Academic year
2022 - 2023

Temporal Causal Discovery with Attention-based Dilated Convolutional Networks

Maurin Voshol

Master's thesis
Master of Science in computer science: data science and artificial intelligence

Supervisors
prof. dr. T. Verdonck, UAntwerpen
prof. dr. T. de Schepper, UAntwerpen
Steven Mortier, UAntwerpen

University of Antwerp
Faculty of Science

Acknowledgements

I would like to express my gratitude to the following individuals and organizations for their support and contributions toward the completion of my Master's thesis. Firstly, I would like to acknowledge that this work was submitted in fulfillment of the requirements for the degree of Master of Science in Computer Science: Data Science and Artificial Intelligence. I would like to express my appreciation to the University of Antwerp (UA) and the Internet & Data Lab (IDLab) research group for providing me with the necessary resources and technical support to carry out this research. I would like to thank my promoters Prof. T. Verdonck and Prof. T. De Schepper for providing me with this topic and making this research possible in the first place, but also for their insightful feedback and critical review of my thesis. I would like to extend my gratitude to my supervisor, Steven Mortier, for his exceptional support, assistance, and guidance throughout the course of my research. His expertise, feedback, and constructive criticism have been crucial in the successful completion of this thesis. I truly appreciate his availability, flexibility, and the great cooperation we had. Lastly, I would like to thank my family and friends, and in particular my wife Hanna, for their encouragement and support throughout this academic year. Their support has been a source of motivation and inspiration to me.

Abstract

In this thesis, we explore the complexities and challenges of causal discovery in time series data using machine learning methods. For example, additive models can identify temporal causal relationships in data. However, due to their inability to fully approximate non-additive relationships, they might overlook certain relationships. Furthermore, temporal models with constrained receptive fields are unable to capture long-range dependencies. Increasing the complexity of the model may expand the receptive field, but it also introduces the risk of overfitting, potentially resulting in less accurate causal predictions. Considering the real-world implications of such predictions, there arises a need to quantify the uncertainty of these models to enhance the robustness and reliability of their output predictions. To address these issues, we make two key contributions: (1) We introduce the Temporal Attention Mechanism for Causal Discovery (TAMCaD) architecture. This framework captures non-linear, non-additive, long-range temporal relationships, and as it produces a causal matrix for every timestep, TAMCaD may be able to identify contemporaneous relationships. (2) By integrating Evidential Deep Learning (EDL) [1, 2], we quantify both aleatoric (data-centric) and epistemic (model-centric) uncertainties, improving the precision and interpretability of the identified causal relationships. In the TAMCaD architecture, a Temporal Convolutional Network (TCN) captures the long-range dependencies and generates a context embedding. This embedding is processed by an attention mechanism, drawing inspiration from the transformer architecture [3]. Additionally, by employing weight-sharing and recurrent layers in the TCN, we decrease the model's complexity, which acts as implicit regularization, forcing the model to focus only on the the relevant relationships in the data and improving the efficiency of the gradient computation during training due to having fewer parameters. We introduce the Soft-AUCROC metric to evaluate predicted causal relationships taking into account the confidence scores, providing a smooth ROC-curve through sampling. Lastly, this research also provides a comprehensive overview of the challenges in temporal causal discovery, covering both general challenges as well as the specific challenges associated methods suggested by prior works. We demonstrate that a TCN can identify long-range dependencies. However, distinguishing genuine relationships from spurious ones within an extensive receptive field remains challenging. By reducing the complexity of our model, we achieve performance equivalent the original model, while drastically reducing the number of learnable parameters, making the training process more efficient. While TAMCaD successfully learns non-additive relationships, its impact on the reconstruction of the causal matrix is not substantial. This suggests that additive models may be adept at identifying the most evident relationships, possibly due to the inherent regularization of their additive nature. Additionally, we show that TAMCaD is able to learn contemporaneous relationships. Furthermore, when our model is combined in an ensemble with a simpler additive model, it achieves even better scores. This underscores the robustness of additive models, and suggests that pairing them with a non-additive model can uncover complex relationships that the additive model alone would miss. Lastly, using evidential learning to express model uncertainty offers insights into the predicted causal matrix and improves the precision by filtering false negatives/positives using the confidence scores.

Contents

1 Introduction

The identification of causal relationships between variables is a fundamental concept in numerous fields, including life sciences and social science. Causal inference enables informed decision-making and a deeper understanding of the world around us [4, 5]. However, the process of inferring causality is not always straightforward, and there are several challenges associated with it. One of these challenges is spurious correlations, where correlations between variables can be mistaken for causal relationships. For example, the number of ice cream sales and the number of sunburns are highly correlated, although they do not cause each other. Additionally, there may be hidden confounding variables that affect the cause-and-effect relationship. In the example of ice cream sales and sunburns, the confounder is the temperature that causes both variables. Despite these challenges, the ability to make accurate causal inferences remains critical for decision making, the development of effective interventions, and the advancement of our understanding of complex systems [4, 6]. Consequently, there has been a growing interest in methods for reliable causal inference. With the recent increase in large available datasets and advances in computational power, machine learning techniques have become more accessible and popular [7], allowing researchers to develop new methods to better understand causal relationships between variables, resulting in more accurate predictions and more effective interventions [4]. As such, the use of machine learning techniques has become a promising avenue for advancing causal inference research.

The source code and documentation of the methods can be found on GitHub[1].

[1] `https://github.com/m4urin/temporal-causal-discovery`

2 Background and related work

2.1 Domains in Causal Machine Learning

Causal machine learning has different subdomains that focus on various aspects of causal relationships in data. These subdomains include causal inference, causal discovery, causal representation learning, causal prediction, and causal reasoning. Each subdomain presents unique challenges that require different approaches to solve. The problems and proposed solutions in the field of causality frequently involve methods that overlap between the subdomains.

Causal Inference. Causal inference refers to the process of determining the causal relationships between variables in a given system [8, 9]. While methods from causal inference are used in other subdomains, such as causal discovery and causal representation learning, the focus of causal inference is specifically on determining the causal relationships between variables in a given system based on hypotheses and existing knowledge, whereas causal discovery aims to find the unknown causal relationships between variables in observational data.

Causal Discovery. Causal discovery aims to identify potential causal relationships between variables in observational data. Identifying these variable pairs assists in determining the focus of future controlled experiments, which can further validate, refute, or adjust our confidence in these relationships. Despite the lower certainty compared to controlled experiments, the detection of cause-effect pairs from observational data can still be a crucial step in the formulation of new hypotheses, guiding further investigation. Methods for causal discovery often rely on graphical models, such as structural causal models (SCMs) or Bayesian networks, to represent the underlying causal structure [5, 10, 11, 12]. The goal is to find the most likely structure that represents the causal model that generated the observed data. This involves filtering out spurious correlations that are present due to the presence of confounders, selection bias, and other complexities in the data. Moreover, deducing the structure from data is an NP-Hard problem, as the search space of possible structure scales super-exponentially [13]. This makes it difficult to establish the causal influences in datasets containing many variables due to the computational complexity. To address this, optimization methods have been proposed that use greedy or heuristic search techniques [11, 12], while others do not rely at all on the combinatorial aspects of the graphical structure [14, 15, 16]. Causal discovery in time-series data poses additional challenges, such as the identification of temporal dependencies between variables and accounting for time lags.

Causal Representation Learning. Causal representation learning focuses on inferring high-level causal variables from low-level observations. For example, latent representations can be obtained from complex and sparse data, such as pixel information in images [17, 18]. The main objective is to distinguish true causal relationships from irrelevant or spurious associations present in the data, which can lead to more accurate models. By leveraging pre-trained causal representations, models become more robust and interpretable. One advantage of causal representations is the ability to transfer knowledge across different domains. Models trained in one domain can utilize the acquired causal understanding as a starting point for learning in other related domains. Furthermore, this transfer of causal knowledge may lead to improved predictive models, especially in domains with limited available data. Overall, causal representation learning offers the potential to make machine learning models more robust and capable of transferring knowledge across different domains.

Causal Prediction. Causal prediction considers developing statistical models that are robust under interventions (for example, a causal prediction model should be able to accurately predict the impact of interest rate changes on future stock prices). The goal is to develop models that are robust enough to generalize beyond the observed data and provide reliable predictions, even under changing conditions. One well-known method is Invariant Causal Prediction (ICP) [19], which identifies causal relationships that remain constant across different environments (e.g., various locations, patients, or timeslices in the data), provided that these environments do not interfere with the variables being studied. Although ICP works well for linear relationships, it is more challenging for nonlinear relationships due to the difficulty of performing non-parametric tests for conditional independence. To overcome this limitation, nonlinear and non-parametric versions of ICP have been proposed [20].

Causal Reasoning. Similar to the other domains, causal reasoning involves identifying cause-and-effect relationships between variables. However, the goal of causal reasoning is predicting outcomes and answering questions in a retrospective or interventional way [21]. This process often involves the use of causal models to simulate interventions, allowing the evaluation of different strategies and their potential consequences. Hence, it is unsurprising that novel approaches in the field of reinforcement learning utilize causal reasoning. In this area, agents need to reason about events retrospectively to learn from their mistakes and maximize future rewards [22]. Causal reasoning also appears in the context of recommender systems in the form of counterfactual reasoning [21, 23]. For instance, popularity bias in recommender systems can be mitigated by identifying the intrinsic properties of items that cause them to be popular.

2.2 Preliminaries and Notations for Causal Discovery

Graphical models. Graphical models are important in causality because they provide a visual representation of the relationships between variables. They allow researchers to formalize their assumptions about causal relationships and to test these assumptions using statistical methods. These models can be formalized using a set of structured equations. In the field of causality, this is often called a SCM. It is important to note that the term SCM is preferred over structural equation model (SEM) in the context of causal inference, as SEM is often used in contexts where the relationships among variables are treated as algebraic equations rather than causal relationships [9].

Definition 1 (Structural Causal Model) *An SCM is a framework used to describe the causal relationships between variables in a system [4]. It consists of a set of equations that describe how each variable is causally influenced by other variables in the system, and can be visualized using a directed graph (Figure 2.1).*

Autonomy and Invariance. Two fundamental concepts in SCMs are autonomy and invariance [9]. Autonomy implies that each variable in the model should be determined by its own set of causes, independent of other variables not directly connected to it. On the other hand, invariance suggests that the causal relationships between variables should remain constant across different populations, environments, or contexts. As shown in Equation 2.1, the value of variable X_i in an SCM is determined by its direct parents ($\mathbf{Pa}_i$) through the function f_i. The concept of autonomy corresponds to the fact that X_i is influenced only by $\mathbf{Pa}_i$, while invariance refers to the consistency of the function f_i across various conditions.

$$X_i = f_i(\mathbf{Pa}_i) \tag{2.1}$$

If invariance is violated, it can lead to misspecification of the model, where either f_i or $\mathbf{Pa}_i$ may not accurately capture the true causal relationship for variable X_i.

Visualization. Graphical representation is a powerful tool to visualize SCMs, where nodes represent variables and edges indicate the causal relationships between them (see Figure 2.1). Each node is associated with a structural equation that describes how the variable's value depends on the values of its parent nodes in the graph [4].

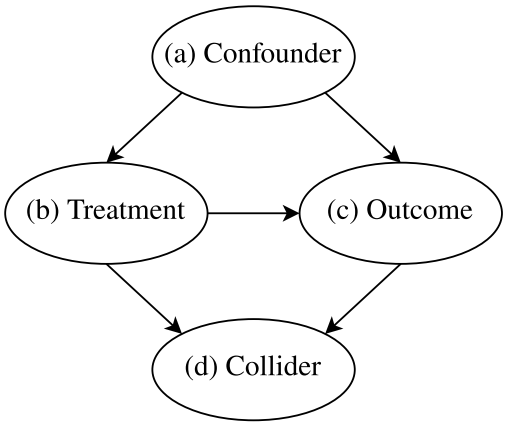

Figure 2.1: An SCM consisting of four variables represented as a DAG.

Intervention in causal models. Intervention in a causal model involves manipulating a variable to achieve a specific desired outcome. This allows for the analysis of cause-and-effect relationships between variables in the system. However, it is important to note that this type of analysis is not applicable to observational data, such as observations in the stock market. Interventions can generally be categorized as either soft or hard interventions. A hard intervention consists of setting a variable to a constant value and severing its causal connections with its parent variable, denoted as do$(X = x)$ [24]. This approach is typically used to unveil direct cause-and-effect relationships. For instance, in a randomized controlled trial (RCT), participants are assigned to either a treatment or a control group. On the other hand, soft interventions involve adjusting a variable while preserving its causal connections to its parents, thereby altering only its conditional probability. Formally, this type of intervention imposes a specific functional relationship $g(z)$ on the variable X in response to a set Z of other variables. It is represented as do$(X = g(z))$, and its effects can be observed by examining the distributions after the intervention [25]. Soft interventions are widely employed in biology and medicine, where completely removing parental influences is challenging, but perturbing them is more feasible [26].

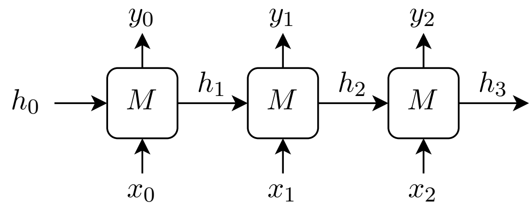

Figure 2.2: A recurrent model (M) processing temporal data (x) sequentially. Every prediction (y) relies on the input as well as a context vector (h), which has a fixed representational capacity.

2.3 Deep Learning for Time Series Prediction

2.3.1 Recurrent Neural Networks

Deep learning architectures have significantly improved the accuracy of time series predictions. These models can effectively identify complex temporal dependencies and non-linear relationships in the data. Most of these models process temporal data sequentially. A recurrent neural network (RNN) is good example of such a model. RNNs have hidden states that enable them to capture temporal dependencies and make predictions based on past observations (see Figure 2.2). However, traditional RNNs suffer from vanishing and exploding gradient problems, limiting their effectiveness for long-range dependencies. To address these challenges, methods such as Long Short-Term Memorys (LSTMs) and Gated Recurrent Units (GRUs) were proposed, aiming to control the flow of information, enabling better long-term memory retention and learning. While the primary strength of an LSTMs is their long-term memory, their retention can be inconsistent over time. This might lead an LSTM to preserve contextual information of specific variables for a longer duration compared to others, potentially undermining the learning of relationships that are expected to remain static throughout the entire time series.

2.3.2 Convolutional Models

Convolutional models were originally designed to process two-dimensional grid structures like images. However, they can also be adapted to (multivariate) time series data by applying them to the time dimension. These models can effectively capture local patterns and dependencies within the time series using convolutional layers, making them useful for tasks where local context is crucial for making predictions [27]. They have demonstrated significant potential across a variety of time series forecasting domains, including energy load forecasting [28], weather forecasting [29], stock market prediction [30], as well as multiple computer vision tasks [31, 32]. While both sequential and convolutional models have their strengths and weaknesses, their effectiveness can vary depending on the specific characteristics of the time series data and the type of prediction task. However, in many cases, a simple convolutional architecture outperforms canonical recurrent networks like LSTMs across a diverse range of tasks and datasets, while demonstrating longer effective memory [27, 29]. For example, utilizing an LSTM did not yield significantly better results compared to using a simple 1-layer convolution within the Neural Additive Vector Autoregression (NAVAR) framework [15]. Therefore, convolutional networks should be regarded as a natural starting point for sequence modeling tasks [27]

Temporal Convolutional Networks. Temporal Convolutional Networks (TCNs) are a specific type of convolutional neural networks that excel in capturing long-term temporal dependencies within time-series data [31]. This architecture relies on stacked dilated convolutions, providing an exponentially growing receptive field the more layers are added, which is crucial for handling long-range dependencies in the data. One significant advantage of TCNs over most sequential models is their ability to process data in parallel across layers, resulting in faster training and inference times. Empirical studies have shown that TCNs outperform traditional recurrent architectures like LSTMs in various tasks [27]. However, it is important to note that the optimal architecture for time series prediction will vary depending on the specific dataset and problem domain.

By default, conventional TCN implementations allow predictions to be influenced by both past and future data. While this is suitable for some applications, it presents challenges for time-series prediction tasks where future data is not available. To address this concern, a causal variant of TCN has been introduced, which relies solely on current and past data to compute outputs, strictly adhering to the temporal order of the input sequence. This property makes the causal TCN particularly suitable for real-world prediction tasks where future information is limited or unknown. Given that the NAVAR framework's main objective is regression, our approach will leverage the causal variant of TCN. Moreover, the inclusion of residual connections in the TCN model enhances its capability to handle temporal dependencies effectively, addressing issues such as vanishing or exploding gradients.

The schematic overview of a TCN can be observed in Figure 2.3. Each layer in the schematic (left) represents a temporal block (right). In its implementation, this block is actually a two-layer 1D convolutional network. This further increases the receptive field and enhances the model's flexibility and expressiveness. However, it's worth noting that a single value in the time series has multiple paths to the final output. Therefore, if the goal is to determine the correct number of lags by inspecting the convolutional weights in a hierarchical setting, like Temporal Causal Discovery Framework (TCDF), it is necessary to either reduce the temporal block to a single layer or address this issue in a different manner.

The receptive field (RF) of a TCN can be computed using the following equation:

$$\mathrm{RF}(k, \ell, b) = (k-1) \cdot \ell \cdot (2^b - 1) + 1 \tag{2.2}$$

Here, k represents the kernel size, which corresponds to the width of the convolutional filters employed in the TCN. The number of layers within each temporal block is denoted with ℓ, and b represents the number of temporal blocks in the network. In the original work the temporal block consists of two convolutional layers, defaulting b to 2.

2.3.3 Attention-Based Models

Attention mechanisms were primarily developed for sequence-to-sequence models, especially in the domain of neural machine translation. These attention mechanisms allowed for dynamically selecting a subset of the information from the source sequence during each step of the target sequence generation. At every decoding step, a context vector was generated as a weighted sum of hidden states. The computed context vector, combined with the decoder's hidden state, was then used to predict the next word in the target sequence. These mechanisms alleviated the model from compressing all source information into a single fixed-size context vector, as seen with LSTMs or GRUs. By dynamically selecting context, models can handle long sequences more efficiently, which results in significant improvements in tasks like machine translation.

Scaled Dot-Product Attention. The Scaled Dot-Product Attention is an improved attention mechanism that was introduced as part of the Transformer model architecture [3], which has now become the foundation for many state-of-the-art models in natural language processing models. One features of this approach is its ability to determine relationships between all elements in a sequence, not just adjacent or

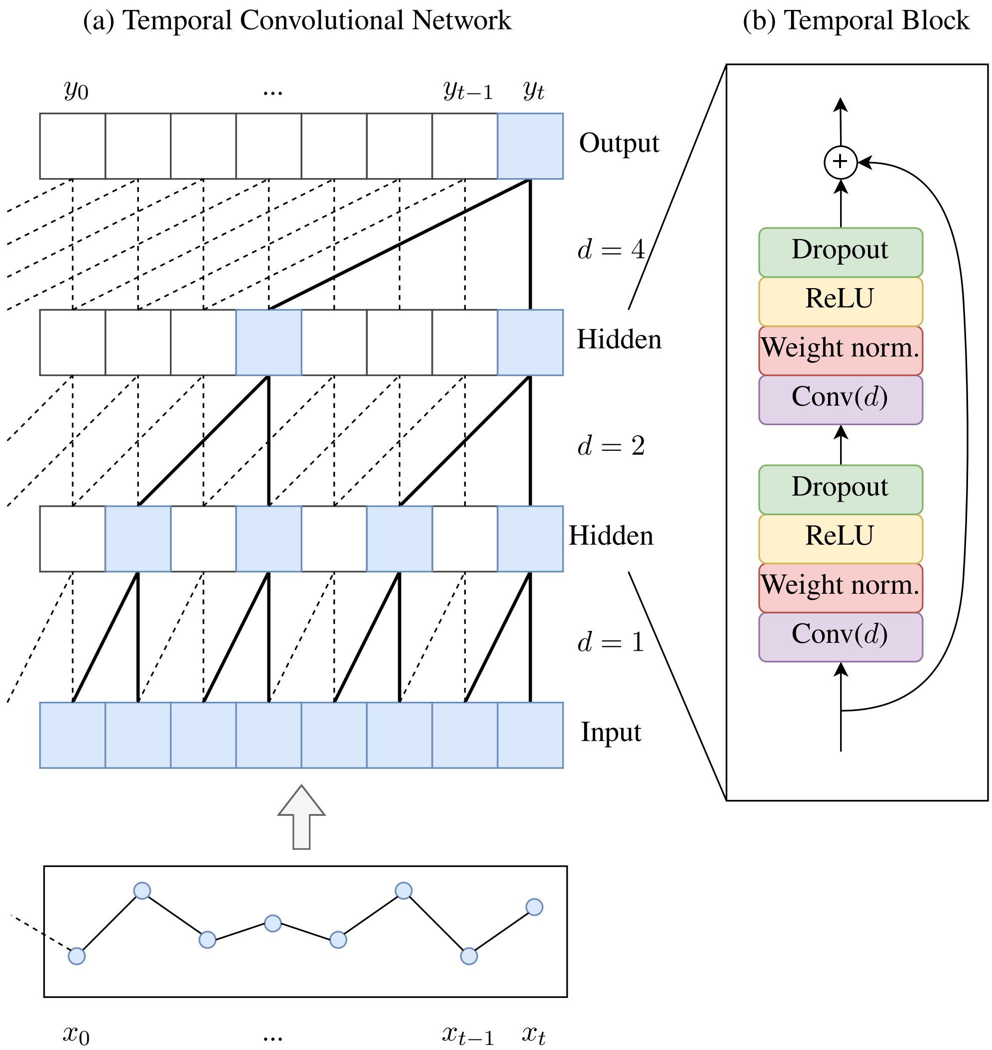

Figure 2.3: A Temporal Convolutional Network comprised of three temporal blocks (b). (a) The architecture stacks multiple layers with filters of increasing dilation d to achieve exponential growth of the receptive field. (b) Each temporal block typically consists of two convolutional layers (ℓ) and incorporates a residual connection.

near-adjacent ones. This contrasts with many traditional sequence processing methods which typically consider local patterns or relationships. Furthermore, these traditional attention mechanisms encountered challenges like computational inefficiency, scaling limitations, and fixed representational capacity, which is problematic for data with longer sequences. The Scaled Dot Product Attention addresses some of these concerns by employing dot product computations for attention scores, which is computationally efficient compared to learning another linear model. Additionally, multi-head attention allows focus on various input parts.

The attention mechanism is given as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V \tag{2.3}$$

A more intuitive illustration of these steps are provided in Figure 2.4. This attention mechanism is defined by three primary matrices: Q, K, and V. These matrices are derived from three distinct linear transformations of the input data, each capturing relevant information. $Q \in \mathbb{R}^{N \times d_q}$ represents the query matrix. Here, N denotes the number of input embeddings, and E represents the embedding size of the input that will be partitioned in h heads of size d. The query matrix is responsible for encapsulating the information that needs attention within the input data. $K \in \mathbb{R}^{N \times d_k}$ and $V \in \mathbb{R}^{N \times d_k}$, on the other hand, may have differing dimensions compared to Q, but often, they are the same size as Q. Matrix V represents the values that the model aims to broadcast to the other inputs. These matrices enable the attention mechanism to effectively capture and aggregate relevant information. The input can be split along the embedding dimension into multiple heads, which are concatenated back to the original embedding size after applying the scaled dot product attention to each head. The output of the softmax operation is an $N \times N$ matrix, representing the attention matrix between N inputs. The softmax function guarantees that an attention vector $\mathbf{a}_i \in \mathbb{R}^N$ from the attention matrix responsible for incorporating data from V to a new embedding is a simplex ($a_{ij} \geq 0 \wedge \sum_j a_{ij} = 1$).

2.4 Temporal Causal Discovery Methods

In scientific research, it is assumed that the cause precedes its effect in time, known as time asymmetry or causal precedence in the context of causal discovery. This assumption is particularly relevant in time series analysis, where observed variables at a given time point are assumed to be influenced by variables at previous time steps, assuming that there are no instantaneous effects [33]. The natural temporal ordering in time series data can also be advantageous for causal discovery, as it narrows down the potential causal relationships. Incorporating the temporal aspect introduces unique challenges for causal discovery, such as endogeneity due to feedback loops, non-stationarity, history-dependent noise, and time lags. These challenges will be explained in more detail in Section 2.6.

Discovering causal relationships among variables in a system from time series data is a complex task. Over time, various methods have been developed, ranging from traditional statistical approaches to more advanced deep learning techniques. This section explores various concepts and methods for temporal causal discovery, including both well-established techniques and novel approaches in the field, highlighting their strengths and limitations.

Granger causality. Granger causality is a statistical concept that measures the predictive power of one time series on another [33]. If timeseries A Granger causes time series B, this indicates that the historical values of A can improve the prediction of future values of B, even after accounting for the past values of B. Granger causality does not imply a direct causal link between A and B, but

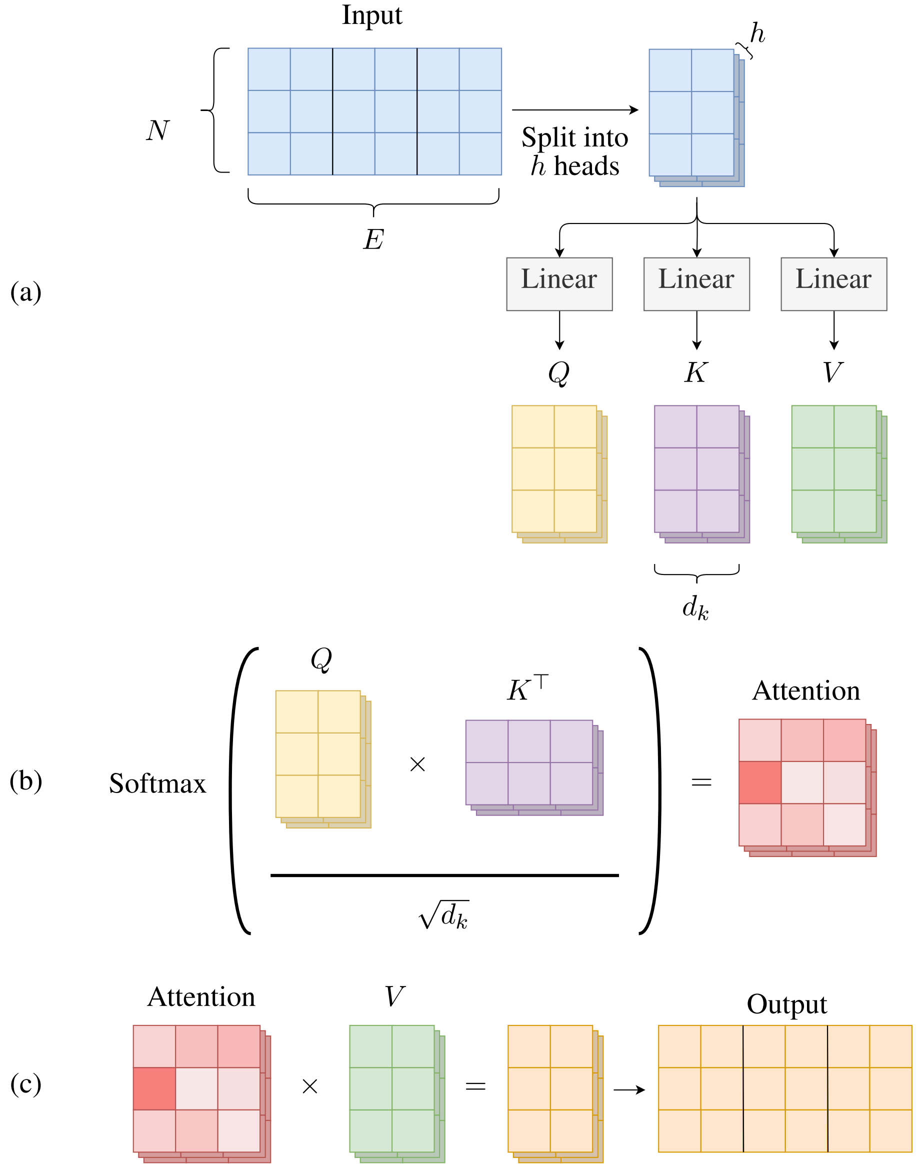

Figure 2.4: Depiction of the Multi-Head Attention Mechanism. (a) The initial input of dimensions $N \times E$, with N representing the number of embeddings and E denoting the size of each embedding, is partitioned into h separate heads across the embedding dimension. Each head is passed through three separate linear transformations, resulting in Query (Q), Key (K), and Value (V) matrices with dimensions $N \times d$. (b) The attention matrix, of size $N \times N$, for each head is calculated through the application of the softmax function to the multiplication of Q and K^T, scaled by the square root of d_k, the embedding dimension of K. (c) The output for each head is generated by matrix multiplication of its individual attention matrix with the corresponding Value (V) matrix. These head-specific outputs are then concatenated to produce the final output, matching the dimensionality of the original input.

rather a statistical association that helps to understand the dependencies between variables in a system. Fields such as economics, finance, and neuroscience often use Granger causality to investigate causal connections between variables in time series data.

Equation 2.4 extends the SCM formulated in Equation 2.1 into a Granger causality framework, where $X \in \mathbb{R}^{N \times T}$, and X_t^i denotes the value of variable i at timestep t. Parents $\mathbf{Pa}_{<t}^i$ denotes all past values of parents that have a causal impact on X_t^i.

$$X_t^i = f_i(\mathbf{Pa}_{<t}^i) \tag{2.4}$$

2.4.1 Additive Models

Generalized additive models. A generalized additive model (GAM) is a type of linear model that incorporates a set of functions f_ij to predict variables based on predictor variables [34]. The challenge is to approximate these functions f_ij, which can be parametric or non-parametric.

$$X_i = \beta_i + \sum_{j=1}^{N} f_{ij}(X_j) \tag{2.5}$$

GAMs have several advantages and limitations. The model enables multivariate analysis, meaning multiple predictor variables can be modeled simultaneously, leading to a better understanding of the relationships between the variables. Missing data can be handled by excluding any function f_ij at prediction time if, for example, variable i is not present. In this way, a prediction can still be made based on the available data. The additive nature of the model allows for the investigation of the interactions between variables, making it easy to interpret the model. However, this additive nature also poses a serious limitation, as important interactions between variables may be missed. Furthermore, GAMs may overfit when the complexity of individual functions f_ij is high, when there is an inherent lack of regularization, or when sample sizes are too small [35].

$$X_t^i = \beta^i + \sum_{j=1}^{N} f_{ij}(X_{<t}^j) \tag{2.6}$$

As shown in Equation 2.6, the notion of time can also be incorporated into the model. In this case, the complete history of a single variable is used as a predictor. However, this can even be extended to a model in which a function is learned for each time lag. This increases the interpretability of the lagged dependencies at the cost of performance and expressiveness of the learned functions.

Vector Auto-Regression. A common approach for identifying relationships between variables within the framework of Granger causality is the use of vector auto-regression (VAR) models, as these can be used for modeling multiple time series together in a joint manner [14]. VAR models identify the dynamic relationships between variables by including the past values (lags) of each variable as predictors of its current value, as well as past values of all other variables in the system. VAR models are particularly well suited to study Granger causality because they allow us to estimate the causal relationships between all variables in a system simultaneously.

$$X_t^i = \beta^i + \sum_{j=1}^{N} \sum_{\tau=1}^{K} [A_\tau]^{ij} X_{t-\tau}^j + \eta_t^i \tag{2.7}$$

In this model, $[A_\tau]$ represents the time-invariant adjacency matrix for variables at time step $t - \tau$. Therefore, the coefficients of this 3-dimensional matrix can be interpreted as an SCM that also includes information about the lagged causal influences. By learning the coefficients of a VAR model, the direction and strength of the causal relationships between variables can be measured within the framework of Granger causality. In essence, a VAR model is a GAM in which the functions f_{ijk} are obtained as linear scalars. The noise term η is often used in linear regression and reflects a constant variance.

While traditional VAR models with coefficients can identify linear relationships between variables, recent studies have attempted to create non-linear VAR models that employ neural networks to approximate non-linear relationships. This is necessary because observed relationships in real-world data are often non-linear [15, 16]. However, solely incorporating deep learning techniques to approximate the causal relationships does not resolve all the challenges related to causal discovery, which will be elaborated on in Section 2.6.

Neural Additive Vector Autoregression. NAVAR is a neural approach to causal structure learning that can discover nonlinear relationships in time series data. The method consists of training a neural network for each of the variables, using the past K values of each variable as input, and attempting to predict the contribution to the other variables at each timestep, denoted by $c_t^{j\to i}$. Subsequently, using an additive model allows for the aggregation of these contributions to produce a final prediction.

$$c_t^{j\to i} = f_{ij}(X^j_{t-K:t-1}) \tag{2.8}$$

$$X_t^i = \beta^i + \sum_{j=1}^{N} c_t^{j\to i} + \eta_t^i \tag{2.9}$$

If a contribution made by one of the neural networks significantly contributes to the final prediction compared to those of other variables, this may suggest a causal relationship between the two variables. To quantify this causal relationship, the standard deviation is computed over all time steps for a single relationship, as shown in Equation 2.10.

$$\text{score}(j \to i) = \sigma(c_{K:T}^{j\to i}) \tag{2.10}$$

The scalability of such models becomes infeasible with an increasing number of variables, as we need to train a separate model for each variable using NAVAR. While the training time and the number of parameters that must be trained may increase linearly with the number of variables, which is an improvement compared to methods where the computation time increases super-exponentially [11], training additive models can still be considered slow due to the significant computational resources that most deep learning approaches require. One potential solution to this challenge is to introduce weight sharing across models [16, 36]. This approach leads to a single model that takes an additional embedding representing a variable as input, which contains information about the relationship of a variable to all other variables in a system (this embedding can be interpreted as a form of representation learning). However, to the best of our knowledge, the trade-offs between the expressiveness of such an embedding and the efficiency of this approach have not been investigated.

Temporal Causal Discovery Framework. The TCDF is a method closely related to NAVAR, as they share a similar architecture. TCDF utilizes an adapted version of the TCN architecture and incorporates attention weights in the first convolutional layer to determine the significance of a variable in predicting another variable [37]. The idea behind TCDF is to train an individual models for each variable within a system, and this training process can be efficiently parallelized using a "depthwise separable" architecture. In the final layer, a pointwise convolution is applied to the model outputs to obtain the regression prediction for a time series representing a specific variable. NAVAR takes a single variable as an input for each model and aims to predict all other variables for each model. In contrast, TCDF takes all variables as inputs and strives to predict just a single variable. Though not explicitly stated by the authors, the TCDF model exhibits an additive nature, though the 1×1 convolution that combines the outputs of the TCN into a single value, without the need for an activation function. However, instead of analyzing the contributions like NAVAR, TCDF examines the learned attention weights during a causal validation step. Additionally, TCDF accounts for instantaneous causal effects by including the current values of all other variables, except the variable predicted, in the input.

Mathematically, the TCDF model can be represented as follows, where the output channels of the TCN are transformed to the N channels using the pointwise convolution:

$$X_t^{(i)} = \sum_{j=1}^{N} \mathbf{w}_{ij} \odot \text{TCN}_{ij}(a_{ij} \odot X_{t-K:t-1}^{(j)}) \tag{2.11}$$

In this equation, $\mathbf{w}ij$ represents the weight vector used to transform the output channels produced by the TCN, aij is the attention weight parameter, and the value of K determines the receptive field of the TCN. By utilizing this weight vector to transform the output channels of various TCNs into a single value for each of the other variables, TCDF operates in a similar manner to the contributions observed in NAVAR, resulting in an additive model.

Furthermore, the authors not only investigate the attention weights but also examine the weights of the convolution kernels to estimate the lags for each variable. This analysis of both attentions and weights poses a strength of the framework and has the potential to improve the analysis of the retrieved causal matrix in other methods as well, such as the interpretation of the contributions in NAVAR. However, one drawback of the TCDF approach is the lack of regularization in the network's objective function. For example, the absence of regularization in the attention weights allows the network to learn high attention weights a_{ij}, suggesting a strong influence of variable i on variable j. Simultaneously, the network may learn $\mathbf{w}_{ij}$ weights of 0, effectively negating the influence entirely. This issue compromises the interpretability of the model and may lead to less reliable causal inferences.

2.4.2 Variational Models

Variational inference. Variational inference is a method that involves approximating complex posterior probability distributions with simpler and more tractable distributions. Given a dataset D, where $x \in D$, learning the posterior probability distribution $p(z|x)$ with a latent representation z can be intractable for larger datasets. Instead, one can learn a simpler distribution from a variational family that minimizes the Kullback-Leibler (KL) divergence, which measures the difference between the learned distribution and a normal distribution $N(0, 1)$. To achieve differentiable computation in neural models, stochastic methods, such as the reparameterization trick, can be used to sample from the learned distribution, enabling the learning of this objective function with gradient descent. This concept has been successfully applied to variational autoencoders (VAEs) [38] to learn a latent representation as a distribution. This distribution can be sampled and then used as input to the decoder. Additionally, β-VAE [39] introduces an additional hyperparameter β in the loss function that balances the decoder loss and the KL divergence loss. This leads to learning a minimal posterior probability distribution that describes the dataset, which extends the interpretability of autoencoders. Furthermore, the variational approach can also be employed in the last layer of a model that makes a final prediction, instead of in a latent representation in an autoencoder. In this way, the probability distribution of the prediction can be learned given a certain input, which can be interpreted as the uncertainty of a prediction.

Rhino. Recently, Rhino was introduced [16], combining VAR, deep learning, and variational inference to effectively model non-linear relationships with instantaneous effects while incorporating historical observations to modulate the noise distribution. This leads to more accurate noise distribution modeling, as Rhino considers past actions that may have influenced the noise distribution. The functional relationships between variables are captured with differentiable functions denoted as f_i and g_i, where g_i transforms the noise term ϵ_t^i. The relationship between f_i and the VAR model is evident:

$$X_t^i = f_i(\mathbf{Pa}_G^i(<t), \mathbf{Pa}_G^i(t)) + g_i(\mathbf{Pa}_G^i(<t), \epsilon_t^i) \tag{2.12}$$

In this formula, $\mathbf{Pa}_G^i$ are the parents of variable i at time steps $< t$ and instantaneous time step t. In essence, function f_i is closely related to the VAR model:

$$f_i(\mathbf{Pa}_G^i(<t), \mathbf{Pa}_G^i(t)) = \zeta_i\left(\sum_{\tau=0}^{K}\sum_{j=1}^{D} G_{\tau,ji}\ell_{\tau j}(X_{t-\tau}^j)\right) \tag{2.13}$$

Here, ζ_i and $\ell_{\tau i}$ represent non-linear neural networks, and G denotes a parameterized causal matrix for K lags. The non-linear function ℓ transforms the input into a latent space, which is then combined in an additive manner using G. Finally, ζ further transforms these values to produce the final prediction. The function g_i, on the other hand, is a conditional spline flow [40], taking a similar form to f_i but excluding the instantaneous parents. Its primary purpose is to transform the noise term ϵ_t^i to ensure a proper density for more accurate noise distribution modeling, ultimately enhancing the overall model performance. Moreover, Rhino applies variational inference over the causal matrix G to learn a distribution, rather than a direct causal matrix.

The experimental results show Rhino's reasonable robustness to history-dependency mismatch and achieves the best performance when correctly specified. However, the research references the NAVAR method without including it in the benchmark results. It is shown that Rhino acquires the best results on the ecoli/yeast benchmark, but only when the experimental results from NAVAR are excluded.

2.5 Methods for Quantifying Uncertainty

Uncertainty in predictions can be divided into two categories: epistemic and aleatoric. Aleatoric uncertainty refers to the unpredictability inherent in the data. Models can account for this type of uncertainty by, for example, outputting both a mean and variance for a prediction, which effectively captures the underlying distribution. Consequently, a model can potentially learn the (history-dependent) noise present in the data. On the other hand, epistemic uncertainty relates to the confidence of the model in its predictions. This raises the question: Does the model recognize when it knows or doesn't know the answer? Addressing this form of uncertainty proves to be more challenging. It's crucial to understand that a probability output or an attention output produced by the softmax operation should not be confused with the confidence of the model.

2.5.1 Aleatoric Uncertainty Methods

Negative Log-likelihood. To learn the uncertainty associated with the data in regression tasks, the Gaussian Negative Log Likelihood (NLL) loss is often used. By treating the output of the model as a probability distribution over the target variable, the NLL gives a measure of how well the model's predictions align with the observed data. Specifically, the model is tasked to predict both the mean and the variance of the target distribution as:

$$\mathcal{L}_{\text{NLL}}(\theta)_i = \log \sigma_i^2 + \frac{(y_i - \mu_i)^2}{\sigma_i^2} \tag{2.14}$$

2.5.2 Epistemic Uncertainty Methods

Approximating Epistemic Uncertainty Through Sampling. Epistemic uncertainty can be approximated using stochastic sampling. This involves evaluating many prediction outputs by using only a subset of model weights for each prediction. Several approaches can achieve this, such as: (1) Monte-Carlo Dropout, which involves activating a dropout layer during the testing phase. (2) Using an ensemble of models where each model provides independent predictions. (3) Implementing a Bayesian Neural Network (Bayesian Neural Network (BNN)) that learns a distribution over all its weights. However, it should be noted that ensemble models and BNNs might introduce significant computational overhead, especially when dealing with larger models.

Evidential Deep Learning Evidential deep learning quantifies uncertainty by treating learning as an evidence-gathering process. The goal is to directly estimate both aleatoric and epistemic uncertainty. This is done by placing prior distributions over the likelihood parameters for predicting aleatoric uncertainty. For example, in cases with low uncertainty, the distribution around the mean (μ) and variance (σ^2) concentrates at a specific point, indicating high confidence. On the other hand, an increased variability in μ values indicate a high epistemic uncertainty. Training neural networks to learn and use these evidential distributions is the challenge.

In Evidential Deep Learning (EDL), a Dirichlet distribution is used as a prior for modeling uncertainties in predicted categorical probabilities for each class [1]. Here, $\mathbf{p}$ is the probability density function that can be sampled from the Dirichlet distribution:

$$\begin{gathered} y \sim \text{Categorical}(\mathbf{p}) \\ \mathbf{p} \sim \text{Dirichlet}(\boldsymbol{\alpha}) \end{gathered} \tag{2.15}$$

Introducing a two-stage learning framework significantly enhances uncertainty estimation in classification tasks, boosting AUC and training robustness [41]. The proposed method can be applied to various types of deep learning models, making it a useful method in various domains.

Deep Evidential Regression (DER) is a method for estimating uncertainty in regression [2]. Here, μ and σ^2 denote the aleatoric uncertainty. The model is also required to learn parameters α, β and γ, to model the distribution over the aleatoric uncertainty:

$$\begin{gathered} y \sim \mathcal{N}(\mu, \sigma^2) \\ \mu \sim \mathcal{N}(\gamma, \sigma^2 v^{-1}) \quad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta) \end{gathered} \tag{2.16}$$

Despite DER's empirical success, gaps in its mathematical foundation raise questions about its workings [42]. DER appears to be a heuristic for uncertainty, not an exact quantification. The authors call for corrections in how aleatoric and epistemic uncertainties should be extracted from NNs and propose a simplified version of the loss function, which is similar to the negative log likelihood loss function:

$$\mathcal{L}_{\text{DER}}(\theta)_i = \log \sigma_i^2 + (1 + \lambda v_i)\frac{(y_i - \mu_i)^2}{\sigma_i^2} \tag{2.17}$$

In this loss function, v can be considered as a scalar related to the error margin. For samples where the error is nearly zero, v has minimal impact. Conversely, for samples with a considerable error margin, v should be minimized. The epistemic uncertainty can be computed as v^{-1}.

Gaussian Processes. Gaussian Processes (GP) are a class of non-parametric Bayesian models used for regression and classification tasks [43]. Their flexible non-parametric nature and computational simplicity makes them popular choice [44]. GPs can be useful for time series prediction, because they not only provide a prediction for future values but also quantify the uncertainty associated with these predictions. However, one drawback is that they are computationally expensive. The complexity for a basic GP model scales as $O(n^3)$, with n being the number of data points, making them infeasible for large time series.

2.6 Challenges in Causal Discovery

Dynamic temporal relationships. Observed relationships can be classified into three categories: static, contemporaneous and sequential [45]. Static relationships remain constant over time, such as the dependence of the current temperature on the previous temperature. Contemporaneous relationships, on the other hand, occur within a finite time window and may require additional contextual information for proper understanding. An example of a contemporaneous relationship is the relationship between temperature and the amount of sunlight, which weakens or disappears when clouds block the sun. Sequential relationships involve spontaneous causal effects that occur at specific time points. For example, the occurrence of a hurricane leads to damage. This relationship is difficult for models to detect because it occurs infrequently in the data. Although the observed relationships can be classified into these categories, we can argue that the true underlying causal structure is inherently static. However, it is impossible to model every variable involved in the causal process. Thus, the problem is abstracted by using only a subset of variables that can be measured, resulting in observed relationships in the data that are not static.

For example, given the following relationship:

$$X_t^{(1)} = X_{t-1}^{(2)} \cdot X_{t-1}^{(3)} \tag{2.18}$$

If $X_t^{(3)}$ converges to 0 over time, the overall contribution to $X_t^{(1)}$ will be 0. In other words, $X_{t-1}^{(3)}$ "regulates" the relationship between $X_{t-1}^{(2)}$ and $X_t^{(1)}$ and vice versa. When both variables are observed, a model may capture this static relationship. However, if one of the variables is not observed, the relationship between $X_t^{(2)}$ and $X_t^{(1)}$ may appear contemporaneous.

Feedback loops. The presence of cycles within a causal graph would create logical inconsistencies that make it difficult to determine the direction of causality or to estimate causal effects [9]. In this case, the causal structure is represented in the form of a directed acyclic graph (DAG). However, when discovering causal relationships in time series data, the graphical representation need not be acyclic since the temporal ordering of variables provides a natural direction for causal effects. Therefore, cyclic causal models can be used to represent feedback loops or other recursive relationships frequently observed in time series data [9]. As shown in Figure 2.5, is still possible to convert an SCM into a DAG by expanding the nodes for each variable at each time step. It is important to note that statistical methods that typically work for DAGs cannot be used on these expanded graphs when working with a time series of variables, since there is only one sample per node, and it is impossible to obtain additional samples, since the data from another time series cannot be matched to the first.

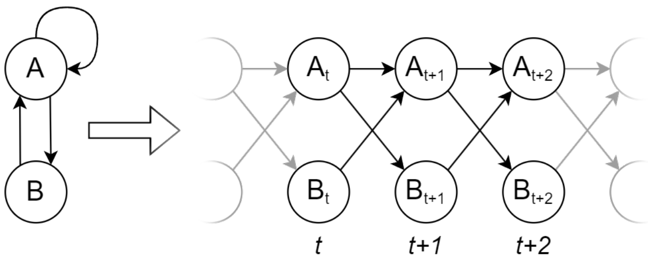

Figure 2.5: Expansion of a temporal SCM into a DAG.

Endogeneity and auto-correlation are two related concepts that can pose challenges for causal inference. Endogeneity occurs when feedback loops exist between variables, making it more difficult to determine the direction of causality between these variables, as all the variables will be correlated. Auto-correlation refers to the correlation between a variable and its past values. For instance, tomorrow's temperature is dependent on today's temperature. Therefore, it is important to consider both endogeneity and auto-correlation in time series data to avoid bias and ensure accurate causal inference. One strength of VAR models is the ability to capture the temporal relationships between all variables simultaneously, which allows for learning bidirectional causality and feedback loops.

Non-additive Relationships. Methods for causal discovery in time series data are often based on a GAM. This approach assumes that all variables are independent to a degree where relationships between these variables can be described as a summation of various functions, which is not usually the case in naturally occurring datasets [46].

$$f(\mathbf{X}) = f_1(X^{(1)}) + f_2(X^{(2)}) + .. + f_n(X^{(n)}) \tag{2.19}$$

In order to fully approximate the model using an additive approach, all possible permutations of sub-variables must be captured by including a larger number of functions. However, since $X \in \mathbb{R}^k$, this results in 2^k functions, and only a fraction of them are actually useful for approximating f. By identifying the functions that have the greatest impact, the causal relationships between sub-variables within a causal model can be uncovered. However, the computational complexity of identifying these functions increases exponentially with the number of variables, making it a challenging topic in the field of causal discovery.

The methods discussed in Section 2.4 follow an additive approach. This is because the effects of variables on other variables must be interpretable and the mixing of features in neural-based models makes it difficult, if not impossible, to interpret these effects. Methods to handle non-additive relationships in the context of multiple regression are proposed by [47]. This involves learning additional terms that consist of subsets of variables, resulting in an interpretable additive model where variables may be coupled. However, this approach introduces a new combinatorial issue as all potential subsets of variables must be included. Furthermore, to the best of our knowledge, there are no robust approaches that explicitly handle non-additive causal relationships in a temporal setting. Investigating to what extent non-additive relationships occur in real-world datasets is essential in addressing this issue.

Transitive Relationships. In some cases, the true causal relationship can be deduced from the past values of other variables. In Equation 2.20, the dependency between $X^{(2)}$ and $X^{(3)}$ in f_1 can be decoupled, given the knowledge about the structural causal model.

$$\begin{aligned} X_t^{(1)} &= f_1(X_{t-1}^{(2)}, X_{t-2}^{(3)}) + \eta_t^{(1)} \\ X_t^{(2)} &= f_2(X_{t-3}^{(3)}) + \eta_t^{(2)} \\ X_t^{(3)} &= f_3(X_{t-5}^{(3)}) + \eta_t^{(3)} \end{aligned} \tag{2.20}$$

As $X^{(2)}$ is dependent on $X^{(3)}$, a new relationship can be deduced for $X^{(1)}$ based on values of signal $X^{(3)}$ alone (Equation 2.21). Rather than learning functions f_1 and f_2, NAVAR can make accurate predictions by learning a simplified function h.

$$\begin{aligned} X_t^{(1)} &= f_1(f_2(X_{t-4}^{(3)}), X_{t-2}^{(3)}) + \eta_t^{(1,2)} \\ &= h(X_{t-4}^{(3)}, X_{t-2}^{(3)}) + \eta_t^{(1,2)} \end{aligned} \tag{2.21}$$

The difficulty here lies in the fact that NAVAR will accurately identify that $X^{(3)}$ has a contribution on $X^{(1)}$. However, as the regularization in NAVAR aims to reduce unnecessary (spurious) contributions, the model will learn to disregard information from signal $X^{(2)}$. This will lead to an incorrect causal matrix where $X^{(2)}$ is considered not to be a contributing factor to $X^{(1)}$.

Instantaneous causal effects. The notion of instantaneous causality refers to the idea that a cause and its effect can occur without any time lag between them. However, missing information due to the sampling and aggregation process of the data can make it seem like there is an instantaneous causal relationship between variables even when there is not. Sampling involves selecting a subset of data points from a time series data set, often because the raw data is too large or too detailed to process efficiently, or only certain time points are of interest. In other scenarios, such as physically measuring a system, sampling with a smaller interval may not be possible. Aggregation, on the other hand, involves summarizing and condensing data over a specific period of time. Nevertheless, these processes can lead to the loss of valuable information that could prove useful in the causal discovery process. To address instantaneous causality in regression analysis, methods may use the most recent data of input variables except for the variable being predicted, such as using $X_{<t}^{(i)}$ and $X_{\leq t}^{(j\neq i)}$ to predict $X_t^{(i)}$ [37].

Discrete and continuous time-series. The type of time-series data used can have a significant impact on the modeling and analysis process. Discrete time-series data is often used when there is a discrete set of time intervals and specific time lags between the causes and effects being studied. For instance, given $X_t^{(1)} = X_{t-1}^{(2)} + X_{t-1}^{(3)}$, the resulting time series for $X_t^{(1)}$ is not continuous and may be very volatile. In contrast, continuous time-series data is measured or generated over a continuous and uninterrupted period of time, such as in differential equations. These SCMs are inherently autocorrelated. For instance, the following continuous time series may be approximated with $X_t^{(1)} = 0.99 \cdot X_{t-1}^{(1)} + 0.01 \cdot X_{t-1}^{(3)}$. As the values in these time series may change very slowly, lags are more of a range rather than a single time step. Although a higher sampling rate can lead to more accurate models, this can be more of an issue for continuous time-series data than for discrete time-series data. However, the models proposed in Section 2.4 do not allow modeling variability in the lags. This is more of a problem for continuous time series and not for discrete time series. Since the values of continuous time series may change very slowly, a possible solution may be to take a subset of the original time series at a larger interval. Another approach could be to apply techniques that have an increased receptive field, and is able to capture these dependencies.

Long-Range Dependencies. Long-range dependencies within temporal data are dependencies where present values are influenced by distant historical values. Such dependencies can be observed in various domains: the stock market's reaction to events from years prior, crucial turning points in the climate system that shape the future patterns of the climate, or textual callbacks to earlier contents in a book. One challenge is to determine the span of historical data impacting the present. Does a value significantly depend on a value from the distant past, or is it primarily influenced by more recent data? Discrete data may show periodic patterns or sudden changes, while continuous data often reflects smoother transitions influenced by longer historical contexts.

Modeling these dependencies may improve the accuracy of predictions and help to understand the underlying causal influences in a system. One approach to discover long-range dependencies is to expand the receptive field of a model. However, excessively extending the receptive field leads to a larger search space, increased computational costs and can obfuscate the model's ability to pinpoint the true causal relationships. Furthermore, there's a risk of overfitting, where the model unnecessarily uses non-representative long-range patterns, compromising its performance on unseen data.

Confounding. Confounding is an important concept in scientific research that can lead to inaccurate estimates of treatment effects. It occurs when a variable, that is not being studied, is associated with both the treatment and the outcome [4, 48] (see Figure 2.1). For example, if a study evaluating the effectiveness of a new drug does not account for confounding factors such as age or sex, any observed improvement in health outcomes could be due to the differences in these variables, rather than the drug itself. This makes it difficult to determine whether the observed effect is due to the drug or the confounding variables. There are several statistical methods that can be used to mitigate the impact of confounding variables in scientific research. These methods include restriction, matching, statistical control, propensity scores, and randomization [49]. In the field of causal inference, specialized methods can be employed when the SCM is known, such as the backdoor adjustment method. This approach involves identifying variables that "block" the path between the treatment and the outcome, enabling the conditioning on these confounding variables to produce an accurate estimation of the causal effect between two variables [4]. The presence of confounding variables can pose a significant challenge in causal discovery, especially when the data is only observational and there are unobserved or unknown variables. Additionally, when analyzing time series data, it is important to consider the potential lagged impacts of confounding variables. These issues could lead to a predictive model learning spurious correlations between variables, potentially resulting in poor generalization and with that, violating the principle of invariance.

An RCT is considered the gold standard for assessing treatment effects because they aim to balance the impact of confounding variables between the experimental and control groups by randomizing participants [50]. However, even in RCTs, confounding variables may still exist if randomization is not performed correctly, resulting in biased estimates of treatment effects [51]. Moreover, RCTs cannot be applied to observational data, since participants are not randomly assigned to treatment groups. Because of this, there may be differences in characteristics between the groups that could affect the results.

Even though observational data can be used to identify potential causal relationships, it is impossible to judge whether a correlation is spurious purely on the analysis of observational data [37]. However, ranking these relationships can help domain experts with directing future experiments to test new hypotheses.

Interpretability. Interpretability is crucial in constructing causal matrices, with various methods adopting different strategies. These methods either depend on the data they produce or analyze the internal model parameters, interpreting them as evidence of causal connections. For example, the additive nature of NAVAR allows for prediction of contributions. By applying the standard deviation to these contributions, the connections can be ranked. TCDF focuses on interpreting attention scalars, applied before convolutional layers, and further investigates convolutional weights to identify influential variables. Rhino directly learns a distribution over the adjacency matrix. There is potential in combining methods to improve the correctness of causal matrix construction. Additionally, advancements in the field of explainable AI could provide further insights to enhance interpretability, leading robust causal discovery.

History-dependent noise. The concept of "noise" refers to any factor that introduces variability into a system. Typically, it is associated with interference or random fluctuations that are not relevant to the system under study. Essentially, noise represents all incoming effects on a variable that are not accounted for by the model. Random fluctuations are often the result of measurement errors and generate noise that is independent and identically distributed (i.i.d.). This noise can be simulated using various methods, such as sampling from a Gaussian distribution (for example, $N(0,1)$). On the other hand, noise that originates from other sources is referred to as history-dependent noise. This type of variability is not necessarily random or unrelated to the system, but instead, it is dependent on past events or conditions within or outside the system. The presence of history-dependent noise poses a challenge when developing causal discovery methods for real-life time series data, particularly in domains where there are relationships between numerous unobserved variables, such as finance or climate data. Throughout this work, we will refer to random fluctuations as η (i.i.d. noise) and history-dependent noise as ϵ (not i.i.d. noise).

Noise, especially of the i.i.d. type, could play a crucial role in uncovering causal relationships between variables. Contrary to common perception, noise is not just a challenge to overcome; it can also prove beneficial in distinguishing between correlation and true causal relationships. For example, consider the following SCM:

$$\begin{aligned} X_t &= f_1(Y_{t-1}) + \eta_1 \\ Y_t &= f_2(X_{t-1}, Y_{t-1}) + \eta_2 \end{aligned} \tag{2.22}$$

In the case of predicting Y_t, the model could find the true causal relationship or can learn the following transitive relationship:

$$Y = f_2(f_1(Y_{t-2}) + \eta_1, Y_{t-1}) + \eta_2 \tag{2.23}$$

The difference is that the first equation is dealing with a single source of i.i.d. noise, η_2. Whereas the second equation is dealing with both η_1 and η_2, which can either amplify or cancel each other out, resulting in less stable predictions. Therefore the model may lean towards the true equation, as it has an easier time predicting the first equation. In this case, the i.i.d. noise can also be interpreted as soft interventions at each timestep in the time-series data. The i.i.d. noise acts as a filter, helping to identify direct, consistent causal effects. However, the complexity of the functions f_1 and f_2 could also impact this. If the "true" relationship is complex, it might be easier for the model to learn the noisier, surrogate relationship.

Finally, in synthetic data, i.d.d. noise is often added to simulate external influences of history-dependent noise. However, in the case of real-world scenarios, you are never sure whether i.d.d. noise is present at all. Therefore, we should look beyond synthetic i.d.d. noise and focus on the history dependent noise.

Scalability and expressiveness. The methods discussed in Section 2.4 have different approaches in their implementation, efficiency, and complexity, which affect how well they can be scaled and be used in practice. TCDF learns a model for each pair of variables. While the depth-wise separable architecture allows for parallel learning for up to N models, it becomes computationally expensive when the number of variables increases. Rhino employs a single model for all variables with the use of weight-sharing and learning representations for each variable. This offers a reduced computational load and memory usage, and helps in finding patterns that are shared across the variables. However, there's a concern whether this weight-sharing hinders the learning of the true functional relationships. NAVAR falls in between these approaches. It learns a separate model for each variable, and tries to predict the outcomes of all other variables. As the hidden layer in the model limits the information flow through the network, this approach might offer a regularizing effect on spurious correlations, as it potentially encourages the model predict only a subset of variables.

All three methods employ an additive framework, which is known for its simplicity and interpretability, it has a limitation in capturing complex relationships between the variables. These models might struggle with non-linear relationships that go beyond simple summation. This trade-off in expressiveness is balanced against the benefits of these methods, such as the prevention of overfitting.

Unreliable predictions due to overfitting. Overfitting is a common issue in machine learning, where models become too complex and start to capture noise or random fluctuations in the training data rather than the underlying patterns or relationships. This problem is particularly important in the context of temporal causal discovery since overfitting can lead to the memorization of specific values or patterns in the time series instead of the causal relationships between variables.

TCNs can be prone to overfitting, as they require many neural layers to increase the number of lags and thus the number of parameters in the model, increasing the risk of memorization. Moreover, using small datasets of time series can contribute directly to overfitting. If we are to employ a TCN in our approach, it is important to ensure that the model is learning the functional relationships between variables rather than memorizing the training data.

3 Methods

3.1 Reducing Model Complexity while Preserving Long-Range Dependencies in TCNs

When dealing with high-frequency sampled data, capturing long-range dependencies can be challenging due to significant time lags between cause and effect. To address this, TCNs provide an efficient solution using parallelism in the convolutions. However, increasing the depth of a TCN to enhance the receptive field results in a more complex model and can lead to overfitting. Recently, NAVAR demonstrated exceptional results on various benchmarks using an additive approach, only utilizing a single-layer neural network. We hypothesize that by employing a simple model with low complexity, this method captures the most straightforward contributions in the data, and augmenting the model complexity with a TCN could potentially lead to poorer performance in learning causal relationships. As demonstrated in Section 2.6, a deep TCN has the capacity to entirely memorize temporal datasets. Since our objective function is regression, there is no assurance that the learned contributions are part of causal relationships rather than memorized noise.

We propose two approaches to modify the model's architecture, which aim to reduce model complexity while maintaining an increased receptive field. These modifications should enhance parameter and memory efficiency and potentially mitigate overfitting. First, we can employ weight-sharing, allowing multiple variables to be learned with the same parameters, potentially within a single model. Second, a recurrent component can be introduced to the TCN, enabling hierarchical pooling of temporal embeddings within the same embedding space. These approaches have the potential to strike a better balance between capturing long-range dependencies and preventing overfitting while learning causal relationships effectively.

3.1.1 Weight-sharing

Weight-sharing is a technique commonly used in deep learning models where certain weights are shared across multiple layers. This technique is particularly useful in convolution neural networks (CNNs) where the same kernel (weights) is used across the entire input image. Extending this approach to temporal convolutions in TCNs can reduce the number of parameters, which is especially beneficial for large models or limited computing resources.

NAVAR [15] learns a separate network for each variable, capturing functional relationships between variables through the model's parameters. On the other hand, Rhino [16] leverages weight-sharing for efficient computation. They introduce an embedding for each variable, which is combined with the time-series data as input to the model. Consequently, a single model is learned for all variables, serving as a general model capturing temporal features, while the functional relationship is encoded in the input embedding specific to each variable.

When relying on embeddings as the differentiating factor between variables, it is important that the time-series follow the same distribution. Otherwise, the model may have difficulty capturing all the relevant temporal characteristics in the various time-series. In other words, it might prevent the model from fully exploiting the unique characteristics or structures present in different inputs, potentially leading to suboptimal performance, particularly in cases of heterogeneous data compared to homogeneous data. Another consideration is whether the embedding can efficiently capture the functional relationship and guide the model in selecting the relevant information from the time series. This includes selecting the correct number of lags and excluding redundant information, which is often a significant portion of the data. Additionally, the model's capacity to learn complex non-linear relationships is also a critical aspect to address. Furthermore, it is essential to consider whether the embeddings, often high-dimensional, store relevant information, especially given benchmark datasets typically involving only a few variables.

In scenarios where the challenge of data distribution does not pose a significant obstacle, using embeddings can be a highly parameter-efficient approach. Moreover, as new variables may be introduced later on, the embeddings can be learned based on a previous model as a starting point. By sharing parameters, the model is encouraged to acquire more robust and general features that are relevant across different inputs, serving as a form of regularization. Furthermore, the acquired embeddings can offer valuable insights into the functional relationships between the variables.

To address the issue of data distribution mismatch among variables, we propose a solution where the first convolutional layer of the model is learned separately for each variable. This approach not only aligns the features within the same distribution but also enables the model to better determine the appropriate number of lags for each variable. For instance, it can identify and eliminate redundant information by factoring it to zero in the convolutional layer. The convolutional transformations can potentially be analyzed to determine the correct number of lags, similar to the approach in TCDF [37]. With this method, we eliminate the need for an embedding altogether, as it does not yield improved results. In the case of the linear layer, where the embedding e is concatenated to the input x, the output y is given by:

$$y = W \begin{bmatrix} x \\ e \end{bmatrix} + b \tag{3.1}$$

By splitting the weight matrix W for x and e individually, we have:

$$\begin{aligned} y &= \begin{bmatrix} W_x & W_e \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + b \qquad (3.2) \\ &= W_e e + W_x x + b \qquad (3.3) \end{aligned}$$

Since $W_e e$ does not interfere with $W_x x$ and is completely parameterized, we can incorporate it into the bias term as b':

$$y = W_x x + b' \tag{3.4}$$

In this scenario, learning an embedding e becomes redundant. However, this approach has a minor drawback in that the biases cannot be directly compared between variables, since the biases are not represented within the same embedding space. Additionally, since the example presented is a linear transformation, incorporating an activation function along with a second linear transformation can effectively ensure that the data lies within the correct distribution. In our experiments, we will investigate the impact of weight-sharing for the first two convolutional layers across different variables in our TCN model. We aim to assess whether this approach effectively reduces model complexity while preserving baseline performance. For the purposes of these experiments, we will refer to this weight-sharing variant as "WS".

3.1.2 Recurrent Temporal Convolutions

As the depth of the TCN increases, the model's receptive field also expands. However, with each additional layer, the model's complexity increases due to a higher number of learnable parameters.

To address this complexity issue without compromising performance, we propose a simplification by repeating the final temporal convolutional block in our TCN model. This recurrent layer aims to efficiently represent temporal data in an embedding, enabling repeated pooling within the same embedding space in the final layer. As a result, the model's receptive field can be increased, while the total number of parameters remains unchanged. This approach draws inspiration from Graph Neural Networks (GNN), where time-series data is represented as a hierarchical graph using dilated convolutions.

However, this simplification might compromise the model's ability to capture distinct levels of abstraction in each layer, potentially hindering the hierarchical learning process. Consequently, the model's expressiveness and flexibility in learning complex relationships could be limited. Additionally, despite the benefits, the increased depth can still lead to vanishing or exploding gradients. To mitigate this issue, we incorporate residual connections to enhance gradient flow within the model. In our experiments, we will evaluate the impact of recurrent layers in our TCN model to determine whether this approach effectively reduces model complexity while maintaining baseline performance. Throughout the experiments, we will refer to this recurrent variant as "Rec".

3.2 Temporal Attention Mechanism

To address the problem of additive models not being able to fully approximate non-additive relationships, we propose a method that uses the attention mechanism from the transformer architecture. Specifically, we adopt the scaled dot-product attention from Equation 2.3, which allows for mixing features between variables, while maintaining interpretability through the produced attention matrix. Originally, this attention mechanism is applied to a sequence of embeddings, e.g. a sentence. However, within the context of causal discovery, we do not apply it to the temporal component of our data, but to the embeddings representing various time series of variables. This produces an attention matrix of size $N \times N$ for each time step t, from which causal relationships can be interpreted.

The attention mechanism that incorporates the aspect of time is given as follows:

$$\text{Attention}(Q, K, V)_t = \text{softmax}\left(\frac{Q_t K_t^\top}{\sqrt{d_k}}\right) V_t \tag{3.5}$$

First, a TCN processes the time series and generates embeddings. These embeddings serve as input for the attention mechanism, which transforms them through the scaled dot product to incorporate information from other variables. Finally, regressive predictions can be made using these embeddings that doe not rely on addition of individual contributions. We name the proposed method Temporal Attention Mechanism for Causal Discovery (TAMCaD). A high-level overview of the proposed architecture is depicted in Figure 3.1.

The transformer Attention mechanism does not inherently account for sequence order. To address this issue in language models, positional encodings are incorporated. However, since we apply the attention over the variables where the order is not of importance, we can leave this out of the implementation.

3.2.1 Learning contemporaneous relationships

Methods like NAVAR aggregate the predictions for each variable to construct the final causal matrix. The attention mechanism offers an advantage of generating an individual causal matrix at each time step, allowing for capturing contemporaneous relationships. In systems consisting only of static relationships, this will likely result in the same matrix at each time step. In systems with contemporaneous relationships,

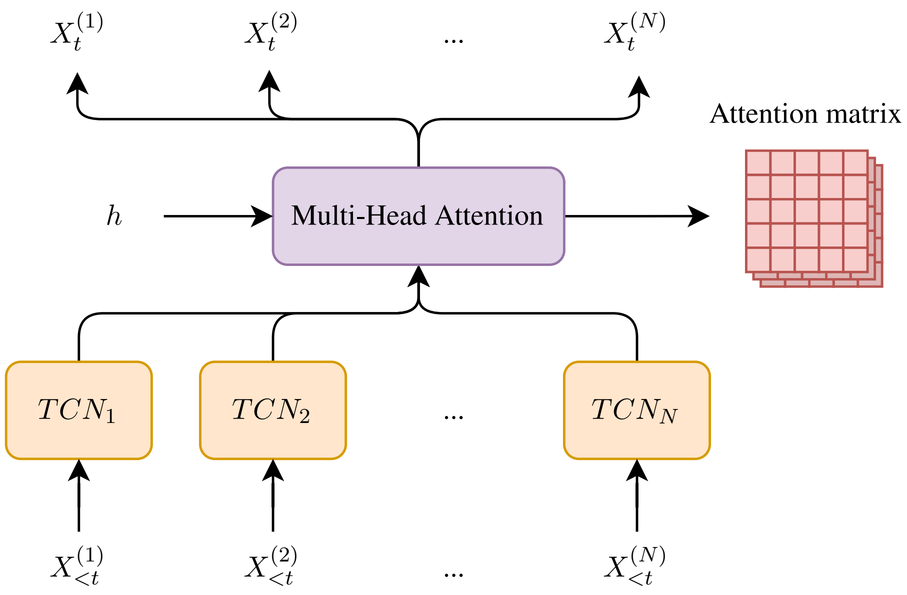

Figure 3.1: Attention-based causal discovery.

we will be able to observe the progression causal matrices across time. This approach allows us even to observe fading or sudden disappearances of causal connections.

Contemporaneous relationships in Additive models. When using NAVAR, a standard deviation (std) is applied over the time series of the contributions to construct a (single) causal matrix. However, this makes it impossible to discover contemporaneous relationships. To address this, we propose applying the std solely to a time window of the contributions, by implementing a sliding window along the time axis. This captures local variability and fluctuations in variability over time, capturing contemporaneous relationships. The larger the window, the larger the sample size, more confident we are of the output. However, this also leads to the changes appearing smooth, whereas some contemporaneous relationships are abrupt. There can be made a trade-off between confidence interval of the std and the accuracy of the contemporaneous relationships.

3.2.2 Multi-head attention

Besides the hypothetical advantages, there are also some issues when using this approach. For example, when dealing with only a few variables in a system, there are only a few embeddings available for which to calculate the dot product. Specifically for high-dimensional embeddings, this will likely result in scores that do not capture the interaction well and learning these embeddings becomes harder. Multi-head attention is a feature of the transformer based attention mechanism that uses multiple “heads” to split up the embeddings to form multiple attention matrices. This would allow one variable to attend to different variables subsets for each head. This helps the embedding attending to other embeddings over various contexts. However, these heads may suffer from attention collapse, where different heads pick up similar features, reducing the model’s effectiveness [52]. In our approach, we employ the number of heads as a hyper-parameter. The final matrix can be obtained by averaging over the various attention matrices from the heads. Given the low dimensional space (simplicity) of our causal relationships, this method will probably not suffer from the attention collapse, but only provide more insight into the causal structure of the data.

3.2.3 Instantaneous Scaled Dot-Product Attention

To deal with instantaneous relationships, data from $t+1$ can be included in the input for our model as well [37]. This can efficiently be incorporated into our Temporal Attention Mechanism by allowing the query to interact with the key values at $t+1$.

This adaptation is defined as follows:

$$\text{InstantAttention}(Q, K, V)_t = \text{softmax}\left(\frac{Q_t \left[K_t^\top \quad K_{t+1}^\top\right]}{\sqrt{d_k}}\right)\begin{bmatrix} V_t \\ V_{t+1} \end{bmatrix} \tag{3.6}$$

The concatenation of the matrices is denoted with $\left[\cdot\right]$. The intermediate attention matrix in this equation is of size $N \times 2N$ and will be referred to as A. Here, it is important that the attentions at time step $t+1$ are masked for the corresponding variable that is being predicted, otherwise variables will attend to their future selves and this would be considered cheating within the auto-regressive context. After masking, the individual attention matrices A_t and A_{t+1} can be averaged to end up with a single matrix of size $N \times N$.

3.2.4 Learning external influences.

Instead of completely masking future self-attentions, we introduce a regularization term to ensure minimal attention to future self-values. This is controlled using the hyper-parameter λ, similar to the one in NAVAR, which also regulates attentions.

The loss is described by:

$$\mathcal{L}_{\text{ext}} = \lambda \sum_{i=1}^{N} A_{t+1}^{(\,i \rightarrow\, i)} \tag{3.7}$$

This formulation implies that if the model cannot learn causal information from other variables, it will refer to its future self, hinting at external influences not present within the current system. As described earlier, self attentions at $t+1$ are masked. When interpreting these values as external influences, the masked values can be repositioned to a separate column, producing a matrix output of size $N \times (N+1)$ (with N variables attending to N + 1 virtual external variable).

A significant benefit of this method is that it overcomes the constraint of a variable being forced to attend to other variables within the system due to the softmax. Here, a variable can attend to its future self and potentially assign zero attention to other system variables. However, the model's sensitivity to the hyper-parameter λ might still result in attending to non-causally related variables, especially when the penalty for future self-attention is high.

3.2.5 Beyond Softmax: Alternatives for scoring Attentions

The scaled dot-product attention typically employs the softmax function to normalize the attention scores. Given an input vector z of logits for each class, the softmax function transforms this input vector to a range that can lead to a probabilistic interpretation of an input belonging to each class. This function is a critical component of the attention mechanism, which ensures that the model's attention weights sum to one.

The default softmax function is given by:

$$\text{Softmax}(\mathbf{z})_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{3.8}$$

While it is true that the output of the softmax function can be treated as a probability vector, it is easy to fall into the trap of that this information represents confidence (in the statistical sense) and might give a wrong interpretation of the retrieved causal matrix. In the context of predicting causal connections, this approach leads to issues when dealing with variables that lack incoming connections within the system under study, as each variable is forced to attend to other variables. Moreover, the softmax function outputs a a prediction for a class relative to the other predictions. Here arises the question whether softmax is a good choice for discovering the causal connections, since a predicted causal connection is present regardless of the other predictions. This may even hinder the learning of new connections, when the model converges to a subset of connections. Several modifications and alternatives to softmax have been proposed to address its limitations or to achieve specific desired properties.

SparseMax. SparseMax is an adaptaion of the softmax that can produce sparse attention weights [53]. The main advantage of SparseMax over softmax is the ability to assign exactly zero weights to certain inputs, which can be beneficial for interpretable models and for situations where only a subset of the inputs should be attended to. It involves projecting the input logits onto a simplex, by subtracting a single (non-vector) value $\tau(\mathbf{z})$. It performs similarly to the traditional softmax in language modeling tasks, but with a selective, more compact, attention focus. One drawback of this approach is that the logits must be sorted to compute $\tau(\mathbf{z})$, thus making it slow to compute if the dimension of $\mathbf{z}$ is large.

Softmax-1. In certain language models, it has been observed that certain word embeddings assign extremely high weights to punctuation characters like spaces and commas. This behavior might indicate that these embeddings are avoiding attending to other words, as the output of the scaled dot product is added to the current embeddings using a residual connection. To address this concern, a modified version can be used, which ensures the sum of attention weights is less than one and allows the model to output zero attention. This modified softmax function is defined by including a 1 in the denominator:

$$\text{Softmax-1}(\mathbf{z})_i = \frac{\exp(z_i)}{1 + \sum_j \exp(z_j)} \tag{3.9}$$

It is important to note that there is no empirical evidence suggesting that this approach performs better than the original softmax function. However, this modified approach is already incorporated into the predefined scaled dot product functionality in the PyTorch library. Leveraging this approach could potentially allow us to learn zero-attentions for variables. Further research and experimentation are required to determine its effectiveness.

Gumbel-Softmax. The Gumbel-Softmax is a distribution that can be smoothly annealed into a categorical distribution [54]. It introduces a Gumbel noise to the input logits before applying the softmax. While a dropout mask might obstruct both values and their gradients, the Gumbel-Softmax approach enables the flow of gradients through these values. This approach might prove useful in preventing the model from becoming trapped in local minima during training. The Gumbel-Softmax function is defined in Equation 3.10. Here, g is sampled from the Gumbel distribution and τ is the temperature parameter.

This temperature parameter can be adjusted to obtain more sharp distributions over the attentions.

$$\text{GumbelSoftmax}(\mathbf{z})_i = \frac{\exp((z_i + g_i)/\tau)}{\sum_j \exp((z_j + g_j)/\tau)} \quad \text{where} \;\; g_0 \ldots g_k \sim \text{Gumbel}(0,1) \tag{3.10}$$

Normalized Sigmoid. The sigmoid function is a frequently used activation function in neural networks and introduces non-linearity in the model. Contrary to the softmax function, which considers all inputs when assigning importance to a value, the normalized sigmoid evaluates each input in isolation. This can be beneficial in the case where the independent treatment of inputs is desired, as with the predictions of causal connections. The function is presented in the following equation, where $\sigma(x)$ is the default sigmoid function:

$$\sigma_{\text{Norm}}(\mathbf{z})_i = \frac{\sigma(x_i)}{\sum_j \sigma(x_j)} \tag{3.11}$$

3.3 Quantifying Aleatoric and Epistemic Uncertainty

3.3.1 Aleatoric Uncertainty in TAMCaD

Integrating aleatory and epistemic uncertainties into TAMCaD poses challenges. For regression problems, the Negative Log-Likelihood (NLL) loss is commonly employed to capture the noise in the data. However, for classification tasks where labels are binary, this approach is infeasible. The cross-entropy loss combined with a softmax can be used instead, as this approach outputs a certainty estimate in the form of a probability distribution. For instance, a prediction might assign 20% probability for a dog and 80% for a cat. However, the absence of the true causal matrix means that cross-entropy is inapplicable in this scenario. The attention logits in our attention mechanism are computed with a dot-product between variables rather than a simple linear layer an might not work well for uncertainty prediction. However, given that softmax already reflects a degree of uncertainty, further adaptations to this layer within our attention mechanism to account for uncertainty might be redundant. As shown in Section 3, including aleatoric estimates into the final regression prediction can help in mitigating overfitting and learning the true contributions. Therefore, we train a simple TCN that runs parallel to the attention mechanism. This TCN is exclusively responsible for predicting aleatoric noise. The aim for this auxiliary network is to help our attention mechanism in identifying the true causal relationships.

3.3.2 Aleatoric Uncertainty in Additive Models.

While incorporating aleatoric uncertainty might help in mitigating overfitting and enhances robust learning, it does not provide insights into the uncertainty of causal contributions. One challenge arises because the NLL loss cannot be directly applied to true causal contributions. We provide a potential workaround, where the final prediction is treated as the true causal contribution. This implies viewing the individual models of the variables as the main predictive model. To support this idea, consider the relationships depicted in Figure 5.2. Here, the values of a variable ($X_t^{(i)}$) in a time series are sorted and plotted against the true values of our target variable ($X_{t+n}^{(j)}$). With the relationships shown visually, we can clearly see their functional forms. Furthermore, it reveals that the spurious relationship from $X^{(0)}$ to $X^{(2)}$ is also being learned. Due to the transitive nature of this relationship, it exhibits a higher aleatoric uncertainty compared to the other relationships. This degree of aleatoric uncertainty could potentially be used as a metric for estimating whether the identified contributions represents a true causal relationships or not.

The NNL loss from equation 2.14 can be adapted as follows:

$$\mathcal{L}_{\text{NLL}}(\theta)_t = \frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{N}\log\sigma^2_{t,i\to j} + \frac{(y_{t,j} - c_{t,i\to j})^2}{\sigma^2_{t,i\to j}} \tag{3.12}$$

In this equation, $c_{t,i\to j}$ and $\sigma^2_{t,i\to j}$ represent the mean contribution and variance of causal link from i to j at time step t, with $y_{t,j}$ being the true value to be predicted. To preserve the additive context of the contributions, the original loss function is also used, including the regularization on $c_{t,i\to j}$.

3.3.3 Epistemic Uncertainty in Additive Models with Deep Evidential Regression

Deep Evidential Regression (DER) can be used for capturing and quantifying uncertainties in regression settings. The same idea for aleatoric uncertainty over the contributions can potentially be applied to DER to obtain estimates for epistemic uncertainty. We can train our model to not only predict the mean contributions, but also α, β and v, that capture the epistemic uncertainty distribution over these predictions. If a causal contribution consistently exhibits high epistemic uncertainty, this may signal the need for further investigation or additional data. Furthermore, the uncertainties may also be used to help construct the causal matrix.

3.4 Synthetic Data Generation Process

Our method aims to address challenges of capturing non-linear, non-additive and contemporaneous relationships with long-range dependencies. To assess the effectiveness of our approach in handling these challenges, we designed a data generation process by constructing temporal causal graphs that encapsulate such relationships. The idea is to represent this causal graph using a three-dimensional adjacency matrix, denoted with $G \in \mathbb{R}^{N\times N\times K}$, where the dimensions correspond to the source node, target node, and time lag.

We can implement the non-linear non-additive relationships between nodes through N simplified two-layer neural networks denoted by f. These networks take as input the past values of all the other variables, which are masked by adjacency matrix G:

$$X^{(i)}_{t+1} = f_i(G \odot X^{(0:N)}_{t-K:t-1}) + \eta^{(i)}_t \tag{3.13}$$

By applying G as filter, we block data that should not contribute to a specific variable. This is an efficient approach, since f can be implemented as a single convolutional neural network. Following this, the data is generated sequentially to obtain the temporal dataset. Furthermore, we mimic contemporaneous relationships by altering certain causal links in G during the generation process. For example, in a generated dataset with $T = 500$, a causal link is severed at $t = 200$ and restored at $t = 300$.

The parameters of f are derived by training on a randomly generated dataset consisting of a few data points. A non-additive model is trained to fit these data points. Concurrently, the additive model showcased in the figure is trained on data generated by this non-additive model. In other words, this additive model is tasked with learning two distinct non-linear functions for variables x_1 and x_2. While the figure indicates that the additive model is able to approximate the data, it does so sub-optimally.

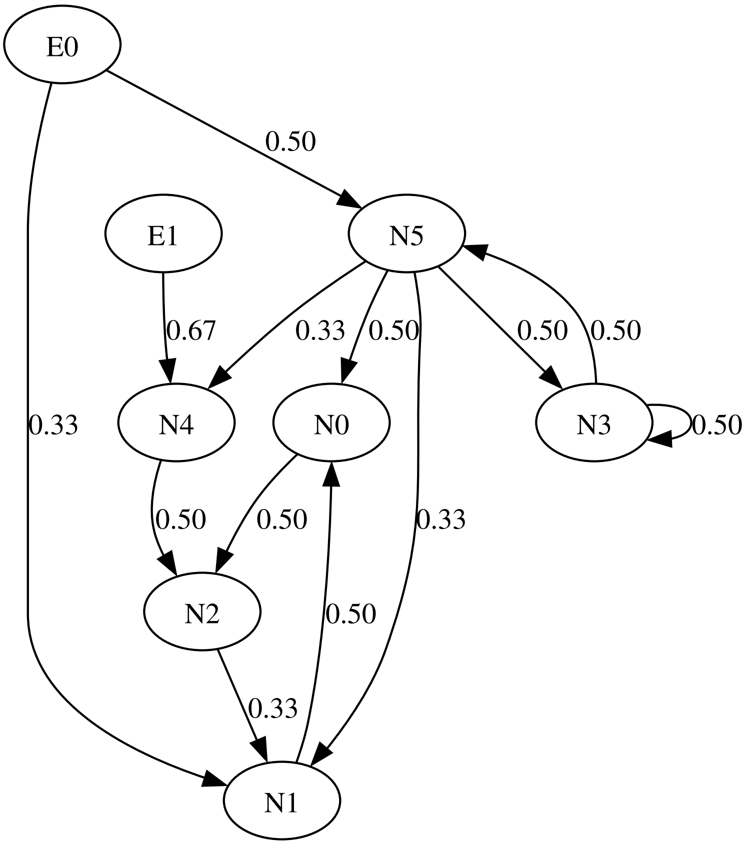

Figure 3.2: A synthetically generated causal graph consisting of 6 variables (N) and 2 external variables (E), which are excluded as input to the models.

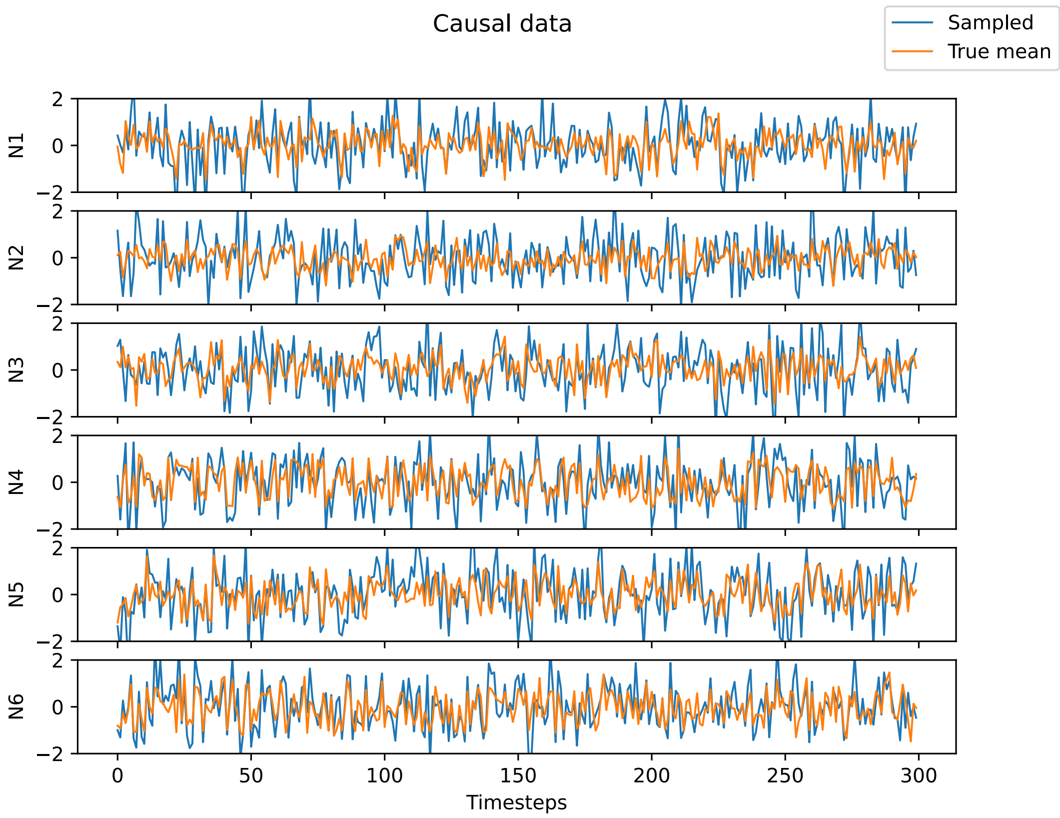

Figure 3.3: A synthetically generated time series data consisting of 6 variables (N) defined by the causal graph illustrated in Figure 3.2. The sampled data (blue) includes static noise and used as training data. The true mean (orange) is what the model should be predicting if accounted for this noise in the data.

4 Experiments

4.1 Quantifying Uncertainty in Causal Predictions

To measure aleatoric or epistemic uncertainty in causal predictions, we use established methods and aim to integrate these approaches into either an additive model or our proposed attention method. First, to illustrate the problem of overfitting in TCNs and its potential impact on causal discovery, we conduct an experiment involving a single instance of a TCN. The task for this TCN is to learn from a randomly generated time series that consists of three variables ($T = 800$). The TCN uses a kernel size of 2 and incorporates three dilated temporal layers, effectively leading to a maximum lag (K) of 15 time steps. The data for these three variables are generated from a normal distribution, specifically $\mathcal{N}(0, 1)$. Each individual TCN corresponding to a variable has a relatively small hidden dimension of 8, resulting in a model with a total of 852 parameters per variable. To provide a comparative perspective, we introduce a second TCN alongside the primary one. This auxiliary network is tasked with predicting the aleatoric uncertainty associated with the data, essentially guiding the primary model to learn the prediction means. The training of causal models is accomplished using the additive framework of NAVAR, yielding a causal matrix as the outcome. Furthermore, the model responsible for predicting aleatoric uncertainty does not operate independently across the variables, as it retains the capability to blend input information from the multiple variables when generating its predictions. Furthermore, we conduct another experiment to investigate whether the epistemic uncertainty can be quantified over the contributions. For this, we train an additive model along with an auxiliary model to predict the parameters following the DER method [42].

4.2 Datasets for Performance Evaluation

To evaluate the performance of our proposed methods, we will conduct a series of experiments on both real-world and synthetic datasets. The CauseMe platform offers benchmarks for evaluating and comparing the effectiveness of methods used to detect causal relationships in time series data. These datasets can be either synthetic, designed to replicate real-world challenges, or real-world datasets where the causal structure has been established with high confidence. This helps identify which methods are best suited for different challenges [46].

As our method aims to address various challenges for discovering causal relationships having certain characteristics, it is crucial to evaluate our method on datasets that contain these characteristics. Knowing that these characteristics might not always be present in existing (real-world) datasets, we will use our proposed synthetic dataset. We will create a number of synthetic datasets with controlled characteristics. For example, by varying the number of variables (N), the number of lags (K), and the number of timesteps (T), we can systematically analyze the strengths and limitations of our proposed methods.

We will compare our method with established baseline methods on the CauseMe benchmark, specifically SELVAR, SLARAC [55], and the original NAVAR. SLARAC employs a VAR model on bootstrap samples of the data. Each time, it selects a random number of lags to include. On the other hand, SELVAR uses a hill-climbing procedure based on the leave-one-out residual sum of squares of a VAR model to select edges.

4.3 Evaluation Metrics and Experimental Setup

In this section, we discuss the evaluation metrics used to assess the performance of our proposed approach and provide details about the experimental setup.

4.3.1 Evaluation Metrics

For evaluating the effectiveness of our approach, we employ several metrics to capture different aspects of model performance. One of the key metrics we utilize is the Area Under the Receiver Operating Characteristic Curve (AUROC). AUROC is a widely used metric in binary classification tasks, including causal inference. It measures the ability of a model to distinguish between positive and negative instances. In the context of our problem, AUROC quantifies how well our approach can rank the true causal relationships. However, there is a potential issue with using AUROC when dealing with causal inference in scenarios where the causal matrix for many variables is sparse. In such cases, there can be a high number of true negatives, which dominate the dataset. This dominance of true negatives can lead to a skewed perspective of model performance. Therefore, the authors behind the CauseMe benchmark underscore the significance of prioritizing a high True Positive Rate (TPR) for evaluating their datasets.

In the subsequent sections, we elaborate on the experimental setup we employed to evaluate our approach and provide insights into the datasets, model configurations, and other relevant details.

Adapting Receiver Operating Characteristic for predictions with uncertainty. Applying ROC analysis to predictions with uncertainty, represented as probability distributions rather than discrete values, requires some modifications. Here, we propose a method to extend ROC analysis to work with probability distributions. Rather than using the predicted values as thresholds, we use k number of thresholds, distributed uniformly from 0 to 1. At each threshold, we can determine the True Positive Rate (TPR) and False Positive Rate (FPR) by calculating the probability of the data points falling above or below the threshold using the Cumulative Distribution Function (CDF) for the predicted distribution. Instead of a prediction being merely true/false positive/negative, samples can fall into both categories at the same time. For example, the True Positive (TP) is usually the count of true positive samples; now it's the sum of all probabilities of the positive predictions for the actual positive samples. Because the probabilities are calculated for each threshold yielding different values, the TPR and FPR also differ at each threshold, resulting in a smoother curve. Furthermore, consider a sample that should be classified as positive having a mean prediction of 0.1 and a standard deviation of 0.01, this prediction would be penalized significantly more compared to having a standard deviation of, let's say, 0.9, as the model admits uncertainty regarding the sample's (wrong) negative classification. If all standard deviations tend towards infinity, the probabilities will approximate 0.5, thereby reflecting a random model (approximating the linear random guess line in the ROC plot). Conversely, if all standard deviations approach 0, the method will be the same as without uncertainty, showing a staircase-like curve.

4.3.2 Training Setup

The models themselves are implemented in PyTorch, as it provides us with the flexibility to customize models according to our preferences and requirements. To optimize the training process, we employ the AdamW optimizer, which is less prone to getting trapped in local optima.

The hyper-parameter optimization involves tuning various parameters that affect the training and performance of the models. The key hyper-parameters we consider include:

- **Hidden Dimension**: This parameter determines the size of the hidden layers within the model, influencing the information flow and its capacity to capture complex relationships.
- **Lambda (regularization)**: Lambda (λ) controls the strength of regularization applied to the model during training, penalizing high contributions in NAVAR and future self-attentions in TAMCaD.
- **Learning Rate**: The learning rate defines the step size at which the model's parameters are updated during gradient descent. It significantly impacts the convergence speed of the training process.
- **Weight Decay**: Weight decay is a regularization technique that discourages overly large weights in the model, contributing to better generalization.
- **Dropout**: Dropout is a form of regularization that randomly sets a fraction of the model's input units to zero during each training iteration, reducing reliability on specific features and enhancing robustness.
- **Architecture of the Model**: The architecture parameters include the kernel size, the number of blocks, and the number of layers in the model. These architectural choices influence the model's receptive field. These parameters can be inferred from the provided maximum lags in the data.
- **Number of Heads (TAMCaD Model)**: For the TAMCaD model, we also consider the number of attention heads. This parameter influences the model's capacity to attend to different parts of the input sequence simultaneously.

To determine optimal hyper-parameter values, we perform a hyper-parameter search using the `hyperopt` library. Our optimization objective is to minimize the loss on the last 20% of the time series of the original dataset, which serves as an unseen data subset for evaluating the model's generalization performance. We set the `hyperopt` library to perform 100 evaluations of different hyper-parameter combinations to find the best set of hyper-parameters for our models. When the number of epochs is set at 3000, a model can be trained on a single subdataset from the CauseMe benchmark within around 30 seconds. Nevertheless, doing hyper-parameter optimization followed by the processing of all datasets could extend the duration to several hours.

Given that each dataset from the CauseMe benchmark consists of many (around 200) sub-datasets, we aggregate the losses from five sub-datasets to compute the final loss for hyper-parameter optimization. This aggregation helps provide a more robust assessment of the models' performance.

5 Results

5.1 Uncertainty Analysis Results

5.1.1 Aleatoric Uncertainty Findings

The predicted time series in Figure 5.1) show that the entire time series could be perfectly reconstructed by the model, with a loss approaching 0. These findings suggest that the noise added to a causal time series may be stored in the model itself. This could potentially explain why the observations of minor contributions from various variables in the causal predictions outputted by NAVAR. Furthermore, the size of hidden dimensions presents a tradeoff. A smaller hidden dimension reduces the number of parameters and mitigates value memorization, while a larger dimension may help with learning more complex relationships and reducing the number of spurious correlations learned. Additionally, regularization techniques such as dropout can prevent overfitting to the training data.

Furthermore, a NAVAR using an extra model to estimate eleatoric uncertainty as an auxiliary task (named NAVAR-A) is also able to correctly learn the aleatoric noise in the data as shown in Figure 5.2. For higher number of hidden dimensions and more layers (a more complex model), the aleatoric is less prone to learn spurious correlations.

5.1.2 Epistemic Uncertainty Findings

Furthermore, the results reveal that it is also possible to incorporate epistemic uncertainty into the contributions. These measures of uncertainty over the predictions can help identify outlier data points. For constructing the causal matrix, predictions related to these outliers might be excluded or their confidence used as a weight. Exploring this approach further will be reserved for future work.

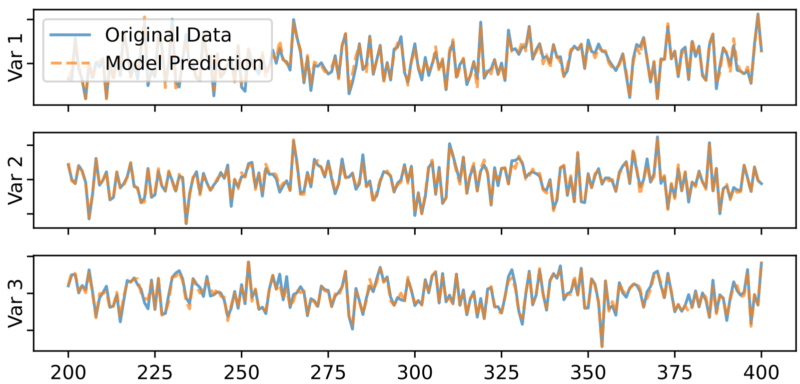

Figure 5.1: A subset of the of the random data ($N = 3,\ T = 1000$) memorized by a TCN model.

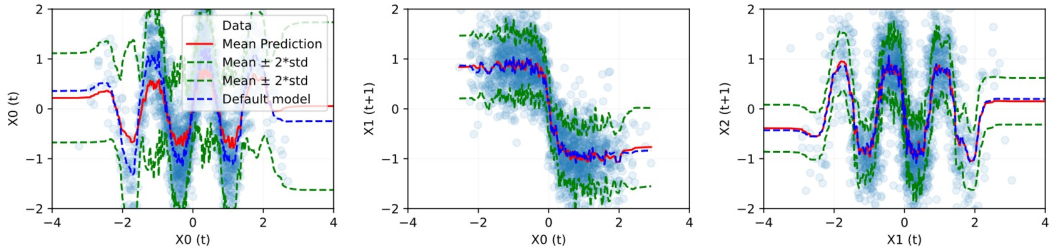

Figure 5.2: Learned causal relationships for causal structure $X_t^{(0)} \rightarrow X_{t+1}^{(1)} \rightarrow X_{t+2}^{(2)}$. The third image shows the learned relationship

	Synthetic data		
	$N = 5$	$N = 12$	$N = 8$
	$K = 6$	$K = 6$	$K = 30$
	$T = 500$	$T = 1000$	$T = 1000$
TAMCaD	0.93	0.99	0.79
TAMCaD-A	0.68	0.91	0.55
TCN-NAVAR	0.95	**1.00**	**0.9**
TCN-NAVAR-A	**0.98**	**1.00**	0.84

Table 5.1: Comparison of results between models with and without the aleatoric auxiliary tasks. The table presents the AUROC scores for different synthetic data configurations.

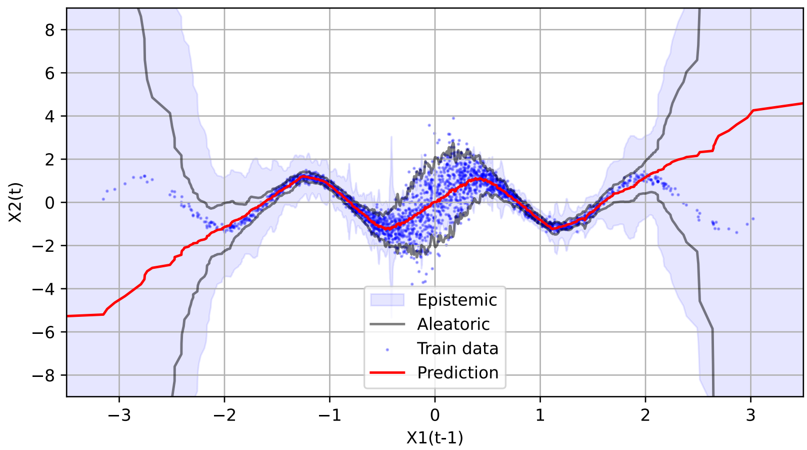

Figure 5.3: .

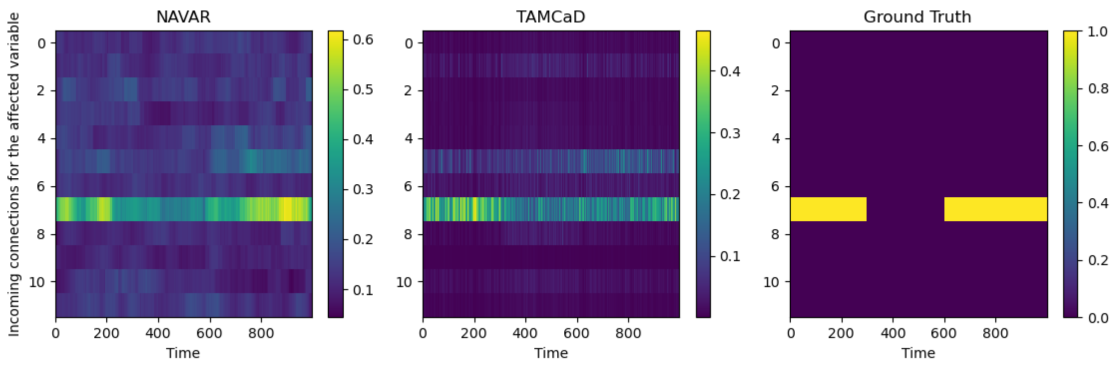

Figure 5.4: .

5.2 Relationship Learning Results

5.2.1 Contemporaneous Relationships Findings

To show how well the adapted NAVAR and TAMCaD can capture the contemporaneous relationships, we plot the attentions and contributions next to the true causal links.

Figure 5.4 shows that the contemporaneous relationship is being learned to some degree, but more than often the causal remains the same bor both NAVAR and TAMCaD.

When NAVAR is given a dataset with contemporaneous relationships, NAVAR might predict -1 from t=1 to t=100, then predict 1 for t=100 to t=200. Normally, a static value can be captured in the bias in the NAVAR framework, but this only works for a single value. When processing the timeseries with the std to construct the causal matrix, this results in an std of 1 (in the case of the example), which may or may not be considered correct.

To demonstrate NAVAR and TAMCaD capturing contemporaneous relationships, we present visualizations of attentions and contributions over time. Figure 5.4 illustrates that both the NAVAR and TAMCaD models are capable of learning contemporaneous relationships to a small extent. However, it is worth noting that often the predicted causal links remains the same for both models.

In scenarios where NAVAR is provided with a dataset containing contemporaneous relationships, it may exhibit a pattern of predictions such as -1 from time step t=1 to t=100, followed by a prediction of 1 from time step t=100 to t=200. Typically, a constant value can be effectively captured by the bias term within the NAVAR framework. However, this only works for singule values. When the causal matrix is constructed, a contribution can result in a standard deviation of 1. Whether this outcome is accurate or not depends on the specific context.

In Figure 5.5, the ability of an additive model to learn a non-additive relationship is depicted. The non-additive model successfully approximates the random data points. In contrast, the additive model identifies two separate one-dimensional functions, which provide a less optimal representation of the data. Even with these dynamics, when applied to our synthetic data, the additive model seems to be the best predictor of causal matrices. This might be attributed to the model's use of temporal history as an information source to augment its predictions. The use of spurious correlations is not a problem in this framework, since the magitude of the correct contribution outranks the spurious contributions. This highlights the efficacy of additive models. We theorize that the additive aspect of such models acts as a form of regularization.

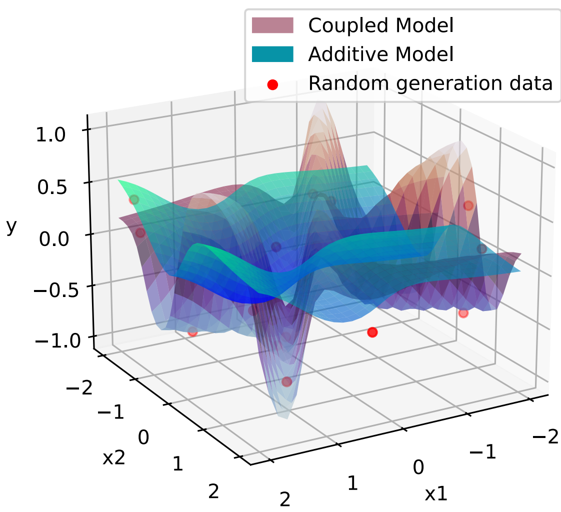

Figure 5.5: The non-additive model (Coupled) is able to approximate the random data points, whereas the additive model learns two separate one-dimensional functions that are approximating the data sub-optimally.

	Synthetic data	
Method	$N = 5$ $K = 6$ $T = 500$	$N = 8$ $K = 30$ $T = 1000$
TAMCaD	0.66	0.72
TAMCaD-WS	0.76	0.64
TAMCaD-Rec	**0.85**	0.74
TAMCaD-WS-Rec	0.63	**0.79**
TCN-NAVAR	**0.98**	0.84
TCN-NAVAR-WS	0.97	**0.90**
TCN-NAVAR-Rec	0.96	0.86
TCN-NAVAR-WS-Rec	0.95	0.88

Table 5.2: Comparison of results between models using weight-sharing (WS) and recurrent layers (Rec). The table presents the AUROC scores for different synthetic data configurations.

Scoring function	AUROC
Softmax	0.66
Normalized Sigmoid	0.80
SparseMax	0.83
Softmax-1	0.70
GumbelSoftmax	0.73
GumbelSoftmax-1	**0.89**

Table 5.3: Comparison of results between TAMCaD models using various attention-scoring functions. The table presents the AUROC scores for $N = 5$, $K = 6$, and $T = 500$.

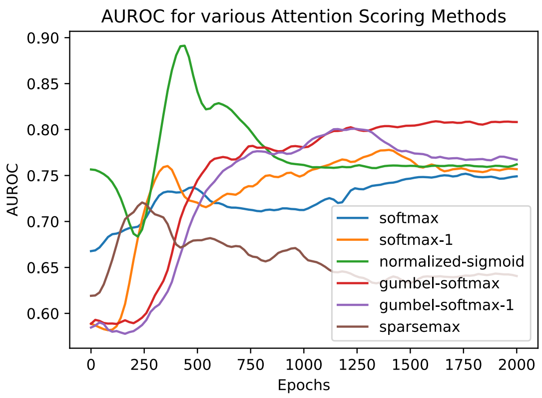

Figure 5.6: The AUROC scores during training for models employing different scoring functions over the attentions.

5.3 Performance Results

5.3.1 Attention Scoring Methods Results

The results in Table 5.3 offer insights into the performance of models employing different categorical distributions over attention scores in the context of a TAMCaD-A model. The models all use the same hyper-parameters to isolate the effect of the categorical distribution. It shows that the standard Softmax function exhibits the best performance when considering the highest achieved AUROC scores during training. This performance gain is mostly present within the initial 100 epochs of training, highlighting the model's capability to quickly learn the most evident causal relationships. However, as training progresses, the model begins to incorporate more distant and transitive relationships, potentially for compensating for noise in the data and enhancing predictive accuracy. The normalized sigmoid function emerges with the most favorable loss. Each prediction is independent of the others, which allows for less constrained learning. Nonetheless, the efficacy of this approach goes with overfitting, as it achieves poor performance in terms of AUROC scores. Some of the well-predicted causal matrices are accompanied by a test loss that only goes up. This underlines the need for finding a balance between achieving low loss in the objective function and maintaining performance. Lastly, the Gumbel-Softmax distribution achieves the best overall scores for constructing a causal matrix due to the regularization effect from the sampling mechanism during training. The use of Gumbel-Softmax helps in preventing overfitting and moreover, the temperature parameter τ of the Gumbel-Softmax can be adjusted to achieve a causal matrix that is less uniform.

	CauseMe						
	Nonlinear VAR				Climate	Weather	River
	$N = 3$	$N = 5$	$N = 10$	$N = 20$	$N = 40$	$N = 10$	$N = 12$
	$T = 300$	$T = 300$	$T = 300$	$T = 300$	$T = 250$	$T = 2000$	$T = 4600$
TAMCaD	0.54	0.58	0.57	0.59	0.68	0.61	0.86
TAMCaD-A	0.59	0.58	0.61	0.62	0.64	0.59	0.74
TCN-NAVAR	0.86	0.82	0.79	0.80	0.68	0.76	0.85
TCN-NAVAR-A	0.65	0.65	0.53	0.53	0.56	0.86	0.90
NAVAR (original)	0.86	**0.86**	**0.89**	**0.89**	0.80	0.89	**0.94**
SELVAR	**0.88**	**0.86**	0.86	0.85	0.81	0.90	0.87
SLARAC	0.74	0.76	0.78	0.78	**0.95**	**0.95**	0.93

Table 5.4: Comparison of results between models using different configurations. The table presents the AUROC scores for different datasets from the CauseMe benchmark.

5.3.2 CauseMe Results

Our method and its adaptations to NAVAR have significantly underperformed in comparison to the baselines. One possible reason for this might be the duration of the training and the extent of hyper-parameter evaluations. However, a primary concern is that we optimize for regression test loss on the test set. This might not necessarily reflect the a good causal model. Additionally, the regularization parameter was not sufficiently constrained. The best model leans towards a λ as low as 1e-3, whereas it should ideally be higher. Although the experiments should be revisited to yield improved outcomes, this is not feasible within the current research timeline.

Bibliography

[1] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018. iii, 17

[2] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937, 2020. iii, 17

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. iii, 9

[4] Judea Pearl. *Causality*. Cambridge university press, 2009. 3, 6, 7, 21

[5] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000. 3, 5

[6] Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005. 3

[7] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. 3

[8] Judea Pearl. The seven tools of causal inference, with reflections on machine learning. *Communications of the ACM*, 62(3):54–60, 2019. 5

[9] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017. 5, 6, 18, 19

[10] Shohei Shimizu, Patrik O Hoyer, Aapo Hyvärinen, Antti Kerminen, and Michael Jordan. A linear non-gaussian acyclic model for causal discovery. *Journal of Machine Learning Research*, 7(10), 2006. 5

[11] Xun Zheng, Bryon Aragam, Pradeep K Ravikumar, and Eric P Xing. Dags with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems*, 31, 2018. 5, 14

[12] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. Dag-gnn: Dag structure learning with graph neural networks. In *International Conference on Machine Learning*, pages 7154–7163. PMLR, 2019. 5

[13] Zhichao Chen and Zhiqiang Ge. Directed acyclic graphs with tears. *arXiv preprint arXiv:2302.02160*, 2023. 5

[14] Christopher A Sims. Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, pages 1–48, 1980. 5, 13

[15] Bart Bussmann, Jannes Nys, and Steven Latré. Neural additive vector autoregression models for

causal discovery in time series. In *International Conference on Discovery Science*, pages 446–460. Springer, 2021. 5, 8, 14, 25

[16] Wenbo Gong, Joel Jennings, Cheng Zhang, and Nick Pawlowski. Rhino: Deep causal temporal relationship learning with history-dependent noise. *arXiv preprint arXiv:2210.14706*, 2022. 5, 14, 16, 25

[17] Bernhard Schölkopf, 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. 5

[18] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020. 5

[19] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016. 6

[20] Christina Heinze-Deml, Jonas Peters, and Nicolai Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 6(2), 2018. 6

[21] Léon Bottou, Jonas Peters, Joaquin Quiñonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14(11), 2013. 6

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

[23] Tianxin Wei, Fuli Feng, Jiawei Chen, Ziwei Wu, Jinfeng Yi, and Xiangnan He. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1791–1800, 2021. 6

[24] Judea Pearl. The causal foundations of structural equation modeling. Technical report, California Univ Los Angeles Dept of Computer Science, 2012. 7

[25] Yaowei Hu, Yongkai Wu, Lu Zhang, and Xintao Wu. Fair multiple decision making through soft interventions. *Advances in Neural Information Processing Systems*, 33:17965–17975, 2020. 7

[26] Murat Kocaoglu, Amin Jaber, Karthikeyan Shanmugam, and Elias Bareinboim. Characterization and learning of causal graphs with latent variables from soft interventions. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 14346–14356. Curran Associates, Inc., 2019. 7

[27] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. 8, 9

[28] Christian Lang, Florian Steinborn, Oliver Steffens, and Elmar W Lang. Electricity load forecasting–an evaluation of simple 1d-cnn network structures. *arXiv preprint arXiv:1911.11536*, 2019. 8

[29] Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an ef-

fective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24:16453–16482, 2020. 8

[30] Sidra Mehtab and Jaydip Sen. Stock price prediction using convolutional neural networks on a multivariate timeseries. *arXiv preprint arXiv:2001.09769*, 2020. 8

[31] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D Hager. Temporal convolutional networks: A unified approach to action segmentation. In *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III 14*, pages 47–54. Springer, 2016. 8

[32] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, Mohammed Bennamoun, Gerard Medioni, and Sven Dickinson. *A guide to convolutional neural networks for computer vision*, volume 8. Springer, 2018. 8

[33] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969. 11

[34] Trevor J Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017. 13

[35] Michele Chambers and Thomas W Dinsmore. *Advanced analytics methodologies: Driving business value with analytics*. Pearson Education, 2014. 13

[36] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. Deep state space models for time series forecasting. *Advances in neural information processing systems*, 31, 2018. 14

[37] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019. 14, 20, 21, 26, 29

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 15

[39] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017. 15

[40] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019. 16

[41] Xue Li, Wei Shen, and Denis Charles. Tedl: A two-stage evidential deep learning method for classification uncertainty quantification. *arXiv preprint arXiv:2209.05522*, 2022. 17

[42] Nis Meinert, Jakob Gawlikowski, and Alexander Lavin. The unreasonable effectiveness of deep evidential regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9134–9142, 2023. 18, 35

[43] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003. 18

[44] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004. 18

[45] John Hicks et al. *Causality in economics*. Australian National University Press, 1980. 18

[46] Jakob Runge, Sebastian Bathiany, Erik Bollt, Gustau Camps-Valls, Dim Coumou, Ethan Deyle, Clark Glymour, Marlene Kretschmer, Miguel D Mahecha, Jordi Muñoz-Marí, et al. Inferring causation from time series in earth system sciences. *Nature communications*, 10(1):1–13, 2019. 19, 35

[47] Christof Wolf and Henning Best. The sage handbook of regression analysis and causal inference. *The SAGE Handbook of Regression Analysis and Causal Inference*, pages 1–424, 2013. 19

[48] Roseanne McNamee. Confounding and confounders. *Occupational and environmental medicine*, 60(3):227–234, 2003. 21

[49] Miguel A Hernán and James M Robins. *Causal Inference: What If*. Boca Raton, FL: Chapman and Hall/CRC, 2020. 21

[50] Peter Markus Spieth, Anne Sophie Kubasch, Ana Isabel Penzlin, Ben Min-Woo Illigens, Kristian Barlinn, and Timo Siepmann. Randomized controlled trials–a matter of design. *Neuropsychiatric disease and treatment*, pages 1341–1349, 2016. 21

[51] Brady Neal. Introduction to causal inference from a machine learning perspective. *Course Lecture Notes (draft)*, 2020. 21

[52] Bang An, Jie Lyu, Zhenyi Wang, Chunyuan Li, Changwei Hu, Fei Tan, Ruiyi Zhang, Yifan Hu, and Changyou Chen. Repulsive attention: Rethinking multi-head attention as bayesian inference. *arXiv preprint arXiv:2009.09364*, 2020. 28

[53] Andre Martins and Ramon Astudillo. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International conference on machine learning*, pages 1614–1623. PMLR, 2016. 30

[54] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 30

[55] Sebastian Weichwald, Martin E Jakobsen, Phillip B Mogensen, Lasse Petersen, Nikolaj Thams, and Gherardo Varando. Causal structure learning from time series: Large regression coefficients may predict causal links better in practice than small p-values. In *NeurIPS 2019 Competition and Demonstration Track*, pages 27–36. PMLR, 2020. 35