University of Sheffield

SYNTHESISE IT AND THEY WILL COME: GAN-based Data Augmentation for Causal Structure Learning

Joel Hogg

Supervisor: Ramsay Taylor

A report submitted in fulfilment of the requirements
for the degree of BSc in Artificial Intelligence and Computer Science

in the

Department of Computer Science

May 10, 2023

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Joel Hogg

Signature:

Date: 10th May 2023

Acknowledgements

I would like to thank **Dr. Ramsay Taylor** for his pragmatic advise throughout the project, and trusting me enough to allow me to diverge from the original plan quite significantly.

I would also like to thank **Hristo Petkov** of The University of Strathclyde, for lending his expertise on a number of occasions about DAG-WGAN, DAG-GNN, and the area as a whole.

Abstract

Discovering causal relations from data is at the core of many areas of science including medicine, genetics, economics, and health [Yu et al., 2019]. Causal Structure Learning is a powerful area of research for going about this task algorithmically, aiming to infer a graph of causal relations from raw data. However, in practice, the field is underdeveloped: this is in-part due to a lack of quality data to develop new techniques with. Previous methods for synthetic data generation in this domain including Bayesian Networks, and SCMs, may have problems that render their structure identifiable from an exploitation of marginal variances [Reisach et al., 2021]. Other ML-based techniques for data generation impose strict constraints [Wen et al., 2021] [Zheng et al., 2018] [Yu et al., 2019] on the structure of the causal relations, which may inhibit more advanced relations from forming within the model. We propose a novel GAN-based method, 'Causal-Implicit GAN' (**CIGAN**) to implicitly learn causal relations by estimating the target data's distribution, and by using fewer constraints. We verify the model's performance with metrics including: Pearson Correlation tests, Kolmogorov Smirnov tests, *varsortability* [Reisach et al., 2021], and empirical demonstrations. We find that the model is not vulnerable to inadvertent structure identifiability in the same way as previous methods, and can produce new, unique samples similar to the target data, and model more complex causal relations without imposing unrealistic assumptions about the data-generating process of the ground-truth.

Contents

List of Figures

Chapter 1

Introduction

The ultimate goal behind many areas of science involves discovering the mechanisms that explain why variables have the values that they do. Indeed, wanting to explain why events occur is innately human, and dates back to the time of Aristotle [Pearl, 2009a], and likely beyond [1]. Yet, the tools to actually discuss *Causality* in science are still limited, and Causal Inference is still in its infancy compared to other areas in computer science and mathematics [Pearl, 2009b]. Beyond Randomised Control Trials, scientists often rely upon traditional statistics and correlation to make judgements, which not only limits the interpretation of results, but it also can lead to incorrect interpretations if the scientist is not apt in the sometimes misleading nature of correlation[2].

For many, the first thing they learn in statistics class is the age-old mantra: "*Correlation does not imply Causation!*"; Causal Inference seeks to go beyond these limits, and explain the causes behind events, even allowing us to question what might have happened had circumstances been slightly different [Pearl, 2010]. The field of Causal Discovery (the area underpinning the research in this project) seeks to enable Causal Inference by discovering the graph representing how causes are associated between variables, from the raw data.

This Project at a Glance

Over the course of this dissertation, we will further justify the need for Causality in science, and then explain the gaps needing to be filled relating to quality data augmentation in the Causal Discovery (CD) field. Finally, assisted by exploratory experimentation, we will introduce a novel GAN-based model for synthetic data generation to pave the way for more sophisticated CD models to come.

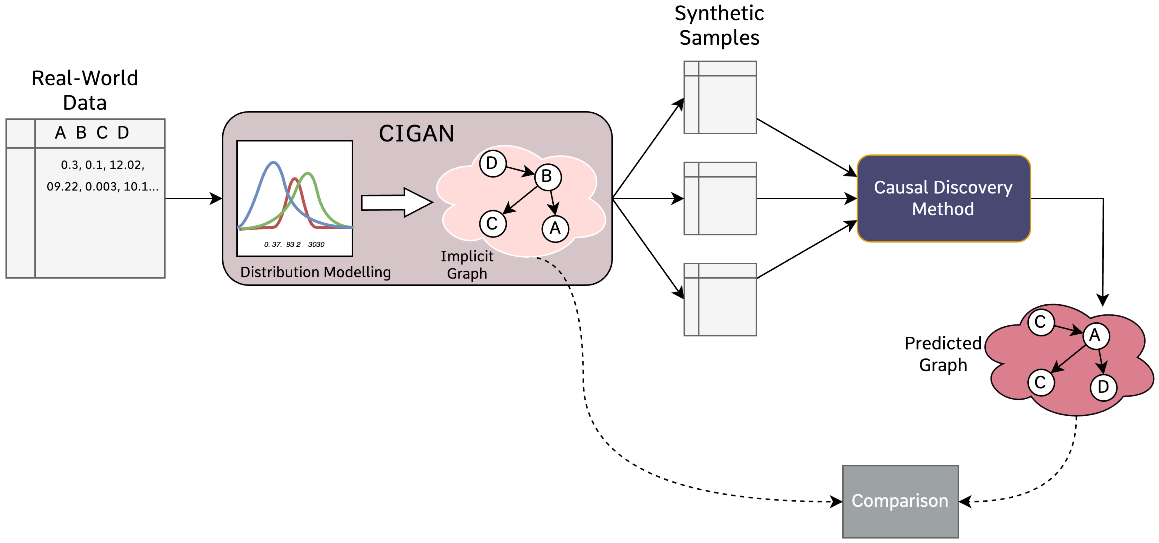


Figure 1.1: An abstracted, high-level representation of the role of 'CIGAN".

1.0.1 Definition of the Task

Shown in 1.1, *'Causal-Implicit GAN'*, (or ***CIGAN*** for short) is part of the pipeline in developing Causal Discovery[3] techniques, and validating that they work on real data. *CIGAN* takes as input some dataset we aim to mimic, and internally (and implicitly), it models the causal relations of each of the input's variables. After it has had sufficient training time to optimise 'closeness' to the target distribution (the input data), it can output samples, which aim to appear as if they had belonged in the original dataset. These additional samples can now be used in conjunction with, or instead of, the original dataset for training, development, or testing on a Causal Discovery (CD) technique. The graph that is modelled inside *CIGAN* is then compared to the prediction from the CD method, for evaluation purposes. At this stage, only a high-level overview of the system is given, and any qualms with the process will likely be addressed later in the report.

1.1 Simpson's Paradox

To highlight the importance of causal models when understanding results, we introduce an example of Simpson's Paradox. Consider the following table, with success rates from heart

[1] Even the University of Sheffield motto is "*Rerum Cognoscere Causas*"-'to discover the causes of things'.

[2] This website gives some good examples of spurious correlations that clearly are not related in causal terms.

[3] Note that Causal Structure Learning is a subset of the more broad Causal Discovery field, but we will use the terms interchangeably throughout this project. See the Literature Review 2.3 for a more detailed description.

surgery when patients are prescribed a new drug[4]:

Recovery	Drug	No Drug
Men	81/87 **93%**	234/270 **87%**
Women	192/163 **73%**	55/80 **69%**
Total	273/350 **78%**	289/350 **83%**

From looking at the effect of the drug in men, it would appear that it has a more favorable outcome (93% successful recovery with the drug vs. 87% without); this is also the case when looking at women. However, when looking at the total success rates, the drug has a rate of 78%, while not taking the drug yields a success rate of 83%- this is a case of Simpson's Paradox [Sprenger and Weinberger, 2021]. These findings suggest that we should only administer the drug if we know the patient's gender- an irrational conclusion. The reason behind these results are due to the proportions being significantly skewed, and the age of the patient being unaccounted for: the success rate of recovery is more strongly influenced by the patient's age and lifestyle than the drug administered[5].

1.1.1 Resolving the Paradox

Pearl [Pearl, 2014] proposes that the paradox can be solved with modern Causal Inference tools and an understanding of the underlying causes. For example, the ground-truth graph of causal effects may look like the following:

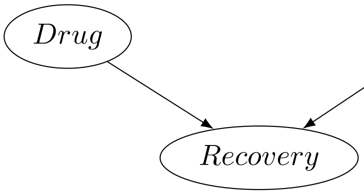


Figure 1.2: Causal Graph depicting the affect of age, and use of the drug, on patient recovery.

It can be seen that the effect of the drug on patient recovery cannot be estimated before we account for age's confounding affect. This highlights the need for an understanding of causal relations in science- the *why*, but the *how* is not yet explained (and will be later in this paper).

1.2 Need for improved data augmentation

Having briefly introduced the need for Causal Inference & Causal Discovery, we will now give a short overview of the gaps that need to be filled in order to improve the capabilities of causal models.

[4] Table adapted from [Charig et al., 1986], a real-life study of success of kidney stones treatments. We adapt the setting for simplicity of exposition, but the numbers remain the same.

[5] This is a fictional underlying cause, but is logical, and there are a number of examples with similar reasons.

Causal Discovery is the task of inferring the causal graph (such as in 1.2) that best explains the data. Without expert knowledge, this is a non-trivial task that requires a great deal of compromise to solve the problem [Nogueira et al., 2022]. Modern techniques which feature Machine Learning are currently limited to synthetic datasets for training, as a result of both the large amount of data needed for ML methods, and the lack of real-world data with a ground-truth graph within the field [Yu et al., 2019]. Although these ML-based techniques [Zheng et al., 2018] [Yu et al., 2021] [Yu et al., 2019] [Petkov et al., 2022] have good performance on synthetic data, questions have been raised over their transferability to real-world settings [Reisach et al., 2021]. To develop more sophisticated models capable of handling more real-world data, there is a necessity for large datasets featuring real-world data with the ground-truth graphs. In the absence of this, data augmentation of existing datasets becomes the next best alternative. We identify the need for more sophisticated synthetic data generation techniques, for which this project aims to satisfy.

1.3 Aims and Objectives

In addition to experimental research, **the key aim for this project is to develop a synthetic data generation method for Causal Discovery that produces more 'realistic' data than its predecessor**. A simple guideline for the goals of this project are as follows:

- Research and document the state-of-the-art in Causal Discovery techniques
- Collect and produce experimental results on CD techniques in relation to performance on synthetic data
- Formulate and propose a novel synthetic data generation technique
- Measure the results of the technique, and discuss the findings critically

1.4 Structure of this Report

Following on from the goals of this project, we disseminate the aims into the following chapters:

- Literature Review
- Analysis of Current Techniques
- Requirements and Analysis
- Design of the Solution
- Findings
- Discussion

- Conclusion

In order to guide an informed decision about the design of the synthetic data generation technique, we need to survey the current techniques, including their shortcomings, and evaluate these techniques using relevant methods. This requirement will be fulfilled in the Literature Review and Analysis of Current Techniques. Although many projects would not have any experimentation before the Requirements and Analysis section, we deemed it necessary to have a greater understanding of the area and its weaknesses before designing a solution. Chapter 4 (Requirements and Analysis) is a more formal statement of the flaws with current techniques, and the need for an improved novel technique. It also formally states the targets for the project, which are used to guide the Findings chapter, in terms of testing and analysis. As a significant portion of this project lies in understanding the domain, proposing the new technique in the Design section is relatively abstracted into the mathematical underpinnings, as these are they key novel contributions of the project- not the structure of the code. The Findings chapter tests the new model against the targets defined in Chapter 4, explaining the meaning of the results, and providing further empirical evaluation. Chapter 6 (Discussion) critically evaluates the results of the project, and discusses additional shortcomings.

Chapter 2

Literature Review

This survey will aim to cover the relevant mathematics and concepts needed to understand the project going forward, assuming the reader has a proficient understanding of Computer Science. Particularly, this project uses several Causality and Statistics techniques which the average computer scientist may not be familiar with, so these will be explained these in further detail.

2.1 Granger Causality

2.1.1 Introduction

One of the first techniques popularised in the field of Causal Inference, Granger Causality [Granger, 1969] Introduces a simplification of causality, and uses the forecasting ability of one time series variable to predict the other as a basis to define if a variable causes another. This technique harnesses the assumption that causes always precede their effects, so if the time-lags of X improve the prediction of the future time series of Y, then X 'Granger-causes' Y. Although this technique will not be used in this project, the concept of using assumptions (such as time linearity) to make causal deductions is important to understand.

2.1.2 Definition

Using 'all the information in the universe' (at time t-1) to predict Y at time t, if removing X from this set of information reduces our forecasting ability at all, it is said that X has some relevant information that can inform us of the state of Y in the future: hence X 'granger-causes' Y.

Let $H_{<t}$ be all the relevant information to predict Y, and $P(Y_t|H_{<t})$ be the optimal prediction of Y given H. We can say X 'granger-causes' Y if:

$$var[Y_t - P(Y_t|H_{<t})] < var[Y_t - P(Y_t|H_{<t} \setminus X_{<t})] \tag{2.1}$$

Meaning that if 'the variance of the optimal prediction error' of Y is increased by including

X, then X 'granger-predicts' Y [Shojaie and Fox, 2021]. (Where we denote $(H_{<t} \setminus X_{<t})$ to mean X is taken out of the set H.

2.1.3 Autoregressive Models

A more commonly used way to measure Granger Causality (GC), is in the form of an autoregressive model [Friston et al., 2012]. Autoregressive models are stochastic probability models used to predict future time series points based on previous data, and can be Uni, Bi, or Multivariate. Kilian defines a Vector Autoregressive model (VAR) as follows:

$$y_t = A_1 y_{t-1} + ... + A_{t-p} y_{t-p} + \epsilon_t \tag{2.2}$$

Where A_n is a coefficient of time lags of y at time t, and ϵ is an error term [Kilian, 2011]. [Granger, 1969] defines a VAR as follows:

$$A_0 x_t = \sum_{k=1}^{d} A^k x_{t-k} + \epsilon_t \tag{2.3}$$

Where d is the time lag, and $A^0..A^k$ are lag matrices. Comparing two time series written in this form, ***if A^0 is diagonal, Granger Causality corresponds to non-zero entries in the autoregressive coefficients*** $(A^0..A^k)$ [Shojaie and Fox, 2021]. Using this basic definition, all relevant information is required to perform a GC test, and the test is limited to the Bi-variate case.

2.1.4 Discussion

In its most basic form, the GC test cannot account for non-linearity between variables, and methods to 'linearize' the data can result in a loss of the causal connection [Shojaie and Fox, 2021]. Furthermore, choosing the correct lag can impact the performance of the test significantly, and this task is left to the researcher, where domain-specific knowledge of the problem is required [Maziarz, 2015]. The test also cannot account for spurious connections, such as in the 'Common Cause Principle', so a rejection of the null hypothesis could mean many different things, and thus is not overly useful for the use case of this project (where a test to determine causality between events will be applied to an observed data set, and a directed graph created).

Non-Linearity

In a causal system, non-linearity is when events interact in a bi-directional way, such as feedback in a complex system [van 't Hof, 2018].

Common Cause Principal

Reichenbach states: *"If coincidences of two events A and B occur more frequently than would correspond to their independent occurrence, that is, if events satisfy relation (1), then there exists a common cause C for these events..."* [Reichenbach, 1956]. If there are 2 events, A, B which appear to be dependent, then it is more likely than coincidence that there is a co-founder: a common cause.

$$A < -C- > B \tag{2.4}$$

2.2 Causal Inference

Separate from notions of Granger Causality, Judea Pearl has fostered his model of causality on a functional, rather than a probabilistic approach. GC is viewed as a statistical method to analyse time series data, whereas Pearl proposes a framework to model the actual causal relations between variables as functions of each other, using a 'Structural Causal Model' (SCM) [White et al., 2010]. Using this model of the causal relations between variables, powerful inference can be performed on the model to reason about questions such as: 'what might happen?', 'why did X happen?', 'what if Y had happened?'. Although important to the field of causality, this project will focus on causal discovery only, thus little prerequisite is needed about causal inference aside from its existence and use-cases.

2.3 Causal Discovery

Causal Discovery is the 'other half' to causal inference, responsible for "analyzing and creating models that illustrate the relationships inherent in the data" [Nogueira et al., 2022]. Prudent to this project's use-case, causal discovery encompasses a range of different techniques for learning relationships (specifically, building a directed graph or SCM) from observational data.

2.3.1 Structural Causal Models

Pearl introduces a Structural Causal Model (SCM) as a set of variables ($x_1 ... x_n \in X$) with unique structural equations as follows:

$$x_i := f_i(pa_i, u_i) \tag{2.5}$$

Where pa_i represents the set of variables that are parents to x_i and u_i is errors due to omitted factors [Pearl, 2009b].

SCM as a tuple

An SCM can be represented more formally as a tuple [Brulé, 2018] $M = \langle U, V, F, P(u) \rangle$ where:

- U is the set of exogenous variables (that cannot be determined within the model)
- set $V = (V_1...V_n)$ of endogenous variables that are determined within the model (*note $U \cup V$)
- F is a set of functions $(f_1...f_n)$ where f_i is a mapping of U_i and pa_i to V_i where $pa_i \subseteq V$
- $P(u)$ is a probability function defined over the domain of U

2.3.2 Discussion

The SCM builds upon the framework of a Directed Acyclic Graph (a directed graph with no cycles), meaning that variables cannot refer back to their ancestors, as this would cause a cycle (which is illegal under a DAG). This limits the usefulness of this model particularly in real-world settings, as causes and effects are often non-linear [Kashif et al., 2020].

Directed Acyclic Graph

A Directed Acyclic Graph (DAG) is a graph in which all edges are directed (one-way), and such that for any path from a node A to a node B, there is no path (following the correct direction) back to A from B [Özkaya and Çatalyürek, 2022]. In the causal application, DAGs are non-parametric representations of the *assumed* data-generating process where variables are depicted as nodes, and an arc between two nodes denotes the existence and direction of a causal relationship [Tennant et al., 2019].

Ancestors

In the causal sense, Pearl defines ancestors as such: "If two nodes are connected by a directed path, then the first node is the ancestor of every node on the path, and every node on the path is the descendant of the first node" [Pearl, 2016].

2.3.3 Mathematical Formulation

Supposing an observed dataset consists of n random variables $V = (V_1...V_n)$, each variable satisfies the SCM formulation as defined in 2.5. The task of causal discovery is to recover the causal adjacency matrix, B that represents the data's underlying SCM, where $B_{ij} = 1$ represents that V_i is a parent of V_j. [Ding et al., 2021]

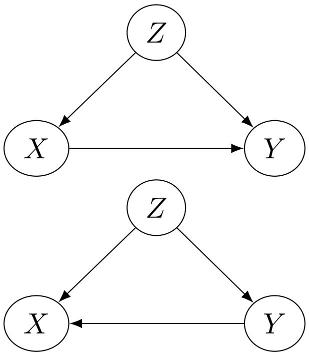
2.3.4 Causal Markov Assumption

The Causal Markov Assumption (also known as the 'Markov Condition') states that: in an acyclic causal graph, every node is independent of its non-descendants [GEIGER and PEARL, 1990]. This means that the assumption is held that apart from the nodes that a node points to (and ones its children point to) in the directed graph, it is statistically independent of all other nodes. This assumption allows us to perform CD techniques such as the PC algorithm (discussed in more detail in 2.4.1), which algorithmically tests for independence.

2.3.5 identifiability with a Markov Equivalence Class

Using the framework of the Markov Condition to encode causal models, we can introduce the Markov Equivalence Class (MEC). An MEC is the set of DAGs that encode the same set of conditional dependencies [He et al., 2015]. This means that DAGs with the same conditional dependencies, but different causal directions between some nodes can exist in the same MEC, which brings rise to the issue of *identifiability*.

Figure 2.1: Two different DAGs within the same Markov Class



Given a joint distribution of data, there is no way to identify the underlying causal model beyond its Markov Equivalence Class, making Causal Discovery impossible from observational data alone [Vowels et al., 2021]. However, if assumptions are made about the data, such as the causal data generating mechanisms (such as assuming the data was generated by a SCM), then a causal graph can be *estimated*. Asymmetries in the data can be drawn out, for example, by asymmetrical complexities between two variables [Nikolaou and Sechidis, 2020]. This idea is key to the field of Causal Discovery, but it also is the reason that CD algorithms struggle with real-world data, as the data-generating mechanisms are infinitely complex, making any model inaccurate [Lawrence et al., 2021]. This topic is discussed in further detail in 5.1.2.

2.4 Approaches to Causal Discovery

In this section, we will break down the different approaches to the problem of Causal Discovery, giving an overview of the classes of methods, and some well-known methods.

2.4.1 Constraint Based Approaches

A group of algorithms that test for conditional independence for variables in a graph, these methods are able to identify the correct Markov Equivalence Class of the true graph from observational data. [Nogueira et al., 2022] [Sadeghi and Soo, 2022].

PC algorithm

An implementation of the conceptual 'inductive causality' (IC) algorithm, the PC algorithm recursively deletes edges from a fully connected graph based on conditional independence tests. While popular, this is a more naive approach to causal discovery, but has been shown to reliable, and can be fast when parallelized [Le et al., 2015].

2.4.2 Score Based Approaches

A separate group of algorithms that use a 'closeness' score to select a relevant graph from a large set of candidate graphs [Nogueira et al., 2022].

Greedy Equivalence Search

Greedy Equivalence Search (GES) iteratively searches a group of MECs with an additional edge to the current MEC, scoring each MEC. The highest-scoring MEC is used, and the process repeats until the score can no longer be improved. The process is repeated again with edge removals [Ramsey et al., 2017].

2.4.3 Exploiting Asymmetry Based Approaches

These methods exploit some type of asymmetry to infer causal direction, specifically using either: time, complexity, or functional. For example, Granger Causality exploits time asymmetry by making causal deductions from the assumption that causes precede their effects [Granger, 1969].

Kolmogorov Complexity

[Nikolaou and Sechidis, 2020] uses the Occam's Razor principle to assume causal direction: 'the most simple explanation is usually the right one' is used by choosing, in a 2-variable problem, the causal direction with the lowest Kolmogorov Complexity.

Independence of Causal Mechanisms Principle

The Independence of Causal Mechanisms (ICM) principle is a key concept that enables Causal Discovery to take place [Parascandolo et al., 2017]. It assumes that in a SCM, each causal mechanism (ie. edge on the graph connecting two nodes) does not influence or contain any information about any other mechanism. This notion can be exploited to estimate the function between two variables, without having to account for the rest of the graph.

2.5 Causality in Machine Learning

2.5.1 i.i.d

Independent and Identically Distributed (i.i.d) is a group of random variables that are samples from the same, nonfluctuating probability distribution, and are independent of each other, meaning for any variable, it gives no information about another variable [Clauset, 2011]. i.i.d is a common assumption in machine learning algorithms, including some causal discovery algorithms [Vonk et al., 2022].

2.5.2 Moving Distributions

Due to the i.i.d assumption, many machine learning algorithms fail with data that experiences a shift in its distribution [Schölkopf et al., 2021]. One example of distribution shift is *Covariate Shift*, where the distribution of the data can change over time, while the variables retain the same labels. The task of causal learning is to account for a changing distribution, by way of modelling the result of interventions on variables (which can change the distribution) [Schölkopf et al., 2021].

2.5.3 Non-Stationarity

A non-stationary process is one with a distribution that changes over time [Gagniuc, 2017].

2.5.4 Discussion

We identify the difficulty of performing Causal Discovery on real-world data, due to the 'non-iidness' of real data: naturally, there are coupling between certain variables, be that a causal, temporal, or semantic coupling [Adhikari, 2020]. [Cao, 2013] identified examples of these couplings in real-world data, demonstrating the difficulty of causal discovery in this domain: to assume a non-i.i.d structure on real-world data imposes a model too restrictive, and can lead to poor results, especially in the causality field, where causal information between variables is important [Adhikari, 2020].

2.6 Generative Adversarial Networks

A Generative Adversarial Network (GAN) is a framework that introduces distribution approximation as an adversarial game [Goodfellow et al., 2014]. Goodfellow et al. [Goodfellow et al., 2014] introduce the Generator and Discriminator, which are trained from each others' loss. This concept will be explained in detail as it is core to the data augmentation tool described in later chapters.

2.6.1 Adversarial Game

The Generator and Discriminator are trained to respectively minimise and maximise the following function:

$$\min_G \max_D E_{x \sim p_{data}(x)}[\log D(x)] + E_{z \sim p_z(z)}[1 - \log D(G(z))]$$

2.6.2 Generator

The goal of the Generator (G) is to take samples of latent noise z, and output samples $G(z)$ that are in the probability distribution of the target dataset (p_{data}). By iterativly receiving its loss score, backpropagation refines the network weights of the G and it ideally outputs more realistic samples. After training, the Generator should converge to a good estimator of p_{data}, the target probability distribution, by refining the distribution $z \sim p_g$ (when $G(z)$).

2.6.3 Discriminator

The Discriminator's (D) role is to be fed at random, either a sample from the target dataset, or a sample from the generator, $G(z)$, and output a scalar probability that the sample belongs to G. Thus, it is the goal of G to *fool* D by producing realistic samples.

2.6.4 Mode Collapse

This is a prevalent issue in GANs since their introduction in 2014, and refers to a situation where D only produces the same few samples that consistently fool D every time, or it only replicates a single sample from the training data [Zhang et al., 2018]. If the Discriminator fails to identify a set of outcomes in the target distribution, these become open for abuse by the Generator.

2.6.5 Auto-encoder

An Auto-encoder is a Neural Network that learns high-dimensional embeddings of its input data, by using an Encoder and Decoder [Kramer, 1991]. The Encoder takes the input data, and transforms it into a smaller, but deeper mapping (z) in latent space. The Decoder then takes this mapping and transforms it into a flat vector, which is an efficient embedding of the original input data. Auto-encoders can be used in the Generators of GANs, to help them learn feature embeddings of the data more efficiently [Ghojogh et al., 2021].

2.6.6 Data Augmentation

Data Augmentation is a technique in machine learning used for increasing the amount of training data by generating new data-points from the existing set [Frid-Adar et al., 2018]. Although not a *feature* of GANs, they are used frequently for this purpose [dos Santos Tanaka and Aranha, 2019 [Xu and Veeramachaneni, 2018a] [Frid-Adar et al., 2018].

2.7 Survey of Relevant Papers

This survey will aim to evaluate several techniques highly relevant to the project area, which the data augmentation tool will be based on, or where inspiration has been drawn. These papers have been chosen to demonstrate the author's understanding of the project domain, and adjacent work, and to explain the techniques that have been drawn on in creating the model. As a prerequisite, it is assumed the reader has read and understood the topics in the prior sections.

2.7.1 DAG-GNN: DAG Structure Learning with Graph Neural Networks

The authors [Yu et al., 2019] employ work from Zheng et al. [Zheng et al., 2018], which transformed the search for an optimal DAG from a super-exponential combinatorial problem to one of continuous optimization. The authors introduce a Graph Neural Network (GNN) which aims to optimize an Evidence Lower Bound score (ELBO). A variational auto-encoder is used, with an explicit acyclicity constraint to enforce a DAG structure.

The authors transform the model for linear SEM: $X = A^T X + Z$ to triangular solve of A: $X = (I - A^T)^{-1} Z$. This can be written with functions: $X = f_2((I - A^T)^{-1} f_1(Z))$. This form becomes a learnable adjacency matrix (A) of the data (X). The authors use evidence lower bound (ELBO) to learn the generative model by maximising this metric:

$$L^k_{ELBO} \equiv -D_{KL}(q(Z|X_k)||p(Z)) + E_{q(Z|X_k)}[\log p(X^k|Z)]$$

A variational auto-encoder is used as the generative model, due to its ability to represent the graph explicitly, and due to their ability to capture complex distributions of data. The authors find better graph size / SHD scalability over DAG-NOTEARS. Additionally, the authors report SHD=19 on the Sachs dataset; however, our results using DAG-GNN yielded a result of 31. This difference of ~ 10 is not an overly concerning amount: it can be put down to difference in model hyper-parameters (as we use the defaults, but the authors do not comment on their set-up), and the stochasticity of the model.

Discussion

Overall, DAG-GNN is an impressive improvement over its successor, DAG-NOTEARS [Zheng et al., 2018]. It shows an improvement in performance across multiple areas, and graph complexities. However, the authors only give their results measured with Structural Hamming Distance (SHD), which can be a limited metric. From this, we aim to incorporate multiple metrics when testing our model, so as not to fall into the same pitfall.

2.7.2 DAG-WGAN: Causal Structure Learning with Wasserstein Generative Adversarial Networks

The authors of this paper [Petkov et al., 2022] improve upon the works of DAG-GNN [Yu et al., 2019] by introducing causal structure learning as an adversarial game. The authors use the variational auto-encoder from DAG-GNN as the generator in the GAN, and introduce their own discriminator. The model is trained with Wasserstein adversarial loss; replacing the 'critic' role of the discriminator with a 'realness' metric for the generator's output [Petkov et al., 2022]. Wasserstein GAN has been shown to improve training stability over standard GANs [**?**]. The authors show improvement in most areas over DAG-GNN and DAG-NOTEARS, particularly with larger graph sizes.

Discussion

The authors show the validity of using GANs for causal structure learning, and implicitly show that it is likely possible for a GAN to learn causal structure internally. However, acyclic constraints are imposed on the model in the encoder and decoder, so it is unclear from this paper if removing the constraint would alter performance.

2.7.3 Synthesizing Tabular Data using Generative Adversarial Networks (TGAN)

The authors [Xu and Veeramachaneni, 2018b] utilise the impressive data-generation abilities of GANs, by widening their domain of image generation to include tabular data. The authors propose that due to the ability of GANs to implicitly learn the probability distribution, they capture the relations between columns better than that of previous generative models. The authors introduce the tabular data generation problem as follows:

- Table T contains n_c continuous random variables, n_d discrete random variables, both of which follow an unknown joint distribution $P(C_{1:n_c}, D_{1:n_d})$, where each row is a sample of the distribution.
- The goal is to learn generative model $M(C_{1:n_c}, D_{1:n_d})$, such that samples from M create a synthetic table T_{synth}, which can satisfy:
  - An ML model trained with T_{synth} can achieve similar performance to that of one trained on T
  - The mutual information between any variables i,j in T and T_{synth} is similar.

The authors find that, through metrics such as macro-F1, and pair-wise mutual information, TGAN improves on other generative models in most areas.

Discussion

This paper introduces the concept of GANs for tabular data generation, and the formulation of the problem for the causal domain will likely be somewhat similar. We aim to build upon ideas introduced in this paper, such as that of implicitly learning the distribution and its relation to the GAN's ability to synthesise data. We identify the use of mutual information as a metric to test the efficacy of generative models and aim to use this metric to test our model.

2.7.4 Causal-TGAN

We acknowledge the work of [Wen et al., 2021], which is in a similar area to that of this project, albeit in a slightly different way; we will disambiguate these differences for the purposes of outlining the novel contributions of our project. Causal-TGAN works with a limited number of variables in the ground-truth dataset, owing to its design of 'sub-generators' for each variable, which generates its values. Furthermore, it uses an SCM-based generator to create values, hence opting for a much more explicit method of relation modelling than this project. We acknowledge the authors' use of metrics such as KL-Divergence and Machine Learning efficacy to test the model, which we aim to use in our project.

2.7.5 Beware of the Simulated DAG! Causal Discovery Benchmarks May Be Easy To Game

The authors of [Reisach et al., 2021] find evidence that synthetic data for Causal Discovery may inadvertently make its causal structure identifiable through an abuse of marginal variances between variables. The authors show that it is possible, in some examples, to identify the direction of causal association between variables by ranking variances. Furthermore, it is demonstrated that continuous structure learning algorithms, such as DAG-GNN [Yu et al., 2019] and DAG-NOTEARS [Zheng et al., 2018] inadvertently exploit this weakness, due to the black-box nature of ML algorithms. The experiments are performed using SCMs generated with Erdos-Renyi graphs (an algorithm for randomly generating graphs). The authors introduce a metric to measure the extent to which the data's structure is identifiable from variances alone: *varsortability*.

Discussion

This paper has important implications for this project because it shows that most synthetic data for the Causal Discovery domain (many of which are generated with SCMs) is much easier for some CD methods than first hypothesised. Thus, scoring well on this synthetic data does not mean the method will produce any meaningful results on real data, as it has learnt mainly to use variances as a means for structure learning. Furthermore, the authors use a comparison of raw synthetic data vs. standardised synthetic data, showing that standardised gives worse performance (due to removed variances), which will be a useful metric for the new data augmentation tool.

2.7.6 Causal Protein-Signaling Networks Derived from Multiparameter Single-Cell Data

We introduce with work of Sachs et al. [Sachs et al., 2005] as a key dataset to the research in this project. The authors use Machine Learning techniques to derive the causal influences between proteins to recreate an accurate Bayesian Network model of the behaviour. The dataset produced from this work, known simply as 'Sachs' within Causal Discovery circles

[Yu et al., 2019] [Zheng et al., 2018], features 11 molecules: ’praf’, ’pmek’, ’plcg’, ’PIP2’, ’PIP3’, ’p44/42’, ’pakts473’, ’PKA’,’PKC’,’P38’,’pjnk’. This dataset is very useful for CD research because it is one of few datasets with ground-truth relations attached.

Chapter 3

Analysis of Current Techniques

In this chapter, we will analyse a current Causal Discovery technique in-depth, chosen to be reflective of current ML-based techniques in the area. We choose a diverse range of synthetic datasets and include results from a limited finding of real datasets. **We use this analysis of the capabilities of current CD methods working with synthetic data to inform how we should build the data augmentation tool**, in order to improve certain aspects (which will later be defined) of the data that the CD methods run on.

3.1 Criteria for Dataset Analysis

In order to justify the need for the novel data augmentation tool, we must demonstrate the performance of a causal discovery technique with an existing suite of synthetic datasets. A number of datasets encompassing different domains and data generation techniques have been selected to give a comprehensive view of Causal Discovery performance as the field stands. The CD method of choice has been selected to be *DAG-GNN* [Yu et al., 2019], as it has good performance, is well-cited in its area, and is generally well-understood by the author.

3.2 Overview of Datasets

The datasets selected can be split into 4 categories: 2-variable problem, artificial graphs, Bayesian networks, and real datasets. This is partly due to the 4 different sources for these datasets, but also how they are generated. The most comprehensive dataset is artificial graphs, as a number of graphs spanning different sizes have been generated to give a picture of how the CD techniques scale.

3.3 Cause-Effect Pairs

This dataset is a collection, from different fields of science, of cause-effect pairs [Mooij et al., 2015]. The dataset includes over 100 examples, but in preparing the dataset, only around 60 were found to work with the CD method. The goal of the CD method is, from data about 2 variables, to discern the causal direction. Meaning, to infer which variable causes the other (for example, in a rising hot-air balloon, altitude causes the temperature to decline, not the other way around)

3.3.1 Results

As shown in 3.3.1, the CD method failed to infer the causal direction correctly more than a coin flip would. With so many samples, and only 2 variables in each, it's clear that the method is unable to discern causal direction when the graph only has 2 variables.

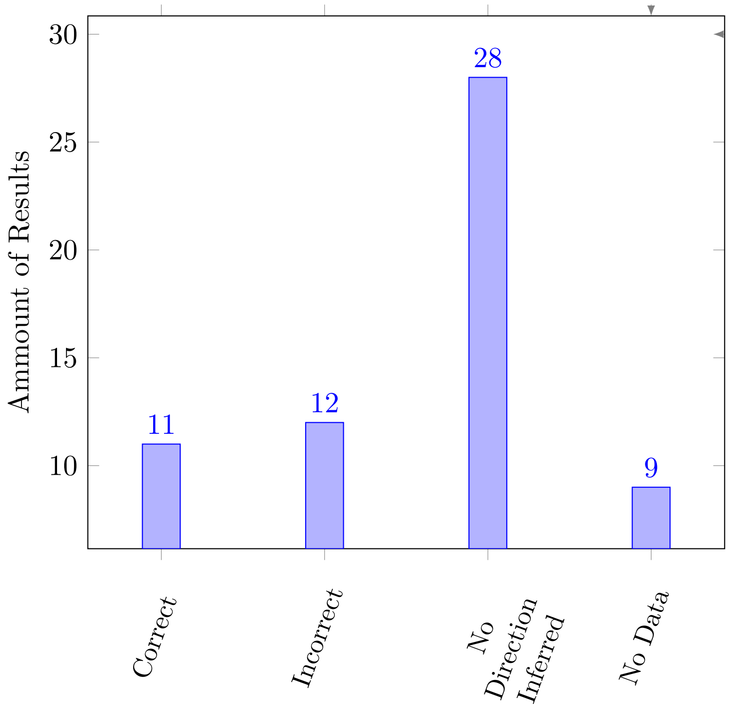


Figure 3.1: Results for cause-effect pairs dataset using DAG-GNN

3.3.2 Discussion

Running this dataset, some samples had more than 2 variables in their graph and had to be discarded, and some did not work altogether. After this, only 60 of the over 100 samples remained, although this is still a good enough quantity to test with. The 'No Direction Inferred' category of results refers to the times when the CD method outputs a result, which did not show a causal direction between any variables (ie: it claims the 2 variables are not related). Clearly, the similar levels of correct and incorrect results demonstrate that the CD method fails with a graph of size 2.

3.4 Artificial Graphs

This dataset features synthetic data from a number of different types of causal graphs, produced using the Causal Discovery Toolbox python package [Kalainathan and Goudet, 2019]. A graph can be defined by its type and the number of nodes in the graph.

```
generator = AcyclicGraphGenerator(graph_type,
    npoints=sample_size, nodes=nodes, parents_max=parents_max)
```

The term '*graph type*' refers to the equation used to describe a causal link between variables, using the Structural Causal Model (SCM). For instance, some types available are linear, Gaussian, sigmoid, and neural network.

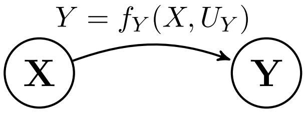


Figure 3.2: An example of a linear SCM

3.4.1 Results

As shown in 3.3, regardless of complexity, the CD method appears to scale similarly among all graph types. The main factor that seems to affect performance is the number of nodes in the graph.

3.4.2 Discussion

It may have been expected that the complexity type of the graph would affect Causal Discovery performance, but this appears not be the case from these specific results. This suggests that data generated from causal graphs is roughly the same difficulty to model for the CD method. This is likely because the artificial graphs' data are all generated with the same underlying SCM process regardless of what function is used between variable. This highlights a key issue with artificial causal data: the CD method is aware of the data generating process that was used, so Causal Discovery becomes much easier. This issue will be expanded on in a later section, but it is important to note that these specific results are not conclusive on the issue with simulated data. In addition, it is also worthy to note only one metric was tested: Structural Hamming Distance (SHD). This metric tests for, between the target DAG and the result DAG, the number of edge removals to get from one graph to another [Peters and Bühlmann, 2014]; this implies that naturally, when there are larger graph sizes, there are bound to be more nodes that are wrong (even though there still may be the same percentage wrong as a smaller graph with a lower score). This finding implies that the CD method may not scale as poorly as the findings suggest, and that a high SHD is not always conducive to poor performance [Peters and Bühlmann, 2014].

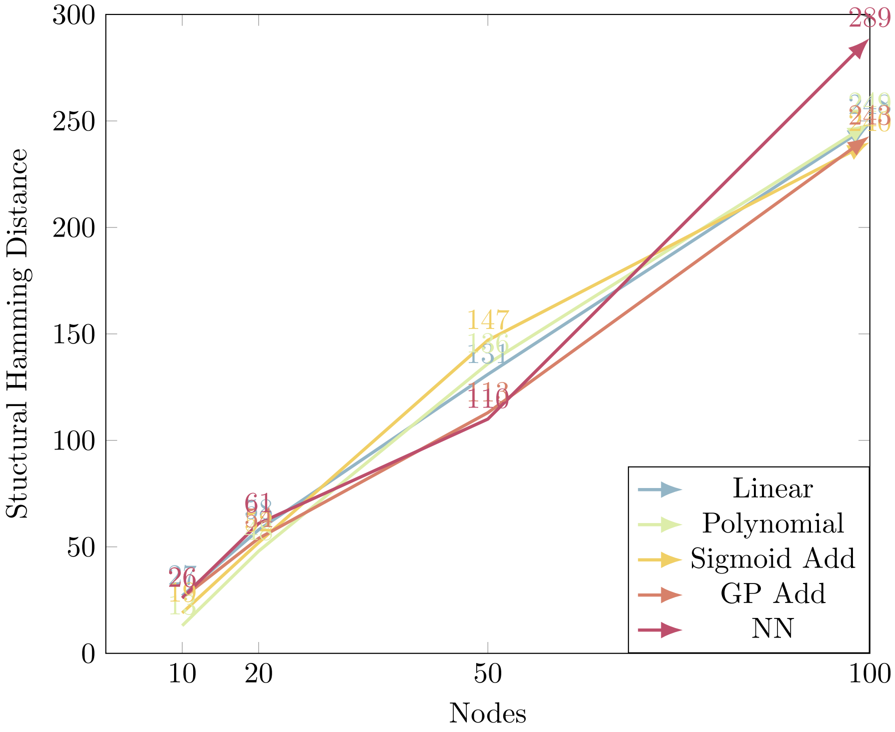


Figure 3.3: A plot of Structural Hamming Distance against number of nodes in ground-truth graph, ran using DAG-GNN.

3.5 Bayesian Networks

This dataset [Scutari,] features Bayesian networks of different sizes, gathered primarily from papers where the authors have produced Bayesian networks to model some data. The data in question is produced by taking the network, and sampling it; artificial 'readings' are then produced for each node and this data will be stored in a tabular format for input into the CD method. The goal of the CD method will be to infer the ground-truth network (represented as a DAG) from this sampled data.

3.5.1 Results

These results are shown in 3.4. It can be seen that nodes and SHD roughly correlate, although only 3 samples have been ran, so the dataset size is likely too small to draw any meaningful conclusions at this time.

3.6 Real Datasets

Real datasets were collected from papers, where experts have uncovered what is likely to be the ground-truth graph and provide the tabular data from which they inferred this. Real-world data with ground-truth graphs are rare [TU, 2023], making this section extremely

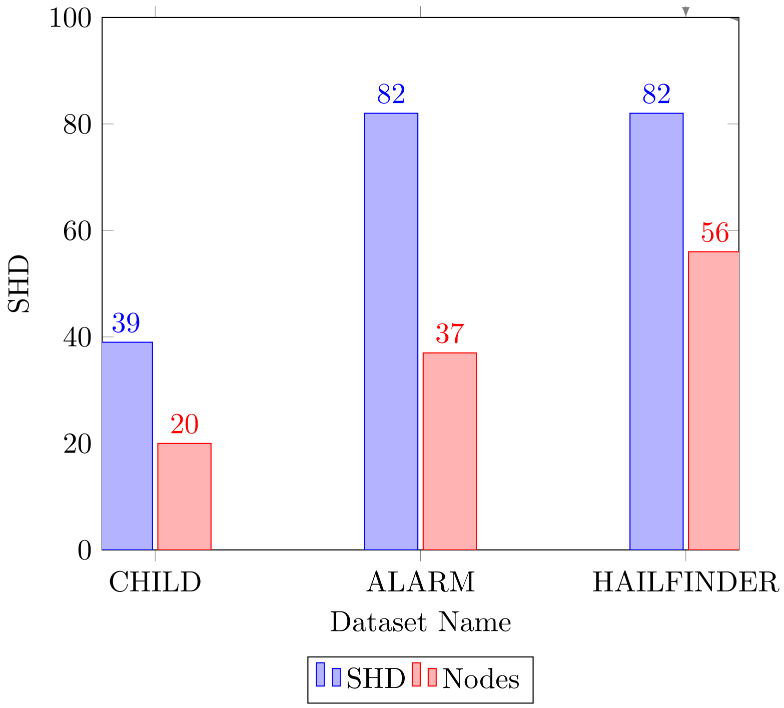


Figure 3.4: A bar chart of Structural Hamming Distance and Nodes in graph for each Bayesian network, ran on DAG-GNN.

Dataset Name	*nodes*	*Result*
sachs	11	31

Figure 3.5: SHD for real data, ran on DAG-GNN

limited. The Sachs [Sachs et al., 2005] dataset is that of a protein-signalling network, where real data from the experiment is provided, and the expert-made suspected ground-truth DAG is provided to test with.

3.6.1 Results

As shown in 3.6.1, datasets are limited. However, this does show that compared to the artificial graphs dataset, the SHD result is slightly higher for Sachs when compared to results of a similar graph size. This suggests that real data may be harder for the CD method in question (DAG-GNN), although the sample size is very small, so no definite conclusions can be drawn.

3.7 Summary

In this section, we have tested a continuous optimization based-structure learning algorithm, 'DAG-GNN'[Yu et al., 2019] against a range of synthetic datasets of different characteristics from different sources. **Our main finding is the scalability invariance to different ’complexities’ of synthetic data**. We have hence identified that synthetic data is not as challenging for the end CD method as intended, and we aim to bridge this gap in capability with the model defined in the coming chapters.

Chapter 4

Requirements and Analysis

Having identified issues with the current synthetic data generation methods in the Causal Discovery domain, **we will define the requirements for an improved method, and the targets needed to reach a satisfactory implementation.**

4.0.1 Recap of Issues with Existing Synthetic Generation Methods

To justify the need for a new synthetic data generation technique, we restate the findings collected in this report to date. Firstly, [Reisach et al., 2021] (as discussed in 2.7.5), finds that training data intended for CD algorithms generated synthetically may unintentionally encode causal structural information in the marginal variances of variables, which can be exploited by black-box CD methods. Secondly, as discussed in 3.4.1, we find that Structural Hamming Distance against the ground-truth does not scale differently (as would be expected) for more 'complex' SCMs. This finding suggests that the complexities of synthetic generation methods do not work as intended, in part perhaps, because the CD method knows the mechanism used to generate the data (and the assumptions that were made)[1].

4.1 Overview

The purpose of this project is to create a new data augmentation technique to create more 'realistic' training data for Causal Discovery models. Thus, the data augmentation technique will need to be useful for the models it is supplying additional training data for; **results obtained with CD methods using this synthetic data should be an accurate representation of the CD method's performance, and should give similar performance to that of comparable 'real-world' data**. This axiom will be used as the basis to define our requirements and targets for the model, such that meetings these targets means that the model is successful.

[1] See 5.1.2 for more detail on the assumptions made in synthetic data generation

On the term 'training data'

We suggest that the technique's synthesised data could be used as 'training data', for Machine Learning models in the Causal Discovery domain. However, we clarify that many models in this domain do not 'train' on some dataset which provides them with domain generalisation; many models [Yu et al., 2019] [Yu et al., 2021] [Zheng et al., 2018] [Petkov et al., 2022] simply rely on exploiting some feature of the data which can be used to recover at least the Markov Equivalence Class. This being said, there is ongoing work [Wang et al., 2022] in achieving out-of-domain generalisation for causal discovery algorithms, so we suggest that the synthetic data generated will be useful for future ML-based algorithms that will require a large set of data to learn from.

4.2 Requirements

We define the requirements of the model in order for it to have been successful as follows:

1. The synthetic data should be 'similar' to that of the real-world data it aims to replicate.
2. The synthetic data's ground-truth DAG should only be recoverable in the same ways that real data is.
3. The synthetic data should be useful for developing Causal Discovery techniques.

These abstract requirements will be expanded on to justify the tests used to verify the success of the technique.

4.3 Targets

We will now define the specific metrics used to test the model and quantify the target range of values on these metrics.

4.3.1 Kolmogorov-Smirnov test

Derived from requirement 1, we will measure 'similarity' as a comparison between the cumulative distributions of each variable, for the real data, and the synthetic. We will use the two-sample Kolmogorov-Smirnov test [Dimitrova et al., 2020], a non-parametric method which measures the distance between two cumulative probability distributions. The p-value of this result will be used in interpreting similarity, by estimating the probability that the two groups were sampled from the same distribution. If this value is less than 0.05 (5%), we reject the null hypothesis and infer that the two groups are from different distributions (and vice-versa for above 5%). We would expect that the samples are above the 5% threshold, as the model is aiming to roughly replicate the range of values for each variable.

4.3.2 Pearson Correlation Coefficient

Derived from requirements 1 and 3, we introduce Pearson Correlation [Freedman et al., 2007] as a means to measure linear correlation between two datasets. Although the test does not account for any non-linear correlation, this test will be useful to verify that the model is producing samples not exactly identical to the truth dataset- which would be an example of mode collapse[2]). As the aim of data augmentation is to create new samples that are useful for increasing training capacity, this 'new-ness' can be measured in part by Pearson Correlation; we expect to see a score around 0 ± 0.1 as the samples should not be linearly correlated.

4.3.3 *varsortability*

Derived from requirement 2, we introduce *varsortability*, a measure of 'realness' proposed by [Reisach et al., 2021]. The authors proposed a metric to specify to what extent some data's underlying DAG could be identified from ranking the variables' marginal variances alone; on a continuous scale of 0-1, 1 indicates the DAG is entirely identifiable. We define the acceptable range as $0.4 - 0.6$; pure chance- the variances should contain no information about the graph's structure, so we allow only for stochasticity from the sample.

4.3.4 Standardised Comparison

Derived from requirement 2, we introduce the Standardised Comparison, a technique used in [Reisach et al., 2021]. Similar to the measure of *varsortability*, the standardised data (which removes any variances beyond a Gaussian fit) is compared to the raw data in terms of its efficacy on CD methods. We expect that given the synthetic data's variances should provide no information about the structure, the result of a CD method using both the raw data and standardised data should be the same.

4.3.5 Machine Learning efficacy

Derived from requirement 3, we introduce ML efficacy as a metric for the utility of the synthetic data for developing CD methods. Thus, the CD method is expected to score poorer on the technique's synthetic data, than on previous techniques for generating synthetic data (because the causal relations being modelled in this new technique are more complex, meaning that the graph should be harder to infer). Due to this, we define the ML efficacy target as: $ML_e(T_{artificialgraphs}) < ML_e(T_{synth}) < ML_e(T)$, where $ML_e(T_{artificialgraphs})$ is the performance of a CD method on the input data of the 'artificial graphs' dataset outlined previously, and $ML_e(T_{synth})$ is the performance on the data generated by the new technique. 'performance' is to be SHD from the target ground-truth causal graph.

[2] Mode Collapse is a situation in a GAN where the discriminator's performance outmatches the generator's too greatly, and the generator collapses and generates only copies of the sample set (see [Zhang et al., 2018] [?]

4.4 Recap of Targets

Targets for CIGAN (T_{synth})		
Target Name	Target Range	Unit
varsortability	0.5 ± 0.1	score (float)
Standardised Comparison	$CD(T^{raw}_{synth}) \approx CD(T^{standardised}_{synth})$	SHD (int)
Kolmogorov-Smirnov	$KS_p(T_{synth}, T) > 0.05$	(float)
Pearson Correlation	$PC(T_{synth}, T) = 0 \pm 0.1$	(float)
ML efficacy	$ML_e(T_{artificialgraphs}) < ML_e(T_{synth}) < ML_e(T)$	SHD (int)

Figure 4.1: Quantified targets for CIGAN. $CD(T)$ refers to the score of the Causal Discovery method ran on table T (measured by SHD).

Here (in figure 4.1 we define exactly the range the model is expected to produce for the targets discussed previously.

Chapter 5

Design of the Solution

In the following section, we will formulate the problem of synthetic data generation for the causal discovery domain, and discuss the need for, and implementation of the data augmentation tool.

5.1 Mathematical Formulation of the Problem

5.1.1 Tabular Synthetic Data Generation

Let a table T contain n_c continuous random variables $C_1 ... C_n$, where each row of T is a sample. The probability distribution D generates the samples belonging to T [Xu and Veeramachaneni, 2018b]. The distribution is defined by Kearns et al. [Kearns et al., 1994], where there can be said to be a true generator G_D for D, which takes a string of random bits y and outputs $G_D[y] \in X$ according to D, where X is the support of D (comparable to T). It is important to disambiguate the 'generator' defined here is different to the generator of a GAN, although the GAN's generator does have a similar role.

The goal of Synthetic Table Generation is to learn a generative model $M(C_{1:n}, D_{1:n})$, such that M generates samples which can be concatenated to synthetic table T_{synth}. The synthetic table should aim to satisfy $D_{KL}(T||T_{synth}) \leq \epsilon$, where ϵ is an arbitrarily small number [Xu and Veeramachaneni, 2018b].

Kullback–Leibler divergence

Kullback-Leibler divergence (denoted as $D_{KL}(A||B)$ measues the statistical distance between two probability distributions (A and B) [Kullback and Leibler, 1951].

5.1.2 Formulation of Causal DAG learning

The process of learning a DAG from observational data will be explained with mathematical notation, in order to make how it is done clearer to the reader.

- Let there be random variables $X = (X_1, ..., X_n)$, which we desire to learn a DAG from.

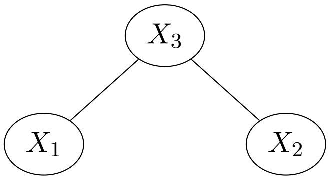
- We assume that X was generated by some joint probability distribution $(X_1, ...X_n) \sim p(X)$.
- $p(X)$ can be re-written using Bayes' rule as follows:

$$p(X) = p(X_n) \prod_{i=0}^{n-1} p(X|X_{i+1}, ..., X_n)$$

- Using the rule of independence *(if $X_1 \perp\!\!\!\perp X_2$ then $p(X_1, X_2) = ... = p(X_1)p(X_2)$)*, $p(X)$ can simplify to the factorization of the joint distribution.
  - For example, the conditional dependencies may look like:

$$p(X_1, X_2, X_3) = p(X_3|X_1)p(X_2|X_3)$$

  - Which corresponds to a dependence graph:



- At this point, to estimate any causal directions between variables requires assumptions about the data-generating process [Nogueira et al., 2022]. Assuming the principal of Independent Causal Mechanisms (ICM) [Parascandolo et al., 2017], we move forward with the assumption that each edge in the graph G is a causal module that is not influenced by, or influences any other in the graph.
- Assuming an SCM, each connected node can be modelled as a function of its parent nodes, allowing causal direction to be inferred.

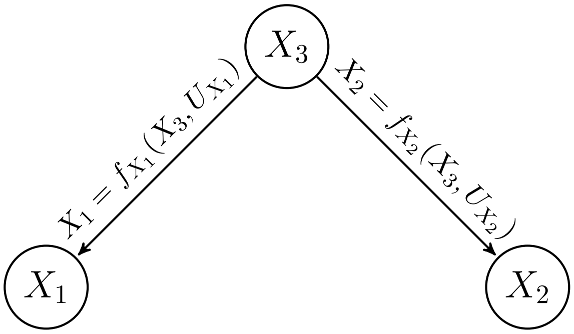


Figure 5.1: SCM with functions annotated (X_3 is the cause of X_1 and X_2)

- This restriction of function classes (eg: assuming all functions in the SCM are linear) can help to identify causal asymmetry between variables [Schölkopf, 2019]. For example, it has been shown that given a distribution over X and Y generated by an additive noise model (of the form $Y = f(X) + V$ where V is noise), one cannot fit an additive noise model in the opposite direction [Schölkopf, 2019].

5.2 Formulation of Causal-Implicit-GAN

5.2.1 The need for real data in Causal Discovery

In the current state of the field, Causal Discovery datasets based on real-world data are very rare, especially those with an associated ground-truth graph [Yu et al., 2021] [Yu et al., 2019]. [Reisach et al., 2021] identifies that synthetic datasets may inadvertently render their structure identifiable through asymmetries in marginal variance. Results indicated that current CD algorithms merely fall within naive baselines for real-world dataset. It is evident that to to obtain utility from the Causality field, CD methods will have to be trained on real data to ensure they can work outside of an i.i.d setting [Reisach et al., 2021]. We have identified the need for real-world data to make useful CD techniques, while concurrently, shown the lack of datasets for this domain. We propose a novel data augmentation tool, which can extend the utility of existing datasets, while overcoming the current identifiability issues with current synthetic data generation techniques.

5.2.2 Mathematical formulation of CIGAN

We refer to the previous definition of the Tabular Synthetic Data Generation problem, where the model, M can be said to have learnt T's joint probability distribution if $D_{KL}(T||T_{synth}) \leq \epsilon$. This is an assumption that will be used in building the model going forward.

We have also demonstrated in this section that learning the joint probability distribution of the target data (X) is a key step in uncovering the DAG representing the causal relations within the data. This observation will be used in a further assumption for the model.

Key Contribution

We propose that learning the joint probability distribution J_X of the target data X is conducive to learning its underlying causal mechanisms. Given some model has been said to have learned J_X, **then we propose that, given the model is sufficiently complex[1], then it has also implicitly modelled X's underlying causal mechanisms, which are responsible for producing the samples in T.** Using this assumption, further samples

[1] The definition of 'sufficient' is limited at this time: this follows Goodfellow's seminal GAN paper[Goodfellow et al., 2014], which suggested GANs learn the distribution given sufficiently large deep nets, but did not expand on the exact size [Arora and Zhang, 2017]. We aim to extend this suspension of detail in our work, to give a more pragmatic definition of 'sufficient' by way of the results of our model.

from the same causal mechanism can be attained trivially to generate a synthetic table T_{synth}, which should be within some measure of 'realness'[2].

The proposed model will utilise a Generative Adversarial Network (GAN) as a key part of the model. GANs have been selected due to their ability to generate realistic results, and work on small amounts of training data, both of which are constraints for this project [Petkov et al., 2022].

5.2.3 Overview of CIGAN

As seen in 5.2, CIGAN is trained using the adversarial set-up defined earlier in this project. The generator is comprised of an auto-encoder, as this yielded good performance in DAG-WGAN [Petkov et al., 2022]. The difference between this model and DAG-WGAN, is that although both are GANs in nature, this model does not aim to recover the adjacency matrix (A) of X, CIGAN instead aims to produce samples $\hat{X}$ that are '*close*'[3] to samples from X. Furthermore, CIGAN's generator takes Z, latent noise as input, and learns to map this to useful output, whereas DAG-WGAN takes samples of the ground-truth table, X as input.

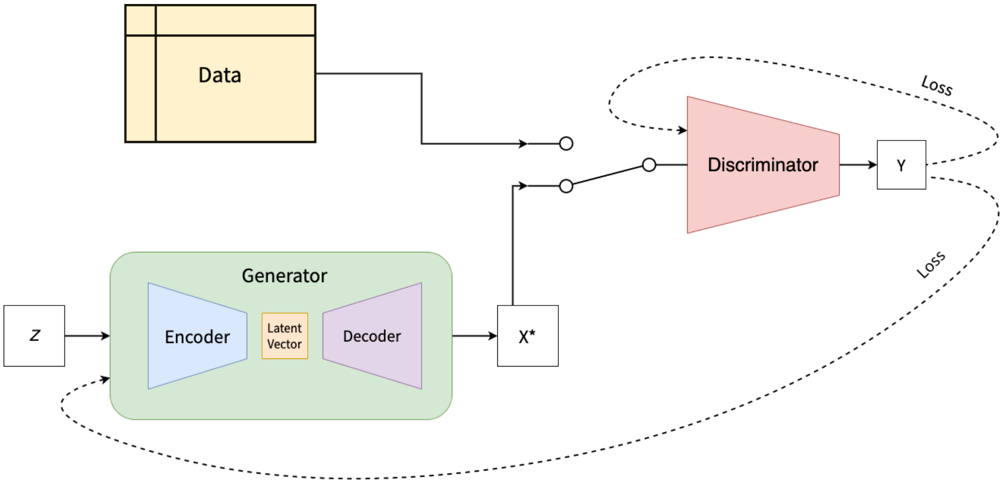


Figure 5.2: Design for Causal-Implicit GAN, showing the adversarial training set-up.

5.3 Discussion

While an interesting approach for causal data augmentation, the model has some potential drawbacks that should be discussed, and other aspects of the design and implementation will also be expanded on to illuminate any potential questions the reader might have.

'Learning a faithful directed acyclic graph (DAG) from samples of a joint distribution

[2] Measures for how *'real'* this synthetic data is will be discussed further under the implementation section.

[3] Measures for '*closeness*' are defined under the 'targets' section of this chapter. **??**

is a challenging combinatorial problem' [Yu et al., 2019], so how can it be guaranteed that CIGAN will be able to do this, especially without any constraints to guide it?
Owing to the Universal Approximation Theorem [Hornik et al., 1989], Neural Networks (NNs) are able to approximate a wide range of continuous functions using a finite amount of neurons in a single layer. As Zheng et al. [Zheng et al., 2018] reformulated the causal structure learning problem to continuous optimisation, and CIGAN aims to implicitly model the causal structure of the input data before outputting samples, it can be said that the function the model is trying to learn is continuous, and thus approximable.

The point of the causal relations being implicit is so that more complex structures beyond the DAG can exist internally to allow for more complex data to be produced on the other end. Does scoring T_{synth} against the input DAG not nullify this, by forcing relations to become linear?
Our contribution is that instead of using SEMs or SCMs[4] to internally model T, the causal relations will be implicit. If SCM constraints were imposed, then the ICM2.4.3 assumption would have to be used, which differs from the real world, and potentially makes the causal structure more identifiable [Reisach et al., 2021]. By not using this assumption, we pave the way for future CD techniques to be tested on datasets more akin to the real world. It should be noted that removing this ICM assumption doesnt mean that CD becomes impossible, as other assumptions can be used, such as time coupling [TU, 2023] (where it is assumed time precedence is an indicator of causal direction).

5.4 Implementation

As a significant portion of the work for this project lies in the understanding of the relevant areas, and the theoretical formulation of the model with the experimentation underlying it, this section is relatively concise. The model itself consists of one Python file, which contains the class for the data augmentation model written with PyTorch.

We use a GAN, with the generator comprising of an auto-encoder[5]. The encoder and decoder are made from 3 linear layers, with ReLU activations in-between. In the '*forward()*' method, the input vector (latent noise) is resized to the shape of the adjacency matrix, A. A is generated randomly at the start of the method, using a sub-method, '*generate_adj_A()*'. This sub-method generates a random erdos-renyi graph, and captures its adjacency matrix. The noise vector x is then multiplied with this adjacency matrix so that values are removed where there are no connections between variables (as this corresponds to 0s in A, multiplying a value in x by this will make it also 0). In doing this, x has been in a way, 'fitted' to match A.

[4] Structural Causal Models (see mathematical survey 2.5 are an evolved version of Structural Equation Models

[5] Relevant code snippets are available in A

From here, x will be fine-tuned by the generator during training to decrease the distance from G's samples to the real data. Although the random graph the generator is working with is unlikely to be similar to the real data's ground truth, by aiming to minimise distance between $G(z)$ and X, we propose that a '*realistic*' graph will be produced, although potentially with different causal mechanisms.

We use the PyTorch library for the implementation as it is written in Python (which is relatively simple to understand and write) and because the library offers simple-to-understand implementations of deep learning techniques, compared with alternatives such as TensorFlow.

Chapter 6

Findings

In this chapter, we will demonstrate the results from CIGAN, and check the model against the targets specified in the Requirements and Analysis chapter. We aim to give a comprehensive view of the characteristics and performance of the model. For most tests, the Sachs [Sachs et al., 2005] dataset is used for comparison, as it is a well-known dataset used in the Causal Discovery domain, and the causal relations are relatively complex.

Targets for CIGAN (T_{synth})		
Target Name	Target Range	Unit
varsortability	0.5 ± 0.1	score (float)
Standardised Comparison	$CD(T_{synth}^{raw}) \approx CD(T_{synth}^{standardised})$	SHD (int)
Kolmogorov-Smirnov	$KS_p(T_{synth}, T) > 0.05$	(float)
Pearson Correlation	$PC(T_{synth}, T) = 0 \pm 0.1$	(float)
ML efficacy	$ML_e(T_{artificialgraphs}) < ML_e(T_{synth}) < ML_e(T)$	SHD (int)

Figure 6.1: Quantified targets for CIGAN. $CD(T)$ refers to the score of the Causal Discovery method ran on table T (measured with Structural Hamming Distance).

To recap the targets previously outlined in 4.1, the model will be deemed a success if it conforms to the specified ranges given in the table 6.1.

6.1 Targets

6.1.1 *varsortability*

As seen in 6.2, the *varsortability* score achieved by CIGAN is 0.455 *3s.f*, which is within the threshold specified in the initial targets. The ’gold standard’ would be 0.5- pure chance:

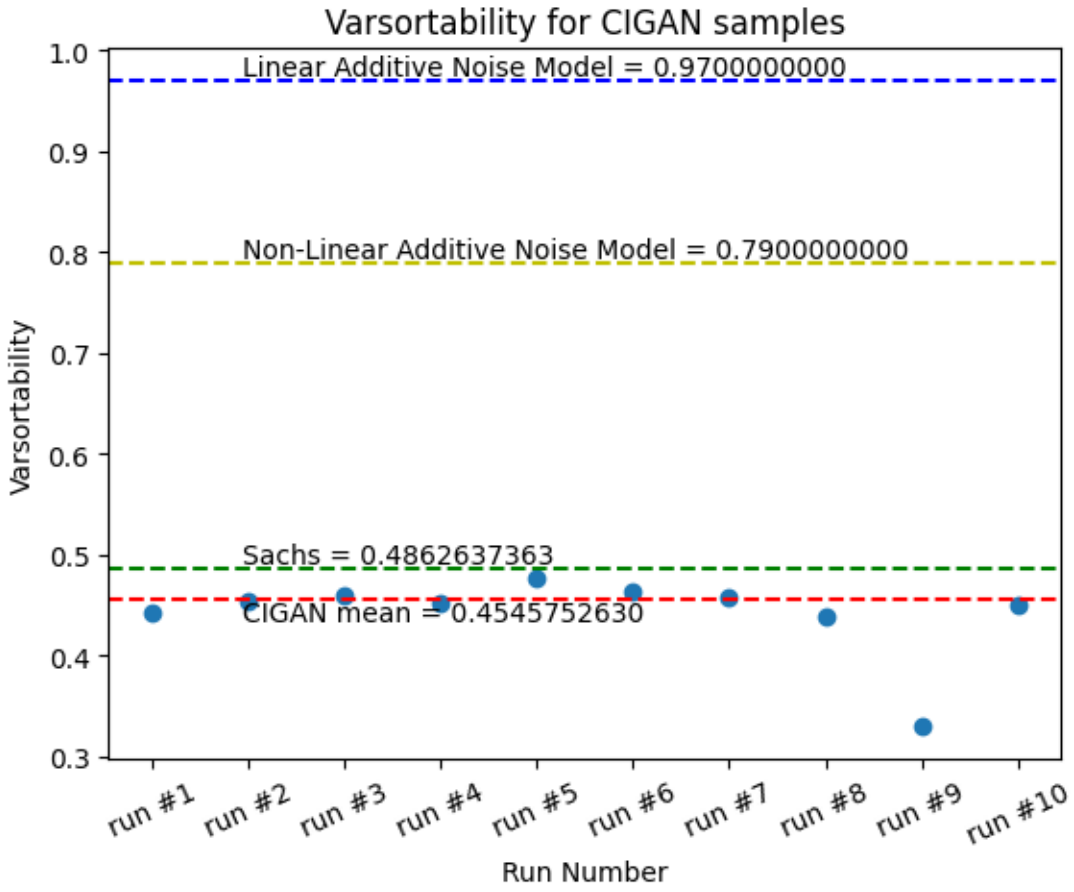


Figure 6.2: Scatter Plot of varsortability for CIGAN samples, from 10 separate runs (ie: re-trained 10 times). Using results from [Reisach et al., 2021], the mean result for a Linear Additive Noise model using gumbel noise, and a Non-Linear Additive Noise model based on a Multi-Layer Perceptron are included also for reference.

Due to the small number of nodes in some real datasets such as Sachs, ranking by any arbitrary metric (such as variance) produces at least some correct connections. CIGAN's training function was ran 10 times, each of which initialises a new random graph that the implicit relations are based on. Due to this, two samples from different instances of a trained CIGAN model will have different underlying graphs. To expand, using Sachs as an example, the model will re-arrange how each variable (column in the ground-truth dataset) relates to other ones. This means that the causal graph will be different; hence the samples will be of different underlying causes. This randomisation of the graph is simply to encourage the model to generate new data, that is 'realistic' in the same aspects that Sachs is, but using a new graph so as to not just replicate the ground-truth data exactly.

Scores of previous synthetic generation methods have been included (taken from [Reisach et al., 2021]) for reference purposes. These methods show that they are partially recoverable, highlighting the success of CIGAN. **To summarise, the model has been successful in this metric, by not being recoverable by variance ordering.**

6.1.2 Standardised Comparison & ML efficacy

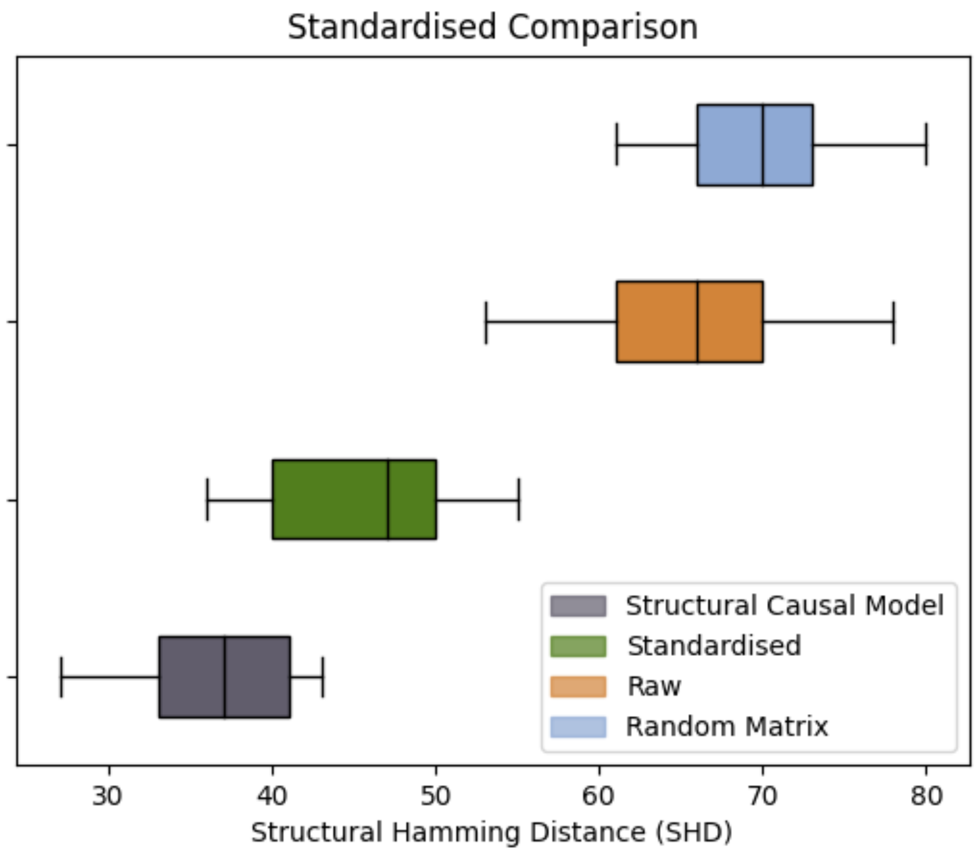


Figure 6.3: A standardised comparison of output samples (n=50) from CIGAN, where the dark green box represents CIGAN samples that have been standardised, and the orange box represents CIGAN raw samples. A random adjacency matrix and samples from a previous method of generating synthetic data (SCM) have also been included.

The figure 6.3 in this section will double, to demonstrate results of both the Standardised Comparison, and of ML efficacy. Firstly, samples from CIGAN have been ran on a CD method (DAG-GNN [Yu et al., 2019]), and the prediction tested against the graph defined in the CIGAN model, much like 1.1. The CD method has been tested both with a raw, unchanged sample from CIGAN, and the same sample, but with standardisation to remove any information that might be encoded in the variances. It can be seen in 6.3 that the standardised data actually performs better than the raw. This result shows empirically, that

CIGAN's synthetic data is not exploitable by its variances. The fact that the standardised data performs better than that of the raw is surprising, and more research is needed to offer an explanation at this time. However, this does not take away from the finding that the raw data cannot be exploited by CD methods. Although this result is positive, it does not fall into the target threshold for success. More work may need to be done to ensure that the data is not exploitable by standardisation.

To analyse the ML efficacy results, we compare the SHD of the SCM data (generated in a similar fashion to the '*artificial graphs*' dataset from Chapter 4), to the standardised and raw CIGAN data. It can be seen, as predicted in the target table 6.1, the SCM data performs better than the CIGAN data. For reference, the CD method achieves a SHD of 47 on Sachs. **To summarise, the CD method performs as expected when comparing previous synthetic generation techniques, and CIGAN**. This implies that CIGAN will be more useful for training new CD techniques, as it gives a closer result to how real data would perform.

6.1.3 Kolmogorov-Smirnov

Variable	$KS_p(T_{synth}, T) > 0.05$**?**
praf	**true**
pmek	**true**
PIP2	**false**
PIP3	**true**
p44/42	**true**
pakts473	**false**
PKA	**false**
PKC	**false**
P38	**false**
pjnk	**true**
total	5/10 (50%)

Figure 6.4: Analysis of p-values from 2-sample Kolmogorov-Smirnov tests for each variable from Sachs vs. CIGAN samples.

It can be seen from 6.4 and 6.5 that only half of the variables between Sachs and CIGAN can be said to originate from the same distribution; in other words, half of CIGAN's sample variables are said to be similar to the Sachs distribution that CIGAN aimed to reproduce. This result is not in line with the specified targets 6.1, however, since the targets were initially specified, the design of CIGAN has been described in more detail. Due to the design of the model, the samples outputted are based on a randomised graph, meaning that it is natural that the samples from CIGAN and Sachs may not match, as they have different underlying graphs. Due to this finding, the 50% score is satisfactory.

```
praf: KstestResult(statistic=0.22, pvalue=0.17858668181221732)
pmek: KstestResult(statistic=0.26, pvalue=0.06779471096995852)
PIP2: KstestResult(statistic=0.3, pvalue=0.02170784069014051)
PIP3: KstestResult(statistic=0.2, pvalue=0.2719135601522248)
p44/42: KstestResult(statistic=0.22, pvalue=0.17858668181221732)
pakts473: KstestResult(statistic=0.3, pvalue=0.02170784069014051)
PKA: KstestResult(statistic=0.36, pvalue=0.002834980581320342)
PKC: KstestResult(statistic=0.42, pvalue=0.0002460240344273171)
P38: KstestResult(statistic=0.36, pvalue=0.002834980581320342)
pjnk: KstestResult(statistic=0.16, pvalue=0.5486851446031328
```

Figure 6.5: Kolmogorov-Smirnov results for each variable.

6.1.4 Pearson Correlation

As seen in 6.6, the Pearson Correlation Coefficient is measured between Sachs and CIGAN, across the 11 variables. The average of all these variables equates to -0.0007 (*4s.f*), meaning the 2 tables are not linearly correlated- the expected finding 6.1. As stated previously, this test does not account for any non-linear relations but is a useful metric for ensuring that the model is not simply producing exact copies from the ground-truth dataset, as this would not be useful for data augmentation.

6.2 Additional Results

6.2.1 Summary Statistics

We include a comparison of the summary statistics 6.7 between a sample from CIGAN, and the first 50 rows of the Sachs dataset, in the form of a box-plot. It can be seen that although Sachs has a greater range of values for most variables, CIGAN does appear to have produced similar results, in terms of the Interquartile range . This result confirms that CIGAN has been successful in producing samples similar to that of the Sachs dataset.

6.2.2 Actual Tabular Output Comparison

We include a comparison of the tabular output of CIGAN and Sachs for transparency purposes. Due to the size and complexity of these tables, little can be inferred without the metrics used in the rest of this chapter, but it may be of interest to the reader to see what the output looks like. We include only one sample from the model, however due to its stochasticity, multiple runs may produce slightly different results. A table displaying the full 50 rows of the sample is available in the appendix B.1.

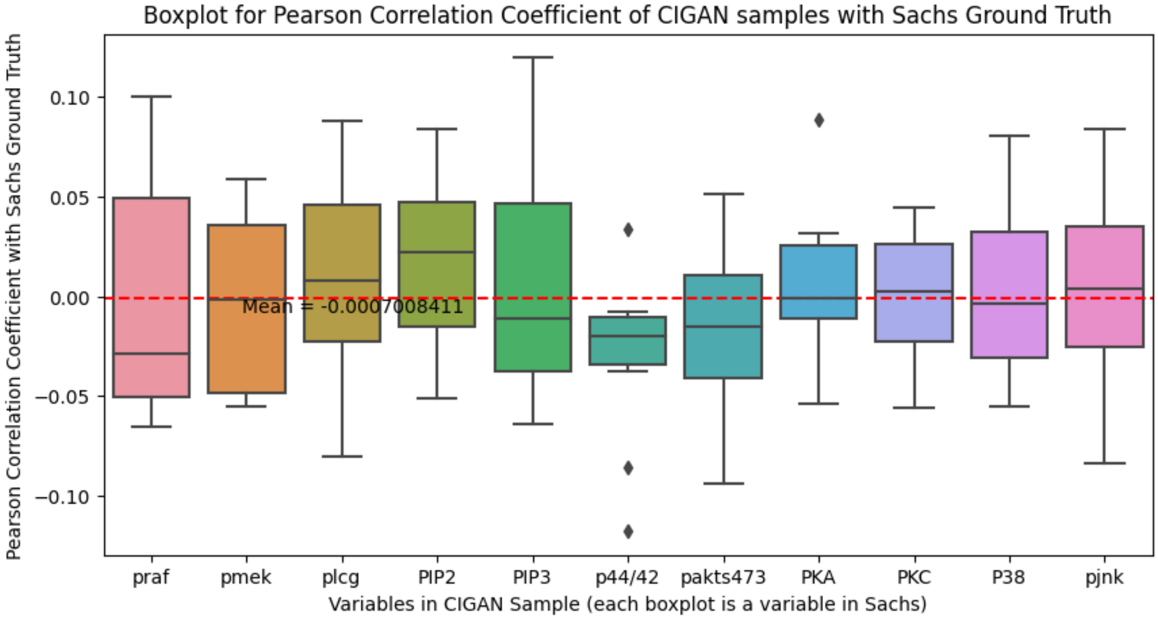


Figure 6.6: Boxplot to demonstrate the Pearson Correlation Coefficient for each variable (ie. column in the sample table) with the associated variable in the Sachs ground truth. Averaged over 10 runs of CIGAN.

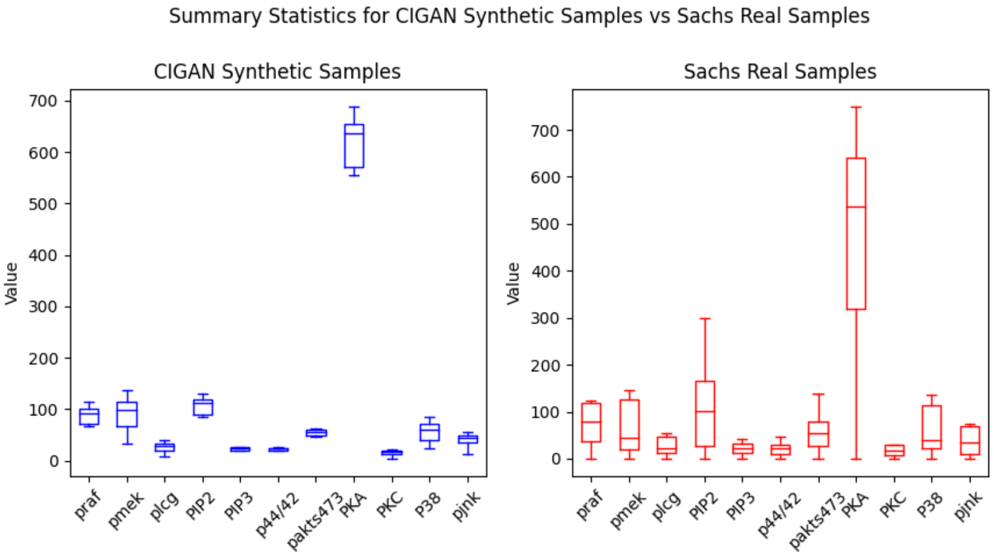


Figure 6.7: Summary statistics box-plot comparing CIGAN samples (**left**) (averaged over 10 runs), and Sachs data (**right**).

	praf	pmek	plcg	PIP2	PIP3	p44/42	pakts473	PKA	PKC	P38	pjnk
mean	102.225105	114.895897	33.426468	118.671188	24.903570	24.358616	59.515678	657.327820	18.861446	71.556313	50.380733
std	21.149761	33.097664	8.942586	18.997259	2.788921	2.305845	7.730553	46.687756	4.663389	23.248390	11.684608
min	67.241379	59.165680	16.700653	84.574295	18.837490	19.591494	46.665947	553.776978	11.968894	33.921394	31.801443
25%	86.246017	87.870049	27.147010	107.085484	22.662432	23.101337	53.890884	625.238556	15.310949	53.911802	41.933522
50%	99.164806	109.705906	31.585348	118.244991	25.316066	24.577830	59.693651	649.215271	18.595778	68.161121	47.087730
75%	113.469347	136.787739	40.258728	130.454266	26.656313	25.545380	63.368661	688.607559	21.455253	85.076891	56.299446

Figure 6.8: Summary Statistics table for **CIGAN synthetic samples** (averaged over 10 runs).

	praf	pmek	plcg	PIP2	PIP3	p44/42	pakts473	PKA	PKC	P38	pjnk
mean	124.237897	145.686468	54.933060	151.284125	27.053650	26.642470	81.271299	625.779949	30.397738	135.022087	73.419219
std	247.762082	377.403216	174.037529	299.633083	43.090940	45.865959	137.891020	644.874579	92.960564	495.053705	215.867190
min	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
25%	30.800000	16.700000	9.390000	18.300000	9.560000	8.510000	23.300000	276.000000	4.530000	19.300000	8.060000
50%	53.800000	26.700000	16.500000	53.300000	17.800000	17.200000	37.200000	449.000000	12.900000	30.500000	18.400000
75%	103.000000	64.400000	27.100000	172.000000	32.800000	32.200000	72.300000	750.000000	23.500000	49.600000	52.800000

Figure 6.9: Summary Statistics table for **Sachs data**.

	praf	pmek	plcg	PIP2	PIP3	p44/42	pakts473	PKA	PKC	P38	pjnk
0	58.153183	67.452667	5.515162	20.526722	14.244808	8.271784	6.851221	126.871246	6.447596	7.948894	11.018434
1	116.401825	70.251732	7.684680	16.912500	20.640703	12.415327	11.305522	797.578857	28.186655	42.559166	6.130466
2	17.173473	31.454037	3.977929	3.796919	-1.575432	22.066626	30.442183	383.120178	8.997263	10.030116	25.172325
3	1266.352539	1842.651611	22.891045	69.898560	24.977407	19.484884	74.780731	283.381439	-1.215165	66.468628	4.772189
4	60.603214	36.644024	4.333175	274.450226	78.493820	12.445873	15.239491	495.064362	12.037351	9.338633	28.963636
5	39.277664	55.828648	-1.045167	6.621567	1.381955	21.989290	59.827206	322.228180	12.422595	17.270878	74.974678
6	32.930721	30.935038	159.734406	555.465332	72.958954	41.507084	52.150665	1249.775513	37.853630	115.444824	135.507294
7	12.294005	60.186028	49.826599	60.091858	0.008515	15.019249	168.014008	7.937132	60.229549	348.914886	56.175076
8	130.957932	78.786812	8.495017	14.782521	27.720823	10.052710	10.336218	925.606995	45.020969	73.697563	15.632478
9	44.450436	26.200974	-0.279134	263.789642	37.819302	81.275932	136.528366	1641.369019	7.568213	19.096899	13.530834

Figure 6.10: A portion of a CIGAN output sample trained on Sachs.

	praf	pmek	plcg	PIP2	PIP3	p44/42	pakts473	PKA	PKC	P38	pjnk
0	26.400000	13.200000	8.820000	18.300000	58.800000	6.610000	17.000000	414.000000	17.000000	44.900000	40.000000
1	35.900000	16.500000	12.300000	16.800000	8.130000	18.600000	32.500000	352.000000	3.370000	16.500000	61.500000
2	59.400000	44.100000	14.600000	10.200000	13.000000	14.900000	32.500000	403.000000	11.400000	31.900000	19.500000
3	73.000000	82.800000	23.100000	13.500000	1.290000	5.830000	11.800000	528.000000	13.700000	28.600000	23.100000
4	33.700000	19.800000	5.190000	9.730000	24.800000	21.100000	46.100000	305.000000	4.660000	25.700000	81.300000
5	18.800000	3.750000	17.600000	22.100000	10.900000	11.900000	25.700000	610.000000	13.700000	49.100000	57.800000
6	44.900000	36.500000	10.400000	132.000000	16.300000	8.660000	17.900000	835.000000	15.000000	35.900000	18.100000
7	47.400000	15.000000	14.600000	30.500000	17.500000	20.200000	45.300000	466.000000	6.440000	24.400000	20.000000
8	104.000000	61.500000	10.600000	21.100000	41.800000	11.500000	23.500000	445.000000	29.200000	61.000000	25.300000
9	21.100000	21.500000	1.880000	205.000000	43.700000	13.200000	135.000000	213.000000	14.600000	26.700000	101.000000

Figure 6.11: The first 10 rows of the Sachs dataset.

Chapter 7

Discussion

The previous chapter presented the model's results evaluated against a collection of metrics to measure the success of the model in relation to the goal of the project: to develop a novel data augmentation tool for the Causal Discovery domain. We discuss the evolution of the project, and the success of the model proposed (CIGAN) based on the prior chapter.

7.1 Discussion of Results

Results for CIGAN (T_{synth})		
Metric Name	Target Range	Result
varsortability	0.5 ± 0.1	0.49
Standardised Comparison	$CD(T^{raw}_{synth}) \approx CD(T^{standardised}_{synth})$	$CD(T^{raw}_{synth)}) > CD(T^{standardised}_{synth})$
Kolmogorov-Smirnov	$KS_p(T_{synth}, T) > 0.05$	50%*
Pearson Correlation	$PC(T_{synth}, T) = 0 \pm 0.1$	0.00070
ML efficacy	$ML_e(T_{artificialgraphs}) < ML_e(T_{synth}) < ML_e(T)$	$ML_e(T_{artificialgraphs}) < ML_e(T_{synth}) < ML_e(T)$

Figure 7.1: Quantified targets for CIGAN. $CD(T)$ refers to the score of the Causal Discovery method ran on table T (measured with Structural Hamming Distance). *: 50% of the variables in a CIGAN sample were found to exceed the p-value of 0.05, meaning that they were said to have been from the same distribution as the Sachs variables.

We find that CIGAN succeeded with meeting 3/4 targets set for it in the Requirements and Analysis chapter, thus the project has been successful in developing a technique that can be useful for development of future CD methods. The standardised comparison did not give the expected results for a success, nor the result expected for a failure. We found that the standardised data performed better than the raw, which is a counter-intuitive finding as any transformation to the data will likely result in causal associations being disturbed (making

CD harder). However, this test was only ran once, with 1 sample, so it may be an anomaly. and inferring any further details about the behaviour of the model would be improper. The remaining metrics were however tested with multiple samples, thus findings from these are more reliable.

7.2 Limitations

Although the testing revealed promising results, the model was only evaluated against one dataset: Sachs. Due to the lack of real-world datasets in this area, and the time constraints of the project, it was not possible to evaluate among other datasets. Furthermore, the model's generation capability is currently limited to the number of rows being set by the 'batch size' hyper-parameter, due to a quirk of the model implementation. As the model was only needed for tests as a proof-of-concept, and not a fully developed system, using a batch size of 50, for 50 rows, was sufficient to test with. At the time of writing, there are few (if any) CD methods that need multi-domain training data, so the utility of CIGAN in the short-term is merely hypothetical, outside of using the model to test existing techniques.

7.3 Future Research

During the testing phase of the project, we identified some additional datasets found in [Chen et al., 2022] that could have been used for more thorough testing of CIGAN, and which would be useful if further work was to be done on the model. To improve the performance of the model, different architectures for the generator could be experimented with, as the architecture used in CIGAN currently is a rather standard auto-encoder design, as a proof-of-concept, and there are likely performance gains to be made. Additionally, to aid to development of the data generation technique, a unified metric specific to causal data augmentation could be devised, which would give a clearer picture of the models success. A further area of research which could be prudent to developing more capable CD systems is data privacy, as GANs naturally give some weak privacy guarantees [Lin et al., 2022], but using this technique for popular CD research areas such as healthcare would require more robust data privacy.

Chapter 8

Conclusions

The aim of this project was to research and identify issues with current data generation and augmentation techniques in the field of Causal Discovery, and to propose a novel solution. First, we surveyed the area of Causal Discovery and its prerequisites, and the best performing papers in the field. We then evaluated one of these methods (DAG-GNN) against a comprehensive collection of datasets, and identified issues with with synthetic data's lack of complexity. Building on research from [Reisach et al., 2021], which found that some classes of CD methods could cheat on some synthetic data set-ups, we identified the issues with the current class of synthetic data generation techniques for Causal Discovery methods. To solve this, we proposed CIGAN, a GAN-based model, and suggested that a model learning the distribution of the target data is conducive to the model learning causal relations implicitly (given sufficient complexity). Using a number of metrics designed to evaluate CIGAN's utility, we found that the model was successful at fulfilling the requirements set out in chapter 4. The results illustrated the characteristics of the model, and identified an oddity with standardisation of the data, which will be pinned for further research.

Bibliography

[Adhikari, 2020] Adhikari, S. (2020). *Causal Structure Learning when Data is not Independent and Identically Distributed (non-IID)*. PhD thesis.

[Arora and Zhang, 2017] Arora, S. and Zhang, Y. (2017). Do gans actually learn the distribution? an empirical study.

[Brulé, 2018] Brulé, J. (2018). Causal programming: inference with structural causal models as finding instances of a relation. *ArXiv*, abs/1805.01960.

[Cao, 2013] Cao, L. (2013). Non-iidness learning in behavioral and social data. *The Computer Journal*, 57:1358–1370.

[Charig et al., 1986] Charig, C. R., Webb, D. R., Payne, S. R., and Wickham, J. E. (1986). Comparison of treatment of renal calculi by open surgery, percutaneous nephrolithotomy, and extracorporeal shockwave lithotripsy. *BMJ*, 292(6524):879–882.

[Chen et al., 2022] Chen, H., Du, K., Yang, X., and Li, C. (2022). A review and roadmap of deep learning causal discovery in different variable paradigms.

[Clauset, 2011] Clauset, A. (2011). A brief primer on probability distributions.

[Dimitrova et al., 2020] Dimitrova, D. S., Kaishev, V. K., and Tan, S. (2020). Computing the kolmogorov-smirnov distribution when the underlying cdf is purely discrete, mixed, or continuous. *Journal of Statistical Software*, 95(10):1–42.

[Ding et al., 2021] Ding, C., Huang, B., Gong, M., Zhang, K., Liu, T., and Tao, D. (2021). Score-based causal discovery from heterogeneous data.

[dos Santos Tanaka and Aranha, 2019] dos Santos Tanaka, F. H. K. and Aranha, C. (2019). Data augmentation using gans. *CoRR*, abs/1904.09135.

[Freedman et al., 2007] Freedman, D., Pisani, R., and Purves, R. (2007). Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

[Frid-Adar et al., 2018] Frid-Adar, M., Klang, E., Amitai, M., Goldberger, J., and Greenspan, H. (2018). Synthetic data augmentation using GAN for improved liver lesion classification. *CoRR*, abs/1801.02385.

[Friston et al., 2012] Friston, K., Moran, R., and Seth, A. (2012). Analysing connectivity with granger causality and dynamic causal modelling. *Current opinion in neurobiology*, 23.

[Gagniuc, 2017] Gagniuc, P. (2017). *Markov Chains: From Theory to Implementation and Experimentation.*

[GEIGER and PEARL, 1990] GEIGER, D. and PEARL, J. (1990). On the logic of causal models* *this work was partially supported by the national science foundation grants iri-8610155, “graphoids: A computer representation for dependencies and relevance in automated reasoning,” and iri-8821444, “probabilistic networks for automated reasoning.”. In SHACHTER, R. D., LEVITT, T. S., KANAL, L. N., and LEMMER, J. F., editors, *Uncertainty in Artificial Intelligence*, volume 9 of *Machine Intelligence and Pattern Recognition*, pages 3–14. North-Holland.

[Ghojogh et al., 2021] Ghojogh, B., Ghodsi, A., Karray, F., and Crowley, M. (2021). Generative adversarial networks and adversarial autoencoders: Tutorial and survey. *CoRR*, abs/2111.13282.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[Granger, 1969] Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-Spectral Methods. *Econometrica*, 37(3):424–438.

[He et al., 2015] He, Y., Jia, J., and Yu, B. (2015). Counting and exploring sizes of markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 16(79):2589–2609.

[Hornik et al., 1989] Hornik, K., Stinchcombe, M., and White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366.

[Kalainathan and Goudet, 2019] Kalainathan, D. and Goudet, O. (2019). Causal discovery toolbox: Uncover causal relationships in python.

[Kashif et al., 2020] Kashif, M., Singh, S. K., Thiyagarajan, S., and Maheshwari, A. (2020). Linear and nonlinear causal relationships between international reserves and economic growth: Evidence from india. *Asia-Pacific Journal of Management Research and Innovation*, 16(1):54–59.

[Kearns et al., 1994] Kearns, M., Mansour, Y., Ron, D., Rubinfeld, R., Schapire, R. E., and Sellie, L. (1994). On the learnability of discrete distributions. In *Symposium on the Theory of Computing.*

[Kilian, 2011] Kilian, L. (2011). Structural vector autoregressions. *Economics 2013*, (8515).

[Kramer, 1991] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, 37(2):233–243.

[Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79 – 86.

[Lawrence et al., 2021] Lawrence, A. R., Kaiser, M., Sampaio, R., and Sipos, M. (2021). Data generating process to evaluate causal discovery techniques for time series data.

[Le et al., 2015] Le, T. D., Hoang, T., Li, J., Liu, L., and Liu, H. (2015). A fast PC algorithm for high dimensional causal discovery with multi-core pcs. *CoRR*, abs/1502.02454.

[Lin et al., 2022] Lin, Z., Sekar, V., and Fanti, G. (2022). On the privacy properties of gan-generated samples.

[Maziarz, 2015] Maziarz, M. (2015). A review of the Granger-causality fallacy. *The Journal of Philosophical Economics*, 8(2).

[Mooij et al., 2015] Mooij, J. M., Peters, J., Janzing, D., Zscheischler, J., and Schölkopf, B. (2015). Distinguishing cause from effect using observational data: methods and benchmarks.

[Nikolaou and Sechidis, 2020] Nikolaou, N. and Sechidis, K. (2020). Inferring causal direction from observational data: A complexity approach. *CoRR*, abs/2010.05635.

[Nogueira et al., 2022] Nogueira, A. R., Pugnana, A., Ruggieri, S., Pedreschi, D., and Gama, J. (2022). Methods and tools for causal discovery and causal inference. *WIREs Data Mining and Knowledge Discovery*, 12(2):e1449.

[Parascandolo et al., 2017] Parascandolo, G., Rojas-Carulla, M., Kilbertus, N., and Schölkopf, B. (2017). Learning independent causal mechanisms. *CoRR*, abs/1712.00961.

[Pearl, 2009a] Pearl, J. (2009a). *The Art and Science of Cause and Effect*, page 401–428. Cambridge University Press, 2 edition.

[Pearl, 2009b] Pearl, J. (2009b). *Causality: Models, Reasoning and Inference.* Cambridge University Press, USA, 2nd edition.

[Pearl, 2010] Pearl, J. (2010). An introduction to causal inference. *The International Journal of Biostatistics*, 6(2).

[Pearl, 2014] Pearl, J. (2014). *Understanding Simpson's Paradox.* The American Statistician.

[Pearl, 2016] Pearl, J. (2016). *Causal inference in statistics : a primer.* Wiley, Chichester, West Sussex.

[Peters and Bühlmann, 2014] Peters, J. and Bühlmann, P. (2014). Structural intervention distance (sid) for evaluating causal graphs.

[Petkov et al., 2022] Petkov, H., Hanley, C., and Dong, F. (2022). DAG-WGAN: Causal structure learning with wasserstein generative adversarial networks. In *Embedded Systems and Applications*. Academy and Industry Research Collaboration Center (AIRCC).

[Ramsey et al., 2017] Ramsey, J., Glymour, M., sanchez romero, R., and Glymour, C. (2017). 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*, 3.

[Reichenbach, 1956] Reichenbach, H. (1956). *The Direction of Time*. Dover Publications.

[Reisach et al., 2021] Reisach, A. G., Seiler, C., and Weichwald, S. (2021). Beware of the simulated dag! causal discovery benchmarks may be easy to game.

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

[Sadeghi and Soo, 2022] Sadeghi, K. and Soo, T. (2022). Conditions and assumptions for constraint-based causal structure learning.

[Schölkopf, 2019] Schölkopf, B. (2019). Causality for machine learning. *CoRR*, abs/1911.10500.

[Schölkopf et al., 2021] Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. (2021). Towards causal representation learning. *CoRR*, abs/2102.11107.

[Scutari,] Scutari, M. bnlearn- bayesian network repository.

[Shojaie and Fox, 2021] Shojaie, A. and Fox, E. (2021). Granger causality: A review and recent advances.

[Sprenger and Weinberger, 2021] Sprenger, J. and Weinberger, N. (2021). Simpson's Paradox. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2021 edition.

[Tennant et al., 2019] Tennant, P., Harrison, W., Murray, E., Arnold, K., Berrie, L., Fox, M., Gadd, S., Keeble, C., Ranker, L., Textor, J., Tomova, G., Gilthorpe, M., and Ellison, G. (2019). Use of directed acyclic graphs (dags) in applied health research: Review and recommendations.

[TU, 2023] TU, R. (2023). *A Further Step of Causal Discovery*. PhD thesis.

[van 't Hof, 2018] van 't Hof, S. (2018). Systems thinking and the nature of reality, wicked solutions.

[Vonk et al., 2022] Vonk, M. C., Malekovic, N., Bäck, T., and Kononova, A. (2022). Disentangling causality: assumptions in causal discovery and inference.

[Vowels et al., 2021] Vowels, M. J., Camgöz, N. C., and Bowden, R. (2021). D'ya like dags? A survey on structure learning and causal discovery. *CoRR*, abs/2103.02582.

[Wang et al., 2022] Wang, R., Yi, M., Chen, Z., and Zhu, S. (2022). Out-of-distribution generalization with causal invariant transformations.

[Wen et al., 2021] Wen, B., Colon, L. O., Subbalakshmi, K. P., and Chandramouli, R. (2021). Causal-tgan: Generating tabular data using causal generative adversarial networks. *CoRR*, abs/2104.10680.

[White et al., 2010] White, H., Chalak, K., and Lu, X. (2010). Linking granger causality and the pearl causal model with settable systems. *Boston College Department of Economics, Boston College Working Papers in Economics*.

[Xu and Veeramachaneni, 2018a] Xu, L. and Veeramachaneni, K. (2018a). Synthesizing tabular data using generative adversarial networks. *CoRR*, abs/1811.11264.

[Xu and Veeramachaneni, 2018b] Xu, L. and Veeramachaneni, K. (2018b). Synthesizing tabular data using generative adversarial networks.

[Yu et al., 2019] Yu, Y., Chen, J., Gao, T., and Yu, M. (2019). Dag-gnn: Dag structure learning with graph neural networks.

[Yu et al., 2021] Yu, Y., Gao, T., Yin, N., and Ji, Q. (2021). Dags with no curl: An efficient DAG structure learning approach. *CoRR*, abs/2106.07197.

[Zhang et al., 2018] Zhang, Z., Li, M., and Yu, J. (2018). On the convergence and mode collapse of gan. In *SIGGRAPH Asia 2018 Technical Briefs*, SA '18, New York, NY, USA. Association for Computing Machinery.

[Zheng et al., 2018] Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning.

[Özkaya and Çatalyürek, 2022] Özkaya, M. Y. and Çatalyürek, V. (2022). A simple and elegant mathematical formulation for the acyclic dag partitioning problem.

Appendices

Appendix A

CIGAN code

Random DAG Generator

```
    def generate_adj_A(data_dim):
        nodes = data_dim
        graph = nx.erdos_renyi_graph(nodes, 0.25, directed=True) #erdos renyi graph
        adj_mat_np = nx.adjacency_matrix(graph).todense()
        adj_mat = torch.from_numpy(adj_mat_np).to(torch.float32)
        #save adj_mat to csv
        adj_mat_np = adj_mat_np.astype(int)
        timestamp = datetime.datetime.now()
        timestamp = str(timestamp.strftime("%Y%m%d%H%M%S"))
        np.savetxt(f'adj_mat_{timestamp}.csv', adj_mat_np, delimiter=',', fmt='%d')
        return adj_mat
```

Generator

```
class Generator(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super(Generator, self).__init__()

        self.adj_A = generate_adj_A(output_dim)

        # Encoder layers
        self.encoder = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.ReLU(True),
            nn.Linear(hidden_dim, hidden_dim),
            nn.ReLU(True),
            nn.Linear(hidden_dim, hidden_dim),
            nn.ReLU(True)
        )

        # Decoder layers
        self.decoder = nn.Sequential(
            nn.Linear(hidden_dim, hidden_dim),
            nn.ReLU(True),
            nn.Linear(hidden_dim, hidden_dim),
            nn.ReLU(True),
            nn.Linear(hidden_dim, output_dim),
            nn.Identity()
        )

        self.resize_to_adj_size = nn.Linear(input_dim, output_dim)
        self.resize_x_back = nn.Linear(output_dim, input_dim)

    def forward(self, x):

        resize_x_for_adj = self.resize_to_adj_size(x)
        x2 = torch.matmul(self.adj_A, resize_x_for_adj.T)
        x = self.resize_x_back(x2.T)
        x = self.encoder(x)
        # logits = torch.matmul(self.adj_A, x)
        x = self.decoder(x)
        return x
```

Discriminator

```
class Discriminator(nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):
        super(Discriminator, self).__init__()

        # Discriminator layers
        self.layers = nn.Sequential(
            nn.Linear(input_dim, hidden_dim),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(hidden_dim, hidden_dim),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(hidden_dim, hidden_dim),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(hidden_dim, output_dim),
            nn.Sigmoid()
        )
    def forward(self, x):
        x = self.layers(x)
        return x
```

Training

```
loss_fn = nn.BCELoss()

#-=-=-=-= training =-=-=-=-=

for epoch in range(epochs):
    for idx, real_data in enumerate(train_dev_loader):

        noise = torch.randn(batch_size,latent_dim)
        fake_data = generator(noise)
        #- Train Discriminator

        discriminator_optimizer.zero_grad()

        real_labels = torch.ones(batch_size, 1)
        fake_labels = torch.zeros(batch_size, 1)
        real_output = discriminator(real_data)
        fake_output = discriminator(fake_data.detach())
        real_loss = loss_fn(real_output, real_labels)
        fake_loss = loss_fn(fake_output, fake_labels)
        discriminator_loss = real_loss + fake_loss

        discriminator_loss.backward()
        discriminator_optimizer.step()

        #- Train Generator

        generator_optimizer.zero_grad()

        noise = torch.randn(batch_size, latent_dim)
        fake_data = generator(noise)

        fake_labels = torch.ones(batch_size, 1)
        fake_output = discriminator(fake_data)
        generator_loss = loss_fn(fake_output, fake_labels)
        # !!! could do ELBO loss here?????????
        generator_loss.backward()
        generator_optimizer.step()

        #-=- output info and save
        batches_done = epoch * len(train_dev_loader) + idx
        if batches_done % sample_interval == 0:
                # save_image(fake_data.data, "images/%d.png" % batches_done, normalize=True)
                timestamp = datetime.datetime.now()
                timestamp = str(timestamp.strftime("%Y%m%d%H%M%S"))
                df = pd.DataFrame(fake_data.detach().numpy(), columns=dataset.get_col_names(
                df.to_csv((f'images/%d_{timestamp}.csv' % batches_done), index=False)
                print("saved a table!")
    print("Epoch %d: Generator loss=%.4f, Discriminator loss=%.4f" % (epoch+1, generator_los
```

Appendix B

Results from CIGAN

B.0.1 Output Sample

	praf	pmek	plcg	PIP2	PIP3	p44/42	pakts473	PKA	PKC	P38	pjnk
0	58.153183	67.452667	5.515162	20.526722	14.244808	8.271784	6.851221	126.871246	6.447596	7.948894	11.018434
1	116.401825	70.251732	7.684680	16.912500	20.640703	12.415327	11.305522	797.578857	28.186655	42.559166	6.130466
2	17.173473	31.454037	3.977929	3.796919	-1.575432	22.066626	30.442183	383.120178	8.997263	10.030116	25.172325
3	1266.352539	1842.651611	22.891045	69.898560	24.977407	19.484884	74.780731	283.381439	-1.215165	66.468628	4.772189
4	60.603214	36.644024	4.333175	274.450226	78.493820	12.445873	15.239491	495.064362	12.037351	9.338633	28.963636
5	39.277664	55.828648	-1.045167	6.621567	1.381955	21.989290	59.827206	322.228180	12.422595	17.270878	74.974678
6	32.930721	30.935038	159.734406	555.465332	72.958954	41.507084	52.150665	1249.775513	37.853630	115.444824	135.507294
7	12.294005	60.186028	49.826599	60.091858	0.008515	15.019249	168.014008	7.937132	60.229549	348.914886	56.175076
8	130.957932	78.786812	8.495017	14.782521	27.720823	10.052710	10.336218	925.606995	45.020969	73.697563	15.632478
9	44.450436	26.200974	-0.279134	263.789642	37.819302	81.275932	136.528366	1641.369019	7.568213	19.096899	13.530834
10	57.391563	127.123283	394.144867	640.444885	2.783427	7.896554	143.219650	30.375118	70.237816	408.545898	171.688782
11	11.325523	42.388367	24.017496	15.764098	48.456398	3.868731	9.725658	401.449432	36.552578	54.804405	5.535397
12	8.959945	18.718325	7.980759	4.537305	-1.915410	20.251642	26.645445	318.853271	4.879082	2.072287	12.256055
13	63.012600	91.331367	8.773459	84.096809	14.822381	11.885002	16.231005	200.537674	16.218569	32.970097	25.283081
14	22.695868	33.415524	44.520718	328.502838	33.770123	31.624298	42.295582	931.715210	19.040026	56.522884	70.442085
15	24.366405	28.056707	4.387264	31.659410	7.037630	11.195791	15.050982	791.913757	16.921206	26.808399	16.295914
16	22.773685	60.586315	22.576332	16.592178	33.863811	25.168171	34.946926	607.068054	74.347084	100.963333	14.137516
17	37.920124	35.910744	4.440981	5.565658	14.358774	8.992680	15.414416	268.059357	55.540955	76.341133	18.005930
18	90.573563	52.540005	-1.722556	81.222771	21.614901	17.536242	21.927534	1073.321777	4.910804	30.653122	137.591980
19	30.469788	22.851442	8.014879	89.944893	48.054779	31.744471	31.251318	1059.023193	8.303728	7.252522	2.465782
20	33.558403	41.144581	3.399037	9.209356	1.342391	18.108913	44.267078	252.251007	8.403893	8.125826	39.589657
21	9.752665	36.098755	0.550584	6.871666	2.157311	25.533432	71.907463	236.231628	9.959721	9.024755	91.702721
22	28.742561	20.105495	3.734645	32.791534	23.027428	22.429853	26.104321	368.375183	7.387754	11.085650	1.983058
23	95.966354	38.672134	0.392444	167.337387	39.013760	4.606812	11.299136	568.854980	5.824029	9.119923	4.452228
24	20.507439	27.278917	6.393077	14.944275	1.743591	9.349148	7.804640	665.828430	24.913202	33.989605	15.146188
25	-2.282738	38.996624	19.915230	8.933353	-0.589643	62.587292	89.359261	654.223206	30.426506	26.853251	53.013790
26	15.520286	48.917171	16.547428	64.959381	11.674776	94.492889	128.779129	2667.290527	9.451260	30.099630	93.961838
27	15.165617	28.841579	79.873161	454.170471	61.654510	35.150238	45.484264	1348.295532	0.753684	18.701670	22.113720
28	81.915321	60.531879	-1.416908	138.278275	18.391884	15.543911	21.762175	872.110107	11.920016	29.459587	64.451820
29	93.984360	45.580177	1.659569	319.801544	53.519356	49.634411	84.196915	1342.030029	8.835462	15.387548	7.630459
30	23.255253	19.627373	7.005527	11.123419	5.510391	15.622953	17.392031	281.838806	2.922947	3.770402	5.067096
31	16.348202	15.097570	3.746908	6.620893	3.124835	10.735710	12.401748	208.862457	2.490445	3.595455	4.716754
32	19.696600	16.749685	1.208852	13.312823	8.295304	9.544537	13.335415	411.801971	11.434542	15.902298	9.117769
33	129.609390	85.960098	0.109493	140.295685	28.936661	27.682686	47.775475	1449.902344	9.771481	31.019161	74.529770
34	20.024092	35.190674	-1.971965	0.744251	6.237179	13.007574	46.312996	161.663239	9.996960	10.515286	65.764786
35	31.953039	25.773472	5.432897	115.408241	19.263624	17.811663	32.607838	1065.062744	7.335739	19.425009	23.887098
36	40.758205	27.482637	13.628500	74.251602	63.421463	41.987633	44.195217	1459.207642	15.837298	13.483350	3.339474
37	22.299166	32.540539	23.640533	16.404629	-2.068664	39.117378	50.519775	730.237854	9.665836	6.730562	15.642346
38	27.732929	29.117907	23.300543	19.916288	5.316463	21.808908	27.311661	657.066223	8.152322	12.752274	2.516130
39	18.454403	48.186275	11.119994	51.953156	10.280031	90.376572	129.934921	2494.800293	9.900083	27.609179	78.338402
40	53.877274	51.169590	0.466492	5.146767	19.646305	12.622629	24.088474	388.528015	21.967093	26.316828	16.476931
41	17.785412	26.424793	32.044628	21.165344	42.519981	2.120649	6.580134	269.298889	15.286672	43.659122	-1.369634
42	47.257835	38.837593	20.047289	115.531479	93.066193	37.873539	44.400730	1046.201294	11.215018	17.731476	2.215217
43	1501.023804	2230.481689	25.079088	71.892380	22.621408	17.749630	59.225605	225.110016	-0.691622	69.134003	2.174077
44	11.157723	42.825687	6.200476	5.488354	22.147131	33.136196	64.487526	341.610626	18.879202	15.634986	73.980324
45	22.367283	15.443619	8.200563	12.511852	7.191856	16.337725	16.557188	253.102753	2.307259	2.832845	3.300633
46	28.933092	28.652618	8.598902	15.867233	10.981715	21.610868	27.982229	365.848907	7.425236	7.768867	9.001924
47	23.991640	32.631554	12.069735	11.453337	2.273368	5.005802	2.075872	1033.856567	25.787872	43.473217	10.987016
48	23.930582	19.180449	7.296494	13.549949	6.501281	20.334988	21.034101	335.125732	2.945505	4.448449	4.619552
49	30.951477	26.680361	6.421465	13.858149	9.647182	13.525999	17.836353	260.708130	8.503348	8.185974	6.306635

Figure B.1: CIGAN's output with 1 sample, when batch size 50 is used (ie: 50 rows)