Learning Stable Causal Explanations from Deep Neural Networks

Sara Pernille Jensen

June 7, 2024
Github repository: `https://github.com/SaraPJensen/FYS9429`

Abstract

In this paper, it was investigated whether machine learning models trained on interventional data, rather than purely observational data, are more reliable and robust to changes in the data generating functions. Further, two methods from explainable AI were used to generate explanations of the different models, and their stability was evaluated as a function of the types of datasets used to train the original ML model. It was found that training models on interventional data greatly enhanced their robustness, which was attributed to their having learned the underlying causal structure of the target system. Additionally, a potential method for learning the causal structure when only observational data is available for training was found. The results also suggested a positive relationship between the robustness of the models and the variation in their corresponding explanations, but more research is needed to confirm this conclusively.

Contents

1 Introduction **1**

2 Theory **2**
2.1 Machine learning . . . 2
2.1.1 Statistical Learning Theory . . . 2
2.1.2 Deep Neural Networks . . . 3
2.1.3 Model Performance . . . 3
2.1.4 Black Box Modelling . . . 4
2.2 Explainable AI . . . 5
2.2.1 Prediction without Explanation . . . 5
2.2.2 Methods and types of explanations . . . 6
2.2.3 Common Criticisms . . . 7
2.3 Correlations and Causality . . . 8
2.3.1 Structural Causal Models . . . 8
2.3.2 Learning Causal Models . . . 10

3 Method **11**
3.1 Datasets . . . 11
3.1.1 Training datasets . . . 12
3.1.2 Testing datasets . . . 13
3.2 Network . . . 14
3.3 XAI Methods . . . 14
3.3.1 Explanations and their variance . . . 15
3.4 Limitations . . . 17

4 Results and Discussion **17**
4.1 Performance . . . 17
4.1.1 Simple Dataset . . . 17
4.1.2 Complex Dataset . . . 19
4.1.3 Discussion . . . 20
4.2 Variance . . . 21
4.2.1 Simple Dataset . . . 21
4.2.2 Complex Dataset . . . 23
4.2.3 Discussion . . . 23

5 Conclusion and Outlook **25**
5.1 Conclusion . . . 25
5.2 Limitations . . . 25
5.3 Future Research . . . 25

References **26**

1 Introduction

The use of machine learning is becoming more and more widespread as a method for data analysis in many fields, including science. Some of the main reasons for this is the high predictive power of many of these methods, their speed and cost-efficiency, as well as an increasing availability of large amounts of observational data, which is what is usually used to train such models. However, there is also an increasing awareness about the potential drawbacks of replacing more traditional methods of scientific investigation and data analysis with machine learning algorithms.

Some of the main criticisms regards their inability to provide understanding of the systems they model, due to their 'black-box' nature [1, 2], as well as their lack of robustness and ability to generalise beyond the domain on which they were trained [3, 4]. In response to the former, *explainable AI* (XAI) is an emerging sub-field in ML research, where the aim is to provide a set of methods for explaining what the trained ML model has learned about the target system. Yet, this field has been met with a set of criticisms on its own, especially regarding the potential heterogeneity of explanations provided of the same model, making them untrustworthy [3, 2]. In the context of this paper, we assume that the ultimate goal is to understand the target system from which the data is gathered. The problem then is twofold. The typical ML models differ from traditional scientific models in that they do not generalise beyond their training domain, and when we try to understand what they do using methods from XAI, we sometimes get conflicting explanations.

The main thesis of this paper is that these issues are not necessarily due to the machine learning algorithms as such, but rather a result of the types of datasets used to train them. Specifically, the datasets are typically *observational* and *static*, two features which are known to hinder scientific learning [4]. Importantly, this is nothing new, and in more traditional scientific approaches, we rarely use such datasets. Thus, although machine learning as such constitutes a novel set of methods in science, some of the problems encountered in their use are traditional problems, in need of traditional solutions.

Firstly, the data is almost always observational, rather than interventional. This is often due to the ease of access of large amounts of such data, which is a requisite for training good ML models. Yet, in order to get stable, reliable and robust models, we need the models to be based on the underlying causal structure of the system, rather than just its correlations. As will be elaborated on in 2.3, interventions are the most reliable way of gaining causal information about a system. Indeed, the causal information is often not even present in the observational data, such that there is no possible way for the network to learn the causal structure from this alone. By including interventions in the data gathering process, more of the causal information will be present in the datasets, making it at least possible for the model to learn it. The aim is then that by including interventional data, the ML model has a better chance of learning the correct causal model, making them more robust and generalisable.

Secondly, the datasets are usually assumed to be static, not dynamic. In traditional science, one rarely assumes that a group of researchers work with the same dataset from beginning to end. Instead, they might start with one dataset, make some models based on this, test the models, and then adjust the dataset

accordingly. Importantly, the data gathering and analysis is an iterative and dynamic process. In machine learning, on the other hand, the datasets are usually assumed to be static, and what is adjusted are the hyperparametres or the learning algorithm, so as to optimise the model performance. This approach may teach you a lot about the machine learning algorithms, though it is doubtful how much one can learn about the target phenomenon in this way.

In this project, we also aim to test whether the lack of robustness which results from training models purely on observational data may be connected to the lack of stability in the explanations we obtain from XAI when trying to understand the models. From this, the three main hypotheses of this project are as follows:

1. Interventional data will improve the reliability and robustness of ML models.
2. If the ML model has learned the underlying causal model, methods from XAI will provide more stable and reliable explanations.
3. In cases where only observational data is available for training, it will be possible to narrow down the space of possible causal structures by making certain causal assumptions and comparing the relative performance of models trained on different combinations of the input features.

This will be tested by training a set of different ML models (neural networks) on a variety of datasets, both observational and interventional. Further, a corresponding set of explanations will be generated, and their variability and accuracy will be analysed and compared with the reliability and robustness of the models themselves.

It should be noted that both the data generating functions and the neural networks considered will be fairly simple. Of course, the state-of-the-art machine learning models for causal inference are far more complex than this. The reason for this choice of relatively simple networks and datasets is to provide a proof-of-principle result rather than to maximise the models' performance. By keeping it simple, the number of variables and potential sources of error are kept low, thereby making it easier to identify the mechanisms and important difference-makers in the datasets and networks.

The structure of the paper is as follows. The Theory section consists of an overview of the machine learning methods used, the different methods from explainable AI, some insight into the ongoing debates regarding the usefulness of these explanations, and an overview of the field of causal inference. In the section on Method, the datasets, network parameters and XAI methods used will be presented. The Results and Discussion section is split into two parts, Performance and Variance. The former looks at the relative performance and robustness of the different ML models based on the datasets they were trained on, and the latter is about the variance and other properties of the XAI explanations obtained from these models. The paper ends with a brief discussion of the results obtained and an outlook for future research in the field.

2 Theory

2.1 Machine learning

2.1.1 Statistical Learning Theory

Statistical learning theory is the underlying theory for most machine learning methods. In essence, it is the theory of how one might learn to make optimal statistical predictions based on data. Assume we have a dataset consisting of input variables $\mathbf{X} \in \mathbb{R}^F$ and output variables $\mathbf{Y} \in \mathbb{R}^M$. We assume that there is some true conditional probability distribution $P(\mathbf{Y}|\mathbf{X})$ underlying the data. If this was known, given any input $\mathbf{X}$, this conditional distribution would provide us with the best possible prediction. This is also known as the *optimal Bayes decision rule*, and it can be shown that no other decision rule will give a lower error on the predictions [5]. In the case of a deterministic system where $\mathbf{Y}$ is only dependent on $\mathbf{X}$, this would always give us the right prediction.

The problem is that in cases where the number of input- and output-features F and M, or the number of datapoints N, are large, calculating this conditional probability distribution becomes extremely computationally heavy, to the point where it is impossible in practice. This is known as the *curse of dimensionality*.

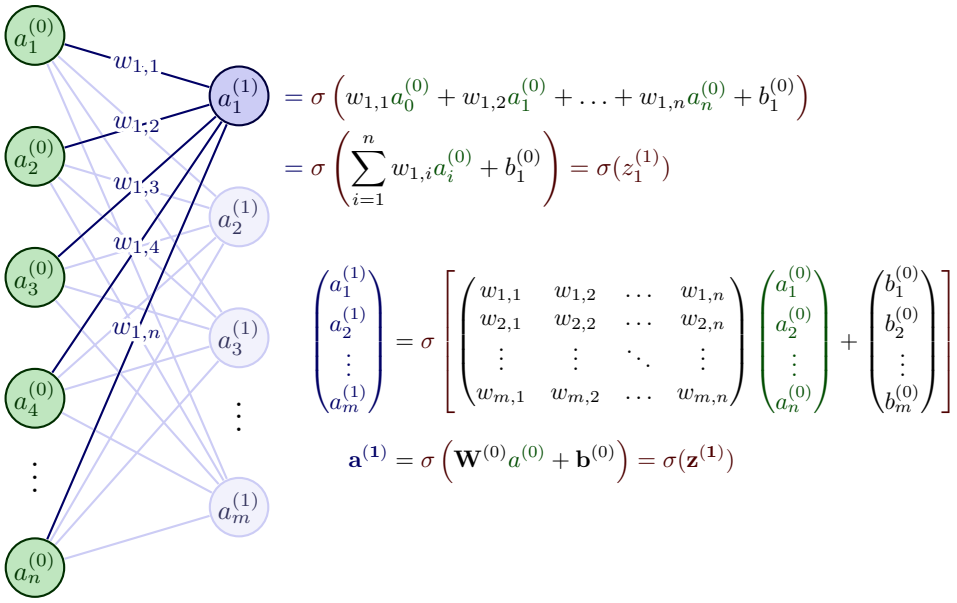

Figure 1: A feed forward neural network showing the input layer and the first hidden layer along with the corresponding calculations. Figure adapted from [7]. Here σ is the Sigmoid function.

Instead, one can find a function f_θ with parameters θ, whose predictions given $\mathbf{X}$ *approximates* the true distribution $P(\mathbf{Y}|\mathbf{X})$. The model's conditional probability distribution is $P_\theta(\mathbf{Y}|\mathbf{X})$, and the goal is that $P_\theta(\mathbf{Y}|\mathbf{X}) \to P(\mathbf{Y}|\mathbf{X})$. In practice this means that given any $\mathbf{X}$ the function produces the same output $\mathbf{Y}$ as the original dataset. Note that many different combinations of parameters θ may give rise to the same probability distribution, meaning that they are functionally equivalent in the given domain.

There are many different methods for approximating such a function f_θ, including a range of different methods from machine learning. For our current purposes, only deep neural networks will be considered, as this is likely the most popular and versatile method used.

2.1.2 Deep Neural Networks

We still assume the dataset of independent variables $\mathbf{X} \in \mathbb{R}^F$ with corresponding dependent variables $\mathbf{Y} \in \mathbb{R}^M$. To make the optimal prediction of $\mathbf{Y}$ given $\mathbf{X}$, we train a network $\mathcal{F}$ with parameters θ. For the current purposes, only a simple feed forward neural network will be considered as the network $\mathcal{F}$. Since the data considered here take continuous values, the loss function used is simple mean squared error (MSE). The trainable parameters θ are then the weights and the biases of the network. These are updated during the *back-propagation* stage, according to the gradients of the weights and biases with respect to the loss of each prediction.

As for the dataset, we assume that all the data is drawn from the same underlying probability distribution, and we split this into a training set, a validation set and a test set, where the network is only updated based on the training data. The validation data is used to pick the best model, and the test data to test the reliability of the model on unseen data. Given the finiteness of the dataset, the model is thus necessarily trained on data from a certain *domain*, as well as from a certain *distribution.*

As the mathematical theory behind such networks have been elaborately outlined in a number of reference works, I simply refer the reader to e.g., [6] for an outline and derivation of the equations used in the feed-forward and back-propagation stages. The feed forward stage is illustrated in fig. 1, with the corresponding equation.

2.1.3 Model Performance

We can identify two desirable traits of the trained machine learning model: *reliability* and *robustness*. Reliability is what is typically tested, and is done by testing the performance of the model on the test-data coming

from the same domain and probability distribution as the model was trained on. This should be reasonably close to the final performance of the model on the training data. If the test error is significantly higher, this means that the model does *not* generalise to unseen data, so has not learned the correct distribution $P(\mathbf{Y}|\mathbf{X})$. Consequently, one cannot expect the model to give correct predictions on new data, so the model is unreliable. Good performance on test data is thus the bare minimum we must expect from a trained machine learning model for it to be the least bit useful. This is also what is usually tested and referred to in the research.

Another potentially desirable performance feature is the model's robustness. This describes how well the model works on different, but related, tasks. This difference may involve a slight change in domain or a change in the underlying probability distribution of the data. A robust model thus has a wider range of applicability, and must thus have learned some structure underlying both the initial and the new domain or distribution.

How different one can expect the domain to be without a reduction in the performance is of course context-dependent, and is a question which applies to the use of other models in science as well. For example, one cannot expect the model of a simple harmonic pendulum to hold in a storm. In general, ML models are less robust than what one might expect and desire [3, 8, 9].

Thus, the two metrics for performance considered here are:

Reliability Performance on unseen data taken from the same domain and distribution.

Robustness Performance on unseen data taken from either (or both) a new domain or from a different distribution.

2.1.4 Black Box Modelling

Assuming we have a reasonably reliable model, what can we learn from this other than the predictions on new data themselves? Not necessarily that much, since the only information easily available to the user are the inputs and their corresponding outputs. For this reason, such models are often called *black box models*, and are described as *opaque* [10, 1]. It is not clear for a user *why* or *how* a given output follows from a given input.

One might respond, then, that it is in theory perfectly possible to spell out the entire final equation which the model calculates. However, this does not seem to solve the problem, as discussed in [11]. Whereas traditional scientific models have mathematical terms that come with a clear physical interpretation, the equations produced by such a spelling out of the ML model would not. Or at least, it would not do so in terms of the same physical variables that we use. For example, assume we train a model to learn the mapping from amplitude A, the spring constant k, mass m and time t to height y, frequency f and period T, given by

$$\begin{aligned} y =& A\sin(\sqrt{\frac{k}{m}}t) \\ \omega =& \sqrt{\frac{k}{m}} \\ f =& \frac{\omega}{2\pi} \\ T =& \frac{1}{f} \end{aligned}$$

Assuming we have a deep network with a reasonable number of nodes in each layer, the equations obtained could consist of arbitrarily many terms, and to rediscover such simple and clearly interpretable relationships as those above would be near impossible. This is a core difference between machine learning models and other scientific models, and one of the main reasons raised for why such ML models are less conducive to understanding.

Thus, from the inputs and outputs alone, it is not possible for the user to identify which features were significant for the given output, or to give any form of explanation thereof, contrary to what a more traditional model would have provided. Indeed, instead of providing the user with information of the output's dependency relations on the different input variables, all that can be inferred from a successful ML model is a certain amount of mutual information between the complete set of input features and the outputs. Note, however, that the absence of success does not imply the absence of mutual information, since there are numerous other reasons for why an ML model may fail in learning the desired mapping. This puts machine learning in a novel epistemic category amongst scientific methods.

2.2 Explainable AI

2.2.1 Prediction without Explanation

The widespread use of machine learning models in a range of different fields, along with the heightened awareness regarding their opacity, has led to an increasing interest in establishing methods for explaining the predictions [10, 12]. Some have called for the use of more interpretable methods instead [1], but most agree that there is often a trade-off between interpretability and the accuracy and flexibility of ML models, such that the more interpretable models tend to perform worse for complex problems, which is where we need them the most [13, 9]. Instead of using more interpretable models, then, the field of explainable AI aims to establish a set of additional methods for explaining the functioning of the opaque ML models.

The range of different methods is vast, and there is some discussion regarding whether they even operate with the same notion of *explanation* [14]. We shall leave these concerns aside here, and limit the discussion and later investigations to a selected few of these methods. Before discussing these, a short note on what is meant by *explanation* and *understanding* in this context will be useful.

Explanations are often taken to be counterfactual and causal. A counterfactual explanation tells you which features would have caused a different outcome, thereby highlighting the relevant difference-makers of the system. Such explanations are usually also causal, where it is the causal difference makers that are taken to be explanatory.

In the natural sciences, understanding a phenomenon is usually associated with having a theory, or at least a model, of the system in question. Such a theory or model usually comes in the form of an equation, thereby making the dependencies between the variables clear and interpretable to the user.

Further, explanations are usually taken to be a necessary, though not sufficient, condition for having understanding. Understanding also requires a more holistic grasping of the system as a whole, not just of individual outcomes [15]. For our purposes, it will suffice to include two requirements for understanding. Firstly, we adopt Knüsel and Baumberger's model-specific adaptation of de Regt and Dieks' [15] account of theoretical understanding, where they take the ability of the model user to qualitatively anticipate model outputs without performing calculations or running a simulation as a measure of understanding. Additionally, if the user understands the model and/or the phenomenon modelled, he or she should be able to *explain* the model and/or specific outputs.

Thus, if we want to understand some natural system through the use of an ML model, it is clearly not sufficient to have a predictively accurate ML model. In addition, we need some conception of *how* the model is making these predictions. In some contexts, local explanations of individual instances may be appropriate, but in natural science, where the goal is to understand a phenomenon as a whole, global explanations which unify all occurrences of the phenomenon are clearly preferable. Thus, a single equation which accounts for all the instances of the phenomenon is usually the ideal.

Understanding the model or understanding the phenomenon? Before elaborating further on the different methods, a quick clarificatory note on the object of explanation in these cases may be of use. The main goal of methods from XAI is to provide explanations of the ML model itself. However, this model is itself often used with the intention of understanding some target phenomenon from which the data was gathered. In such cases, the ML model is used as a mediator between the phenomenon and the explanation. Thus if the model has learned what we hope it has, it should somehow represent the phenomenon truthfully, such that the explanation provided by the XAI method does not only explain the model, but also the target phenomenon itself.

2.2.2 Methods and types of explanations

There are two main types of explanations obtainable from XAI, *local* and *global*. Local explanations are explanations of individual instances, and usually provide some indication of which input features were most important for the resulting prediction. Thus, different datapoints may be accompanied by very different explanations. Typically, this could be an explanation of why a single loan application was granted or denied.

Global explanations aim to give an explanation of the network as a whole, such that the explanation should summarise and provide understanding of the behaviour of the network for all relevant datapoints. A number of different methods exist in both categories, but surprisingly few are directly applicable to scientific contexts where the aim is to discover an underlying mathematical model. Rather, many of these methods have been developed either for explaining different types of image analysis or to explain classifications of individual datapoints [16]. The two methods considered here were chosen for their applicability to the problem at hand, which is meant to mimic a typical scientific investigation from the natural sciences.

Shapley values Shapley values is a local method which provides explanations of the relative contribution of the different input features to individual outputs, first introduced in 1951 [17], but later adopted as a method of ML interpretability [18]. Since the method is meant to generalise beyond simple linear models, and does not require the underlying model to be continuous, it does not provide straightforward linear coefficients like in linear regression. Instead, for each datapoint, each feature is assigned a *Shapley value*. This value quantifies that feature's contribution to the output's deviation from its mean.

Without going into too much detail (which can be found in a number of resources, e.g., [19]), the Shapley values are calculated using methods from *coalitional game theory*, where the players are assumed to be cooperating to maximise some gain. In this case, the game is the prediction task for the given instance, the gain is the model's prediction minus the average prediction, and the players are the feature values who collaborate to recreate the correct gain. Given the input values and the ML model to be explained, the method tests different combinations (coalitions) of changes to the input features, where simultaneous changes to multiple variables are also taken into account. The Shapley values are then the average marginal contribution from each feature value across the whole set of coalitions considered.

In cases where a linear model with independent contributions from the different input variables can be assumed, it is possible to convert the Shapley values into linear coefficients for each input feature. Assuming we have an input feature F with elements f_i, where for each datapoint i we calculate the corresponding Shapley value $s_{F,i}$. We may then convert this into the corresponding estimated coefficient $c_{F,i}$ according to the relation

$$c_{F,i} = \frac{s_{F,i}}{(f_i - \overline{F})}. \tag{1}$$

This provides us with a clearly interpretable expression. Further, if we can assume that there are no discontinuities in the dependencies, such that the same coefficients can be expected for all datapoints, a global explanation in terms of these coefficients can be obtained by taking the average of all the local explanations for each variable.

Symbolic Regression Symbolic regression is a supervised learning task where the aim is to discover an analytic expression to describe the data. Compared with other methods of XAI, which often focus on local explanations and different types of saliency mapping, this method is much more analogous to the traditional goals of natural science. Here, we wish to obtain an interpretable and universal equation describing the system of interest, in terms of the chosen variables. A number of different methods for such symbolic regression have been developed, but it was here implemented using the `PySR` (Symbolic Regression for Python) python package as described in [20]. This method uses a genetic algorithm to identify the most suitable equation, thereby providing a global explanation of the entire model. This package allows the user to define a set of binary and unary mathematical operators which may be included in the model solutions, as well as certain bounds on the complexity of the solutions.

For training, `PySR` uses a dataset of inputs and corresponding predictions from the model to be explained. In very broad terms, in the first generation, an initial set of candidate solutions are provided, and their fitness is calculated according to some fitness function (MSE by default). At each generation (iteration) a random

subsample of the total population is selected, and the fittest individuals of these are then more likely to be passed on to the next generation. Before passing them on, one may apply different operations to the individuals, such as random mutations, crossover between the individuals or simplifications. This ensures diversity amongst the individuals at each generation.

There is an underlying assumption here, known as the 'building block hypothesis', which is that the different individuals are built up of smaller and separable 'building blocks' or elements [21]. The building blocks in this case are mathematical operators and constants. These building blocks must, at least partly independently of each other, provide some selective advantage to the the individual, such that they may be selected for independently of the other elements. Thus, combining the building blocks from fit individuals should lead to as fit or fitter an offspring. Through successive generations, cumulative selection ensures that the best building blocks become more widespread in the population.

After a given number of generations, or when a certain level of accuracy or complexity has been reached, the user is free to choose from a set of the "fittest equations", depending on the desired balance between accuracy and complexity.

2.2.3 Common Criticisms

There are three main criticisms commonly raised against such methods from XAI.

Firstly, they compute different functions to the original ML model, such that they are not truthful representations of the original model. In many cases the XAI algorithm provides one with a simplified function, for example linear, which is meant to represent the ML model. However, the original ML model will almost always compute a non-linear function which is far more complex, and cannot be replicated using a linear function. Thus, the "explanation-function" is necessarily different to the original function, and will sometimes produce different outputs given the same inputs. For this reason, some [2, 1] argue, the model cannot be an explanation of something which it misrepresents and disagrees with. Babic *et al.* [2] go as far as calling the understanding provided by such methods "ersatz-understanding", claiming they merely provide us with an illusion of explanations and understanding. Furthermore, there are cases where different XAI methods produce different and incompatible explanations of the same target model. This seemingly undermines their reliability, as we are left unable to decide which explanation to trust.

Secondly, some argue that the locality of the explanations provided by methods such as Shapley does not provide real understanding of the phenomenon, but only of specific instances thereof [3]. Instead of providing us with an explanation of the global properties of the model, they only describe how the model works in a small region of the feature-space. Underlying this critique is an assumption that there are some unifying features of the target phenomenon, such as a simple law of nature, which the model should have learned. Thus, they see it as a drawback that the XAI methods may produce very different explanations for datapoints which are close in feature-space, assuming that such discontinuities do not occur in nature.

Finally, in cases where the goal of the XAI method is not primarily to understand the ML model itself, but first and foremost the phenomenon modelled, it is essential that the ML model is at least reliable, and preferably also robust. We therefore need to be certain that we can trust the original model in order to justify extrapolating the explanation provided beyond the ML model and onto the target phenomenon itself, and as mentioned, ML models are notorious for their lack of robustness. This is not really an issue with XAI methods as such, but may or may not be an issue, depending on the final target of explanation.

XAI Models as Idealized models The latter two objections are the ones this project aims to solve. The first, however, I believe Fleisher to have an excellent response to in his [13]. He there makes a compelling argument for why the explanations provided by XAI are analogous to idealised scientific models, which people generally agree are conducive to scientific understanding, regardless of their falsity. Fleisher points to three salient features shared by both idealised scientific models and XAI explanations to support his argument:

- *Simplification*: the model provides a simplified version of the target system, which is both interpretable and understandable to the user.
- *Flagging*: the model identifies and highlights features of the target system which are present, but negligible in the given context, e.g., by setting their value to zero.

- *Focusing on specific causal patterns*: the model suppresses some of the causal patterns which are present, so as to highlight certain patterns and their mechanisms in isolation.

Thus, if one believes that idealised scientific models, such as the harmonic oscillator or Snell's law, provides scientific understanding of their target phenomena, one should accept that explanations from XAI provide true understanding of their target ML models.

2.3 Correlations and Causality

In very general terms, the aims of science are two-fold. On the one hand, we aim to make accurate predictions, and on the other, to explain why these predictions are as they are. In the absence of any grand unifying theory, we tend to make do with specific theories for specified domains, and with models with even more restricted domains. As for ML models, however, there is a goal of having models that are both reliable and robust, where, as before, reliability is the predictive power in the original domain, whereas robustness is the predictive power of the model in related domains. The question, then, is what features are needed of the model to ensure these properties, beyond predictive power on the original data used to construct it.

Assuming we have a dataset of observational data, as is commonly the case in data-driven research such as ML, we can imagine using a linear regression model to generate a reasonably simple and interpretable model. If the phenomenon from which the data was gathered is not too complex, there is a good chance that this provides us with a fairly good model. We are then justified in believing that the model will provide us with good predictions on new data points gathered from the same distribution, but may we also assume that the model will be predictively accurate for data gathered from different domains? No. This is because such a model is purely correlation-based and does not assume any underlying causal structure of the phenomenon. It is what we call a *phenomenological* model. McMullin describes such model as follows:

> in general a phenomenological model appears to be an arbitrarily-chosen mathematically-expressed correlation of physical parameters from which the empirical laws of some domains can be derived. [...] From the purely logical point of view, there is no difference between a theory and a phenomenological model. Both can be axiomatised, from both; the desired empirical generalisations can be derived.
> But for the physicist there is a crucial difference between them. This difference can be put in one or other of two ways. The physical theory makes an assertion about a physical sub-structure which can account for the data; the phenomenological model makes no such assertion. [22, p. 391]

Thus, if we want models which generalise beyond the specific dataset chosen, we need them to represent the underlying causal structure of the phenomenon [23] or the *causal capacities* of the different features of the system [24]. This will provide us with more stable and robust models which we may expect to apply more generally, at least in the absence of other causal difference-makers. Pearl, the founder of the field of causal inference, argues that we "expect such difference in stability because causal relationships are *ontological*, describing objective physical constrains in our world, whereas probabilistic relationships are *epistemic*, reflecting what we know or believe about the world. Therefore, causal relationships should remain unaltered as long as no change has taken place in the environment, even when our knowledge about the environment undergoes changes." [23, p. 25]

Thus, if we want reliable, robust, and informative models of dynamical systems, models reflecting the underlying causal structure of the phenomena are preferable to those simply tracking the stable correlations between the system variables. This view has become increasingly widespread, and has lead to the establishment of an entire field of research, *causal inference*, dedicated to establishing methods and the empirical evidence needed to learn such causal relationships [23, 25, 26].

2.3.1 Structural Causal Models

The underlying causal structure of a system can be described in terms of a *structural causal model*, (SCM). The defining feature of such models is the *asymmetric* relations between the variables. Contrary to correlations, which are symmetric, causes and effects are not, as they come with a directedness. Such an SCM

represents all the relevant variables in a system, along with a set of relations representing the direction of the causal influences between the variables. Such a model can be represented as either a set of equations, the *structural equations*, or in a causal diagram. An example of such a diagram is shown in fig. 2, with its corresponding structural equations given in eq. (2). The notation ":=" signifies the asymmetry of the relation, where the right-hand-side is the cause of the left-hand-side of the equation.

$$\begin{aligned} B &:= f_B(A, U_B) \\ C &:= f_C(A, B, U_C) \\ D &:= f_D(A, C, U_D) \end{aligned} \tag{2}$$

With adjacent nodes, A and B, when there is an arrow from A to B, A if referred to as the *parent* of B, and B as the *child* of A. If there is a further arrow that goes from B to a third variable C, then A is the *ancestor* of C, and C the *descendant* of A. Variables without any modelled parents, i.e., variables whose values are not affected by any of the other variables, are called *exogenous*. Variables *with* parents are called *endogenous*. Finally, we always assume that there is some noise or background conditions which have not been accounted for in the model. These are grouped together in some random variable U, whose causal influence on the different variables are represented using dotted lines.

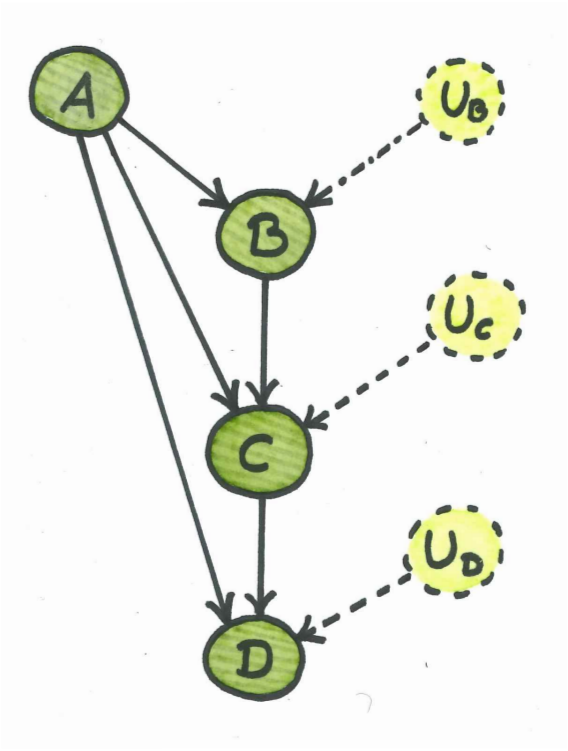

Figure 2: A diagram of a structural causal model. The variables *A, B, C* and *D* are the observed variables, whereas the variables in the dashed circles are their respective noise variables, which remain unobserved.

The notion of causality is closely linked to the notions of counterfactuals and interventions. This is because the causal structure is what determines the assymmetric dependencies between the variables, and these are most easily defined in terms of interventions and counterfactuals. Assuming there are no causal loops in the structure, a change in the value of a variable will lead to a corresponding change in the value of all its descendants, but not to any of its ancestors. In other words, given the model in fig. 2, regardless of what actually happens, this model tells us that if we *had* intervened on variable B, this *would have* caused a change in the variables C and D, but not in A.

2.3.2 Learning Causal Models

What we need is some method for differentiating between correlational and causal relationships between measured variables. Given only observational data of the final state of variables, this is usually not possible. To illustrate this, three different causal diagrams are depicted in fig. 3. Assuming only the variables X and Y are measured, using only the conditional probability distributions $P(X|Y)$ or $P(Y|X)$, there is no way to identify the correct causal graph. This is because these probabilities may be the same for all three SCMs. This may not cause trouble as long as the variable Z is kept constant, but if any change occurs in Z this is likely to cause the model to fail. Z is here an unobserved variable which causally influences the system, which is a common source of error in such modelling tasks. Thus, more than observational data is usually needed to learn the right causal model.

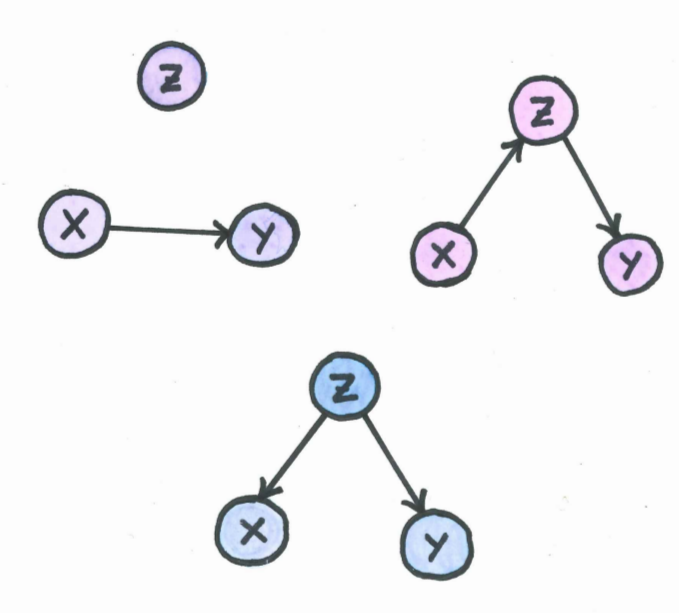

Figure 3: Three causal diagrams in the same Markov equivalence class. Given only observational data of the three variables, the three different causal structures may give rise to the same conditional probability distributions, so are indistinguishable.

As mentioned, causal dependencies are closely linked with interventions. Pearl [23] has introduced his "do-calculus" to formulate the conditions for causal inference in a more formal, mathematical language. Given a variable X, we denote the intervention where it is given the value x as *do(X=x)*. In general, it is assumed that all other variables are left unperturbed, so that only one variable is intervened on at a time. By investigating the changes in the probability distributions of the other variables, it is then possible to identify the causal effect of X on the other variables. From this, Pearl defines the causal effect as follows:

Definition 2.1: Causal Effect

"Given two disjoint sets of variables, X and Y, the causal effect of X on Y, denoted either as $P(y|\hat{x})$ or as $P(y|do(x))$, is a function from X to the space of probability distributions on Y. For each realisation x of X, $P(y|\hat{x})$ gives the probability of $Y = y$ induced by deleting from the structural equations of X all equations corresponding to variables in X and substituting $X = x$ in the remaining equations." [23, p. 70]

Thus, given interventional data, one is more likely to be able to identify the underlying causal structure of the phenomenon, of course depending on the amount and type of data available. However, some progress

can still be made even if only observational data is available.

Observational Data In many cases, it might be either impossible to intervene on the system (for example for ethical reasons), or one might not have enough data to train the network. In fields where this is the case, such as social science, it is common to exclude potential causal relationships by conditioning on the different features. This gives an indication of independencies between variables in the datasets, which can be used to exclude potential causal dependencies, and helps to identify the true ancestors of the dependent target variables. One might imagine the process of identifying the structural causal diagram as starting out drawing edges between every single variable. By identifying independencies between the variables, it is then possibly to remove some of the edges, thereby reducing the space of possible causal diagrams.

3 Method

3.1 Datasets

Each training dataset was generated from the same structural causal model, illustrated in fig. 4. This defined the causal dependencies between all the variables in the dataset. For simplicity, no noise was included, such that the system was purely deterministic.

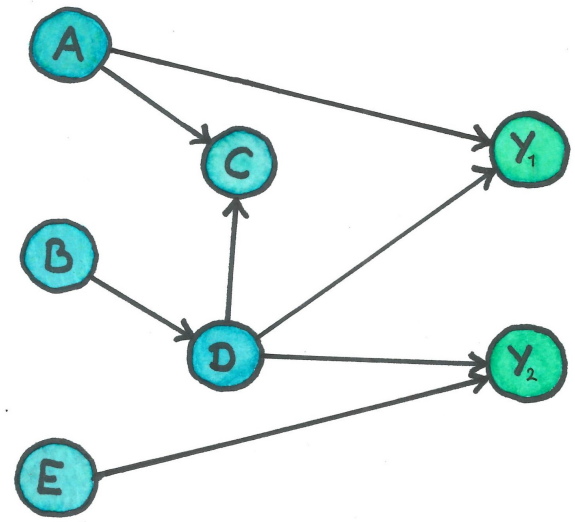

Figure 4: Model of the causal structure underlying the dataset used throughout. The input variables are illustrated with blue circles, the output variables with red circles. The arrows indicate the causal dependencies between the variables.

A, B and E are exogenous variables, and are drawn randomly from a uniform distribution with given bounds. The structural equations for the remaining variables are as follows

$$\begin{aligned} D &:= f_D(B) \\ C &:= f_C(A, D) \\ y_1 &:= f_{Y_1}(A, D) \\ y_2 &:= f_{Y_2}(D, E) \end{aligned} \tag{3}$$

Two different datasets were tested, a very simple one, where all the dependencies between the variables were linear, and a more complex one. The very simple case was included to make it possible to evaluate the methods, comparisons and explanations themselves, since this set of equations should be easily learned by the network. The equations were as follows

$$
\begin{aligned}
D &:= 2B \\
C &:= 5A + D \\
y_1 &:= 3.5A + 0.5D \\
y_2 &:= -2D + 0.2E
\end{aligned} \tag{4}
$$

Further, to make the analysis more meaningful and to also be able to study the limitations of the methods, a more complex dataset was generated using the following equations

$$
\begin{aligned}
D &:= 2B \\
C &:= 5A - \frac{A}{D} \\
y_1 &:= 3A^2 + D^3 + AD - D^2 \\
y_2 &:= 4D - D^2 + \sqrt{E}
\end{aligned} \tag{5}
$$

To be able to distinguish the models' ability to predict variables Y_1 and Y_2, these were considered separately in training the networks, such that each network only had one output variable.

Part of the main aim of this research was to investigate the effect of the dataset used to train the ML models on the models' robustness and the variance in the explanations provided of their behaviour. To do this, a range of slightly different training and test datasets were used, though the underlying causal model and the numeric ranges of the variables were always kept constant for the training data. To compare their robustness, a separate set testing datasets were included, with different variations.

3.1.1 Training datasets

Five different training datasets were used, shortened as follows:

- *Obsv*
- *Intv*
- *Intv_ C_ D*
- *Indp*
- *Simple*

The two main categories of training datasets which were used were observational and interventional, all generated using the same data generating function. In the observational dataset, all the exogenous variables were drawn randomly from uniform distributions, and the endogenous variables calculated based on this. This dataset was labelled *Obsv*. In the interventional datasets, interventions were performed on one variable at a time. The values of all the descendants of the variable intervened on were then calculated as a function of the value set at the intervention, whereas the other variables were left unperturbed. Graphically, this has the effect of removing all incoming edges to the variable intervened on.

Further, three different types of interventional datasets were considered. First, one where each input variable was intervened on one at a time, with equal probability. Thus, each input feature was intervened on in 20 % of the datapoints. This dataset was labelled *Intv*. Next, the effect of only intervening on a subset of the variables was tested by only intervening on the variables C and D. In this context, this was done because the underlying causal model was known, and one would expect that these interventions would be the most useful in identifying the independencies between the variables. The idea was that by comparing the predictive accuracy and robustness of different models, where different subsets of the input variables had been intervened on, this could be used to identify the direct parents of the output variables, since the performance should increase when confounders and other endogenous variables are intervened on. In this case, one would expect that intervening on C and D would be optimal, since this should make it the easiest

for the network to disregard confounders and colliders amongst the input variables, and thereby only rely on the direct causes in the predictions. This dataset was labelled *Intv_ C_ D*.

In addition, a dataset where all the input variables were independent of each other was used. This is at extreme version of the interventional dataset, at the other end of the scale compared to the observational one. In one sense, this is 'ideal', since it does not contain any confounding dependencies. This was included both as a reference point for comparison, and to study the effect on the performance of simply including more input variables than needed, which presumably makes it harder for the network to learn the right dependencies. This dataset was labelled *Indp*.

Finally, to test the network's performance under ideal conditions, a simplified dataset where only the parents of the target variable were included amongst the input features. This meant that for y_1, only A and D, whereas for y_2, only D and E were included. As well as being a test of the maximum potential of the network, subsampling from the input features in this way may be used as a method in itself for identifying the direct causes of the target variables, as mentioned in section 2.3.2, since one would expect the model trained on only the parents to be the most robust. This process requires certain considerations and background assumptions about the causal structure of the phenomenon. The relative performance and robustness of the models may be taken as a relative measure of the interaction information between the joint set of input features and the output features. Thus, by experimenting with including and excluding different combinations of input features for the same dataset and network, it should be possible to gain further information about the dependencies on the different input features by comparing the relative performance of these models. This dataset was labelled *Simple*.

3.1.2 Testing datasets

The following datasets were used for testing, to be described below

- *Test*
- *Obsv*
- *Intv*
- *OOD* (observational or interventional, depending on the model)
- *Diff. mod.* (different model)
- *Rand. mod.* (random model)

In addition to the test set taken from the same distribution as the training set (thus differing from model to model, depending on which training set was used), a set of different datasets were used to compare the robustness of the models. This was done in a few different ways. Firstly, two main changes were tested: changes in the numerical ranges of the variables, and changes in the dependencies between the input variables. Note that the generating process for the dependent variables y_1 and y_2 were always kept the same.

For the out-of-domain numerical ranges, the ranges of the independent variables were shifted slightly, and all the dependent variables were calculated thereafter. For the interventional datasets, the interventions were also included in this dataset. Thus, the relations between the variables were identical to the training sets, only the range on their numerical values were shifted. This dataset was labelled *OOD*.

The other datasets included changes in the generating functions of the input variables, with some changes in the causal relationships between the input variables. Thus, the generating functions of the output variables were the same, such that a model which had learned the right causal model should perform equally well here. For the dataset labelled *Diff. mod.*, the new structural equations were as follows, for the simple and complex datasets respectively

$$\begin{aligned} D &:= 0.8B + 0.5A \\ C &:= 3A - 0.8D \\ y_1 &:= 3.5A + 0.5D \\ y_2 &:= -2D + 0.2E. \end{aligned} \tag{6}$$

$$
\begin{aligned}
D &:= 0.8B + 0.5A \\
C &:= 3A + 0.8D \\
y_1 &:= 3A^2 + D^3 + AD - D^2 \\
y_2 &:= 4D - D^2 + \sqrt{E},
\end{aligned}
\tag{7}
$$

Note that although the ranges of the exogenous variables were kept as in the training set, this has the effect of changing the ranges of the endogenous input variables, and thereby also the target variables. To avoid changes in the loss due to shifts in the numerical ranges, the endogenous inputs variables were rescaled to their original ranges before calculating the target variables.

The other dataset with a different generating function was the same as the independent one used for training. Here, all the input variables were independent of each other, so the target variables were only correlated with their parents. This was labelled *Rand. mod.*.

Finally, all the models were tested on standard observational and interventional datasets, as from the training datasets, labelled *Obsv* and *Intv*.

The reason for including so many different test datasets was to be able to get a more nuanced picture of the conditions for the robustness of the models, as well as to get more datapoints for comparison, so that the conclusions drawn are more generalisable.

3.2 Network

A reasonably simple feed forward neural network was used, where two different depths were considered. Both had 120 nodes in the hidden layers, and the shallow network had two hidden layers, whereas the deep network had 8 hidden layers. ReLU was used as the activation function throughout, given by

$$
\mathrm{ReLU}(z) = max(0, z). \tag{8}
$$

All weights were initialised with `kaiming_normal`, as described in [27], and all biases were initialised to 0.0. The learning rate was kept constant at 0.001, and the optimiser used was Adam [28]. Mean squared error was used as the loss. The initial dataset was split into a training, validation and test-set, in total 3000 datapoints, with 70 % of the data used for training, and 15 % each for validation and testing. Each model was trained for 1000 epochs, and tested on the validation set for each epoch. The model which performed best on the validation set was kept.

3.3 XAI Methods

As discussed in section 2.2.2, two different methods from XAI were used. These include a local and a global method, namely Shapley values [18] and PySR [20]. Shapley values provides a single explanation for each datapoint considered. In the case of a simple linear model, such as the simple datasets here, the Shapley values can be converted into the linear coefficients, so it was reasonable to apply this method to the models trained on the simple datasets. This was implemented using the `shap` python package. This gives a straightforward interpretation of the Shapley values, and since we know that the data generating function remained the same, it is reasonable to take constant coefficients as the ideal, since these should be the same for every datapoint. Thus, calculating the variance in the predicted coefficients for different datapoints will give an indication of the stability of the explanations.

However, this interpretation does not hold for the models trained on the complex datasets, since we can neither assume that the contribution from each variable is linearly dependent on its value nor that it is independent of the contributions from the other variables. The latter is due to the product terms in the structural equations. Thus, there is no straightforward way of converting the Shapley values to a global equation of the system. Of course, this is the case with most physical systems, and in the rare cases where it is not, we might not be aware of it, so that we cannot make the assumption regardless.

Thus, the more complex method of symbolic regression, here using the `PySR` python package, was used for the complex datasets. In all cases, the optimal model according to the algorithm was chosen in every case.

The criteria for optimality are discussed in more detail in [20]. This will in most cases be a rather complex model when considering the number of terms etc., but since no attempt was made at direct inspection of the functional expression, this was considered somewhat irrelevant in this context. The main aim here was to perform a relative comparison between the resulting functions.

When generating the symbolic expressions, the following binary and unary operators were included as possible options: `+`, `-`, `*`, `/`, `^`, `sqrt()`, `sin()`, `cos()` and `exp()` (the natural exponent). In addition, only integer values of exponents were allowed. 50 iterations were used.

Finally, the *Simple* models were excluded from these analyses, due to their different number of input features, making a comparison with the other models difficult to evaluate.

3.3.1 Explanations and their variance

Both for the simple and the complex datasets, a set of explanations were generated for each model. Specifically, each model (trained on its specific dataset) was exposed to a separate set of different datasets, and one explanation was generated from each of these. Specifically, each model was given the following datasets:

- observational
- interventional
- observational, OOD
- interventional, OOD
- diff. mod.
- rand. mod (independent),

as described in section 3.1.2. If the ML model had learned the correct data generating process, you would expect the explanations provided to be identical, regardless of the datasets provided. Hence, the variance of the explanations is to some extent a measure of the deviation from this ideal. Given their differences, different methods had to be used for Shap and PySR in calculating the variance.

In evaluating the explanations, a combined dataset of 600 datapoints, equally sampled from the observational, interventional, observational_OOD, interventional_OOD, diff. mod. and rand. mod. datasets was used.

Shapley values In calculating the Shapley values, 500 datapoints were generated for each dataset considered, and a set of Shapley values was found for each datapoint. 100 randomly chosen datapoints from each dataset were used as the background data considered by the `ShapExplainer`. The linear coefficients were then generated for each datapoint, and for each dataset the average of these coefficients were taken as a global explanation of the model. For each input variable, the variance was calculated across all datapoints and datasets for a given ML model. The overall variance of the explanations of each model was taken as the average of these variances.

In addition, the average of each coefficient was calculated for each model, such that a global explanation was obtained. This global explanation was then used to produce predictions for the combined dataset, and the MSE loss of each model explanation was calculated.

To translate into an equation, we introduce the following notation. To represent the different datasets, datapoints and variables, we use $\mathcal{D}_{ov,t}$ to represent the dataset consisting of datapoints $d_{ov,i,t}$, where ov is the output-variable, $ov \in \{y_1, y_2\}$, i is the index of the datapoint, e.g. $i \in [1, 500]$ and t is the type of dataset. Typically, $t \in \{obsv, intv, obsv\_OOD, intv\_OOD, diff.mod, rand.mod.\}$. For each dataset $\mathcal{D}_{ov,t}$, we have a corresponding set of Shapley values $\mathcal{S}_{M_T,ov,t}$, consisting of the Shapley values $s_{M_T,ov,iv,i,t}$, one for each input variable $iv \in \{A, B, C, D, E\}$ and datapoint i. We label each trained ML model $\mathcal{M}_{T,ov}$, where T represents the training set used, such that $T \in \{Obsv, Intv, Intv_{C_D}, Indp\}$.

The average Shapley values for a given model $\mathcal{M}_{T,ov}$ was thus given by:

$$s_{\mathcal{M}_{T,ov,iv,avg}} = \sum_{t,i=1}^{i=500} s_{\mathcal{M}_{T,ov,iv,i,t}},$$
$$\text{for } t \in \{obsv, intv, obsv\_OOD, intv\_OOD, diff.mod, rand.mod\}.$$

And the corresponding complete set of average Shapley values is $\mathcal{S}_{ov,avg} = \{s_{\mathcal{M}_T,ov,t,avg\}}$. These were the coefficients used to construct the global explanation for each model.

The variance for each Shapley value is thus given by

$$var_{\mathcal{M}_{T,ov,iv}} = \text{var}(s_{\mathcal{M}_{T,ov,iv,i,t}})$$
$$\text{for } t \in \{obsv, intv, obsv\_OOD, intv\_OOD, diff.mod, rand.mod.\}, i \in [1, 500].$$

The average variance of each model was taken as the average of these, i.e.;

$$var_{\mathcal{M}_{T,ov}} = \text{avg}(var_{\mathcal{M}_{T,ov,iv}}) \text{ for } iv \in \{A, B, C, D, E\}. \tag{9}$$

Symbolic regression Since PySR provides global explanations, for each ML model, a separate explanation was calculated for each dataset considered. This was done by creating a separate dataset consisting of the input variables from the dataset in question, and the predictions provided by the model. The PySR model was then trained on this new dataset, consisting of the real input variables and the ML model predictions. The resulting explanation was then an equation in symbolic form, where the independent variables were the original input variables.

As it turned out, none of the models trained on the complex datasets generalised to the *OOD* data. None of the models were even remotely reliable on this dataset. Therefore, when generating explanations and analysing the models trained on the complex datasets, all *OOD* models and datasets were ignored. By excluding them, we are certain that the difference in performance between the models and explanations is due to significant results, and not irrelevant differences in their awful performance on the *OOD* datasets.

Thus, for each model, *four* different explanations were generated, one for each of the datasets, *obsv*, *intv*, *diff. mod* and *rand. mod*. To measure the variance and accuracy of these explanations, a new and combined dataset of datapoints sampled evenly from the different distributions was used. This consisted of 600 datapoints, and the input variables were then passed through the set of explanations to generate four separate sets of predictions. If the explanations had been functionally equivalent, these would have been identical. The variance was then calculated for each datapoint across explanations, and a global average was taken of these. This was then used as a measure of the overall variance of the model explanations. In addition, the explanations themselves were tested on the same dataset, and their MSE loss was calculated and used as a measure of the accuracy of the explanations with respect to the target system.

As above, we denote each ML model with output variable ov, trained on dataset T as $\mathcal{M}_{T,ov}$. We then generate one explanation function $\mathcal{E}_{\mathcal{M}_{T,ov},t}$ for each test dataset. In this case, $t \in \{obsv, intv, diff.mod, rand.mod.\}$. To evaluate the variance and performance of the explanation, we constructed the combined datasets $\mathcal{D}_{ov,comb}$ with elements datapoints $d_{ov,i,comb}$, with $i \in [1, 600]$ and $ov \in \{y_1, y_2\}$. With some liberation in the notation, let us denote the prediction from an explanation of a datapoint as $\mathcal{E}_{\mathcal{M}_{T,ov},t}(d_{ov,i,comb})$. The variation is then first calculated across *explanations*, for each datapoint i as

$$var_{\mathcal{E}_{i,\mathcal{M}_{T,ov}}} = \text{var}(\mathcal{E}_{\mathcal{M}_{T,ov},t}(d_{ov,i,comb}))$$
$$\text{for } t \in \{obsv, intv, diff.mod, rand.mod.\}.$$

The average variance is then taken as the average of the variances for all datapoints i, given by

$$var_{\mathcal{E}_{avg},\mathcal{M}_{T,ov}} = \text{avg}(var_{\mathcal{E}_{i,\mathcal{M}_{T,ov}}}) \text{ for } i \in [1, 600]. \tag{10}$$

3.4 Limitations

There are certain causal structures which both the observational and the interventional model will be unable to discover due to unobserved confounding variables. These are variables whose existence or values remain unknown, but that causally affect observed variables in the datasets. For example, in fig. 4, we could imagine an additional unobserved variable F, which is a parent to both E and y_1. In such cases, models which base their predictions on correlations will perform better than those solely basing their predictions on causal relations, since no "pure" information about the cause is available in the input data. Thus, one is better off making predictions based on the other child of the unobserved parent, even though this will fail under interventions. More advanced methods than simple MSE loss are required to train and evaluate models to learn (or not to learn) such correlational structures.

Since many of the methods used here are based on a comparison of the relative performance between different models, these would not generalise to such cases of unobserved causal variables, since we cannot assume that a model which has learned the true causal model would necessarily be more reliable and robust in such cases.

This framework also assumes that there are no causal cycles amongst the variables, and that the output variables are not the parents of any of the input variables. These are assumptions commonly made in the literature [29, 23], but are nevertheless important to keep in mind when evaluating the scope of the suggested methods.

4 Results and Discussion

4.1 Performance

The overall aim of this part was to investigate what features of the datasets, networks and scaling might improve the generalisation properties of the ML models, measured by the models' performance on different test datasets from different domains than what the models were initially trained on.

Five different models were trained on five different datasets, one set of models for the target variable y_1 and another for the target variable y_2. Additionally, this was repeated with and without min-max scaling of the data, and with a deep and a shallow network. Thus, for each variable, a total of 20 models were trained. This was to be able to infer more global trends in the dependencies. Further, each model was tested on six different datasets, as outlined in section 3.1.2. Because of the large span in results between the models, all losses have been plotted on a logarithmic scale throughout the rest of this paper.

4.1.1 Simple Dataset

For the simple datasets, generated from eq. (4) and eq. (6), the results are shown in fig. 5 and fig. 6. There are many things to discuss in these results, and there are some general trends to be found. However, just as interesting is the fact that most of these trends have exceptions, in most cases for no known reason. Given the limitations of this project, the exceptions to the rules will be highlighted, but little attempt will be made at explaining these.

Firstly, it is very clear the the *Simple* models, which were only given the direct parents of the dependent variable reigns supreme in almost all cases. This is to be expected, since this makes the problem extremely simple, without any confounding correlations, and no changes to the conditional probabilities of the included variables across the datasets.

Next, the *Obsv* models are generally second best to the *Simple* models on the observational data (and its own test-set). This suggests that this was an easier distribution to learn compared to the interventional and independent datasets. This can likely be ascribed to the lack of irrelevant information in the input data, since the models may rely on all the input variables which are interrelated with the parents of the dependent variable without problem, so there are fewer irrelevant variables amongst the inputs for the model to incorrectly rely on. However, the *Obsv* models clearly perform the worst with respect to generalisation, as they are completely unreliable for the *intv*, *diff. mod.* and *rand. mod.* datasets. Somewhat surprisingly, they perform well on the (numeric) out-of-domain datasets, as do the other models. This holds for both y_1

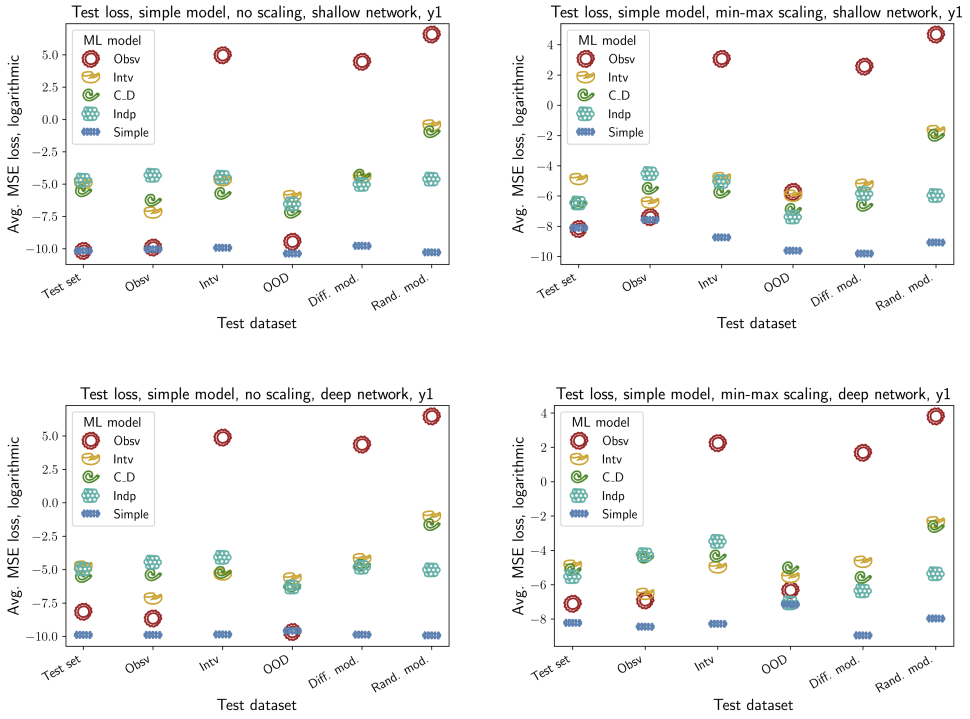

Figure 5: Test MSE loss for the different models trained to predict y_1 for the simple dataset. Each sub-graph represents a different combination of network depth and scaling, as described in the figure titles.

and y_2 (with the exception of the models trained on scaled data for y_2). As will be discussed below, this is not the case for the complex datasets, where all the models fail on the *ood* dataset.

The *Indp* models (no dependencies between the input variables), show almost the opposite trend to the *Obsv* models. They perform worse than what one might have expected for the observational and interventional datasets, but are the models which generalise the best to the other datasets (with the exception of the *Simple* models). The only difference between the training datasets of the *Indp* models and the *Simple* models was the number of input features, where the *Indp* models were given additional, random information. One could think that the models would easily learn not to depend on this random noise. The fact that the *Indp* models perform worse on their test-sets than both the *Obsv* and the interventional models suggests that ignoring such irrelevant features is not necessarily such an easy learning task, and shows that including additional random and irrelevant input features slows the learning process and reduces the predictive power of the model even on within-domain data.

Finally, the models trained on the interventional datasets, which include *Intv* and *C_D*, generalise significantly better than the *Obsv* models, as expected. They perform worse that the *Obsv* models on their own test-set and on the observational data, which shows that its learning problem is harder. This makes sense, since it had a more heterogeneous training dataset, where the dependencies differed from datapoint to datapoint, since the dependencies between the input features differed depending on which variable was intervened on. This is also manifested in the fact that they generalise less than the *Indp* models, especially on the *diff. mod* and *rand. mod.* datasets. However, the interventional models perform better on the observational dataset than the *Indp* models, suggesting they have learnt some generalisation properties better. Further, which of the two interventional models generalise the best differs depending on output variable, network depth and scaling, suggesting that experimenting with which variables to intervene on may not be a reliable method for identifying independencies in the network, since there are clearly other features which contribute significantly to their relative performance.

Regarding the effect of scaling and network depth, a visual inspection of the plots suggests that for y_1, the depth of the network made little difference, whereas the min-max scaling improved the results significantly.

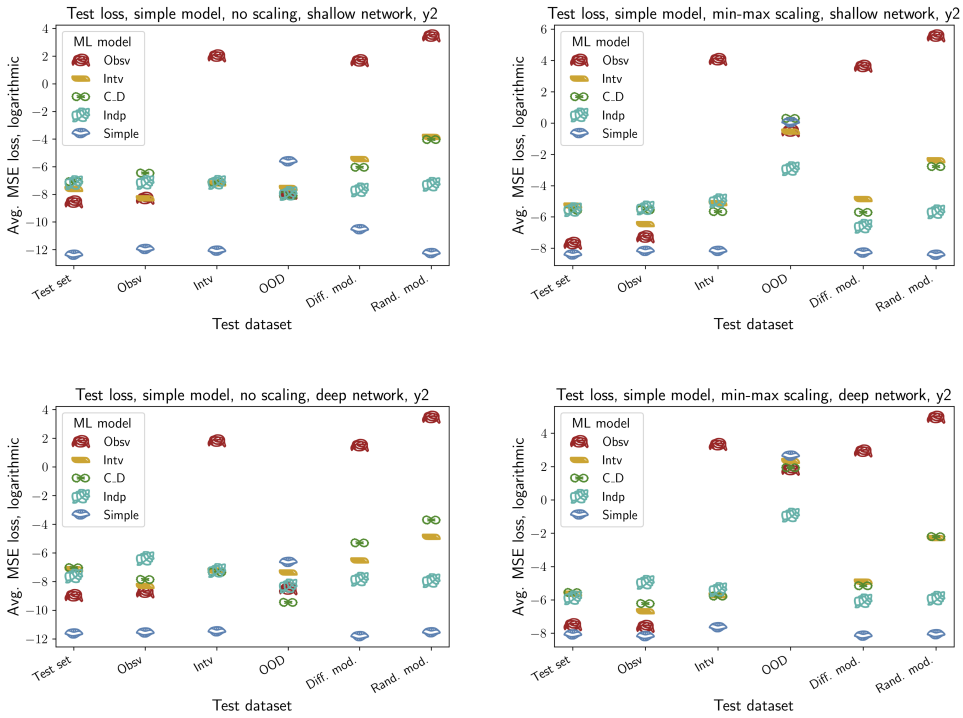

Figure 6: Test MSE loss for the different models trained to predict y_2 for the simple dataset. Each sub-graph represents a different combination of network depth and scaling, as described in the figure titles.

However, for y_2, the opposite was the case. Again, the depth of the network made little difference, but here the models trained on unscaled data generally performed better.

4.1.2 Complex Dataset

The results from the models trained on the complex datasets show many of the same trends as the simple ones, with some notable exceptions.

Again, the *Simple* models perform best on all the test-sets, with the *ood* set as the exception. This is especially interesting, since all the complex models fail on the *ood* dataset, whereas the *Simple* models could handle this shift in numerical range of the variables without problem. Thus, there seems to be a significant difference between the potential for generalisation between the simple linear structural equations eq. (4) and the more complex structural equations eq. (5), which included powers and product terms, making the contribution of the different input variables dependent on each other.

As before, the *Obsv* models do not generalise beyond the observational data (here they also fail on the *ood* datasets), the *Indp* models generalise the best (except compared with the *Simple* models), but are outperformed by the *Obsv* and *Intv* models on the test-sets and the observational sets. Again, there were no general trends in the relative performance between the *Intv* models and the C_D models.

Regarding the dependence on scaling and network depth, the trends are somewhat different to those found for the simple datasets. For y_1, the shallow networks trained on unscaled data seems to perform the worst, and the deep networks with scaling the best. In general, these datasets seem to have benefited from a deeper network, but for the deep networks, the scaling seems to have had little effect, except for the *Obsv* model employed on the other datasets. However, this should be ignored here, since the errors are so extremely high that the model is useless regardless of a slight relative decrease in the error. Similar trends are found for y_2, where the deeper network performed significantly better, but the effect of scaling was mostly noticeable for the shallow network.

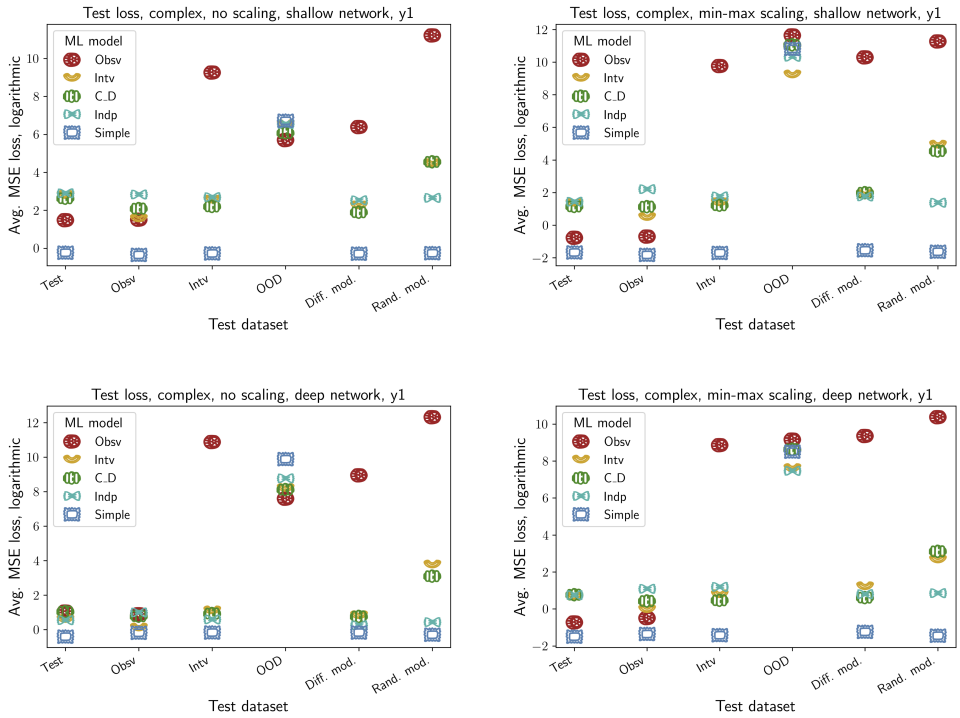

Figure 7: Test MSE loss for the different models trained to predict y_1 for the complex dataset. Each subgraph represents a different combination of network depth and scaling, as described in the figure titles.

4.1.3 Discussion

In general, there are a few significant conclusions to be drawn from these results. Most importantly, interventions work. Throughout, the interventional models (*Intv* and *C_D)*) vastly outperform the *Obsv* models on generalisability, making them far more reliable and robust. Importantly, this was the case even without including the information about which variables were intervened on, which is generally taken to be necessary for causal discovery [23].

Further, the lack of a stable difference in performance between the *Intv* and the *C_D* models suggests that a comparison of the performance of models where different subsets of variables have been intervened on does not help in identifying the true causal parents of the dependent variables. Instead, a combined interventional dataset where different variables have been intervened on in equal proportion is sufficient for ensuring robust performance.

Finally, in cases where not enough interventional data is available to train models on this, experimenting with excluding different subsets of the input features in the training phase works well, as shown by the performance of the *Simple* model. Importantly, it is still necessary to test the different models on data taken from a different domains in order to identify the one which relies on the true parents amongst the input features, and not just on correlations. This method is more data-efficient, but requires the training of far more models.

As for the dependency on scaling and depth of the network, the optimal combination seemed to differ from instance to instance. This is indeed a known phenomenon in machine learning, known as the "no free lunch theorem", which states that there is no one single ML method which is universally the best. The fact that this can even be seen in such simple models and datasets as the ones considered here shows how ubiquitous this phenomenon is, and how hard it is to make general claims about what might improve a network's performance.

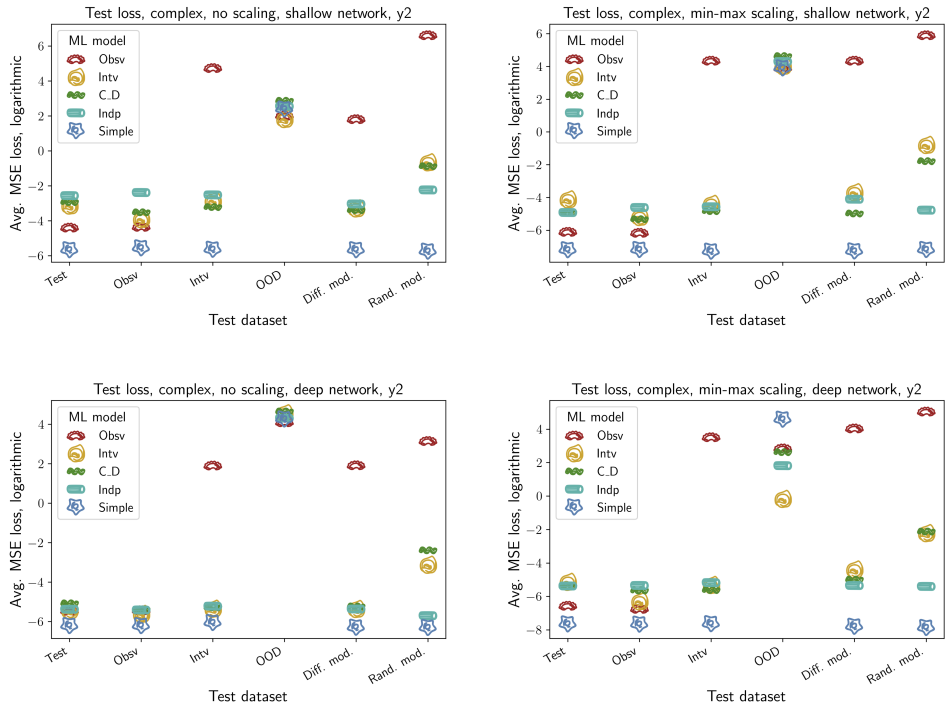

Figure 8: Test MSE loss for the different models trained to predict y_2 for the complex dataset. Each subgraph represents a different combination of network depth and scaling, as described in the figure titles.

4.2 Variance

The second part of this project aimed to establish whether there is a link between the reliability and robustness of an ML model, and the variance (stability) of the explanations provided of that model. In addition, the reliability of the explanations themselves was also tested and compared with the explanation and model loss. Specifically, this was a comparison between the predictions generated from the explanation model and the true targets (*not* the original ML model predictions).

4.2.1 Simple Dataset

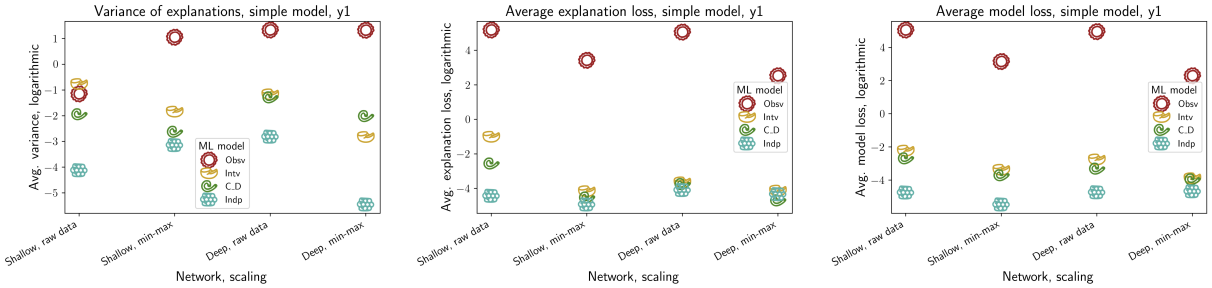

Figure 9: Variance of explanations, average loss of these explanations, and average loss of the corresponding ML models. All for ML models trained on the simple model, output variable y_1. Different combinations of network depth and scaling are plotted along the x-axes.

For the simple dataset, the explanations consisted of local Shapley values, converted into linear coefficients of the input variables, according to eq. (1). The variance of each model's explanations were given by eq. (9).

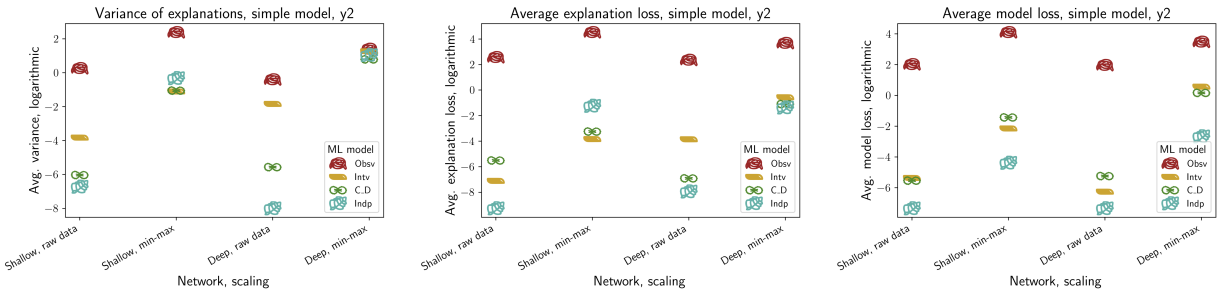

Figure 10: Variance of explanations, average loss of these explanations, and average loss of the corresponding ML models. All for ML models trained on the simple model, output variable y_2. Different combinations of network depth and scaling are plotted along the x-axes.

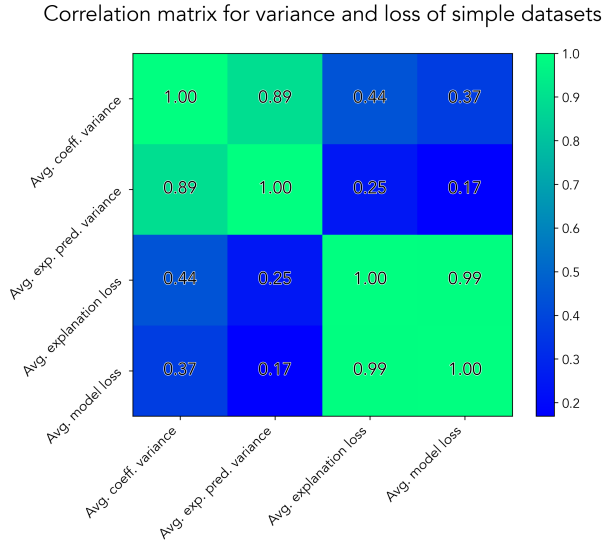

Figure 11: Correlation matrix of the variance of coefficients, variance of explanation predictions, average loss of these explanations, and average loss of the corresponding ML models. This is for simple models of y_1 and y_2 combined, and for all network- and scaling combinations.

Further, the average value of the coefficients for each model, as calculated and explained for the different datasets, were taken as a global explanation of the model. This was then used as a "prediction model", and the average loss of this on a mixed dataset was taken as a measure of the accuracy and reliability of the global explanation. This was again compared with the average loss of the models themselves.

The variance of the explanations (coefficients), explanation loss and model loss, for the different combinations of scaling and network depth, are shown in fig. 9 and fig. 10 for y_1 and y_2 respectively. Though not perfect, just from a visual inspection, we can see that the average explanation loss and the model loss follow each other very closely for both variables. There is a greater difference between the variances and the losses, but there is some similarity in the patterns, as can be seen in the relative difference between the different models and the different network- and scaling combinations.

For a more precise measure of the similarities, the correlation matrix was calculated for the different variables, showing the correlation coefficients between all the included variables. Here, all the data from the different models and network combinations, as well as y_1 and y_2 were combined. For comparison with the complex model, where the explanation variance was calculated as described in section 3.3, the same method was also applied to the global Shap explanations, where the variance of the different predictions from the different explanations of the same model was calculated. The correlation matrix is shown in fig. 11. From

this, we see that the two measures of variance are fairly closely correlated, and so are the two types of losses. However, the correlation between the losses and the variance is not that strong, for neither measure of variance.

Finally, the explanations generated from the *Obsv* models have a higher loss throughout, showing that the explanations generated from models trained on interventional data are more robust and reliable, as could be expected. An interesting find is that the average explanation loss is in fact *lower* than the corresponding average model loss for several of the models. However, it is hard to spot any general trend in which models this holds for. It mostly holds for both interventional models, but not for the *Indp* and *Obsv* models.

4.2.2 Complex Dataset

Looking at the networks trained on the complex datasets, the correlation between the variance and explanation/model loss is far stronger. This can be seen from a visual inspection of the plots of average variance, explanation loss and model loss, as shown in fig. 12 and fig. 13 for y_1 and y_2 respectively. This is further strengthened by the precise correlation coefficients, as shown in fig. 14, with a near perfect correlation between the explanation variance and both types of losses. The explanation loss and model loss are also closely correlated, suggesting that the explanations are functionally similar to the target ML models.

However, looking at the different datasets, it is clear that this generalisation does not always hold. Whereas most of the models seem to follow similar paths for the three variables for y_1, this is not the case for y_2, which shows much more diversity in the behaviours. Here, the pattern is almost identical between the explanation variance and explanation loss, suggesting that the variance of the explanations is an indication of the quality thereof. However, it does not show the same pattern for the model losses, and the same close relationship between the explanation loss and model loss is not found here. For example, the *Indp* explanation loss is much higher than its' model loss, whereas the opposite is the case for the *Intv* explanation and model. This suggests that some of the explanations have been unable to learn the correct model function.

As for the simple datasets, the explanations generated from interventional models are more reliable and robust than those generated from the *Obsv* models.

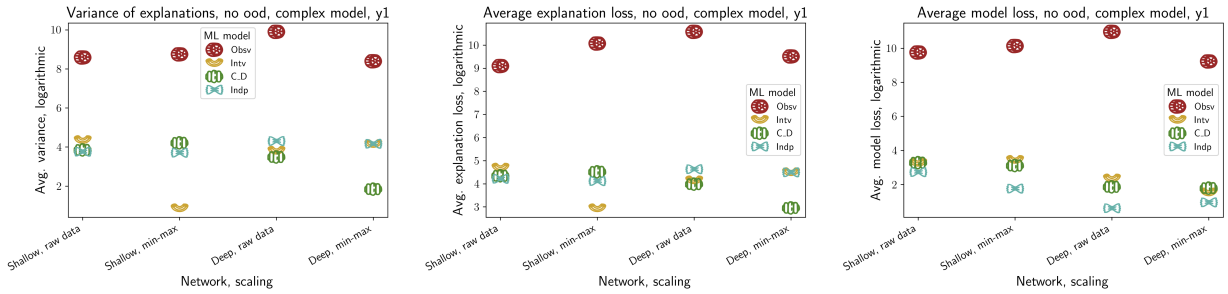

Figure 12: Variance of explanations, average loss of these explanations, and average loss of the corresponding ML models. All for ML models trained on the complex model, output variable y_1. All *OOD* datasets and models have been excluded from this analysis. Different combinations of network depth and scaling are plotted along the x-axes.

4.2.3 Discussion

In sum, the only conclusive result was that both models and explanations derived from interventional data were more reliable and robust than those derived from purely observational data. Beyond that, no definite results were obtained regarding the dependency between the variance of explanations and the reliability of the explanations and ML models. There was a positive correlation between all three, but this was in general not strong enough to justify a strong inference from explanation variance to model or explanation reliability and robustness.

Although there is no direct link from explanation variance to model performance, all the exceptions are cases where the variance and explanation losses are high, but the model loss low. These are cases where

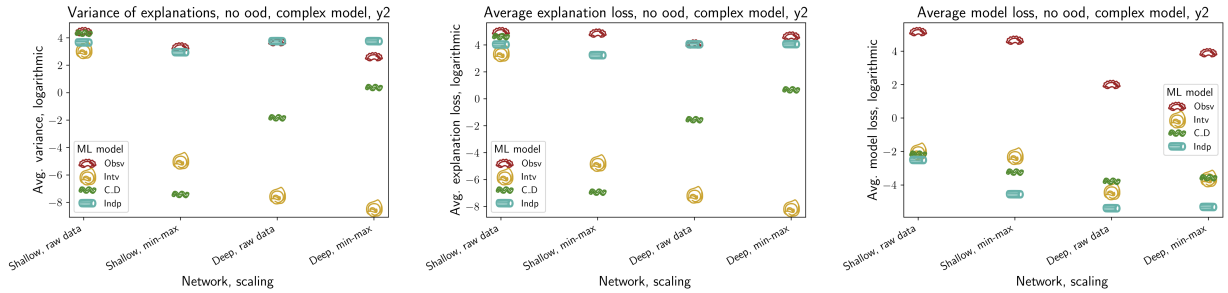

Figure 13: Variance of explanations, average loss of these explanations, and average loss of the corresponding ML models. All for ML models trained on the complex model, output variable y_2. All *OOD* datasets and models have been excluded from this analysis. Different combinations of network depth and scaling are plotted along the x-axes.

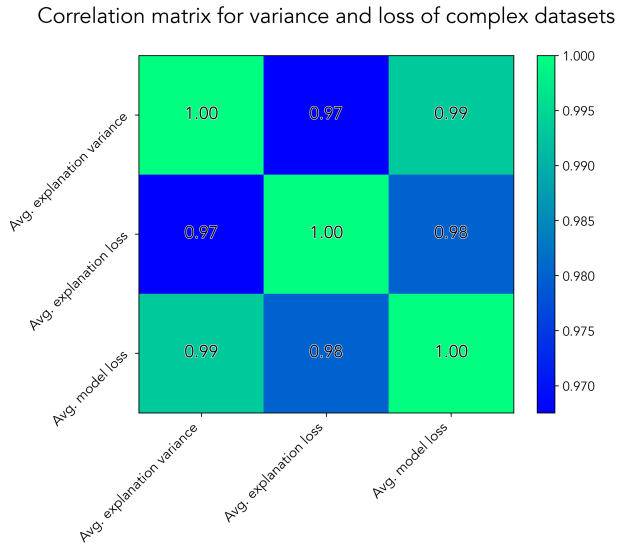

Figure 14: Correlation matrix between variance of explanation predictions, average loss of these explanations, and average loss of the corresponding ML models. This is for simple models of y_1 and y_2 combined, and for all network- and scaling combinations.

the explanation algorithm has failed at capturing the true functionality of the model, leading to a large discrepancy in performance between the explanation and the original ML model. With the exception of the *Obsv*, shallow, raw data y_1 model, there are no cases of a low explanation variance, but with a high explanation and model loss. In sum, we cannot make the inference from high variance to high explanation/model loss, since there are multiple cases of fairly high explanation variance, but low explanation/model loss. Yet, the results here suggest that we *can* generally make the opposite inference, and assume that an explanation with low variance is reliable and fairly robust.

This is an especially useful inference, since it can be tested without actually gathering more data. It only requires that one generates additional (possibly synthetic) datasets of input variables and pass these through both the ML models and their explanation models. Thus, the variance can be calculated across explanations, and reliable explanations can possibly be assumed simply from this.

Finally, the cases where the model explanations perform better than the models themselves are especially interesting. A potential explanation is that the somewhat simplified explanations have managed to idealise away features which the models have learned due to overfitting, which do not generalise. If this is right, one could expect this effect to be even more beneficial in cases where the datasets contain a lot of noise, but

the explanation we seek is a more simplified model. This fits well with Fleisher's [13] analogy between XAI explanations and idealised models in science.

5 Conclusion and Outlook

5.1 Conclusion

From this research, it is clear that including interventional data in the training datasets has great potential for improving the reliability and robustness of both the ML models themselves and the explanations we may obtain of them using methods from XAI. Thus, the first hypothesis is answered in the affirmative. An additional finding was that this improvement was significant, even though information about which variables were intervened on was not given as input to the ML model.

The third hypothesis, which was that experimenting with different combinations of input variables in the training phase will help us to identify the true causal parents, was also confirmed by the success and robustness of the *Simple* models, where only the parent variables were given as inputs.

It is harder to decide conclusively on the second hypothesis, since the results here were more heterogeneous. This regards the connection between the explanation variance and the explanation and model loss. As discussed above, no simple one-to-one mapping was found, and although there was a strong link in many of the instances, there were also multiple unexplained exceptions to the general trend.

5.2 Limitations

The main limitation of this research is likely the limited datasets used. Because of the many levels at which the claims to generalisation applies, it is not sufficient to simply include more datapoints from the same distributions. Instead, a much greater variation of structural causal models, causal equations and variations applied to the out-of-domain datasets would be needed in order for the results to be representative enough to make any strong claims regarding the generalisability of these results.

Finally, these results are also limited by the types and simplicity of the causal structures considered, and do not necessarily generalise beyond those. These limitations were here both necessary and intentional, since this study was mostly meant as a proof-of-principle approach to the significance of interventional data and explanation stability.

5.3 Future Research

The aim of this project was to make a contribution to issues pertaining to the properties of generalisation, reliability and robustness of ML models and their explanations. These are essential properties for ascertaining the domain and reliability of the models. Knowing the domain of a model is absolutely essential in science, since it is not possible to apply the model reliably and responsibly without knowing in what situations it can be trusted.

As things currently stand, there is a general lack of both practical and theoretical knowledge regarding the generalisation properties of ML models [3]. Instead, there is an abundance of rules of thumb regarding network structure, scaling methods, learning rates, optimisers, pre-processing-methods and so on. However, as established by the "no free lunch theorem", none of these are without exceptions, meaning that implementing these is insufficient for ensuring a reliable and robust model.

Of course, in many cases, the reliability of the models and explanations can be tested empirically, but it is in the cases where this is not possible that they have the greatest potential for enhancing our scientific research. Thus, to establish generalisable methods for ascertaining the usefulness and domain of the ML models and their corresponding explanations, both empirically and theoretically, is essential for the future applicability of ML models in science and beyond.

References

[1] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature machine intelligence, 1(5):206–215, 2019.

[2] Boris Babic, Sara Gerke, Theodoros Evgeniou, and I. Glenn Cohen. Beware explanations from AI in health care. Science (New York, N.Y.), 373(6552):284–286, July 2021.

[3] Tim Räz and Claus Beisbart. The Importance of Understanding Deep Learning. Erkenntnis, 2022.

[4] Judea Pearl and Dana Mackenzie. The book of why: the new science of cause and effect. Allen Lane, London, 2018.

[5] Sanjeev Kulkarni and Gilbert Harman. An Elementary Introduction to Statistical Learning Theory. In An Elementary Introduction to Statistical Learning Theory, volume 853. John Wiley & Sons, Incorporated, United States, 2011.

[6] M. A. Nielsen. Neural networks and deep learning, 2018.

[7] I. Neutelings. Neural networks.

[8] Michael Tamir and Elay Shech. Machine understanding and deep learning representation. Synthese, 201(2):51, 2023.

[9] Julie Jebeile, Vincent Lam, and Tim Räz. Understanding climate change with statistical downscaling and machine learning. Synthese (Dordrecht), 199(1-2):1877–1897, 2021.

[10] Florian J. Boge and Michael Poznic. Machine Learning and the Future of Scientific Explanation. Journal for General Philosophy of Science, 52(1):171–176, March 2021.

[11] Florian J. Boge. Two Dimensions of Opacity and the Deep Learning Predicament. Minds and Machines, 32(1):43–75, March 2022.

[12] Sanja Srećković, Andrea Berber, and Nenad Filipović. The Automated Laplacean Demon: How ML Challenges Our Views on Prediction and Explanation. Minds and Machines, 32(1):159–183, March 2022.

[13] Will Fleisher. Understanding, Idealization, and Explainable AI. Episteme, 19(4):534–560, 2022.

[14] Zachary C. Lipton. The mythos of model interpretability. Communications of the ACM, 61(10):36–43, September 2018.

[15] Henk W. De Regt and Dennis Dieks. A Contextual Approach to Scientific Understanding. Synthese (Dordrecht), 144(1):137–170, 2005.

[16] Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, and Peter Battaglia. Rediscovering orbital mechanics with machine learning. Machine Learning: Science and Technology, 4(4):045002, 2023. Publisher: IOP Publishing.

[17] Lloyd Shapley. Book title: Notes on n-person games vii. cores of convex games.

[18] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions.

[19] Christoph Molnar. Interpretable Machine Learning. 2 edition, 2022.

[20] Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl.

[21] Eyal Wirsansky. Hands-On Genetic Algorithms with Python. Packt Publishing, 2020.

[22] E. Mcmullin. What do Physical Models Tell us? In B. Van Rootselaar and J. F. Staal, editors, Studies in Logic and the Foundations of Mathematics, volume 52 of Logic, Methodology and Philosophy of Science III, pages 385–396. Elsevier, January 1968.

[23] Judea Pearl. Causality: models, reasoning, and inference. University Press, Cambridge, 2nd edition. edition, 2013.

[24] Nancy Cartwright. The dappled world: a study of the boundaries of science. University Press, Cambridge, 1999.

[25] Bernhard Scholkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. Proceedings of the IEEE, 109(5):612–634, 2021.

[26] Stefan Buijsman. Causal scientific explanations from machine learning. Synthese (Dordrecht), 202(6):202–, 2023.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In 2015 IEEE International Conference on Computer Vision (ICCV), pages 1026–1034, 2015. ISSN: 2380-7504.

[28] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization.

[29] Jonas Peters. Elements of causal inference: foundations and learning algorithms. Adaptive computation and machine learning series. The MIT Press, 2017.