutual information test. In the direction learning phase, the direction is determined based on a specific V-structure. In the possible orientation stage, we use three rules to orient undirected edges as many as possible. The main problem with this family of methods is that the number of conditional independent tests grows exponentially as the number of nodes increases, and the computational cost is very high. So the main research direction of such algorithms is to reduce the number of tests.

We briefly introduce the Peter Clark(PC [61]) algorithm here. At the first stage, the skeleton of DAG with undirected edges is estimated. We start with a completed connected graph with no oriented edges and search depth equals 0(depth=0 means the neighbor nodes of test nodes). For each pair of nodes $X_i$ and $X_j$, test one by one that given neighbor node $X_k$ of the two in the graph, whether these two nodes are conditionally independent. If yes, then remove the edge of these two nodes $X_i$ and $X_j$ and add neighbor node $X_k$ to the set of $d$-separated $S_{ij}$. When all edges are removed with depth=0, increase the depth to 1 and repeat this process until the number of neighbors of the node is less than the depth. In the second stage of PC algorithm, For each pair of unconnected nodes $X_i$ and $X_j$ with a common connected neighbour $X_k$, if $X_k$ is not in $d$-separate set $S_{ij}$ then the undirected V-Structure $X_i - X_k - X_j$ is orientated to $X_i \rightarrow X_k \leftarrow X_j$. Otherwise $X_k$ is not a collider of the V-Structure. In the third stage, we continue to check if there is new edges can be oriented with three rules avoiding new V-Structure discovered and new cycles(the graph is acyclic): (1) we point from $X_i$ to $X_j$ if $X_k$ pointing to $X_j$ and $X_i$ is not the neighbour node of $X_i$; (2) we point from $X_i$ to $X_j$ if there exists a chain $X_i \rightarrow X_k \rightarrow X_j$; (3) we point from $X_i$ to $X_j$ if $X_i - X_k \rightarrow X_j$ and $X_i - X_l \rightarrow X_j$.

The Inductive Causation(IC [62]) algorithm and its variants [63] are similar to the PC algorithm in which they also use three stages to learn the causal network structure. However, most independence tests are chi-square test or partial correlation tests based on Gaussian distribution or multinomial distribution. To overcome these limitations, many effective methods have been proposed to handle more complex data distributions. For example, using Kernel-based Hilbert-Schmidt Norms and Kernel-based conditional independence test for more complex distributed data. Furthermore, when Causal Faithfulness Assumption is violated, there may be unobservable confounding factors. The FCI(Fast Causal Inference [64]) algorithm and FCI improved RFCI (Really Fast Causal Inference [65]) algorithm are proposed to the discovery of causality with hidden variables through extended graphs.

Constraint-based methods are effective for discovering causality and can be widely used with given reliable conditional independence tests. However, it is not possible to determine the direction of all edges through conditional independence tests and V-structures. Therefore, we need other types of methods to do causal learning.

### 3.5.2 Scored-based Methods

Score-based methods are an alternative to learning causal structures. A score-based approach uses a scoring function to quantify how well a Bayesian network fits a given distribution of data and then uses a search algorithm to find the graph structure that best fits the data. In this approach, the choice of the scoring function is crucial, the scoring function maps the candidate causal graph to a certain scalar based on a given structure. Bayesian Information Criterion(BIC [66]) is commonly and widely used one and its formula is $BIC(X, G) = k \ln(n) - 2\ln(L)$, where $L$ is the maximized value of the likelihood function of given graph $G$ and n is the number of the samples and $k$ denotes the number of the variables. However, BIC failed to do feature selection in high-dimension data. Another popular one of the Bayesian score function is the Bayesian Dirichlet equivalent uniform (BDeu [67]) score which has the form of

$$S(X, G) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha/q_i)}{\Gamma(\alpha/q_i + \sum_{k=1}^{r_i} s_{ijk})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha/q_i + s_{ijk})}{\Gamma(\alpha/q_i)},$$

where $r_i$ is the number of stats of $X_i$; $q_i$ indicate the number of configurations of the parents of $X_i$; $s_{ijk}$ denotes the number of observation data that $X_i$ is its $k$-th value and the parents of $X_i$ took the $j$-th sample.

However, the number of candidature graph structures grows exponentially as the number of variables increases, and the problem becomes NP-hard due to the large search space. Therefore, heuristic search algorithms such as Greedy Equivalence Search(GES [14]) and its extension Fast GES(FGES [16]) are often used to find a locally optimal graph. In the GES algorithm, there are two stages, a forward phase where edges are added and a backward phase where edges are removed. In the forward phase, edges are added in a greedy manner (i.e., maximizing the score which is calculated by a score function defined by GES) until the score can not be further increased. In the second phase, the edges are greedily removed until the score is optimal. GES can search the graph space in a very efficient way because it includes a greedy algorithm. However, the scoring process of the algorithm is too redundant, and adding edges causes the number of scoring to increase exponentially. It means that adding edges can make the time complexity grow exponentially and it becomes impractical as the number of variables increases. The FEGS algorithm improves the GES algorithm by decreasing the computational complexity when adding a new edge. Moreover, FEGS parallelizes special steps and does not depend on the order of operations, which makes the scoring processes much faster than the GES algorithm.

The hybrid approach combines Scored-based Methods and Constraint-based methods to overcome their respective drawbacks by using conditional independence tests to reduce the complexity of the candidate graph search space, followed by a scoring-based approach to find the best network structure. For example, the Max-Min Hill-Climbing(MMHC [68]) algorithm first learns a skeleton of a graph by the Max-Min Parents and Children(MMPC [69]) algorithm, which is equivalent to a constraint-based approach, followed by a greedy Bayesian score climbing search method to orient the graphs. This approach is not only suitable for high-dimensional data, but also improves the effectiveness of learning causal structures.

### 3.5.3 Structural Causal Function Model-based Methods

Constraint-based methods have Markov equivalence class problems and cannot orient all edges while score-based methods are not efficient due to the large search space, therefore, many studies have proposed structural causal models from the perspective of data generation or causal mechanisms between the variables of data. The general form of the structural causal model has the form of $X_j := f_j(X_{pa_j}, N_j)$ Where $X_{pa_j}$ is for the set of parent nodes of $X_j$ and $N_j$ is for mutually independent noise. The structural causal model describes the mechanism for generating data between variables rather than an algebraic equation describing the equality of the left and right sides. However, different SEMs to entail a same distribution $P_X$ on $X(X_1, X_2, \cdots, X_d)$. Therefore, more information such as a stronger assumption of the data generation method $f_j$ should be provided. These algorithms with stronger assumptions include Linear Non-Gaussian Acyclic Model(LiNGAM), Post-NonLinear(PNL), Additive Noise Model(ANM) in non-linear cases and its extensions, Information-Geometric Causal Inference(IGCI), and hybrid algorithms combining Constraint-based methods and Structural Causal Function Model-based Methods. The detail of LiNGAM, ANM, LGMEER can be seen in subsection of 3.2.3.

## 3.6 Experiments

In this section, we experimentally verify whether it is possible to derive dynamic physical systems from observed data via Neural ODEs and then read the causal structure from the physical systems. For ODE Sovler, we used the Pytorch implementation from the public Github repository of Nueral ODEs. Deriving dynamic physical systems with SCMs from static data has not been done before, so it is not possible to compare with previous works. But previous works of learning the causal structure between variables allows us to get some benchmarks. The aim of this chapter is to verify the idea of obtaining a dynamic physical system from observed data and then reading out the causal structure of the variables in the data from the physical system. However, for learning the causal structure of the variables, this approach outperforms previous works to learn the causal structure in some datasets.

**Baselines** we choose the following algorithms as baselines for comparison: two gradient-based methods GraN-DAG [24] and Sparse-DAG [26] using weights in neural networks as causal dependencies; CAM [46] for non-linear additive structural causal models based method; NOTEARS for linear structural causal models and its non-linear extension DAG-GNN [23]. Other algorithms such as PC, GES, and FGS have been shown to be poor performance in multiple experiments [24, 26, 23], so we omitted.

**Metrics** we choose the following metrics to evaluate the causal structure learned observed data: True Positive Rate(TPR) and the structural hamming distance. The former is the number of correctly identified oriented edges divided by the total number of oriented edges in true DAG and the latter counts the number of falsely adding, deleting, and orienting edges.

### 3.6.1 Synthetic Data

In the synthetic data experiments, we used Erdös–Rényi(ER) as the graph type to generate random graphs $G$ and generated data from the random graphs $G$ in which the causal order defined. We generated datasets $X_1, X_2, \cdots, X_d$ with $d = 10$ and $1d$ and $4d$ edges denoted by $ER1$ and $ER4$ respectively. The data generating process we choose is Non-linear Gaussian ANMs with the form of $X_j := f_j(X_{pa_j}) + N_j \;\; j = 1, ...d$, Where $X_{pa_j}$ is for the set of parent nodes of $X_j$ and $N_j$ is for mutually independent unit Gaussian noise and $f_j$ we used is Gaussian Process(GP) with a unit bandwidth RBF kernel. Due to the non-linear assignment of $f_j$ and Gaussian noise, the DAG is identifiable from the distribution $P_X$ over data $X$. The results of comparisons among different methods are shown in Table 3.1, in which we can see that our proposal method DAG-ODE outperforms other algorithms in all aspects in the dataset of 10 nodes.

Table 3.1. Comparison of different methods on non-linear SCMs generated from Gaussian processes(GPs) with unit independent Gaussian noise. The lower the better for SHD and the higher the better for TPR. Our method is DAG-ODE.

| | ER1 with 10 nodes | | ER4 with 10 nodes | |
|---|---|---|---|---|
| | SHD | TPR | SHD | TPR |
| **DAG-ODE** | **2.3±1.9** | **0.86±0.22** | **10.2±3.5** | **0.86±0.15** |
| GraN-DAG | 2.4±2.2 | 0.85±0.13 | 18.6±4.1 | 0.66±0.11 |
| Sparse-DAG | 3.6±2.7 | 0.82±0.22 | 20.1±6.7 | 0.63±0.10 |
| CAM | 5.1±2.1 | 0.90±0.06 | 20.8±1.6 | 0.61±0.08 |
| NOTEARS | 4.8±3.0 | 0.62±0.18 | 35.2±2.7 | 0.16±0.04 |
| DAG-GNN | 7.0±3.5 | 0.51±0.26 | 37.0±2.2 | 0.12±0.09 |

### 3.6.2 Real Data

We evaluate the real dataset that is generally accepted by the biological community and is often used as a benchmark. The data consists of 11 continuous variables corresponding to different proteins and phospholipids in cells of the human immune system with 7466 observations, each of which indicates the measured level of each biological molecule in a single cell under different experimental interventions [29]. The ground truth of consensus network as a causal graph is shown in Figure 3.2.

While the ground truth of the consensus network is 17 edges, we report SHD of 13 estimated 4 edges which are all expected edges as shown in Figure 3.1. For detail, the 4 true positives are Raf → Mek, Plcg → PIP2, PIP3 → PIP2, Erk → Akt. By comparison, while DAG-GNN reports SHD of 19 with 18 edges predicted, GraN-DAG estimated 16 edges with SHD of 13 and Sparce-DAG predicted 13 edges with SHD of 16.

## 3.7 Summary

In this chapter, we extend the Jacobian-based to the physical system which is the method humans explore and reason the world and it is "the most detailed model" [2] of causal learning. By functions fitting with Neural ODE, we can read out the causal structure from functions. Our approach also enforces an important acyclicity constraint on the continuous adjacency matrix of graph nodes and significantly reduces the computational complexity of the search space of graphs. For the task of structure learning, our method outperforms other current state-of-art methods for learning causal structures in experiments of datasets of 10 nodes and improves the performance in datasets with more dense causal relationships. In the next chapter, we will apply our Jacobian-based approach to time series data which is non-IID.

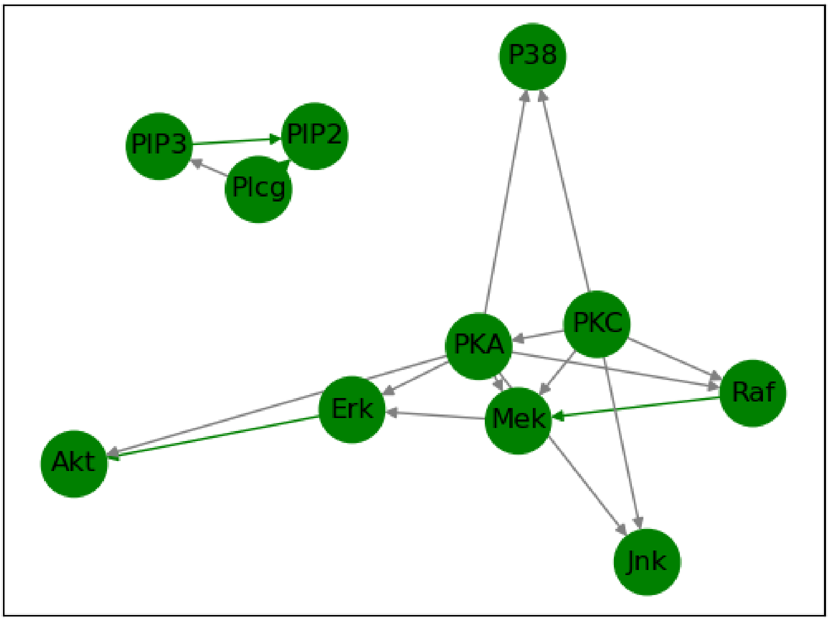


Figure 3.1. The causal graph of Sachs dataset estimated by our methods, in which the gray arrows represent missing edges from the ground truth.

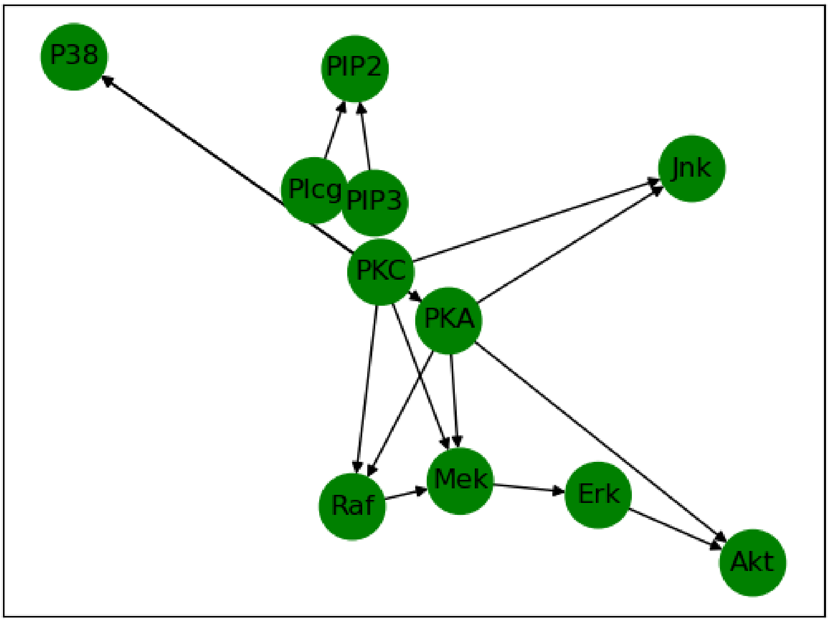


Figure 3.2. The ground truth causal graph of Sachs dataset

# Chapter 4

# Causal Structure Learning for Time Series Data

## 4.1 Introduction

Time series data are widely available in the natural science and social science fields and causal structure learning of time-series data is important for these areas. In financial markets, for example, as a result of technological advances and globalization, financial links between countries are becoming more and more strong and form a complex system. Therefore, we need to analyze the immediate and ongoing effects within individual financial markets as well as the temporal impacts between multiple financial markets. For example, the impact of changes in the Federal Reserve's financial policy on US financial markets and on other countries can last for a long time. The various indicators of daily economic life, such as GDP, CPI, electricity consumption, weather indicators of temperature, humidity are all existing in the form of time series. Compared to static data, if we can dig out the unknown and valuable structures and mechanisms behind the time series data in time dynamics and then predict or intervene in the causal direction, it will be significant for scientific research, commercial marketing, engineering production, and other aspects. Current correlation and statistical methods based on machine learning in CV, NLP, and other fields have achieved great success, especially in the accuracy of prediction and some may have exceeded the human capabilities [1]. However, it is difficult for these models to answer questions about cause and effect. In financial markets, for example, what is at work in the current U.S. stock booming during the pandemic of Covid-19, is it quantitative easing policy by the Federal Reserve? So where do the markets go if the Fed doesn't implement monetary easing? In terms of time, how long will the impact of this policy be, a year or a month? Correlation helps machine learning algorithms to do predictive tasks but does not answer the above questions due to a lack of direction in dependencies.

Most machine learning algorithms including deep learning are based on correlation, which is encoded and learned to improve accuracy in prediction [2]. However, correlation only shows that there is a relationship between the variables and does not give information about the dependencies in directions. For example, two variables may have a common causal variable, then the two variables are correlated but do not have a causal dependency. Such non-causal relationship models are less explainable and weak robust. When the value of variables interference by other factors outside of the environment, such as human intervention, the model will not get the expected results or break down. Most scientific research requires learning causality rather than correlation between variables. The natural sciences, for example, need to know the direct causes of global warming, the interactions of cells and viruses, and the effects of policy on climate change and in finance and we need to know the direct effects of policy on markets. Causality ensures that the model is more interpretable and sufficiently robust. At the same time, a sufficiently robust model based on causality can also answer and solve the problem of external intervention. Since most of the data in machine learning algorithms learned comes from a closed environment, if the closed environment is cracked and the variation in the variables is likely to be due to external interventions, the predictive models encoding correlations between variables in the closed environment data will collapse. Causal models can predict the effect of an externally intervening variable on other variables because they understand the mechanism by which the variable takes its value (usually represented by the Structure Causal Models).

Traditionally, causal structures can be obtained by the intervention that changes the values of variables and sees their impacts on other variables in experiments or by randomized experiments. However, due to the limitations of the experimental setting, we can not do human interventions or the cost of the interventions is huge. For example, we cannot allow patients to use drugs that have not been clinically validated, which is illegal and unethical. We also can't arbitrarily change financial policy to see how it affects the market, because the costs of doing so are enormous. So most experiments can only get passive observations and learn causal relationships between variables relying on observed data. Observed data can generally be divided into two categories: static data that are non-time-sequence and time-series data. For learning structure from static data, there are three main families of methods, namely, constraint-based methods, score-based methods, and structural causal function model-based methods. Causal structure learning based on temporal data is simpler than causal reasoning on static data because temporal data itself contains information about the causal direction in the time dimension, in which the effect factors cannot occur ahead of the cause factors [3]. However, there are also three main issues with causal learning based on temporal sequential data. Firstly, the set of variables is not causally sufficient. Secondly, there are instantaneous effects on which variables are not the time ordered a priori. Lastly, it is often only possible to obtain repeated observations at different times. Generally, the causal learning problem based on temporal data can be defined as follows. Given a time series data $X_t^d$ with $d$-variate and time length $T$, where at time $t$ in time length $T$ we have the vector $(X_t^1, X_t^2, \cdot, X_t^d)$. $X_t^j$ represents the $j$-th variable at time $t$ and we aim to learn the causal structure between all the variables $X_t^j$ where $j \in (0, d)$ and $t \in (0, T)$. Depending on whether there is an instantaneous effect or not, causal structure learning problems based on temporal data are divided into two types, which are with instantaneous effects and without instantaneous effects. In the causal graph without instantaneous effects, there is no arrows from $X_t^j$ to $X_t^k$ where $j, k \in (1, d)$ and $X_t$ nodes only impact other nodes $X_s$ in the future where $s > t$, as shown in Figure 4.1. For causal graph with instantaneous effects(see Figure 4.2), arrows not only point from nodes $X_t^j$ to other nodes $X_t^k$ at the same time $t$ but also go to the future nodes $X_s$, where $s > t$.

We define the temporal directed causal graph as $G = (V, E, D, L)$ over observed time series data $(X^1, X^2, \cdots, X^d)$, where $V_i$ is vertex for the time series $(X^i, i \in (0, d))$ and an edge $e_{ij}$ pointing from vertex $V_i$ to $V_j$ represents a causal relationship that $V_i$ has an effect on $V_j$, and $D_{ij}$ and $L_{ij}$ annotating the edge $e_{ij}$ denote the time delay between occurrence of $V_i$ to occurrence of $V_j$ and the time lag $l$(it is also called time order in auto-regression models) between $V_i$ and $V_i$ since $V_{t+1}^j, V_{t+2}^j, \cdots, V_{t+l}^j$ have impacts on $V_j$, respectively. For example, the Vector AutoRegressive models VAR(2) with order 2 has the following form,

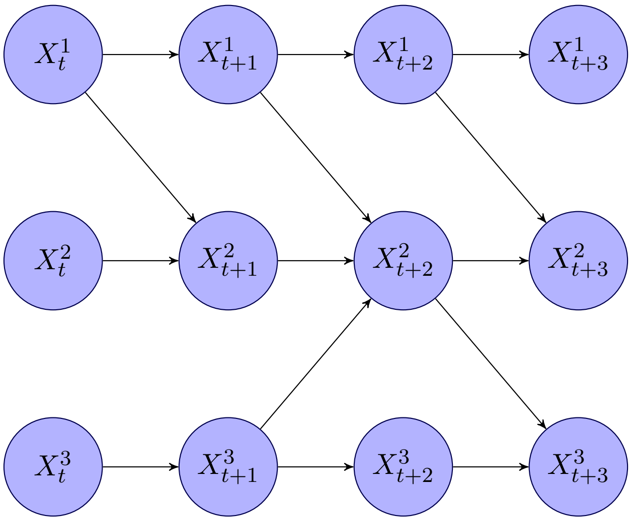


FIGURE 4.1. A Temporal Structure Example without Instantaneous Effects

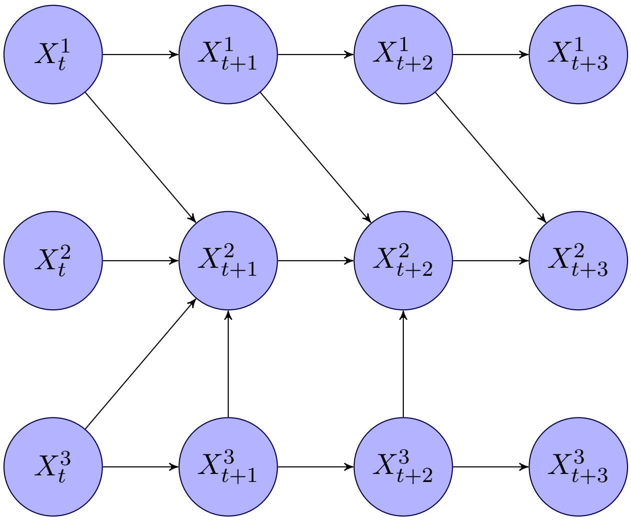


FIGURE 4.2. A Temporal Structure Example with Instantaneous Effects

$$
\begin{aligned}
X_t^1 &= \alpha_1 + \phi_{11} X_{t-1}^1 + \phi_{12} X_{t-1}^2 + \phi_{13} X_{t-1}^3 + N_t^1, \\
X_t^2 &= \alpha_2 + \phi_{21} X_{t-1}^2 + \phi_{22} X_{t-1}^3 + \phi_{23} X_{t-2}^3 + N_t^2, \\
X_t^3 &= \alpha_3 + \phi_{31} X_{t-1}^3 + \phi_{32} X_{t-2}^3 + N_t^3,
\end{aligned}
\tag{4.1}
$$

where $\phi_{ij}$ is the coefficient of cause variables and $N_t^j$ is mutually independent noise, and we have the temporal directed causal graph of the VAR(2) as shown in Figure 4.3

In this work, we assume that the following structural causal models [3] can describe the time series,

$$X_t^j := f^j((PA_q^j)_{t-l-d}), \cdots, (PA_q^j)_{t-d-1}, (PA_q^j)_{t-d}), N_t^j), \tag{4.2}$$

where

$$\cdots, N_{t-1}^1, \cdots, N_{t-1}^d, N_t^1, \cdots, N_t^d, N_{t+1}^1, \cdots, N_{t+1}^d, \cdots$$

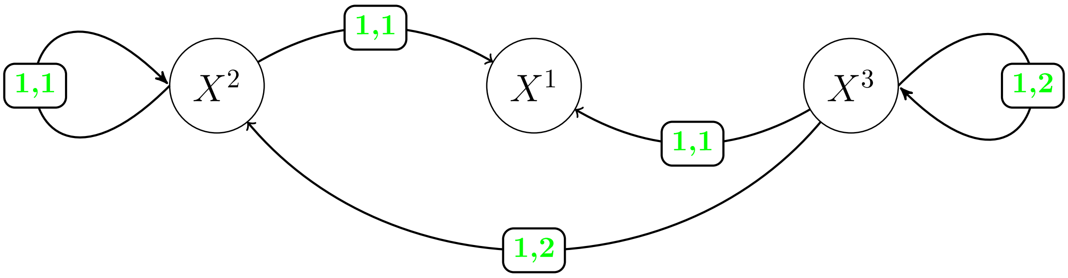


FIGURE 4.3. The temporal directed causal graph of the VAR(2). From the graph, $X^2$ caused by $X^3$ with 1 time delay and 2 time lag and itself with 1 time delay and 1 time lag.

are mutually independent noises, and $d$ and $l$ are time delay between occurrence of the cause and occurrence of the effect and the time lag(order), respectively. The term $(PA_q^j)_{t-s}$ denotes the cause variables which influence $X_t^j$. And we can read out the causal structure from structural causal models by Jacobian matrix.

**Contributions** The main contributions of this work can be summarized as follows,

- Unlike studies that use attentions in neural networks as potential causal relationships for time series and require the use of independence tests for validation, our approach proposes partial differentials as causal relationships. This is the first research to use this method for causal learning of time series data, and it is sufficient to calculate the partial differential as to whether one variable is causally related to other variables.
- It is the first paper to propose the time delay and the time lag together and use the Jacobian matrix to derive these two values in proposed structural causal models.

## 4.2 Related Work

This section is organized as follows: the first subsection will discuss structural learning models based on time series from previous work; the second subsection will explain existing time series prediction methods; the third subsection will describe dynamic causal models based on differential equations.

### 4.2.1 Temporal Causal Structure Learning

#### Granger Causality

Granger Causality(GC [70]) is the first method of analyzing the causal relationships between time-series variables, mainly by detecting techniques to reveal the temporal dependencies between different variables. The basic idea of Granger Causality is that using past values of $X$ and past values of $Y$ can predict future values of $Y$ better than just using past values of $Y$, then the time-series $X$ Granger causes $Y$. It can be written in the following form,

$$X \text{ Granger-causes } Y :\Longleftrightarrow Y_t \not\perp\!\!\!\perp X_{<t} \mid Y_{<t}$$

For $Y$ which can be modeled as an autoregressive process, we can compare the following two time-series prediction models,

$$\begin{aligned} Y_t &= \sum_{i=1}^{q} a_i Y_{t-1} + N_t, \\ Y_t &= \sum_{i=1}^{q} a_i Y_{t-1} + \sum_{i=1}^{q} b_i X_{t-1} + \hat{N}_t. \end{aligned} \tag{4.3}$$

If the noise term $\hat{N}_t$ obtained by regression with $X$ included is significantly smaller than the term $N_t$ obtained by prediction models without $X$, then $X$ Granger causes $Y$. Since Bivariate's GC method can not solve the problem of multiple time-series, conditional Granger Causality [71] made the following extension,

$$X^j \text{ Granger-causes } X^k :\Longleftarrow X_t^k \not\perp\!\!\!\perp X_{<t}^j \mid X_{<t}^{-j}.$$

However, there are limitations in Granger causality. Firstly, violation of causal sufficiency is one of the more serious problems [3]. Secondly, if there are unobserved variables, Granger causality is misleading because it does not deal with unobserved variables or hidden confounders. Besides, Granger's causality only deals with linear dependencies between time series. In the presence of non-linear functions, Granger causality is prone to detect spurious causal relations based on observed local correlations [3].

### Constraint-based Methods

The constraint-based approaches learn a set of DAGs that satisfy the Causal Faithfulness Assumption by conditional independent tests between variables and then determine the optimal DAG by a particular structure in the graphs. The process of these methods contains three stages, which are the skeleton learning stage, direction learning stage, and possible orientation stage. The well-known constraint-based methods are the Peter Clark(PC [61]) algorithm and the Inductive Causation(IC [62]), but these two can not deal with unobserved confounders. The FCI (Fast Causal Inference [64]) algorithm and the FCI improved version RFCI (Really Fast Causal Inference [65]) algorithm are designed to overcome the limitations of the PC and the IC when the Causal Faithfulness Assumptions are violated. The PCMCI [72] and the tsFCI [73] are the time series extensions of the PC and the FCI respectively.

### Information Theoretic Methods

The basic idea of the Information Theoretic approach is that if values of the lag time of variable $X$ provides information about the current state of variable $Y$, then $X$ is likely to be the cause of $Y$. In the case of two variables, then the pairwise dependency can be computed from mutual information $I(X_{t-\tau}, Y_t) = H(Y_t) - H(Y_t \mid X_{t-\tau})$, where $H(Y_t \mid X_{t-\tau})$ is the conditional entropy. For the causal inference of the pairwise dependency of two variables, we also need to take into account the influence of additional information in the system, which can be solved by conditional mutual information [74] $I(Y_t; X_t \mid Z_t)$. Conditional mutual information calculates the mutual information given other information in the system. Many other methods take into account the influence of additional information in the system( [75, 76]). For example, Transfer Entropy(TE [75] calculates the information that is transferred from other variables $X$ to $Y_t$ given the $Y_t$ historical state and it is given by

$$TE_{X\to Y}(\tau) = I(Y_t; X_{t-1}, X_{t-2}, \cdots, X_{t-\tau} \mid Y_{t-1}, Y_{t-2}, \cdots, Y_{t-\tau}).$$

However, TE can only deal with stationary time series data and Partial Symbolic Transfer Entropy(PSTE [77]) is proposed to solve non-stationary sequential data problems.

### Structural Causal Model Methods

Due to the limitations of Constraint-based Methods and Information Theoretic Methods, a new family structure learning methods structural causal Models is proposed, and the form of structural causal Model methods is given by $X_j := f_j(X_{pa_j}, N_j)$ Where $X_{pa_j}$ is for the set of parent nodes of $X_j$ and $N_j$ denotes mutually independent noise. For the time series version of the structural causal Model, please see Equation 4.2. In SCMs, making different assumptions on assignments $f_j$ can lead to different models. For example, Linear Non-Gaussian Acyclic Model(LiNGAM [17]) assumes

$$X_j := \sum_{k\in Pa_j} \beta_{jk} X_k + N_j \qquad j = 1, ...d, \tag{4.4}$$

where all $N_j$ follow non-Gaussian distribution or only a $N_j$ is Gaussian distributed and all $\beta_{jk}$ are non-zero for all $k \in Pa_j$. The extension of LiNGAM for time series is TS-LiNGAM [78], which has the linear assumption and it can have instantaneous effects. For identification of the Structural Vector Auto-Regression(SVAR), VAR-LiNGAM [78] is proposed. LiNGAM based methods are only for linear data, however, TiMINo [79] can handle non-linear relationships but TiMINo is not suitable for large datasets.

## 4.2.2 Dynamic Causal Modeling

Dynamic Causal Modeling(DCM [80]) is a technique that uses differential equations to analyze the interactions between different brain regions and can learn causal relationships between brain regions. Whereas previous understanding of the mechanism between brain regions was based on statistical correlation, DCM uses differential equations to advance this mechanism into the realm of dynamical systems to depict the relationship between variables in nonlinear systems. Assuming that the form of the differential equation which describes the dynamics of activity of different brain regions is given by

$$\frac{d}{dt} z = F(z, u, \theta),$$

where $z$ denotes the activity of different brain regions and $F$ is an unknown function, $u$ is the external input, and $\theta$ refers to the effective connection parameter to be estimated. Also, the equation can be linearly approximated as

$$\frac{d}{dt} z = Az + \sum_{j=1}^{m} u_j B^j z + Cu,$$

where matrix $A$ represents the intrinsic connections between brain regions in the absence of external stimuli; $u_j$ denotes the $j$-th external input; $B_j$ is for the change in connections caused by the $j$-th effective input. And $C$ represents the direct effect of the external input on neural activity.

## 4.3 Gradient-based Temporal Causal Structure Learning

### 4.3.1 Recurrent Neural Networks For Time Series

For time series modeling over input $(X_1, X_2, \cdots, X_t)$, Recurrent Neural Networks(RNN) can generally be used, which has an architecture shown below,

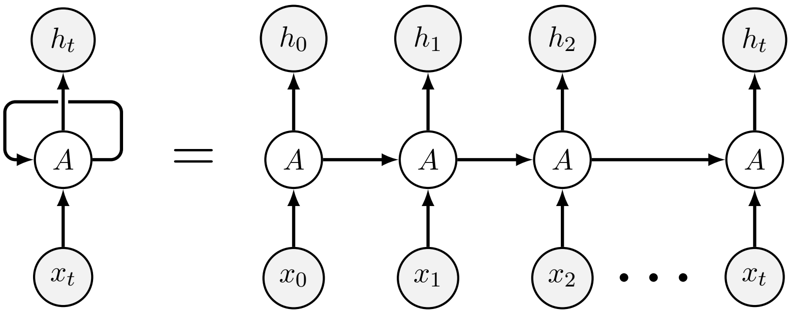


FIGURE 4.4. Vanilla RNN Architecture

In the Figure 4.4, $h_t$ is the hidden state of the neural networks, containing information from $x_0$ to $x_{t-1}$. The output $o_t$ can be predicted using $h_t$ and $x_t$. This process can also be described mathematically by the following equations,

$$\begin{aligned} h_t &= \phi(Ux_t + Wh_{t-1} + b) \\ o_t &= Vh_t + c \\ \hat{x_t} &= \sigma(o_t) \end{aligned} \tag{4.5}$$

where $\phi$ and $\sigma$ are activation functions, $U, W, V$ are weight matrix as parameters to be learned in neural networks and $\hat{x_t}$ is the final predicted output by RNNs. However, RNNs do not work well for modeling long sequences due to the problem of gradient vanishing. Therefore, LSTM(Long short-term memory) [82] is proposed to solve this problem, and the architecture is shown in Figure 4.5.

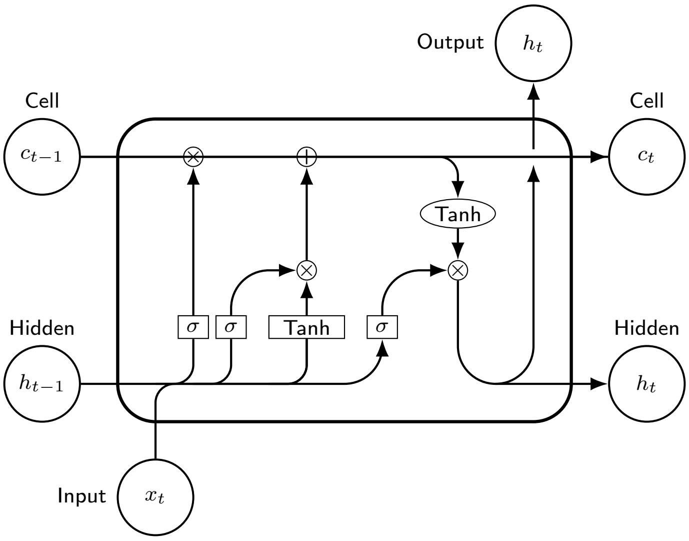


FIGURE 4.5. LSTM Architecture

LSTM consists of three parts: the forget gate, that controls which parts are kept and which parts are discarded; the Input gate, where the sigmoid function is responsible for selecting newer information and the tanh function is responsible for adding new candidates; and the Output gate controls which parts should be output. The GRU(Gated Recurrent Unit) simplifies the LSTM and it consists of two parts: a reset gate, which combines the input with the previous state, and an update gate, which decides how much of the input and state is retained.

### 4.3.2 Neural Ordinary Differential Equations(ODEs)

Neural ODEs use ordinary differential equations to redefine the continuous time hidden state $h_t$ and then solve the Initial Value Problem using the ODE solver(IVP):

$$\frac{d\mathbf{h}(t)}{dt} := f_\theta(t, \mathbf{h}(t)), \text{ where } \mathbf{h}(t_0) = \mathbf{h}_0, \tag{4.6}$$

where the function $f_\theta$ describes the dynamics of hidden state $\mathbf{h}_0$ by a neural network with parameters $\theta$. The hidden state $\mathbf{h}$ at time $t_1$ can be solved by integral from $t_0$ to $t_1$ and the numerical solutions can be solved by an ODESolver [51] with the form of

$$\mathbf{h}_{t_1} = \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt. \tag{4.7}$$

### 4.3.3 ODE-RNNs

We can combine ODEs and RNNs by calculating the state for $i$ observed date from the previous hidden state via an ODESolver as,

$$\mathbf{h}_i' = \text{ODESolve}(\mathbf{h}_{i-1}, f, \theta, t_{i-1}, t_i) \tag{4.8}$$

and updating the latent state $\mathbf{h}_{i-1}$ in a Standard RNN cell by

$$\mathbf{h}_i = \text{RNNCell}(\mathbf{h}_i', x_i). \tag{4.9}$$

ORE-RNNs [51] can be used to do sequence modeling. If there is a sequence $\{X_i\}_{i=0}^T$ where for each $X_i$ is a vector of $\{X^j\}_{j=0}^d$, the task of the model is to make predictions for the next time step from past historical data, i.e. maximizing the conditional probability,

$$p(x) = \prod_i p_\theta(X_i \mid X_{i-1}, \cdots, x_0). \tag{4.10}$$

### 4.3.4 Temporal Causal Structure Learning with Jacobian Matrix

For each $x_t^j$, we learn an ORD-RNN with parameters $\theta$ to fit the assignment of Structural Causal Models as,

$$x_t^j := f_\theta^j(X_{t-1}, X_{t-2}, \cdots, X_0, N_t^j), \text{ where } X_t = \{X_t^j\}_{j=0}^d. \tag{4.11}$$

We now can compute the Jacobian matrix of $x_t^j$ over input variables as

$$J^j = \left[\frac{\partial f^j}{\partial x_1} \cdots \frac{\partial f^j}{\partial x_d}\right]^T = \begin{bmatrix} \frac{\partial f^j}{\partial x_0^1} & \cdots & \frac{\partial f^j}{\partial x_t^0} \\ \vdots & \ddots & \vdots \\ \frac{\partial f^j}{\partial x_0^d} & \cdots & \frac{\partial f^j}{\partial x_t^d} \end{bmatrix}. \tag{4.12}$$

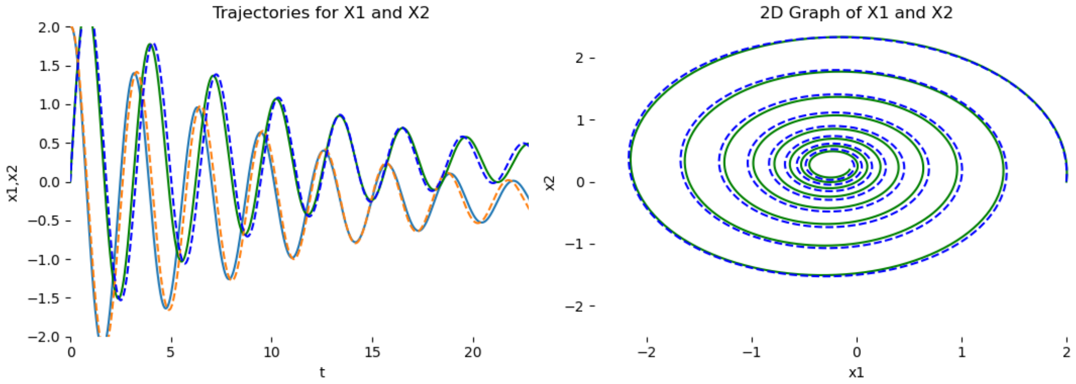


FIGURE 4.6. Use a simple Neural ODE to fit the data sampled from VAR(1) models and dashed lines for trajectories of learned Neural ODE networks.

If $||\frac{\partial f^j}{\partial x_t^d}||_{L^2} = 0$ then $x_t^j$ is independent of $x_t^d$, where $|| \cdot ||_{L^2} = 0$ denotes the usual $L^2$-norm.

## 4.4 Experiments

### 4.4.1 A Toy Example

Considering the time series data sampled from the following VAR(1) without the term of noise,

$$\begin{aligned} x_t^1 &= \alpha_1 + \phi_{11} x_{t-1}^1 + \phi_{12} x_{t-1}^2, \\ x_t^2 &= \alpha_2 + \phi_{21} x_{t-1}^1 + \phi_{22} x_{t-1}^2. \end{aligned} \tag{4.13}$$

We can rewrite the Equation (4.13) as the form of

$$X_t = A + \Phi X_{t-1}, \tag{4.14}$$

where $A = (\alpha_1, \alpha_2)^T$ and $\Phi$ for the matrix $\{\phi_{ij},\ i,j = \{1,2\}\}$ in the Equation (4.13). We assume that the formula is one unit of time variation. In general, the time step can be any unit of time and we can change the time step in the formula to $\Delta t$ by changing the time step. In this way, the Equation (4.14) can be rewritten to $\frac{X_t - X_{t-\Delta t}}{\Delta t} = A + (\Phi - 1)X_{t-1}$. As $\Delta t$ approaches 0, the formula becomes a first-order time-dependent ordinary differential equation,

$$\frac{dX}{dt} = A + (\Phi - 1)X. \tag{4.15}$$

We can use a simple Neural ODE $f_\theta$ with parameters $\theta$ to fit the function $\frac{dX}{dt}$. The performance of function fitting can be viewed in Figure 4.6. The structure can be easily read out by the Jacobian matrix of $f_\theta$ w.r.t. input, please see in Figure 4.7.

### 4.4.2 Stock Market Data

This time series was generated through a simulation of the FAMA Three-factor model [81], in which the stock portfolio's return series is based on three factors: 'volatility',

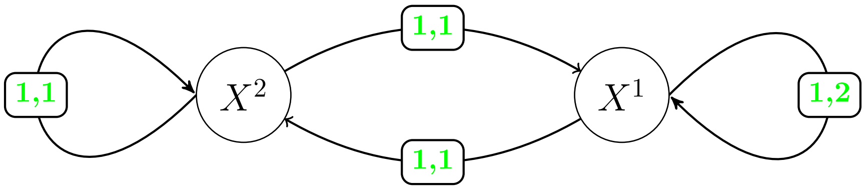


FIGURE 4.7. The temporal directed causal graph of the VAR(1).

'size' and 'value', and the return of stock portfolio at time t is given by

$$R_{i,t} = \sum_j \beta_{ij} f_{jt} + N_{it},$$

where $f_{ij}$ denotes value of one of the three factors at time $t$, $\beta_{ij}$ is weight for $f_{ij}$ at the portfolio and $N_{it}$ represents the portfolio specific noise term. We used the generated data with 6 portfolios and 4000 time points and then split 80% and 20% of the observation length of data for the training and test datasets, respectively. The structure can also be easily read out by the Jacobian matrix of functions learned by neural networks w.r.t. input, please see in Figure 4.8.

## 4.5 Summary

In this chapter, we connect machine learning techniques to causality on time series data which widely exist in our world with the Jacobian matrix of the function $f^j$ on input variables $X$ as causal relationships. It is the first paper to propose it in causal inference experiments on time series data. we also use a full-time causal graph with the time delay and the time lag together to replace the traditional temporal causal structure. Future more, we use ORE-RNN to do function fitting and with experiments, the results show that the success of temporal causal structure learning of time series data.

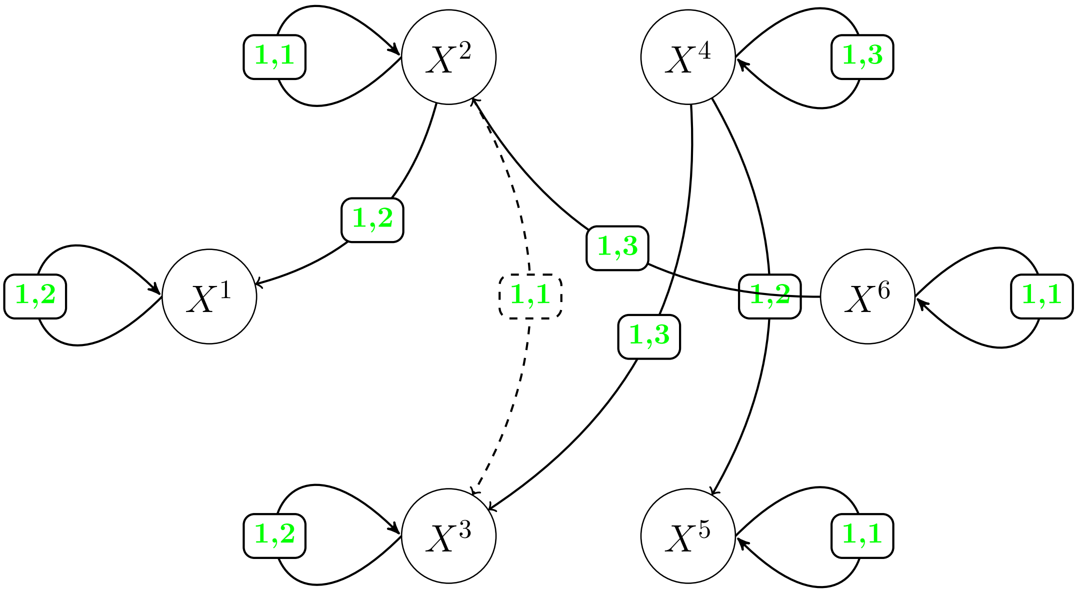


Figure 4.8. The temporal directed causal graph of the financial dataset. The dashed line for the missing edge.

# Chapter 5

# Conclusions and Future Work

In this thesis, a Jacobian matrix based approach is proposed to connect machine learning to causal structure learning on both static data and time-series data. Unlike traditional methods, this method interfaces with deep learning which is a powerful tool for function fitting. And unlike previous machine learning methods that use weights to verify input-output connectivity, our Jacobian-based approach is more straightforward and effective. Furthermore, by using a more flexible approach, the state-of-art deep learning techniques can be adopted instead of simply using multiple layer perceptrons, such as the MAF [34] and Neural ODE [51] we use in this paper, or adding a BatchNorm layer [34] in the neural networks, which was not possible in previous approaches. For static data, the combination of these new deep learning techniques, which increase the model's fitting ability, also allows the method to handle more complex causal structures, which is demonstrated in our experiments. For time-series data, there has traditionally not been a clear general technical framework, and we argue that the framework of our method in this paper is general and clear. The variables in previous data are whether direct causes of future predictions or not, calculating the partial differential of the predictions over the variables can get the answer and this method has also been shown to be valid in experiments. Due to the limited research time for the master's degree, we did not go much deeper to apply our method on time series data, but this thesis has already pointed out the direction of future research, which is to find a better neural network architecture of deep learning and then read out the causal structure between variables through the Jacobian matrix.

# Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. Nature, 521(7553): 436–444, 2015

[2] B. Schölkopf. Causality for Machine Learning. arXiv:1911.10500v2.

[3] J. Peters, D. Janzing, and B. Schölkopf. Elements of Causal Inference - Foundations and Learning Algorithms. MIT Press, Cambridge, MA, USA, 2017

[4] J. Pearl. Causality: models, reasoning and inference. in datasets with extended spatial objects. Cambridge, United Kingdom: Cambridge University Press, 2009

[5] Robert F. Engle. Autoregressive conditional heteroskedasticity with estimates of the variance of UK inflation.. Econometrica, 50 (1982), pp. 987-1008

[6] Pilar Abada, Sonia Benitob and Carmen Lópezc. A comprehensive review of Value at Risk methodologies. The Spanish Review of Financial Economics Vol. 12. Issue 1. pages 15-32 (January - June 2014)

[7] Kingma and Welling Auto-encoding variational Bayes In ICLR, 2014.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, pages 2672–2680, 2014.

[9] L. Dinh, D. Krueger, Y. Bengio. NICE: Non-linear Independent Components Estimation iN ICLR 2015.

[10] Diederik P. Kingma, P. Dhariwal. Glow: Generative Flow with Invertible 1x1 Convolutions. In NeurIPS 2018.

[11] G. Montesi and G. Papiro. Bank Stress Testing: A Stochastic Simulation Framework to Assess Banks' Financial Fragility. Risks 2018, 6, 82; doi:10.3390/risks6030082.

[12] P. Spirtes, C. Glymour, and R. Scheines. Causation, Prediction, and Search, 2nd ed. Cambridge, MA: MIT Press, 2000

[13] J. Pearl and T. Verma. A theory of inferred causation. Principles of Knowledge Representation and Reasoning. Proc. of the Second International Conference, Cambridge, Massachusetts, 1991.

[14] C. Meek. Graphical Models: Selecting causal and statistical models. PhD thesis, Carnegie Mellon University, 1997.

[15] Tsamardinos, I., Brown, L.E. C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. Mach. Learn. 65(1), 31–78, 2006

[16] J. Ramsey, M. Glymour, R. Sanchez-Romero, and C. Glymour. A million variables and more: the fast greedy equivalence search algorithm for learning high-dimensional graphical causal models, with an application to functional magnetic resonance images. International Journal of Data Science and Analytics, pages 1–9, 2016.

[17] S. Shimizu, P. O. Hoyer, A. Hyvarinen, and A. J. Kerminen. A linear non-Gaussian acyclic model for causal discovery. Journal of Machine Learning Research, 7:2003–2030, 2006.

[18] K. Zhang and A. Hyvärinen. On the Identifiability of the Post-Nonlinear Causal Model. In Uncertainty in Artificial Intelligence, 2009.

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

[20] J. Peters, J. Mooij, D. Janzing, and B. Schölkopf. Causal Discovery with Continuous Additive Noise Models. Journal of Machine Learning Research, 2014.

[21] D. Janzing, J. Mooij, K. Zhang, J. Lemeire, J. Zscheischler, P. Daniusis, B. Steudel, and B. Scholkopf. Information-geometric approach to inferring causal directions. Artificial Intelligence, 182–183:1–31, 2012.

[22] Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. DAGs with NO TEARS: Continuous optimization for structure learning. In NIPS, 2018.

[23] Y. Yu, J. Chen, T. Gao and M. Yu. DAG-GNN: DAG Structure Learning with Graph Neural Networks. In ICML 2019

[24] S. Lachapelle, P. Brouillard, T. Deleu, and S. Lacoste-Julien. Gradient-Based Neural DAG Learning. In ICLR, 2020.

[25] I. Ng, Z. Fang, S. Zhu, Z Chen, and J. Wang. Masked Gradient-Based Causal Structure Learning ICLR, 2020.

[26] X. Zheng, C. Dan, B. Aragam, P. Ravikumar, and E. P. Xing Learning Sparse Nonparametric DAGs AISTATS, 2020.

[27] Sanford A. and Moosa I. A Bayesian network structure for operational risk modelling in structured finance operations. J Oper Res Soc 63, 431–444, 2011.

[28] S. Mani and G. F Cooper. Causal discovery from medical textual data. In Proceedings of the AMIA Symposium, 542, 2000.

[29] K. Sachs, O. Perez, D. Pe'er, D. A. Lauffenburger, and G. P. Nolan. Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data. Science, 2005.

[30] M. A. Hernán, B. Brumback, and J. M Robins. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. Epidemiology, 561–570, 2000.

[31] R. H Dehejia and S. Wahba. Causal effects in nonexperimental studies: Reevaluating the evaluation of training programs. J. Amer. Statist. Assoc. 94, 448, 1999.

[32] J Li, O. R Zaïane, and A. Osornio-Vargas. Discovering statistically significant co-location rules in datasets with extended spatial objects. In DaWaK, 2014.

[33] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In ICCV 2017.

[34] G. Papamakarios, T. Pavlakou, and I. Murray Masked autoregressive flow for density estimation. In NIPS, 2017.

[35] H. DMA. On the choice of a model to fit data from an exponential family. The Annals of Statistics, 16(1):342–355, 1988.

[36] Geiger and D. Heckerman. Learning Gaussian networks. In Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI), 1994.

[37] D. M. Chickering. Learning Bayesian networks is NP-complete. In Learning from Data: Artificial Intelligence and Statistics V, pages 121–130. Springer, New York, NY, 1996.

[38] D. M. Chickering. Optimal structure identification with greedy search. Journal of Machine Learning Research, 3:507–554, 2002.

[39] J. Gu, F. Fu, and Q. Zhou. Penalized estimation of directed acyclic graphs from discrete data. Statistics and Computing, 2018.

[40] P. Comon. Independent component analysis — a new concept? Signal Processing, 36:287–314, 1994.

[41] K. Zhang and A. Hyvärinen. On the identifiability of the post-nonlinear causal model. Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009). Montreal, Canada, 2009.

[42] K. Zhang and L. Chan. Extensions of ICA for Causality Discovery in the Hong Kong Stock Market. Proceedings of the 13th International Conference on Neural Information (ICONIP). Hong Kong, China, 2006.

[43] J. Peters, D. Janzing, and B. Schölkopf. Causal inference on discrete data using additive noise models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[44] P. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf. Nonlinear causal discovery with additive noise models. Proceedings of the 23rd Annual Conference on Neural Information Processing Systems, 2009.

[45] P. Daniusis, D. Janzing, J. M. Mooij, and Zscheischler. Inferring deterministic causal relations. Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, 2010.

[46] P. Buhlmann, J. Peters, and J. Ernest. CAM: Causal additive models, high-dimensional order search. high-dimensional order search and penalized regression. Annals of Statistics, 2014.

[47] D. Kalainathan, O. Goudet, I. Guyon, D. Lopez-Paz, and M. Sebag. Structural Agnostic Modeling: Adversarial Learning of Causal Graphs. preprint arXiv:1803.04929.

[48] M. Germain, K. Gregor, I. Murray, and H. Larochelle. MADE: Masked Autoencoder for Distribution Estimation. International Conference on Machine Learning, 2015.

[49] R. M. Anderson, and R. M. May Population biology of infectious diseases: Part I. Nature 180, 361-367, 1979

[50] E. A Coddington and N. Levinson. Theory of ordinary differential equations. Tata McGrawHill Education, 1955.

[51] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural Ordinary Differential Equations. In NIPS 2018.

[52] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[53] Y. Lu, A. Zhong, Q. Li, and B. Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. arXiv preprint arXiv:1710.10121, 2017.

[54] W. KQ and S. LK. Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research, 2009.

[55] J. LaffertyAndrew, A. Mccallum, and F. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Proc. 18th International Conf. on Machine Learning, 2001.

[56] F. Zheng and G.I. Webb. Averaged One-Dependence Estimators. Encyclopedia of Machine Learning. Springer, Boston, MA, 2011.

[57] M. Blei, Y. Ng, and I. Jordan. Latent dirichlet allocation. The Journal of Machine Learning Research, 2003.

[58] R. R Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. Proceedings of the 24th international conference on Machine learning, 2007.

[59] J. Peters and P. Bühlmann. Identifiability of Gaussian structural equation models with equal error variances. Biometrika, Volume 101, Issue 1, March, 2014.

[60] W. Grathwohl, R. T. Q. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud. FFJORD: FREE-FORM CONTINUOUS DYNAMICS FORSCALABLE REVERSIBLE GENERATIVE MODELS. In ICLR 2019.

[61] P. Spirtes, C. N Glymour, R. Scheines, D. Heckerman, C. Meek, G. Cooper, and T. Richardson. Causation, prediction, and search. MIT press, 2000.

[62] J. Pearl. Causal diagrams for empirical research. Biometrika 82, 1995.

[63] M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. JMLR 8, 2007.

[64] P. Spirtes, C. Glymour, and R. Scheines. Causation, Prediction and Search. Lecture Notes in Statistics 81, Springer-Verlag, 1993.

[65] D. Colombo, M. Maathius, M. Kalisch, and T. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. Annals of Statistics 40, 2012.

[66] G. Schwarz. Estimating the dimension of a model. Ann. Stat, 1978.

[67] D. Heckerman, D. Geiger, and D. M Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Machine learning, 1995.

[68] I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. Machine Learning, 2006.

[69] I. Tsamardinos, C. F Aliferis, A. R Statnikov, and E. Statnikov. Algorithms for large scale Markov blanket discovery. In FLAIRS conference, Vol. 2. 376–380, 2003.

[70] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. J.Econom, 1969.

[71] Y. Chen, S. L. Bressler, and M. Ding. Frequency decomposition of conditional Granger causality and application to multivariate neural field potential data. J. Neurosci. Methods, 2006.

[72] J. Runge, D. Sejdinovic, and S. Flaxman. Detecting causal associations in large nonlinear time series datasets. arXiv:1702.07007, 2017.

[73] D. Entner and P. O. Hoyer. On causal discovery from time series data using FCI. In Proceedings of the Fifth European Workshop on Probabilistic Graphical Models, 2010.

[74] A. D. Wyner. A definition of conditional mutual information for arbitrary ensembles. Information and Control, 1978.

[75] T. Schreiber. Measuring Information Transfer. Phys. Rev. Lett, 2000.

[76] J. Sun and E. Bollt. Causal Network Inference by Optimal Causation Entropy. Physica D: Nonlinear Phenomena, 2014.

[77] A. Papana, K. Kyrtsou, D. Kugiumtzis, and C. G. H. Diks. Partial Symbolic Transfer Entropy. CeNDEF Working Papers, Universiteit van Amsterdam, Center for Nonlinear Dynamics in Economics and Finance.

[78] A. Hyvarinen, S. Shimizu, and P. Hoyer. Causal modelling combining instantaneous and lagged effects: an identifiable model based on non-gaussianity. In ICML, 2008.

[79] J. Peters, D. Janzing, and B. Schölkopf. Causal inference on time series using restricted structural equation models. In Advances in Neural Information Processing Systems, 2013.

[80] K. Friston, R. Moran, and A. K. Seth. Analysing connectivity with Granger causality and dynamic causal modelling. Curr. Opin. Neurobiol, 2013.

[81] E. F. Fama and K. R. French. The cross-section of expected stock returns. J. Finance. 1992.

[82] S. Hochreiter and J. A Schmidhuber. Long Short-Term Memory. Neural Computation, 1997.

[83] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In ICLR, 2017.

[84] J. Pearl and E. Bareinboim. External validity: From do-calculus to transportability across populations. Statistical Science, 2014.

[85] K. Ilyes, M. R. Pio, L. Robert, and H. Aapo. Causal Autoregressive Flows. In AISTATS, 2021.