Academic year
2023 - 2024

Temporal Causal Discovery with Machine Learning

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 and daughter Elin, for their encouragement and loving support throughout this academic year. Their support has been a source of motivation and inspiration to me.

Abstract

In this study, we explore the complexities and challenges in temporal causal discovery using deep learning. Additive models can identify temporal causal relationships in data [1]. However, due to their inability to effectively approximate interactive (non-additive) relationships, they might overlook a relationship and incorrectly assign causal effects to variables. Furthermore, expanding the receptive field of a model to capture long-range relationships increases the complexity and potentially results in inaccurate 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 causal predictions. In this study, we provide a comprehensive overview of the challenges in temporal causal discovery, covering both general challenges as well as specific challenges associated with methods suggested by prior works. To address these challenges, we make three key contributions: (1) We incorporate a Temporal Convolutional Network (TCN) to process time series data. This architecture expands the receptive field and increases the complexity, which allows the model to learn more complex, non-linear, and long-range relationships. (2) We introduce the Temporal Attention Mechanism for Causal Discovery (TAMCaD) architecture. This framework is capable of capturing interactive relationships. Furthermore, as it produces a causal matrix for every timestep, TAMCaD can also identify contemporaneous relationships. (3) We describe the process of generating synthetic time series data that hold all of these properties. (4) By integrating predictive uncertainty into attention logits and causal contributions [2], we quantify both aleatoric (data-centric) and epistemic (model-centric) uncertainties, paving the way for future research to enhance the precision and interpretability of the identified causal relationships. By reducing the complexity of a TCN using weight-sharing and recurrent layers, we achieve comparable performance, while reducing the number of learnable parameters. While TAMCaD shows the ability to learn interactive relationships, we find that the interpretability of attentions remains a challenge. Our findings further suggest that additive models are adept at identifying the most evident relationships, which currently makes them more robust than our proposed attention-based method. Nonetheless, our findings also suggest that the attention-based approach holds promise for improving temporal causal discovery.

Contents

1 Introduction

Causal discovery is becoming increasingly important in various fields such as healthcare, economics, and social sciences. Understanding the causal relationships between variables in a system can help make informed decisions, develop predictive models, and gain insights into complex systems. In particular, temporal causal discovery, which infers causal relationships from time series data, helps in understanding various dynamic temporal systems. For example, identifying causal links between lifestyle choices and health outcomes can lead to better preventive measures and treatments. In economics, understanding causal factors behind market trends can improve policy-making and financial forecasting. However, inferring causal relationships from time series data is a difficult task and poses several challenges related to complex relationships, time lags and non-stationarity. Furthermore, since decisions based on these causal discoveries can have substantial implications, it is important to have a reliable measure for quantifying uncertainty in the model and the data. Existing methods for temporal causal discovery have limitations and caveats, and may not always produce reliable and interpretable results. Therefore, there is a need to develop approaches for reliable temporal causal discovery with machine learning that addresses these identified challenges. The goal of this thesis is to provide insights into dynamic temporal causal relationships which contribute towards a robust and effective framework for temporal causal discovery.

This thesis makes contributions by addressing various identified challenges in the field of temporal causal discovery. One of the challenges is the handling of contemporaneous relationships in observational data, where certain causal connections appear only temporarily or under specific conditions. For instance, a heavy rainstorm, a rare event, may temporarily influence various environmental factors. Similarly, the introduction of new law enforcement policies can drastically alter market dynamics, creating or disrupting existing causal links. These examples highlight how higher-level abstractions in observational data can result in contemporaneous relationships and necessitate improvements to existing methods that only assume a static underlying causal structure. Another challenge involves the complexity of interactive relationships. Simple additive models, which consider variables individually, often fail to capture the full extent of these relationships. For example, a relationship between two variables involving a multiplication of their values may not be accurately represented. Further research is needed to determine to what extent such interactive relationships are present in observational data and to assess whether the aspect of temporal data processing allows additive models to identify these complexities within the set number of lags. Given the complexity of real-world data and the inherent uncertainty in the discovery process, our findings might not have universal applicability.

To address these challenges, our approach moves away from traditional additive methods, implementing an attention-based causal discovery framework. The attention mechanism allows for mixing of features, while maintaining interpretability, effectively addressing interactive variables. Unlike methods such as Neural Additive Vector Autoregression (NAVAR)[1], which process an entire time series to generate a single causal matrix, our attention-based approach generates a causal matrix at each time step, addressing contemporaneous relationships. We also improve our proposed method Temporal Attention Mechanism for Causal Discovery (TAMCaD) and NAVAR to incorporate uncertainty in the predicted causal structure. Often, these models provide scores indicating causal relationships, but the interpretation of these scores and the confidence level of the model's predictions often remain ambiguous. This aspect is particularly important in fields like medical research, where reliable causal discovery between health indicators is vital. To address this, we have integrated variational inference techniques alongside an ensemble of

models within TAMCaD and NAVAR. This approach aims to quantify the "predictive uncertainty", which can be disentangled into two distinct types of uncertainty: aleatoric uncertainty, which pertains to the inherent randomness in the data, and epistemic uncertainty, which is related to the uncertainty of the model. This enhances the interpretability and reliability of the causal inferences made by these models. Furthermore, we address the challenges associated with the complexity and computational demands of deep learning models, such as Temporal Convolutional Networks (TCNs), which are employed to efficiently capture long-range dependencies. While these models are powerful, the large number of parameters can lead to overfitting, and they often require substantial computational resources. This complexity presents a challenge in fields like finance, where real-time analysis of market trends is essential, or in autonomous driving systems, where rapid processing of sensor data for decision-making is required. We hypothesize that employing an overly-complex model for a relatively small time series dataset reduces the performance for reliable causal discovery due to overfitting.

The novelty of this work centers on the application of an attention mechanism directly within the causal mechanism. This stands in contrast to recent studies, which have primarily applied attention as scalar factors only in the initial layer of the model [3]. Additionally, our methodology extends this concept by incorporating a scaled dot-product attention mechanism, derived from the transformer architecture. Another distinctive feature of our research is the integration of uncertainty methods directly into the causal discovery process. Unlike other recent studies that have implemented variational inference on a parameterized causal matrix [4], our approach embeds uncertainty quantification within the causal discovery model itself.

The thesis is structured as follows: Chapter 2 provides background information on causal discovery, including domains in causal machine learning, preliminaries and notations for causal discovery, time series forecasting with deep learning, methods for uncertainty quantification, and methods for temporal causal discovery using machine learning approaches. This chapter also further discusses the challenges associated with temporal causal discovery. In Chapter 3, we introduce our proposed method for temporal causal discovery and discuss improvements for efficient temporal processing using low-complexity TCNs and the incorporation of uncertainty methods in the causal discovery process for both TAMCaD and NAVAR. Chapter 4 details the experimental setup, including descriptions of datasets, implemented models, training approaches, and other implementation specifics. The results of these experiments are then presented and discussed in Chapter 5 and determine the efficacy of our proposed methods in addressing these challenges by evaluating the performance of TAMCaD, NAVAR on synthetic and real-world datasets. This initial testing phase on the synthetic data shows how the methods perform under controlled conditions. Following this, we extend the application of our methods to real-world datasets, validating their practicality and effectiveness in more complex, uncontrolled environments. Potential future research directions and our other approaches that did not make this thesis are listed in Chapter 6. Finally, our work is concluded in Chapter 7.

2 Background and related work

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 [5, 6]. However, the process of inferring causality is not always straightforward, and there are several challenges associated with it, which will be discussed in this Chapter. Consequently, there has been a growing interest in methods for causal discovery that address these challenges. With the recent increase in large available datasets and advances in computational power, machine learning techniques have become more accessible and popular [7]. It allows for developing new methods to better understand causal relationships between variables, resulting in more accurate predictions and more effective interventions [5]. As such, the use of machine learning techniques has become a promising avenue for advancing causal discovery research.

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 [6, 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, 1, 4]. 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 [15, 16]. 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) [17], 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 [18].

Causal Reasoning. Similar to the other domains, causal reasoning involves identifying cause-effect relationships between variables. However, the goal of causal reasoning is predicting outcomes and answering questions in a retrospective or interventional way [19]. 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 [20]. Causal reasoning also appears in the context of recommender systems in the form of counterfactual reasoning [19, 21]. 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 [5]. 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.

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

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. 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 [5].

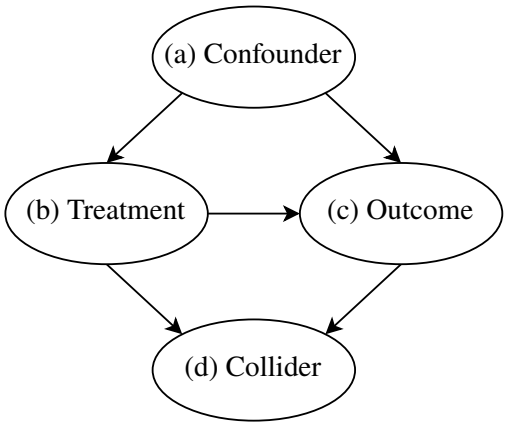

Figure 2.1: An SCM consisting of four variables represented as a DAG.

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-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. Furthermore, 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 $\text{do}(X = x)$ [22]. This approach is typically used to unveil direct cause-and-effect relationships. For instance, in a randomized controlled trial (RCT), participants are randomly 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 $\text{do}(X = g(z))$, and its effects can be observed by examining the distributions after the intervention [23]. Soft interventions are widely employed in biology and medicine, where completely removing parental influences is challenging, but perturbing them is more feasible [24].

2.3 Time Series Forecasting with Deep Learning

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 traditional 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 Memory (LSTM) and Gated Recurrent Unit (GRU) 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.

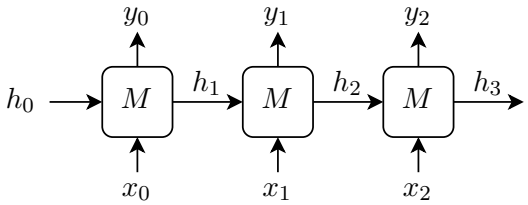

Figure 2.2: A recurrent model (M) processing temporal data (x) sequentially. Each prediction (y) relies on the input as well as a context vector (h), which has a fixed representational capacity.

Temporal Convolutional Networks. Convolutional networks 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 [25]. They have demonstrated significant potential across a variety of time series forecasting domains, including energy load forecasting [26], weather forecasting [27], stock market prediction [28], as well as multiple computer vision tasks [29, 30]. 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 [25, 27]. For example, utilizing an LSTM did not yield significantly better results compared to using a simple 1-layer convolution within the NAVAR framework [1]. Therefore, we regard convolutional networks as a natural starting point for sequence modeling tasks within the context of causal discovery [25].

TCNs are a specific type of convolutional neural networks that excel in capturing long-term temporal dependencies within time series data [29]. 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 [25]. 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 when more layers are added.

The schematic overview of a TCN is depicted in Figure 2.3. Each layer (left) in the schematic 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. Even though discovering the number of lags is not addressed in this work, it is 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.

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

The receptive field (RF) of a TCN can be computed using Equation 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. The original work on TCNs, the temporal block consists of two convolutional layers, defaulting ℓ to 2.

Attention-Based Networks. 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.

The Scaled Dot-Product Attention is an improved attention mechanism that was introduced as part of the Transformer model architecture [31], 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 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 Scaled

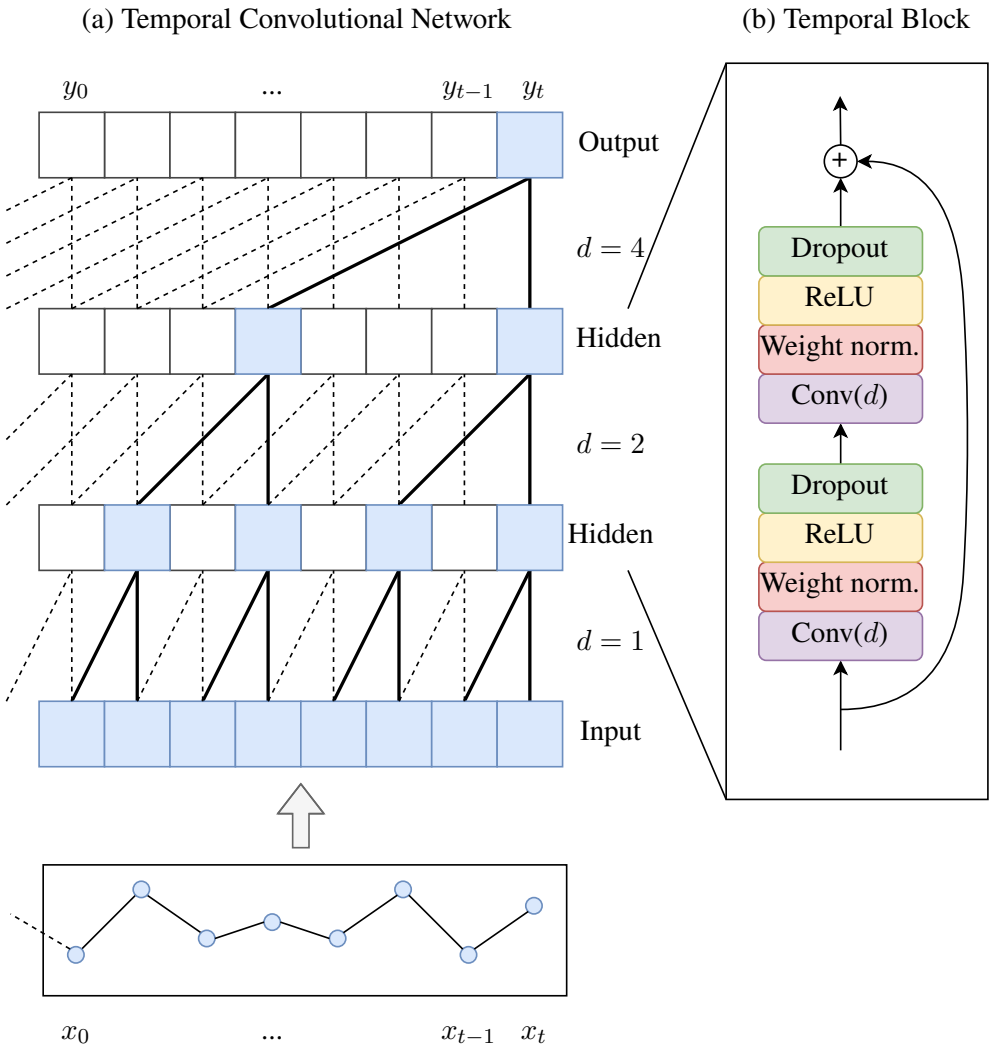

Figure 2.3: A Temporal Convolutional Network comprised of $b = 3$ temporal blocks. (a) The architecture stacks multiple layers with filters of increasing dilation d to achieve exponential growth of the receptive field. (b) Each temporal block typically consists of two convolutional layers $\ell = 2$, and incorporates a residual connection.

Dot-Product 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 d_q represents the embedding size of the (query) input that will be partitioned into h heads of size d_q/h.

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, which is responsible for incorporating data from V to a new embedding, is a simplex ($a_{ij} \geq 0 \wedge \sum_j a_{ij} = 1$).

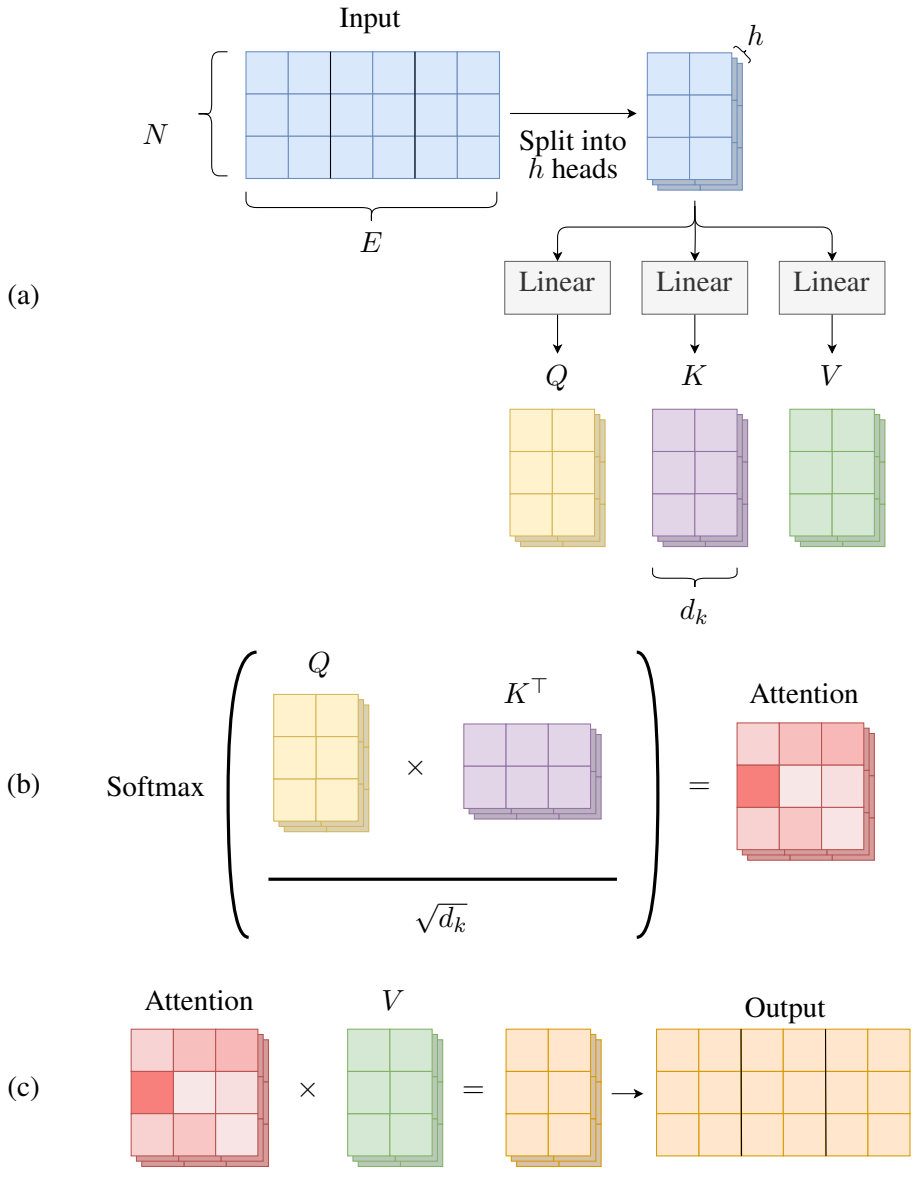

Figure 2.4: Depiction of the Multi-Head Attention Mechanism. (a) The initial input of dimensions $N \times E$, with N representing the number of embeddings and E denoting the size of each embedding, is partitioned into h separate heads across the embedding dimension. Each head is passed through three 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.

2.4 Methods for Uncertainty Quantification

Quantifying uncertainty in causal discovery is important for robust and reliable predictions, particularly in fields where decisions based on causal relationships have significant consequences. It may help in distinguishing between genuine causal connections and spurious correlations. Furthermore, addressing uncertainty can guide research where additional data may clarify uncertain causal conclusions. 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, this may give the model the tools to capture the history-dependent noise and inherent variability present in the data. On the other hand, epistemic uncertainty stems from a lack of knowledge or information in the data and can be interpreted 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. While the aleatoric uncertainty can help in defining the distribution of the data, epistemic uncertainty is still important as a model may make random predictions for unseen or out-of-distribution (OOD) input data.

Gaussian Processes. A Gaussian Process (GP) is a class of non-parametric Bayesian models used for regression and classification tasks [32]. Their flexibility and computational simplicity makes them popular choice [33]. 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. These uncertainty estimates can be interpreted as aleatoric as well as epistemic uncertainty; the upper and lower bounds of the Gaussian estimates may encapsulate the distribution over the data but also predict the bounds for sparse or missing data. 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.

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. Using a Gaussian posterior distribution, the reparameterization trick denoted in Equation 2.4. Here, $\hat{\mathbf{z}}_i$ is a sample derived by adding the predicted mean to the scaled variance, drawn from $\mathcal{N}(0, 1)$.

$$\hat{\mathbf{z}}_i = \mu(\mathbf{x}) + \epsilon\sigma^2(\mathbf{x}), \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1) \tag{2.4}$$

This concept has been successfully applied to variational autoencoders (VAEs) [34] to learn a latent representation as a distribution. This distribution can be sampled and then used as input to the decoder. Additionally, β-VAE [35] 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. This way, the probability distribution of a prediction can be learned given a certain input, which can be interpreted as the uncertainty of a prediction.

Gaussian Negative Log-likelihood. In contrast to variational inference, which is often applied on a latent space in an autoencoder, aleatoric uncertainty associated with the data can be learned in a regression context. For this, 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:

$$\mathcal{L}_{\text{NLL}}(\theta)_i = \log \sigma_i^2 + \frac{(y_i - \mu_i)^2}{\sigma_i^2} \tag{2.5}$$

However, this loss has a tendency to underestimate the variance of the data [2]. Therefore, β-NLL has been proposed by [36] to address this issue:

$$\mathcal{L}_{\beta-\text{NLL}}(\theta)_i = stop(\sigma_i^{2\beta})\mathcal{L}_{\text{NLL}}(\theta)_i \tag{2.6}$$

Here, stop$(\cdot)$ indicates that the flow of gradients through $\sigma_i^{2\beta}$ is blocked. Thus, the predicted variance is used as a weight to each data point, with β regulating the strength, giving more weight to predictions with larger variances.

Stochastic 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. The benefit is that only one model has to be trained. However, this approach tends to estimate higher epistemic uncertainty in and outside of the distribution [2]. (2) Using an ensemble of models where each model provides independent predictions. This approach provides better epistemic uncertainty measures outside of the distribution [2]. (3) Implementing a Bayesian Neural Network (BNN) that learns a distribution over all its weights. 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 (EDL) quantifies epistemic 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 EDL, a Dirichlet distribution is used as a prior for modeling uncertainties in predicted categorical probabilities for each class [37]. Here, $\mathbf{p}$ is the probability density function that can be sampled from the Dirichlet distribution:

$$\begin{aligned} y &\sim \text{Categorical}(\mathbf{p}) \\ \mathbf{p} &\sim \text{Dirichlet}(\boldsymbol{\alpha}) \end{aligned} \tag{2.7}$$

Introducing a two-stage learning framework significantly enhances uncertainty estimation in classification tasks, boosting Area Under the Curve (AUC) and training robustness [38]. The proposed method can be applied to various types of deep learning models, making it a useful method in various domains.

Deep Evidential Regression. In regression tasks, Deep Evidential Regression (DER) can be used for estimating uncertainty [39]. Here, μ and σ^2 denote the aleatoric uncertainty. The model is required to only learn parameters α, β, γ and v, to model the distribution over the aleatoric uncertainty:

$$\mu \sim \mathcal{N}(\gamma, \sigma^2 v^{-1}) \quad \sigma^2 \sim \Gamma^{-1}(\alpha, \beta) \tag{2.8}$$

$$y \sim \mathcal{N}(\mu, \sigma^2)$$

Despite DER's empirical success, gaps in its mathematical foundation raise questions about its workings [40]. 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:

$$\mathcal{L}_{\text{DER}}(\theta)_i = \log \sigma_i^2 + (1 + \lambda v_i) \frac{(y_i - \mu_i)^2}{\sigma_i^2} \tag{2.9}$$

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}.

Uncertainty-Aware Attention Mechanism. While EDL has been used in classification tasks, exploring its use in attention mechanisms within causal discovery contexts presents an interesting direction. Attention scores in these mechanisms could be interpreted similarly to classification probabilities. However, the attention strengths, typically derived from weakly-supervised training models, can be inaccurately allocated [41]. This misallocation makes the probabilistic interpretation of attention scores unreliable and complicates the quantification of causal links. Therefore, they propose an uncertainty-aware attention mechanism that integrates variational inference into the attention layer, which enables direct uncertainty estimation in attention scores. They show improved calibration of predicted probabilities and with that, improved interpretability of attention mechanisms, which may be useful when implementing attention mechanisms for causal discovery. However, the Expected Calibration Error (ECE) that was used faces limitations in causal discovery, as the true causal relationships are often unknown or only represented as binary labels.

Disentangling Predictive Uncertainty. There is still ongoing research in quantifying and disentangling uncertainty in neural networks. Therefore, another view of uncertainty in deep learning models is presented in [2]. They argue that learned posterior distributions represent a "predictive uncertainty", which incorporates both aleatoric and epistemic uncertainty:

$$\sigma_*^2(x) = \mathbb{E}_i \left[\sigma_i^2(x)\right] + \text{Var}_i \left[\mu_i(x)\right] \tag{2.10}$$

In Equation 2.10, the predictive uncertainty $\sigma_*^2(x)$ is decomposed into aleatoric uncertainty, represented by the expected variance across predictions $\mathbb{E}_i \left[\sigma_i^2(x)\right]$, and epistemic uncertainty, denoted by the variance of mean predictions $\text{Var}_i \left[\mu_i(x)\right]$. Here, index i is a sample instance from, for instance, Monte-Carlo dropout (in a single model) or a model prediction within an ensemble (across multiple models). This approach aligns with the principles of DER. However, it avoids the need for complex prior distributions over the aleatoric uncertainty distribution. For classification tasks, approximating a Dirichlet distribution as part of the evidence-gathering process is theoretically correct but challenging to implement. Therefore, [2] suggests using sampling on softmax outputs to obtain mean predictions for each instance and using entropy as a measure of uncertainty. However, it should be noted that the disentanglement of predictive uncertainty is feasible only when applied to logits prior to the softmax operation. Their findings further indicate that using Monte-Carlo dropout tends to overestimate epistemic uncertainty across the entire training data distribution, while ensemble methods demonstrate lower uncertainty on OOD training

data. However, using β-NLL (Eq. 2.6) over the standard NLL improves the outcomes, with ensembles providing the most reliable estimations for both epistemic and aleatoric uncertainties, including OOD data. The idea of β-NLL aligns again with the use of v in DER.

2.5 Methods for Temporal Causal Discovery

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 [42]. 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 [42]. 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 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 2.11 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.

$$X_t^i = f_i(\mathbf{Pa}_{<t}^i) \tag{2.11}$$

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 [43]. The challenge is to approximate these functions f_{ij}, which can be parametric or non-parametric.

$$X_i = \beta_i + \sum_{j=1}^{N} f_{ij}(X_j) \tag{2.12}$$

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 [44].

$$X_t^i = \beta^i + \sum_{j=1}^{N} f_{ij}(X_{<t}^j) \tag{2.13}$$

As shown in Equation 2.13, 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.

$$X_t^i = \beta^i + \sum_{j=1}^{N} \sum_{\tau=1}^{K} [A_\tau]^{ij} X_{t-\tau}^j + \eta_t^i \tag{2.14}$$

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 [1, 4]. 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.

$$c_{t,j\to i} = f_{ij}(X_{t-K:t-1}^{(j)}) \tag{2.15}$$

$$X_t^{(i)} = \beta^i + \sum_{j=1}^{N} c_{t,j\to i} + \eta_t^i \tag{2.16}$$

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:

$$\text{score}(j \to i) = \sigma_t \left[c_{t,j \to i}\right] \tag{2.17}$$

However, the model might learn spurious or static outputs for $\mathbf{c}$, which could vary across experiments, leading to inconsistent results. To address this issue, a regularization term was introduced to minimize contributions to zero. The sum of the absolute values of all $i \to j$ contributions over t time steps is calculated as follows:

$$\mathcal{L}_{\text{reg}} = \sum |\mathbf{c}| \tag{2.18}$$

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.

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 [3]. 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, through the 1×1 convolution that combines the outputs of the TCN into a single value, without using 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:

$$X_t^{(i)} = \sum_{j=1}^{N} \mathbf{w}_{ij} \odot \text{TCN}_{ij}(a_{ij} \odot X_{t-K:t-1}^{(j)}) \tag{2.19}$$

In this equation, $\mathbf{w}_{ij}$ represents the weight vector used to transform the output channels produced by the TCN, a_{ij} 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.

Rhino. Recently, Rhino was introduced [4], 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:

$$X_t^i = f_i(\mathbf{Pa}_G^i(< t), \mathbf{Pa}_G^i(t)) + g_i(\mathbf{Pa}_G^i(< t), \epsilon_t^i) \tag{2.20}$$

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:

$$f_i(\mathbf{Pa}_G^i(< t), \mathbf{Pa}_G^i(t)) = \zeta_i\left(\sum_{\tau=0}^{K}\sum_{j=1}^{D} G_{\tau,ji}\ell_{\tau j}(X_{t-\tau}^j)\right) \tag{2.21}$$

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 [45], 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.

2.6 Challenges in Temporal Causal Discovery

Dynamic temporal relationships. Observed relationships can be classified into three categories: static, contemporaneous and sequential [46]. 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:

$$X_t^{(1)} = X_{t-1}^{(2)} \cdot X_{t-1}^{(3)} \tag{2.22}$$

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.

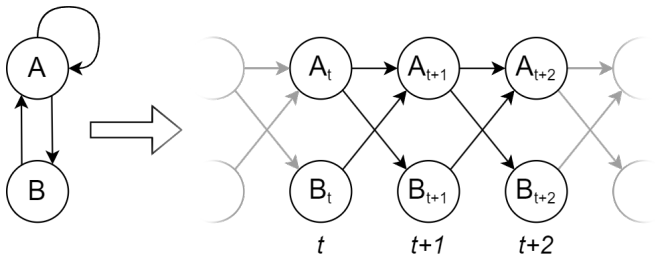

Figure 2.5: Expansion of a temporal SCM into a DAG.

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.

Interactive Relationships. Interactions between variables refer to the dependencies and influences that one variable may have on another over time. For example, the following relationship is non-additive, meaning that X cannot be accurately predicted when one of the variables is missing.

$$X_t^{(0)} = X_{t-1}^{(1)} \cdot X_{t-1}^{(2)} \tag{2.23}$$

Methods for causal discovery in time series data are often based on a GAM (Eq. 2.12). These approaches assume independent variables, where they can be approximated as the sum of various complex independent functions, which is not always the case in naturally occurring datasets [47]. 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 individual predictor functions, of which only a fraction 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 super-exponentially with the number of variables, making it a challenging topic in the field of causal discovery.

The methods discussed in Section 2.5 all 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 [48]. This involves learning additional terms that consist of subsets of variables, resulting in an interpretable additive model where variables are allowed to depend on each other. However, this 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.

One type of real-world data that can exhibit interactions between variables is epistasis in gene expression data. Epistasis refers to the interaction between multiple genes, where the effect of one gene is regulated by other genes. In time series data, epistasis can manifest through the dynamic interactions of genes over time, regulating gene expression and its impact on phenotypes. However, there are very few methods that can disentangle the effects of selection (including epistasis), mutation, recombination, genetic drift, and genetic linkage in evolving populations [49]. In the context of gene expression data, interactions between genes over time can be modeled using dynamic Bayesian networks [49], Granger causality analysis [1], or other time series modeling approaches [50]. These methods can reveal how the expression of one gene may influence or be influenced by the expression of other genes at different time points. Similarly, in fields such as finance, economics, and environmental science, interactions between variables over time are present, such as the relationships between stock prices, macroeconomic indicators, or environmental factors.

Transitive Relationships. In some cases, the true causal relationship can be deduced from the past values of other variables. In Equation 2.24, the dependency between $X^{(2)}$ and $X^{(3)}$ in f_1 can be decoupled, given the knowledge about the structural causal model.

$$X_t^{(1)} = f_1(X_{t-1}^{(2)}, X_{t-2}^{(3)}) + \eta_t^{(1)} \tag{2.24}$$
$$X_t^{(2)} = f_2(X_{t-3}^{(3)}) + \eta_t^{(2)} \tag{2.25}$$
$$X_t^{(3)} = f_3(X_{t-5}^{(3)}) + \eta_t^{(3)} \tag{2.26}$$

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 (Eq. 2.27). Rather than learning functions f_1 and f_2, NAVAR can make accurate predictions by learning a simplified function h.

$$\begin{aligned} X_t^{(1)} &= f_1(f_2(X_{t-4}^{(3)}), X_{t-2}^{(3)}) + \eta_t^{(1,2)} \\ &= h(X_{t-4}^{(3)}, X_{t-2}^{(3)}) + \eta_t^{(1,2)} \end{aligned} \tag{2.27}$$

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 $X^{(i)}_{<t}$ and $X^{(j\neq i)}_{\leq t}$ to predict $X^{(i)}_t$ [3].

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 $X^{(1)}_t = X^{(2)}_{t-1} + X^{(3)}_{t-1}$, the resulting time series for $X^{(1)}_t$ 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 $X^{(1)}_t = 0.99 \cdot X^{(1)}_{t-1} + 0.01 \cdot X^{(3)}_{t-1}$. 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 2.5 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 [5, 51] (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 [52]. 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 [5]. 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. RCTs are 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 [53]. However, even in RCTs, confounding variables may still exist if randomization is not performed correctly, resulting in biased estimates of treatment effects [54]. 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 [3]. 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. The difficulty with the construction of a causal matrix is that it is often not directly produced by the method. Instead, the produced data of the model must be interpreted to construct a causal matrix. These methods either depend on the data they produce (direct contributions [1] or attention scores) or analyze the internal model parameters (find the number of lags from the convolution weights), interpreting them as evidence of causal connections. 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:

$$\begin{aligned} X_t &= f_X(Y_{t-1}) + \eta_1 \\ Y_t &= f_Y(X_{t-1}, Y_{t-1}) + \eta_2 \end{aligned} \tag{2.28}$$

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

$$Y_t = f_Y(f_X(Y_{t-2}) + \eta_1, Y_{t-1}) + \eta_2 \tag{2.29}$$

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 2.5 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 Temporal Attention Mechanism for Causal Discovery (TAMCaD)

To address the limitations of additive models in capturing non-additive relationships, we introduce the Temporal Attention Mechanism for Causal Discovery (TAMCaD) for learning non-additive causal relationships. This method enables the model to differentiate between the context required to make a prediction and the contribution between variables. Unlike traditional attention mechanisms that focus on the time axis in data, such as in seq2seq models, our approach applies attention across a set of causal variables, which is computed at each time step independently. By opting for an attention-based approach, considering the complexity and expressiveness of such a mechanism is important. Therefore, we implement this approach in two-fold. First, we apply a simplistic version of an attention mechanism. Then, we increase the complexity of the attention mechanism with a scaled dot-product to allow for more expressiveness.

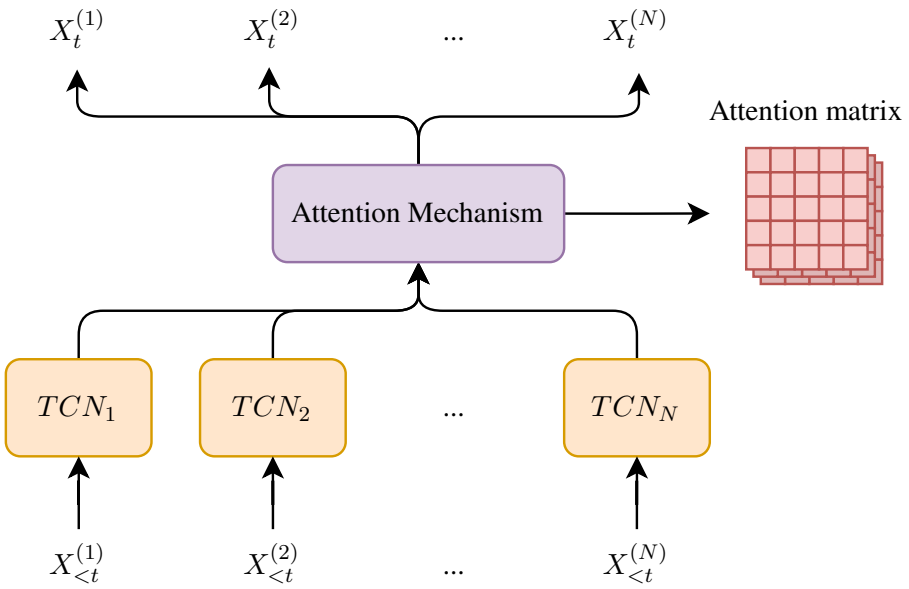

Figure 3.1: Attention-based causal discovery.

3.1.1 Architecture Overview

A high-level overview of the proposed architecture is presented in Figure 3.1. First, the time series data for each variable ($X_{<t}^{(i)}$) is processed by N distinct TCNs, resulting in a context embedding for each time series. These context embeddings are then used to generate attentions and additional embeddings. These elements are aggregated within the attention mechanism, leading to an updated context embedding. Subsequently, a regression prediction is made for $X_t^{(i)}$ based on this updated context. Although the primary focus of our study is on the attentions generated, the regression serves as the learning objective.

The attentions produced by the model are interpreted to construct a causal matrix. Contemporaneous relationships are captured over time, as the model generate an attention matrix at each point in time. Further details on this aspect will be discussed in Section 3.1.3.

3.1.2 Cross-variable attention.

In our initial strategy to implement a temporal attention mechanism, the temporal embeddings generated by N TCNs are converted into context embeddings V and attention logits $\mathbf{a}'$. These logits are then transformed into attention scores through the softmax function. These scores represent the incoming variables for making predictions. Subsequently, an updated context embedding C is computed as follows:

$$[\mathbf{a}'^{(i)}_t, V^{(i)}_t] = \text{TCN}_i(X^{(i)}_{<t}) \tag{3.1}$$

$$\mathbf{a}^{(i)}_t = \text{softmax}(\mathbf{a}'^{(i)}_t) \tag{3.2}$$

$$C^i_t = \sum_{j=1}^{N} \mathbf{a}^{j\to i}_t V^j_t \tag{3.3}$$

$$V \in \mathbb{R}^{N\times D_V\times T},\ \mathbf{a}' \in \mathbb{R}^{N\times N\times T},\ C \in \mathbb{R}^{N\times D_C\times T} \tag{3.4}$$

Here, C^i_t is the aggregation of all context embeddings. This allows for mixing features across variables, while ensuring interpretability through the produced attention matrix.

3.1.3 Learning Contemporaneous Relationships

Methods like NAVAR process predictions for each variable post-hoc to contruct the final causal matrix. In contrast, the attention mechanism offers the advantage of generating an individual causal matrix at each time step. This feature is particularly helpful for identifying 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, 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. Additionally, the learning process can be regularized to maintain a degree of similarity between causal matrices in consecutive time steps, which mitigates fluctuations or random noise in the causal links, ensuring continuous temporal progression in the causal structure.

$$\mathcal{L}_{\text{Continuous}} = \gamma \sum (\mathbf{a}_{0:T-1} - \mathbf{a}_{1:T})^2 \tag{3.5}$$

Contemporaneous Relationships in NAVAR. When using NAVAR, a standard deviation (std) is applied over the time series of the contributions to construct a causal matrix (Eq. 2.17). However, this makes it impossible to discover contemporaneous relationships, as the time axis is projected to a single value. To address this, we propose to apply this operation solely to a window in the time axis of the contributions. By implementing a sliding window along the time axis, the local variability and fluctuations are captured, capturing contemporaneous relationships. The larger the window size, the larger the sample size, the more confident we are of the output. However, a limitations is that this also leads to the changes appearing smooth, whereas some contemporaneous relationships may be abrupt. There can be made a trade-off between confidence interval of the std and the accuracy of the contemporaneous relationships.

In situations where it is known that the causal matrix is time invariant, learning a single, parameterized causal matrix —similar to [4]— may be more computationally efficient and lead to better results. However, we leave this exploration to future work.

3.1.4 Extending to Scaled Dot-Product Attention

We can extend the attention mechanism proposed in Section 3.1 by incorporating a scaled dot-product attention, from the transformers model proposed by [55], to capture the attention scalars between the various variables. This may have several benefits as well as various disadvantages. We adopt the scaled dot-product attention as described in Equation 2.3. Originally, this attention mechanism is applied to a sequence of embeddings, e.g. tokens from a sentence. However, as with our first approach, we do not apply it to the time axis of our data, but to the embeddings representing various time series of variables. This produces again an attention matrix of size $N \times N$ for each time step t, from which a causal matrix can be interpreted. Instead of simply aggregating the produced embeddings with attention provide by the TCN, the embeddings serve as input for the attention mechanism to generate the attention scores. It transforms the embeddings through the scaled dot product to incorporate information from other variables. The attention mechanism that incorporates the aspect of time is denoted in Equation 3.6. In the original work on transformers, the 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.

$$\text{Attention}(Q, K, V)_t = \text{softmax}\left(\frac{Q_t K_t^\top}{\sqrt{d_k}}\right) V_t \tag{3.6}$$

Multi-head Attention to Improve Representation Besides the hypothetical advantages of working with attentions, 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 [56]. 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. In our experiments, we exclude it for a more interpretable evaluation of the attention mechanism itself and leave this for future work.

3.1.5 Causal Interpretability of Attention Scores

TCDF focuses on interpreting the attention scalars, applied prior to convolutional layers, and analyze the convolutional weights to pinpoint influential variables [3]. Rhino, on the other hand, directly learns a distribution over the causal matrix [4]. In the context of NAVAR, its additive framework allows for learning causal contributions of variables in a regressive prediction model [1]. By computing the variability of these contributions, the causal links can be prioritized and ranked. For NAVAR, it is important to note that this variability serves not as a confidence score or a probability indicator of the causal link, but as a direct measure of a variable's quantitative contribution to another. This interpretation

can be skewed if the time series data are not normalized, leading to disproportionately high or low scores for certain variables. Additionally, consider a scenario where a variable is influenced by two causal links: one strong and dominant, and the other subtle yet significant. The latter may register a lower score compared to the former, yet both are crucial for accurate predictions. Therefore, a zero-scored link can be interpreted as non-causal, while a non-zero score only suggests a potential causal link. However, the challenge in real-world applications lies in calibrating these scores appropriately, as it is often unclear whether a minor contribution is genuine or merely spurious. This uncertainty requires cautious interpretation of the causal links inferred from these scores. We will delve deeper into uncertainty in Section 3.3.

On the other hand, our approach produces an attention matrix, which could be directly used as an indication of a causal link. However, there are also issues associated with this approach. First, these attentions scores should also not be interpreted as confidence scores, probabilities or confidence scores, or as indicative of NAVAR's quantitative contributions. As these attentions aim aggregate context embeddings in a weighted manner, they rather present the capacity of information required to make a good prediction. Again, considering the scenario where a variable is influenced by a strong and dominant, and a subtle yet significant causal relationship, the model may now assign a higher attention to the second relationship as it requires more information to make an accurate prediction.

In cases where a variable lacks valid incoming causal links, the model will still allocate attention to other variables due to two reasons: (1) the softmax function inherently prevents attention all attention to become zero, and (2) the model's parameters in individual models may memorize the data to improve the regression loss. Furthermore, for a variable influenced by all other variables, the optimal attention it should receive from each is 1/N. In contrast, a variable with a single valid incoming link can receive a full attention score of 1. This discrepancy in attention allocation among variables prevents effective ranking and requires a method to align them more appropriately.

The first proposal involves applying the softmax function across the outgoing axis, which allows for zero-attention for incoming causal links, but this reintroduces constraints on the outgoing links. The second proposal is to apply softmax over the entire attention matrix. However, this might also present challenges, as the weights of the prediction layer may be inflated to scale up embedding for variables receiving almost no incoming attention. Further exploration of these interpretations, including how they could be effectively combined or used to construct a reliable causal matrix, are left for future research.

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

Traditionally, attention in the context of time series is implemented with an LSTM. However, as the LSTM will be applied over each time step, the model can only take the locality of the causal links into account through the hidden state, which may not work sufficiently for long-range dependencies. Therefore, we opt for a TCN, since the locality of the causal link is tied to the parameters in the model, which address specific amount of lags. Moreover, when dealing with high-frequency sampled data, capturing long-range dependencies can be challenging due to significant time lags between cause and effect. While LSTMs consider the locality of causal links through their hidden state, they often struggle with long-range dependencies due to the limited scope of this local context. These hidden states also hinder the analysis of the origin of causal effects. Additionally, the complexity of LSTMs can lead to overfitting when analyzing time series of a single variable.

To address these limitations, we propose the adoption of a TCN, as this architecture addresses the locality of causal links more effectively, with each parameter in the model targeting specific lags or ranges of lags.

This approach not only captures the context of these ranges, but also extends the scope to include long-range temporal dependencies. Moreover, the parallelism of a TCN makes it a more efficient approach compared to the sequential LSTM. On the other hand, increasing the depth of a TCN to enhance the receptive field results in a more complex model and can also lead to overfitting.

Recently, NAVAR demonstrated good 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. Increasing the complexity of the model 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 architecture of the TCN, both aiming to reduce the complexity of the model, while maintaining an increased receptive field. These modifications should enhance parameter and memory efficiency and potentially partially 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.

3.2.1 Weight-sharing Across Variables

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 convolutional neural networks (CNNs) and TCNs, where the same kernel (weights) are used across the entire input image or time series. As each variable is processed independently by a TCN, extending this weight-sharing over multiple variables can help reduce the number of parameters of the final model, which is especially beneficial for large datasets or limited computing resources. NAVAR [1] learns a separate network for each variable, capturing functional relationships between variables through the model's parameters. On the other hand, Rhino [4] 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 as part of causal representation learning, but we will leave this aspect for future work.

To address the issue of data distribution mismatch among variables, we propose a solution in which 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 [3]. With this method, we eliminate the need for a context embedding for each variable altogether. For example, in the case of a linear layer, where the embedding e is concatenated to the input x, the output y is given by:

$$y = W \begin{bmatrix} x \\ e \end{bmatrix} + b \tag{3.7}$$

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

$$y = \begin{bmatrix} W_x & W_e \end{bmatrix} \begin{bmatrix} x \\ e \end{bmatrix} + b \tag{3.8}$$

$$= W_e e + W_x x + b \tag{3.9}$$

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

$$y = W_x x + b' \tag{3.10}$$

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".

3.2.2 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".

3.3 Quantifying Uncertainty in Causal Discovery

In our efforts to interpret contributions and attentions for the reliable construction of a causal matrix, we have developed a method to quantify epistemic uncertainty in the causal discovery process [1]. This aspect is important, as it reflects the model's inherent uncertainty in the learned causal relationships, which directly impacts the accuracy and reliability of the causal matrix. Although aleatoric uncertainty can offer insights into the extent of history-dependent, external, and static noise in each variable, it tends to be less informative for the attentions in our proposed attention mechanism. This is because aleatoric uncertainty within the attentions does not directly relate to specific characteristics of causal relationships. Therefore, our primary focus is on addressing epistemic uncertainty.

To address this challenge, we employ a method that disentangles a "predictive uncertainty" into aleatoric and epistemic components using an ensemble of stochastic models [2]. The underlying idea is that models which learn a distribution over data (for instance, using the NLL loss) are not merely capturing aleatoric uncertainty, but rather a form of predictive uncertainty aimed at optimizing predictions. This predictive uncertainty can then be disentangled into its epistemic and aleatoric components. We adapt the models to integrate a predictive uncertainty into the attention mechanism of TAMCaD and the contribution layer of NAVAR. This integration allows us to incorporate both aleatoric and epistemic uncertainties in the analysis by applying the disentanglement process as outlined in Equation 2.10. This approach has several advantages. It adapts well within the attention mechanism framework (as classification uncertainty) and additive models (as regression uncertainty), encourages robust uncertainty quantification through ensembles, and enables comparative analysis between NAVAR and TAMCaD.

3.3.1 Robustness with Ensemble Learning

We leverage ensemble learning for estimating predictive uncertainties, as the disentanglement process is optimal compared to alternative sampling methods, such as Monte Carlo Dropout [2]. Ensemble models can discover different patterns, enhancing the robustness of uncertainty quantification and helping with imbalanced datasets. We hypothesize that single models using evidential learning or Monte Carlo dropout might not robustly eliminate spurious correlations, as we observed differing across multiple training runs in preliminary investigations. The use of a TCN offers the advantage of parallelization, extending not only to variables in the model but also across the ensemble models. Despite the computational intensity often associated with stochastic model sampling, our TCNs low-complexity design (see Section 3.2) is able to reduce excessive memory and computational requirements. Nonetheless, we demonstrate the feasibility of this approach even without the low-complexity design.

Evidential Learning Considerations Our preliminary investigations with non-ensemble (single) models, such as EDL and DER, revealed numerical and performance instability during training. These methods, which approximate prior distributions over probability distributions, produced inconsistent results and were ineffective in defining epistemic uncertainty. In contrast, averaging results across

[1] Epistemic uncertainty can also be interpreted as a metric for quantifying OOD data.

multiple models resulted in more consistent results, highlighting the strength of an ensemble approach. However, Evidential Learning remains an area for future research.

3.3.2 Predictive Uncertainty with Stochastic Variational Inference

To achieve uncertainty-aware causal discovery, our approach is to estimate the predictive uncertainty in individual models by integrating variational inference into the contribution and attention layers. However, this presents new challenges. Each variable's contributions and attentions must be represented as distinct distributions, which we term variational contributions and variational attentions, respectively. These distributions aim to encapsulate the predictive uncertainty in the influence of variables on each other. Learning these contributions can be approached through deterministic or stochastic methods.

Stochastic vs Deterministic. Typically, a deterministic objective function like the NLL loss is used to estimate Gaussian distributions in the final regression layer. However, applying it for estimating distributions over contributions or attentions presents challenges. This is primarily due to the unavailability of true values for contributions and attentions, which are required for estimating the prior distribution in these layers. As a result, deterministic methods become infeasible for our proposed attention mechanism. To overcome this, we employ a stochastic approach for variational inference with the reparameterization trick (see Eq. 2.4), which introduces randomness into the learning process, allowing the model to capture the inherent variability in the contributions and attentions. One advantage of the reparameterization trick is that it is applicable across all model layers. This contrasts with the NLL, which is confined to the final regression layer. Another advantage is that it facilitates backpropagation by allowing gradients to flow through random nodes.

Gaussian Posterior Distribution. In our approach, we opt for a Gaussian distribution to model the predictive uncertainty of contributions and attention logits, maximizing the likelihood of the predictive uncertainty. The choice of Gaussian distribution for estimating the predictive uncertainty does limit the complexity of density functions that can be approximated. While more expressive distributions exist (e.g., Gamma or Beta), their application is left for future work. We cannot use ELBO/KL regularization because our goal is to avoid constraining the model's attentions to follow a normal distribution $N(0, 1)$. In our preliminary experiments, such constraints caused all attentions to converge to zero, rendering them uninterpretable However, not using regularization might lead to a model learning a zero variance, resulting in non-stochastic behavior. However, previous studies have successfully employed variational inference without regularization by using a softplus function with a small bias to prevent variance collapse [36]. In our experiments, we observed that the variance does not converge to zero when using the softplus function.

3.3.3 Integrating Predictive Uncertainty in Causal Discovery Methods.

Integrating uncertainty in additive models. Given that the true causal contributions are unknown, deterministic variational inference cannot be directly applied. A possible approach is to consider each variable's individual predictions as a primary model for predicting other variables. This involves adjusting the error term in the NLL loss to $(y_i - \mu_{j \to i})^2$, enabling the estimation of uncertainty. However, this approach may lead to the learning of incorrect relationships and unrepresentative variances. In particular, a zero causal contribution could result in a constant error term, leading to a meaningless variance. Because of the limitations of this deterministic approach, we suggest modifying the NLL loss function to include

an ensemble that utilizes stochastic variational inference.

$$\mathcal{L}_{\text{NLL}}(\theta)_i = \log \sigma_i^2 + \frac{(y_i - \mu_i)^2}{\sigma_i^2} \tag{3.11}$$

Where the total predictive uncertainty is constructed with individual predictive uncertainties:

$$\sigma_i^2 = \sum_j \sigma_{j \to i}^2 \tag{3.12}$$

$$\mu_i = \sum_j (\mu_{j \to i} + \epsilon \sigma_{j \to i}^2), \quad \text{where} \quad \epsilon \sim \mathcal{N}(0, 1) \tag{3.13}$$

Here, the subscript denoting the time step t is omitted for simplification, but it is included in the summation across all time steps. The model sums the variances of individual contributions to calculate the total variance, allowing for identifying the source of noise for each variable. However, this approach assumes that there are no dependencies or interactions between variables (assumption of variance additivity). Directly using the summed mean predictions μ is not viable as it would not result in the model correctly learning variability across individual contributions. For instance, an individual model might capture the uncertainty for the final prediction, while other variables contribute zero. To ensure correct modeling of each variable's variability, we employ stochastic sampling of the variational contributions, combined with the NLL loss for regression. The application of the NLL loss imposes an additional constraint on the variance of individual contributions, requiring them to collectively conform to the correct total variance.

This approach allows then for the disentanglement of predictive uncertainty into aleatoric and epistemic components for each causal link $i \to j$ at time t across m models [2]:

$$\text{Aleatoric:} \quad \boldsymbol{\sigma}_{t,i \to j}^2 = \mathbb{E}_m \left[\sigma_{m,t,i \to j}^2(x) \right] \tag{3.14}$$

$$\text{Epistemic:} \quad \boldsymbol{\sigma}_{t,i \to j}^2 = \text{Var}_m \left[\mu_{m,t,i \to j}(x) \right] \tag{3.15}$$

To provide the aleatoric and epistemic uncertainties in static relationships, we calculate the mean over the time dimension:

$$\text{Aleatoric:} \quad \boldsymbol{\sigma}_{i \to j}^2 = \mathbb{E}_{m,t} \left[\sigma_{m,t,i \to j}^2(x) \right] \tag{3.16}$$

$$\text{Epistemic:} \quad \boldsymbol{\sigma}_{i \to j}^2 = \mathbb{E}_t \Big[\text{Var}_m \left[\mu_{m,t,i \to j} \right] \Big] \tag{3.17}$$

Furthermore, by averaging predictions from k ensemble models, we can estimate contributions and effectively score causal links as implemented in NAVAR:

$$c_{t,i \to j} = \mathbb{E}_m \left[\mu_{m,t,i \to j} \right] \tag{3.18}$$

$$\text{score}(i \to j) = \text{Var}_t \left[c_{t,i \to j} \right] \tag{3.19}$$

Integrating uncertainty in attention mechanisms. As noted by [41], models often generate attention strengths in a weakly-supervised manner, as in our model, where regression is the primary objective, but the focus lies on internal contributions and attentions. Incorrect allocation of attention scores might lead to unreliable predictions. Therefore, integrating uncertainty in the attention layer enhances calibration and helps with more robust predictions.

To be able to capture predictive uncertainty at the classification level within our attention mechanism without the use of a BNN, we use the method outlined by [2, 57]. As with the integration of uncertainty

in the contribution layer of NAVAR, stochastic variational inference is employed for the attention layer of TAMCaD. However, instead of learning a Gaussian posterior distribution over the contributions, it is learned for attention logits prior to the softmax function. The disentangling process is applied to the attention logits to estimate the aleatoric and epistemic uncertainties as in Eq. 3.16.

In [2], a technique called sampling softmax is used to average mean probability scores across different models. This method then employs the standard Shannon entropy score to estimate epistemic uncertainty over p, defined as $\text{entropy}(p) = -\sum_i p_i \log(p_i)$. However, this approach is originally applied in a classification context. In contrast, uncertainty in our attention mechanism presents a different challenge: low entropy does not necessarily signify low uncertainty in individual predictions. Furthermore, the sampling softmax approach can obscure certain nuances, such as the possibility of high uncertainty in one variable being overshadowed by another variable's high logit scores, or a variable with low inherent uncertainty exhibiting increased variance during sampling. These issues make probability density and entropy score unsuitable for accurately representing epistemic uncertainties in individual attention scores in our context.

To address these challenges, we propose to use the attention logits directly for scoring causal relationships. The monotonic nature of softmax maintains the ranking of each individual attention logit, resulting in a more reliable assessment of causal impacts. When comparing attention scores for variables with varying degrees of causal connections, the softmax function leads to inconsistent comparisons. In contrast, the attention logits are not affected by each other and provide a more distinct and interpretable comparison. To address the potential issue of extreme logit values, be they overly positive or negative, our approach includes the regularization of model weights. Specifically, we leverage the weight decay feature already present in the Adam optimizer. This regularization strategy is designed to keep the logits neutral, ensuring comparability across different variables.

Causal masks across ensembles to filter out spurious correlations. Employing a dropout mask during training improves the generalization capability of the model, and Monte Carlo Dropout allows for uncertainty estimation by sampling predictions. However, using dropout masks within the attention or contribution layers results in variables attending to all other variables to make better predictions. This approach conflicts with our aim to identify true causal relationships between variables, as it encourages learning spurious correlations as well. Another challenge arises in cases where variables either have no parents or have all other variables as parents. As the softmax will transform the attention logits to a simplex during training, both cases will lead to uniformly distributed attentions, which prevents us from differentiating between them. In NAVAR, the number of spurious correlations is reduced with a regularization term that forces contributions toward zero. However, this method is not applicable to our attention mechanism, as the attention outputs rely on the softmax function, which prevents all attention outputs to become zero. Therefore, we propose a regularization term based on masked attention outputs. It encourages a model to learn zero attentions, but only for a subset of the causal links. Accordingly, we introduce a mask for each model within the m-model ensemble:

$$\mathbf{a} \in \mathbb{R}^{M \times T \times N \times N}, \ \mathbf{d} \in \mathbb{R}^{M \times N \times N} \tag{3.20}$$

$$\mathcal{L}_{(\text{reg})} = \frac{\lambda}{T} \sum_{t=1}^{T} \mathbf{d} \odot \mathbf{a}_t, \quad \mathbf{d} \sim \text{Bernoulli}(p) \tag{3.21}$$

Here, the computed attentions $a_{m,t,i \to j} \in \mathbf{a}$ for m models, time step t, and causal link $i \to j$, are multiplied by the dropout mask $\mathbf{d}$, which originates from a Bernoulli distribution with the hyperparameter p. The hyperparameter λ is adjustable to modulate the regularization effect. Enforcing stochasticity across models in the ensemble can magnify uncertainties and provide a different perspective on relationship identification.

We hypothesize that through this regularization, spurious relationships or distant parent variables are more likely to be disregarded, while true causal relationships are preserved. For variables with no parent variables, the attention mechanism will consider all other variables, where the omission of individual causal links potentially have lower impact on the regression loss, which leads to high variance in attentions. For variables where all variables are parents, the model will also focus on all variables, but the variance across these attentions will be lower since all variables are essential for prediction. By leveraging a substantial number of ensembles, which is computationally feasible in our implementation, allows for a more accurate estimation of uncertainty across different attention patterns. However, it is important to note that enforcing stochasticity across models in the ensemble introduces additional complexities.

3.4 Synthetic Data Generation Process

3.4.1 Constructing Temporal Causal Graphs

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 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 $t = 300$.

3.4.2 Implementing Non-Linear Causal Relationships with Neural Networks

We can implement the non-linear and 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 the adjacency matrix G:

$$X_{t+1}^{(i)} = f_i(G \odot X_{t-K:t-1}^{(0:N)}) + \eta_t^{(i)} \tag{3.22}$$

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.

4 Experiments

4.1 Datasets

Synthetic time series data under controlled conditions. To evaluate the performance of our proposed methods, we will conduct a series of experiments on both real-world and synthetic datasets. As our method aims to address the challenges involving various characteristics of causal relationships, it is important to evaluate our method on datasets that contain these characteristics. Knowing that these relationships 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 under controlled conditions. 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. In scenarios with contemporaneous relationships, causal links are simply severed and restored at intervals (e.g., from 0.4T to 0.6T). We then visualize and compare these simulated connections with those identified by our model over time. For variables with multiple incoming connections, we simulate interactive relationships by feeding their values into a two-layer neural network.

CauseMe: Benchmark platform for real data challenges. 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 [47].

DREAM3: Gene expression data. The DREAM challenges [58] contain a collection of simulated time series gene expression datasets and offer the opportunity to evaluate our approaches on a real-world biomedical use-case. We evaluate our method on five datasets from the DREAM3 benchmark, representing E. Coli and Yeast gene networks. Each dataset contains $N = 100$ variables with 46 available time series, each spanning $T = 21$ time steps.

4.2 Evaluation Metrics

For measuring the effectiveness of our approaches, we employ several metrics to capture different aspects of model performance. These metrics will be computed for the test set, which comprises 30% of the total data. This data split is taken along the time axis to preserve the temporal dependencies of the data. The training set comprises the initial 70% of the time series (from 0 to $0.7T$), while the test set encompasses the remaining 30% (from $0.7T$ to T), where T represents the total length of the time series.

AUROC for Causal Matrix Assessment. One of the key metrics we utilize is the Area Under the Receiver Operating Characteristic Curve (AUROC). The 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, the AUROC quantifies how well our approach can rank true causal relationships. However, there is a potential issue with using the 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 on model performance. Therefore, the authors behind the CauseMe benchmark underscore the significance of prioritizing a high True Positive Rate (TPR) for evaluating their datasets [59]. In the case of experiments with contemporaneous relationships, the AUROC will be computed over all the constructed causal matrices at each timestep, where a prediction vector is of size $N \times N \times T$.

Soft-AUROC for Uncertainty Assessment. Applying ROC analysis to predictions with uncertainty, represented as probability distributions rather than discrete values, requires some modifications. Instead of a prediction being classified strictly as true/false positive/negative, samples may fall into both categories simultaneously. In this context, the True Positive (TP) count becomes the sum of all probabilities of positive predictions for the actual positive samples. Given that the probabilities are calculated for many thresholds, yielding different values, the TPR and FPR also differ at each threshold, resulting in a smooth curve. For example, consider a sample that should be classified as positive with a mean prediction of 0.1 and a standard deviation of 0.01; this prediction would be penalized significantly more compared to a prediction with the same mean but a larger standard deviation, as the model admits uncertainty regarding the sample's (incorrect) negative classification. Here, we propose a method to adapt ROC analysis using uncertainty estimates. To the best of our knowledge, there have been no other studies that calculate an AUROC score in this way.

Rather than using the predicted values as thresholds, we define k thresholds, $\{\tau_i\}_{i=1}^{k}$, uniformly distributed between 0 and 1. At each threshold τ_i, the True Positive Rate (TPR) and False Positive Rate (FPR) are computed. Let $P = \{p_1, p_2, \ldots, p_n\}$ be the set of positive samples and $N = \{n_1, n_2, \ldots, n_m\}$ be the set of negative samples, along with their (Gaussian) uncertainty scores. For a given threshold τ_i, the True Positive (TP) and False Positive (FP) contributions of a sample are calculated using its Cumulative Distribution Function (CDF), denoted as $CDF(x)$. The True Positive Rate (TPR_i) at threshold t_i is calculated as:

$$TPR_i = \frac{1}{|P|} \sum_{p_j \in P} 1 - CDF_{p_j}(\tau_i) \tag{4.1}$$

The False Positive Rate (FPR_i) at threshold τ_i is calculated as:

$$FPR_i = \frac{1}{|N|} \sum_{n_j \in N} 1 - CDF_{n_j}(\tau_i) \tag{4.2}$$

The Soft-AUROC curve is then obtained by plotting TPR_i against FPR_i for each threshold τ_i. This approach accommodates the probabilistic nature of predictions, allowing for a more nuanced assessment of model performance under uncertainty. The area under this curve provides the Soft-AUROC metric. Notably, the uncertainty in predictions is reflected in the shape of the curve; higher uncertainty leads to a smoother curve, approaching a linear random guess line, while lower uncertainty yields a curve closer to traditional, discrete ROC analysis.

Expected Calibration Error (ECE) is a metric that measures how well a model's predicted probabilities correspond to the actual frequencies of samples in the training data. Given that our causal discovery methods produce a single uncertainty matrix rather than individual sample uncertainty scores, standard approaches like ECE or calibration plots cannot be used to evaluate our uncertainty estimations. Moreover, we also need to address the fact that different methods might produce different ranges of uncertainty

scores, and that distributions over attentions and contributions have different meanings. The uncertainty regarding an attention score may differ from the uncertainty for a contribution. As such, directly comparing methods using Soft-AUROC is challenging when the uncertainty scores for NAVAR's contributions and TAMCaD's attention logits are incompatible. Standardizing or normalizing the uncertainty scores across the different methods before computing the Soft-AUROC can be considered to enable a fair comparison by aligning the scales of uncertainty, ensuring that differences in the Soft-AUROC scores are due to the model's predictive performance and its ability to handle uncertainty, rather than the raw scale of the uncertainty scores. Other approaches might consist of explicit regularization methods to align uncertainty estimates. For now, we normalize the uncertainty scores by dividing them by their mean, thereby placing the uncertainties within a comparable range. Since the scores are inherently positive, normalizing by dividing by the mean is preferred over traditional standardization, which involves subtracting the mean and then dividing by the standard deviation.

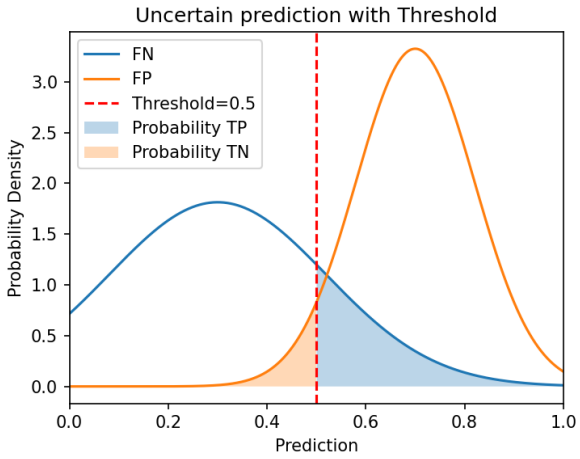

Figure 4.1: Considering uncertainty assigns a small probability to each sample of being correctly classified at a specific threshold ($\tau = 0.5$).

Adjusted Regression Loss Analysis. To evaluate the balance between memorization and performance in causal discovery, we focus on the regression loss metric. In this approach, we exclude regularization terms from the final loss calculation, facilitating an unbiased comparison across various model architectures. This is based on our observations where a model shows a lower regression loss compared to other models, while also having a worse AUROC score. This might indicate that the model memorizes data better than it uncovers underlying causal relationships. Our training data is inherently noisy, so a zero loss is not desired, as it would indicate memorization. To address this, we introduce a regression loss that is 'adjusted' for noise in the data. Here, a zero loss would be ideal as the model perfectly predicts the true means. Such a metric is particularly useful for tracking the model's performance during the training phase, especially to detect when it starts to overfit to noise, deviating from learning the actual causal relationships. During the synthetic data generation process, we have precise control over the noise variables ($\boldsymbol{\eta}$). We employ a noise-adjusted Mean Squared Error ($\text{MSE}_{\boldsymbol{\eta}}$) defined as follows:

$$\text{MSE}_{\boldsymbol{\eta}}(Y, \hat{Y}) = \text{MSE}(Y - \boldsymbol{\eta}, \hat{Y}) \tag{4.3}$$

Here, Y is the training data, $\hat{Y}$ is the model's predictions, and $\boldsymbol{\eta}$ represents the quantitative noise from the data generation process.

Number of Parameters to Indicate Model Complexity. This metric helps to quantify the complexity of each model. Generally, a model with a greater number of parameters is considered more complex. It

affects both the model's performance in terms of its learning and prediction capabilities, and its storage demands. By evaluating the number of parameters, we can gain insight into the trade-offs between model performance and resource requirements.

Compute Time per Training Epoch. The time each model takes to complete a single training epoch is particularly relevant when considering the adoption of ensemble methods. It helps in determining whether the potential improvements in performance offered by ensembles justify the additional training time required. Alternatively, this metric can reveal if the use of ensembles is practically feasible, given the available computational resources. By examining the compute time per training epoch, we can make informed decisions about the models' operational viability in different scenarios.

4.3 Models to be Implemented

In this study, we will implement and evaluate a variety of models, each specifically designed to meet the challenges in temporal causal discovery. These models comprise established architectures along with our adaptations and improvements to answer our research questions. Furthermore, they are implemented in PyTorch, as it provides us with the flexibility to customize them according to our preferences and requirements. We will compare our method with established baseline methods on the CauseMe benchmark, specifically SELVAR, SLARAC [60], and the original NAVAR. SLARAC employs a VAR model on bootstrap samples of the data, selecting a random number of lags each time. 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.

Although detailed in section 3, we list them here for easy reference:

TCN Variants

- **TCN** (Temporal Convolutional Network): A neural network architecture that uses convolutions over time series data to effectively capture dependencies defined by a number of lags.
- **TCN-WS** (TCN with Weight Sharing): An extension of the standard TCN that implements weight sharing across variables. This technique reduces model complexity and parameter count.
- **TCN-Rec** (TCN with Recurrent Layers): An adaptation of the TCN architecture by incorporating recurrent convolutional blocks, allowing repeated pooling within the same embedding space. This design increases the model's receptive field without adding extra parameters.

NAVAR Variants

- **NAVAR** (Neural Additive Vector Autoregression): NAVAR is able to identify additive causal relationships in time series data by outputting a quantitative causal matrix. It acts as a baseline model for comparison in our experiments.
- **NAVAR-UA** (NAVAR with Uncertainty Awareness): This variant addresses epistemic uncertainty by adopting stochastic variational inference in combination with the deterministic NLL loss function, also accommodating aleatoric uncertainty in variable contributions.

TAMCaD Variants

- **TAMCaD** (Temporal Attention Mechanism for Causal Discovery): At the core of this work, TAMCaD identifies non-additive, interactive causal relationships in time series data using an attention mechanism, resulting in a causal matrix based on the attention logits.
- **TAMCaD-UA** (TAMCaD with Uncertainty Awareness): Addresses epistemic uncertainty by learning distributions over the attention logits using stochastic variational inference.
- **TAMCaD-T** (TAMCaD based on the Transformer Architecture): This version uses scaled dot-product attention (from transformers), improving the embedding space of the causal variables.

4.4 Hyperparameter Optimization

To identify the optimal hyperparameter values for our models, we employ the `hyperopt` library to conduct a hyperparameter search. Hyperparameter optimization involves tuning various parameters that affect the training and performance of the models. We prioritize parameters that significantly influence the model's performance, while also considering computational feasibility and the specific requirements of causal discovery. The primary hyperparameters we focus on are listed in Table 4.1. To optimize the model training process, we use the AdamW optimizer, which is less prone to getting trapped in local optima and facilitates faster learning by employing a momentum on the gradients. Another benefit of this optimizer is the built-in weight decay.

In our experiments with synthetic data, our focus is on optimizing the AUROC score for the causal matrix derived from 30% of the test set. This metric helps assess the model's generalization to data created using the same causal graph but with different noise. To identify the most effective hyperparameter combinations for our models, the hyperparameter optimization process performs 100 different evaluations.

In scenarios where the true causal matrix is unavailable, such as with the CauseMe benchmark, we rely on using the regression loss, excluding its regularization terms, otherwise the lambda parameter itself contributes to the loss. However, it remains to be seen whether this approach effectively identifies the best hyperparameters for constructing the causal matrix, as opposed to simply optimizing for regression loss.

The process of optimizing parameters is computationally intensive and does not always yield a model effective in causal discovery. Moreover, doing hyperparameter optimization for the CauseMe benchmark, followed by processing all 200 datasets, stretches over several hours. Consequently, we have chosen to optimize only a selected set of hyperparameters. Through additional experiments, we determine which hyperparameters our model is particularly sensitive to and which ones can remain fixed. This insight led us to exclude certain hyperparameters from our optimization process, as they consistently yielded satisfactory results across various experiments. Nevertheless, our repository still provides the code to perform a comprehensive hyperparameter search if required.

Hyperparameter	Selection	Description
Learning Rate	[1e-5, 1e-1]	The learning rate represents the size of the steps taken during gradient descent, influencing how quickly the model converges during training.
Epochs	2000	Number of steps over the training data.
Hidden Dimension	{8, 16, 32, 64, 128, 256}	Determines the size of the model's hidden layers, which in turn influences the flow of information and the model's ability to capture complex relationships.
Lambda λ (Regularization)	[1e-3, 1]	Strength of regularization applied, affecting contributions in NAVAR and attentions in TAMCaD.
Weight Decay	1e-2	Acts as a regularization technique, preventing the model from learning excessively large weights, which helps with generalization and regulates the attention logits in TAMCaD.
Dropout	0.2	Dropout is another form of regularization where a fraction of the input is randomly set to zero in each training iteration. This technique reduces dependency on specific features and improves robustness of the model. Dropout is typically set at 0.2 in machine learning, as this rate strikes a good balance between generalizability and specificity.

Table 4.1: Hyperparameter selections for optimization. Fixed values are denoted by a single value, choice of values by {} (set notation), and ranges for lognormal distribution in hyperopt by [].

4.5 Resources

The resources for this research are organized and accessible for replication and further exploration.

Repository: The repository for this project is hosted on GitHub[1] and is publicly available. It contains Jupyter notebooks that outline the data generation process and the experimental setups used throughout the study. Additionally, it includes the source code for all implemented models along with others that have been developed in this thesis.

Data Availability: To facilitate reproducibility, the synthetically generated data used in the experiments is also made available through the same GitHub repository. This allows others to directly use the data for a comparative study or to replicate the experimental results presented in this thesis.

Hardware: The computational aspect of this research, particularly the training of models, was conducted using an NVIDIA GeForce GTX 950M GPU.

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

4.6 Problem Statements and Hypotheses

Impact of model complexity on performance. For reliable causal discovery, the balance between model complexity and performance is essential. While a more complex model can capture more complex relationships, there is a risk of overfitting or data memorization, potentially leading to unreliable causal predictions. We hypothesize that increased complexity leads to overfitting, affecting the model's ability to generalize and perform accurate causal discovery. Our primary goal is to determine whether increased complexity undermines effectiveness in causal discovery, especially in our attention-based model and for simpler causal structures.

A possible approach to finding a suitable model complexity is by performing hyperparameter optimization on a test set. However, this approach is insufficient when a long-range TCN will overfit on data regardless of the hyperparameter settings. In such cases, the test loss might not accurately reflect the model's effectiveness and could lead to selecting hyperparameters that do not actually maximize generalization.

A common method to determine an appropriate model complexity involves performing hyperparameter optimization on a test set. However, this approach is insufficient when a long-range TCN will overfit on data regardless of the hyperparameter settings. Under these conditions, the test loss may not accurately reflect the effectiveness of the model and could lead to selecting hyperparameters that do not maximize generalization.

To investigate model complexity, we compare the performance of a complex model, such as a TCN with multiple layers, against simpler models, like TCN with Weight Sharing (TCN-WS) and TCN with Recurrent Layers (TCN-Rec), using the NAVAR framework. Each of these architectures is set up to have a large receptive field. For example, a deep TCN model comprising four dilated temporal layers with a kernel size of 2 and a hidden dimension of 16 for five variables enables a maximum lag of 31 time steps and totals 4144 parameters per variable. For the weight-sharing variant, recurrent variant, and the combined version, this is 1532, 1968, and 1097 parameters per variable respectively.

First, we conduct an experiment to assess the models' propensity for data memorization. We hypothesize that simpler models may be less prone to memorization compared to the more complex ones. The data for this experiment, representing three variables, is generated from a normal distribution, $\mathcal{N}(0, 1)$, over 800 time steps and does not have a causal structure. This setup will allow us to evaluate how well each model can memorize and reproduce a random time series and how this affects the reconstruction of a causal structure. A second experiment focuses on the models' ability to reconstruct a causal structure and how this process evolves during training. We synthetically generate data comprising five variables (N) over 500 time steps (T), with a maximum of 10 lags (K). This experiment helps in understanding how well each model, with varying complexities, can uncover and reconstruct underlying causal relationships in noisy data. For both experiments, models are trained with the objective of minimizing regression loss with the corresponding regularization terms, aiming to reconstruct the input time series as precisely as possible.

Efficacy of attention in learning interactive relationships. This experiment investigates the effectiveness of attention mechanisms in learning and identifying interactive relationships within data. To test this, we will generate synthetic data featuring complex, non-linear interactive relationships among variables. These relationships will be modeled using a two-layer neural network. Our focus will be on comparing the capabilities of TAMCaD and NAVAR in learning these interactive relationships. We hypothesize that attention mechanisms that leverage feature-mixing will be more adept at capturing these interactive relationships compared to models that rely on additive approaches.

The DREAM benchmark provides an opportunity to explore interactions, like epistasis, in real-world

data. Additive models may encounter limitations when dealing with epistatic effects in gene expression data, as they generally focus on individual gene effects and may not account for the interactive effects of these genes. The complexity introduced by epistasis can obscure or alter gene expression patterns, which poses challenges in identifying causal relationships within genetic networks. Our hypothesis is that the non-additive model will outperform the additive model when interactions like epistasis are present. Additionally, we hypothesize that the regulatory influence of genes may vary over time, potentially being more pronounced during certain intervals due to biological processes, which could be captured with our attention-based approach.

Evaluation of TAMCaD and NAVAR in discovering contemporaneous relationships. We aim to explore the effectiveness of TAMCaD and NAVAR in identifying contemporaneous relationships within data. These relationships are not static, but vary over time. To simulate this, we generate data where causal links are severed and restored at specific intervals. TAMCaD's inherent ability to output a causal matrix at each time step potentially makes it an effective method for accomplishing this. We compare this with NAVAR, where we have designed a method to compute a causal matrix over time with a sliding window. The hypothesis is that TAMCaD, with its time-step specific matrices, will be more adept at recognizing and adapting to these sudden changes in causal relationships. NAVAR's sliding window-based approach might lag in responsiveness to such temporal dynamics, but is potentially more stable against individual time-step outliers.

Advantages of dot-product attention. We further explore the benefits of incorporating dot-product attention, particularly in a causal model setting. We compare the performance of TAMCaD and TAMCaD-T, which involves learning embeddings that are used to generate dot-product attentions. Our focus is on whether these embeddings contribute to improved performance or expressiveness, especially in scenarios involving a large number of variables ($N = 16$). The practicality and computational feasibility of such an approach, given the N^2 growth of the attention mechanism, are also considerations. Furthermore, we intend to visually interpret the utility of these embeddings through dimensionality reduction techniques like t-SNE. Here, the Key and Query embeddings can be placed against each other. A particular point of interest is whether the high dimensionality, typically advantageous in contexts like word embeddings, translates effectively to this domain, especially considering the challenges posed by large variable counts and the handling of missing data.

Applicability to real-world data. Another essential aspect of our research is to assess the applicability of our theoretical improvements in the context of real-world data. While our synthetic data allows for controlled comparison of our approaches, the true test lies in their efficacy in practical scenarios. Evaluations should determine whether the improvements, particularly in learning contemporaneous and interactive relationships, are relevant and beneficial in real-world datasets. Improved results would suggest that our approach is successful in addressing these complex data characteristics. Conversely, a decrease in performance could imply either an absence of these complex characteristics in the dataset or an over-complexity in our models. This might suggest the need for simpler models for effective causal discovery in real-world scenarios, such as NAVAR. However, it is important to note that without access to the original structural causal models used to generate the data, our conclusions will primarily focus on the observable performance metrics rather than the efficacy of our theoretical improvements for the underlying causal structures. Therefore, we evaluate our method on the DREAM3 benchmark, as discussed in Section 4.1, where we have access to the ground truth causal structure.

5 Results and Discussion

5.1 Model Complexity vs. Performance Evaluation

We evaluated different model variants within the NAVAR and TAMCaD frameworks for processing temporal dependencies in low-complexity architectures. The primary objective is to analyze the memorization capabilities of these models and their efficiency in reconstructing causal models at varying levels of complexity.

Training on noise data. First, we investigate how models overfit under various conditions, with a particular focus on the sensitivity to hyperparameters that regulate complexity. To quantify overfitting, we use training loss on random data as a metric, which is modeled as a Gaussian distribution. It serves as a proxy for noise to understand how overfitting thresholds change with different model sizes, including the number of layers and hidden dimensions. Specifically, we evaluate the performance of the models by their ability to minimize the Mean Squared Error (MSE) loss when learning random data. A zero loss, which indicates perfect reconstruction of the noise, is undesirable as it signifies complete memorization of irrelevant data patterns. Conversely, a higher loss is preferable; it suggests that the model does not memorize the noise, with an MSE loss around 1.0 being ideal in cases where the data follows a Gaussian distribution. This metric helps to ensure that the models are robust enough to capture underlying causal structures without being influenced by spurious correlations.

We find that there are differences in memorization capabilities among the models. Figure 5.1 shows that some models can perfectly reconstruct the entire time series, resulting in a loss approaching zero. This suggests that the models are capable of storing most of the noise present in a causal time series. In the

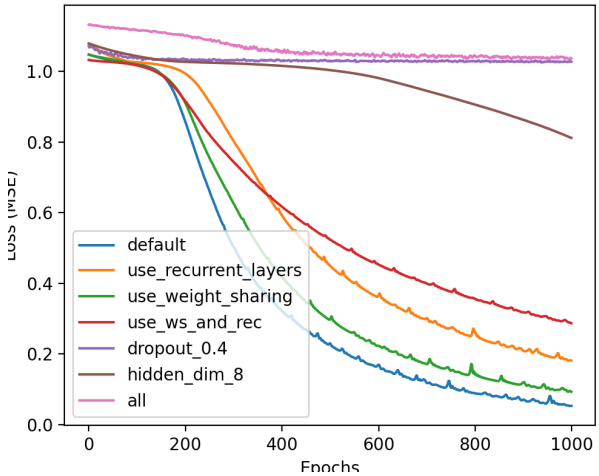

Figure 5.1: The memorization capability for various model complexities is presented. The models each consist of 8 layers with 32 hidden dimensions, a receptive field of 61, and are trained on 2400 data points.

case of NAVAR, this potentially explains why minor contributions from various variables are observed. However, as the variance is computed over the entire time series, these outliers are automatically filtered.

As expected, we observe that deep models tend to exhibit greater memorization capabilities. In contrast, models that incorporate weight-sharing and recurrent layers, such as the NAVAR-WS-Rec, tend to show reduced memorization capabilities compared to the default TCN, yet they still manage to memorize most of the data. Therefore, these models should still have the capacity for learning the causal relationships from data just as well as the default TCN. Furthermore, NAVAR-WS-Rec requires significantly shorter training times due to having fewer parameters.

We identified that a high dropout rate can hinder the model's learning ability, particularly in models with fewer hidden dimensions. Given that both the dropout rate and the number of hidden dimensions constrain the information flow within the model, it appears that the number of hidden dimensions is the most critical factor in overall model performance. We recommend optimizing for the number of hidden dimensions with a sufficiently low number of hidden dimensions in the hyperparameter search space, while maintaining the dropout rate at the default (0.2). This approach ensures that the models are robust enough to capture underlying causal structures without incorporating spurious correlations due to noise in the data. For NAVAR, hyperparameters can be selected such that the contribution is zero when training on random data.

Although weight-sharing and recurrent layers did not perform better as expected in finding causal relationships due to their lower complexity, our experiments demonstrate that these models maintain comparable learning capabilities to traditional TCNs. Additionally, these models maintain a long-range receptive field and benefit from reduced training times and fewer parameters. Furthermore, our findings indicate that a sufficiently low number of hidden dimensions can effectively prevent a model from learning noise.

Performance on synthetic data. We evaluated different model variants, each varying in complexity, on two synthetic datasets: a small dataset with parameters ($N = 5, K = 6, T = 500$) and a larger dataset characterized by long-range dependencies ($N = 8, K = 30, T = 1000$). The MLP-NAVAR (baseline) model performs adequately only when it comprises at least two neural layers; with fewer layers, the model becomes linear and fails to capture the relationships within the synthetic datasets. The TCN variants are constructed with multiple dilated layers (at least four) and have a kernel size of two, ensuring that the receptive field spans the maximum lag observed in the dataset. Each model variant maintains eight

	$N = 5, K = 6, T = 500$			$N = 8, K = 30, T = 1000$		
Method	AUROC	Number of parameters per variable	Training time per epoch (ms)	AUROC	Number of parameters per variable	Training time per epoch (ms)
MLP-NAVAR (baseline)	**0.99**	1470	10	**0.99**	10056	10
TCN-NAVAR	0.98	1830	11	0.78	5448	13
TCN-NAVAR-WS	0.94	678	10	0.70	1416	12
TCN-NAVAR-Rec	0.96	1110	11	0.70	1992	13
TCN-NAVAR-WS-Rec	0.92	534	10	0.69	984	12
TAMCaD	0.96	2390	13	0.83	6344	15
TAMCaD-WS	0.88	1238	12	0.67	2312	14
TAMCaD-Rec	0.89	1670	13	0.68	2888	15
TAMCaD-WS-Rec	0.91	1094	13	0.78	1880	14

Table 5.1: Comparison of results between models using weight-sharing (WS) and recurrent layers (Rec) with a receptive field of 31. The table presents the AUROC scores for different synthetic data configurations.

hidden dimensions.

Table 5.1 compares these models in their ability to reconstruct causal structures from synthetic data, measured by the AUROC. The MLP-NAVAR (baseline) model demonstrates the best performance, achieving high AUROC scores of 0.99 on both small and large datasets. This shows its robustness even with long-range dependencies. Models incorporating TCNs showed varied performance, with significantly lower AUROC scores on the larger dataset. Although TCN-NAVAR only differs in the number of layers and dilations from the baseline, it does not perform as well. However, as shown in Figure 5.3, extended training may benefit models that use multiple layers to achieve their optimal AUROC score, as indicated by the unplateaued learning curve. Models like NAVAR-WS and NAVAR-Rec, which significantly reduce the number of model parameters, underperform in scenarios involving long-range relationships. NAVAR-WS-Rec, though initially slow to learn relationships, often demonstrates improvement with extended training. TAMCaD tends to identify the simplest causal links at the initial stages of training, as indicated by the peaks in the AUROC learning curve, but it may learn spurious relationships and memorize noise over time, leading to lower performance in the end.

The uncertainty-aware variant of NAVAR achieved the lowest noise-adjusted regression loss, whereas the performance of the MLP baseline model worsened as training continued, indicating overfitting. Interestingly, this better performance in noise-adjusted regression loss did not translate into a better AUROC score for the uncertainty-aware variant, suggesting that overfitting does not necessarily imply worse model performance. A possible explanation for this, is that by learning the correct relationships, the model can free up capacity to learn noise.

Through these experiments, we demonstrate that model complexity indeed plays a role in causal discovery performance. When a broader receptive field is required and a deep neural model like a TCN is used, higher complexity is inevitable, potentially decreasing the learning capabilities and interpretability of the causal structure. Simpler models like NAVAR-WS, NAVAR-Rec, and in particular NAVAR-WS-Rec, may offer a more balanced approach. These models generally maintain comparable performance in reconstructing causal relationships and improve computational efficiency.

5.2 Effectiveness of Attention in Learning Interactive Relationships

In this section, we discuss the effectiveness of our method in predicting interactive relationships between variables. Specifically, we compare the performance of TAMCaD and NAVAR in reconstructing the causal structure. The experiments are conducted on synthetic datasets, each consisting of 16 variables over 2000 time steps with a maximum lag of 5, as illustrated in Figure 5.4. In this setup, we ensured the presence of sufficient interactive relationships among the variables. The parameters of the generation model f (Section 3.4.2) are initialized by training on a randomly generated dataset consisting of as few as 30 data points. A non-additive model is trained to fit these data points. Concurrently, the additive model

	$N = 5, K = 6, T = 500$				$N = 8, K = 30, T = 1000$			
Method	AUROC (test)	Soft-AUROC (train)	Soft-AUROC (test)	Training time per epoch (ms)	AUROC	Soft-AUROC (train)	Soft-AUROC (test)	Training time per epoch (ms)
TCN-NAVAR-UA	0.95	0.92	0.94	24	0.93	0.84	0.85	45
TAMCaD-UA	0.98	0.59	0.62	28	0.98	0.55	0.57	46

Table 5.2: Comparison of results between models using an uncertainty-aware mechanism with a receptive field of 32. The table presents the (Soft) AUROC scores for different synthetic data configurations.

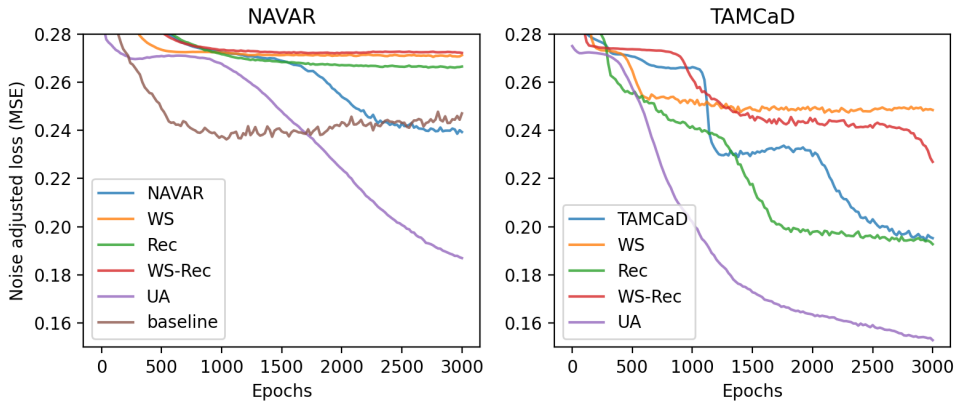

Figure 5.2: Noise-adjusted regression loss for causal data.

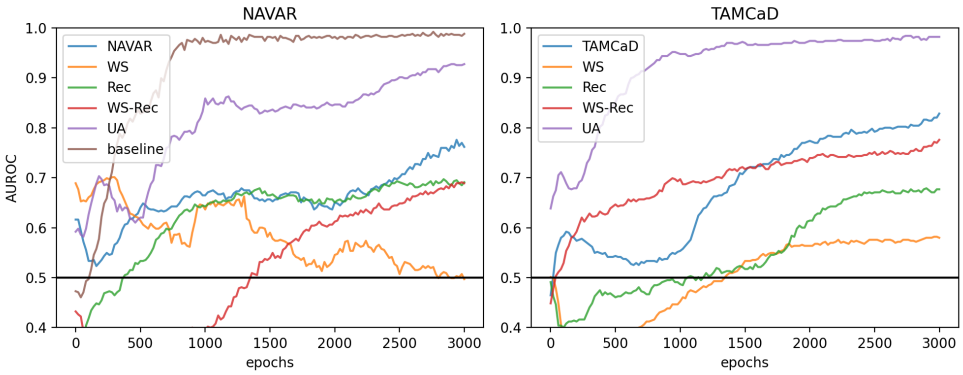

Figure 5.3: AUROC scores during training for models with various complexities trained on synthetic data.

is trained on data generated by this non-additive model. While the additive model can approximate the data by learning two separate one-dimensional functions, it does so suboptimally for strong interactive relationships, as demonstrated in Figure 5.5.

Figure 5.6 presents the noise-adjusted loss and AUROC scores over time, for 10 synthetic datasets. Notably, TAMCaD outperforms the additive model (NAVAR) in predicting the causal structure. The constructed causal graph is visually represented in Figure 5.7. Variables with two incoming connections are correctly predicted by the attention-based model (TAMCaD), whereas NAVAR struggles to learn both incoming connections and frequently predicts only one. However, it is worth noting that NAVAR still identifies many causal relationships, capturing at least one of the two interaction variables. This highlights the effectiveness of additive models in discovering causal relationships. We theorize that the additive aspect of such models acts as a strong form of regularization, helping the additive model with susceptibility to extremely high or low values, thereby contributing to its robust performance (see Figure 5.5).

Interaction variables in temporal contexts. We demonstrated that an additive model like NAVAR is effective in predicting interaction variables in certain situations. We argue that this effectiveness is primarily due to its ability to process recent historical data within a time series, as it enables the model to capture the current value of another interaction variable through traces in the data, resulting in more precise predictions. However, while this aspect of additive models is advantageous for predicting interaction variables, it can present challenges when determining the optimal number of time lags for

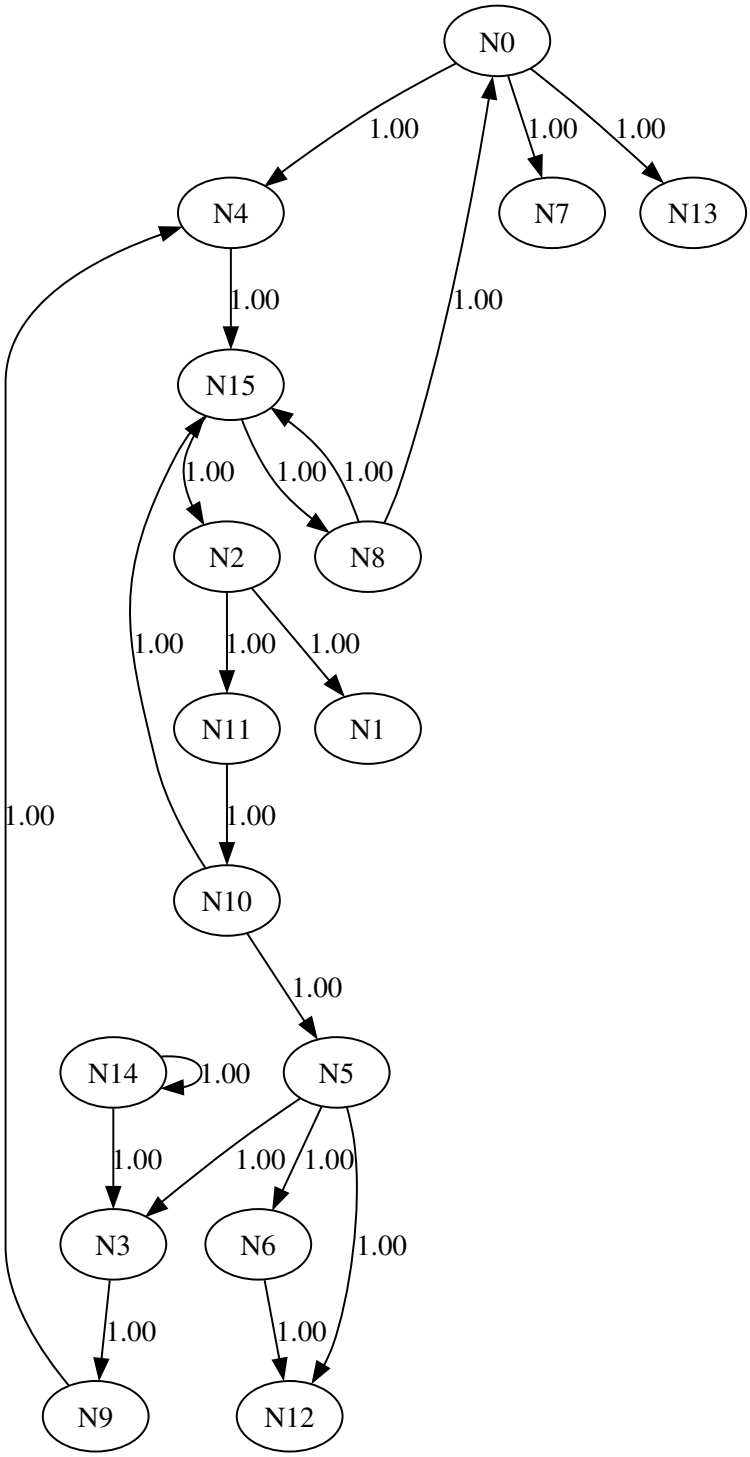

Figure 5.4: A synthetically generated causal graph consisting of 16 variables (N)

prediction accuracy. Furthermore, it might cause models to assign significance to non-causal variables, as these variables contain crucial information required for accurate additive predictions in scenarios involving interaction relationships.

We used the CauseMe benchmark to evaluate the effectiveness of our attention mechanism in capturing interaction variables. However, our approach did not yield the increase in performance we had hoped for. It is important to note that the datasets in this benchmark are predominantly additive in nature.

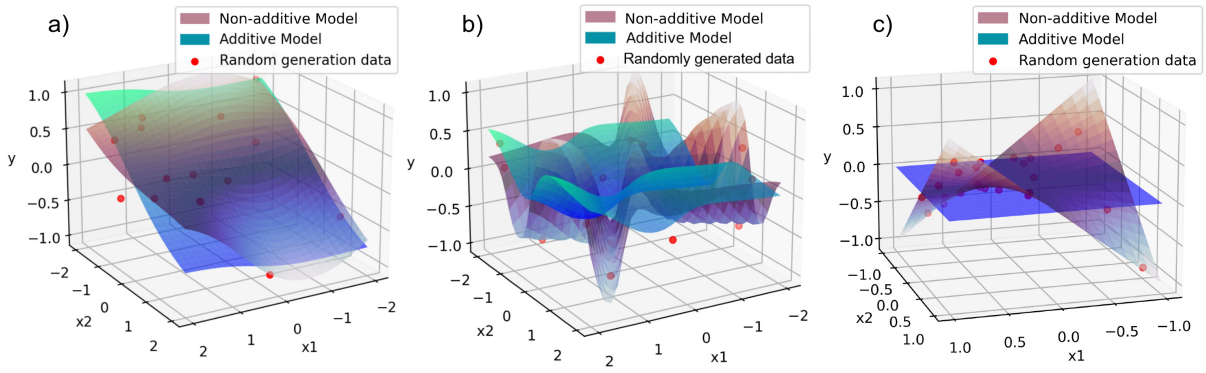

Figure 5.5: Illustration of various interaction relationships that can be produced by fitting a non-additive model to random data points, with subsequent learning by additive models on data generated by the non-additive model. a) Weak interaction: The additive model approximates the relationship effectively. b) Interaction: The additive model learns two separate one-dimensional functions that provide a suboptimal approximation of the relationship. c) Strong interaction: The additive model is unable to learn the relationship (Eq. 2.23) and minimizes its error by outputting zero contributions.

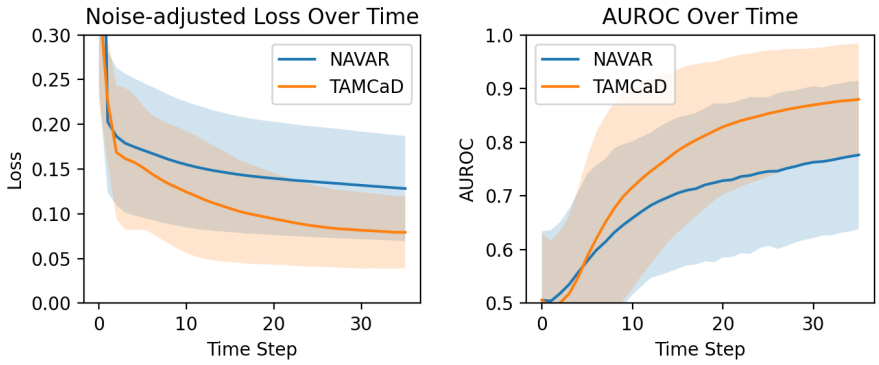

Figure 5.6: Noise-adjusted loss and AUROC over time for 10 synthetic datasets.

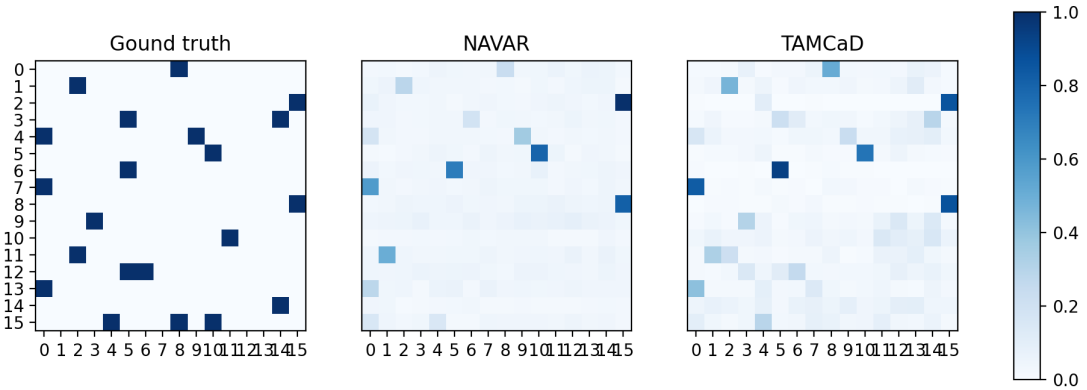

Figure 5.7: The learned causal structure for 16 variables with interactive relationships. Here, the x-axis represents the 'from' variables, and the y-axis represents the 'to' variables.

For example, Nonlinear-VAR employs an additive approach (the VAR component) for data generation. Furthermore, we hypothesize that in real-world benchmarks related to climate and weather, non-additive relationships may exist, but they are typically not significant enough to cause interactions that completely flip outcomes. This is because these benchmarks are based on physical systems where forces tend to influence each other in an additive manner. These weak interactive relationships can be approximated well by an additive model, as demonstrated with NAVAR. With its strong regularization, this makes it better suited for causal discovery in these datasets. Within the CauseMe benchmark, there are other datasets, such as the bivariate structural causal model characteristics data (bSCMC [61]), which describe various functional dependencies as multiplicative or complex. However, these descriptions refer to individual variables and not interactions between them. We recommend that future research focusing on temporal interaction dependencies considers using one of our synthetic datasets, which incorporate strong interactive relationships. Alternatively, researchers can generate data from observational sources that contain specific instances of interactive phenomena.

5.3 TAMCaD vs. NAVAR in Identifying Contemporaneous Relationships

This section presents an experiment comparing TAMCaD and NAVAR in their ability to identify contemporaneous relationships. To demonstrate how NAVAR and TAMCaD capture contemporaneous relationships, we provide visualizations of attention and contributions over time. We generated data for 5 variables with a maximum lag of 3 over 2000 timesteps, where causal links are severed and restored at specified intervals. The concrete changes in the data structure are visualized in Figure 5.8.

Figure 5.9 illustrates that both the NAVAR and TAMCaD models are capable of learning contemporaneous relationships to some extent. Moreover, TAMCaD appears to benefit from using ensembles in its uncertainty-aware variant. It is clear that contemporaneous relationships are being learned to some degree, but often the causal relationship remains unchanged for both NAVAR and TAMCaD. The ability to capture these relationships can be heavily influenced by hyperparameters such as hidden dimensions, which may result in less visible relationships in this plot.

Contemporaneous relationships can influence the causal matrix predicted by NAVAR if this prediction is based on the entire time series. For example, when NAVAR is provided a dataset featuring contemporaneous relationships, it might predict -1 for the first half of the dataset, followed by a prediction of 1

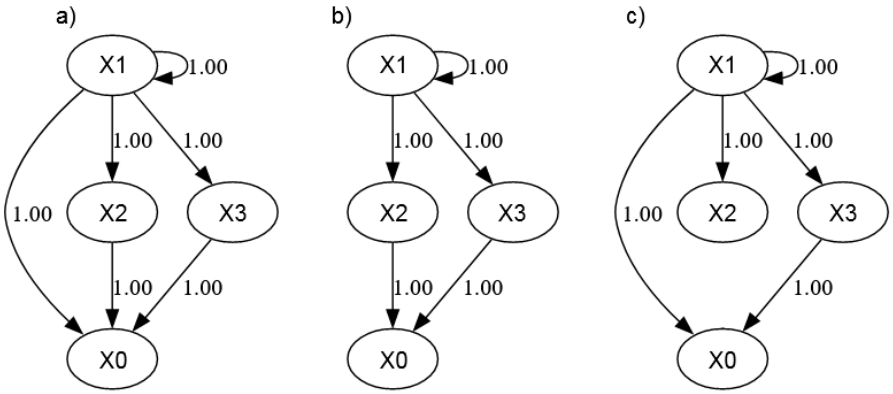

Figure 5.8: Visualization of changes in the data structure over time, showing severed and restored causal links at set intervals.

for the rest of the dataset. While the NAVAR framework can capture static values in the bias term, this approach only works for static time series. Consequently, when using the standard deviation to construct the causal matrix from the time series data, it results in a standard deviation of 1, which may be incorrect.

5.4 Analysis of Dot-Product Attention in TAMCaD and Embedding Characteristics

When evaluating the performance of TAMCaD-T, which uses dot-product attention, we did not achieve the improved results on datasets with a large number of variables that we were expecting. Notably, the TAMCaD-T model performed worse compared to other models in causal discovery when evaluating the AUROC score. Furthermore, the TAMCaD-T model exhibited significantly slower computational performance compared to other models, taking 25 ms per epoch as opposed to the 10 ms per epoch required by the other models.

When we visualized the embeddings using t-SNE, the results were as expected (see Figure 5.10). The query embeddings of the 'from' variables tended to cluster with the key embeddings of the 'to' variables. This clustering behavior was in line with our expectations but did not reveal any particularly new or

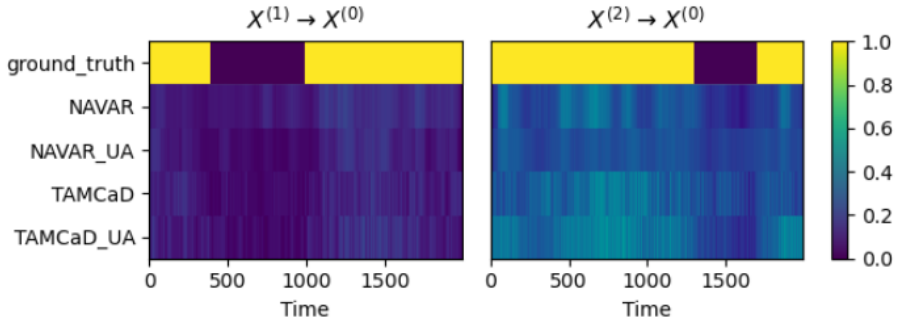

Figure 5.9: Contemporaneous relationships over time for NAVAR, TAMCaD, and their Uncertainty-Aware variants.

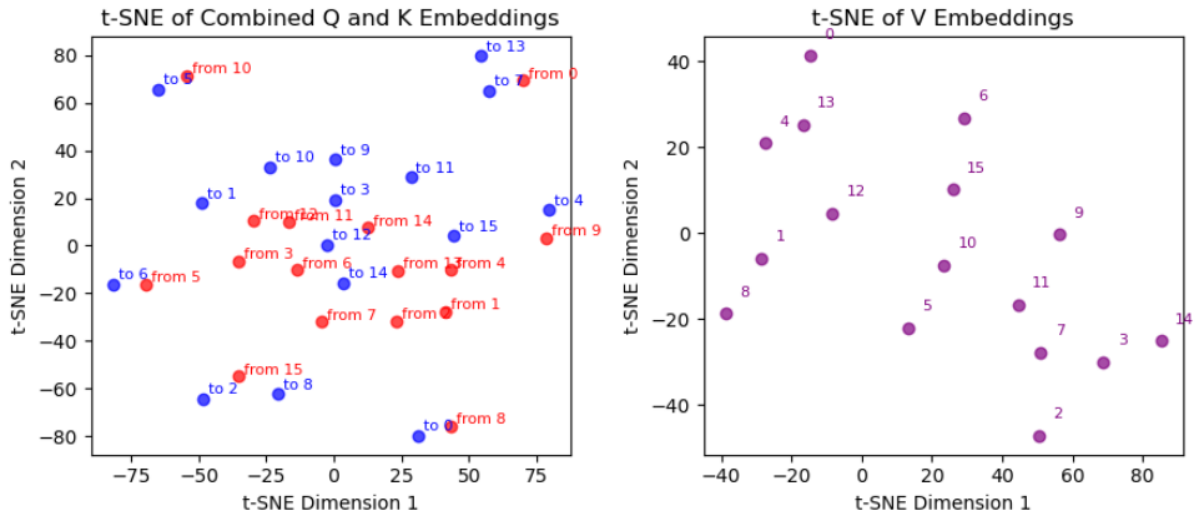

Figure 5.10: t-SNE visualization of TAMCaD-T embeddings, showing expected clustering of 'from' variable queries with 'to' variable keys.

insightful patterns when the value embeddings were clustered.

Consequently, while dot-product attention may have potential in the broader field of causal representation learning, its application in our current framework did not lead to improvements in performance. These findings emphasize the delicate balance between using a simple model that performs well versus a complex model that predicts time series accurately, but makes it harder to interpret the causal structure.

5.5 Impact of Uncertainty-Aware Mechanism

In this section, we assess how the incorporation of uncertainty-aware mechanisms in both TAMCaD and NAVAR influences their capacity to capture causal relationships. We evaluate both the performance and the interpretability of the generated causal matrices of these models with their uncertainty scores.

The introduction of mask regularization in TAMCaD, designed to encourage variance across ensembles, has proven to be highly effective. This concept arose from the observation that TAMCaD alone did not consistently yield results, while ensembles demonstrated a higher degree of consistency. By introducing a regularization term that stochastically penalizes models within the ensembles for learning specific relationships, we achieved greater variability between models. This variability, coupled with variational inference within the models, has led to an approach better equipped to approximate uncertainty.

Although a detailed discussion of the CauseMe benchmark results will be discussed in the following section, we present uncertainty scores obtained from the River Runoff benchmark to illustrate the produced uncertainties, as shown in Figure 5.11. The model exhibits higher uncertainty in predicting links associated with self-causation, while it indicates lower uncertainty in other areas.

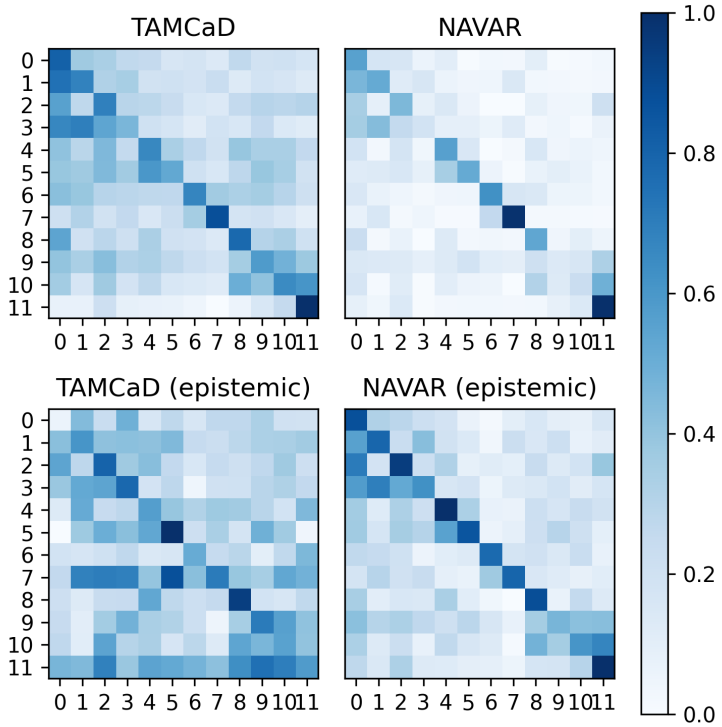

Figure 5.11: Predicted causal structures and their epistemic uncertainty scores for the River Runoff dataset from the CauseMe benchmark.

We adopted the method described by [2] and applied a softmax function to Gaussian-distributed logits to learn predictive uncertainty. This approach is similar to the findings of [62], who demonstrated that when each component z_i of a vector is sampled from a Gamma distribution Gamma$(\alpha_i, 1)$, the resulting normalized vector $\mathbf{z}$ aligns with a Dirichlet distribution characterized by the parameters $\alpha_1, \ldots, \alpha_D$. Given the similarity between the Gamma and log-normal distributions, primarily in scale, a Dirichlet distribution can potentially be approximated using a log-normal distribution for the logits, followed by normalization. Moreover, the softmax function, with its inherent exponential operation and vector normalization, allows for the approximation of a Dirichlet distribution by using a normal distribution for the logits, followed by a softmax transformation. However, it is important to note that approximating the Dirichlet distribution with the exponential scaling of logits in this approach may lead to numerical instability during gradient-based training processes.

5.6 Real-World Data Applicability

In this section, we assess the practical applicability of our models on real-world data. We assess whether the improvements made in learning contemporaneous and coupled relationships are effective in real-world scenarios. Additionally, we discuss the implications of the observed results.

	CauseMe						
	Nonlinear VAR				Climate	Weather	River
	$N = 3$ $T = 300$	$N = 5$ $T = 300$	$N = 10$ $T = 300$	$N = 20$ $T = 300$	$N = 40$ $T = 250$	$N = 10$ $T = 2000$	$N = 12$ $T = 4600$
TAMCaD	0.54	0.58	0.57	0.59	0.68	0.61	0.86
TAMCaD-UA	0.68	0.69	0.75	-	-	0.76	0.91
TCN-NAVAR	0.86	0.82	0.79	0.80	0.68	0.76	0.85
MLP-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 5.3: Comparison of results between models using different configurations. The table presents the AUROC scores for different datasets from the CauseMe benchmark. The original NAVAR variant consists of an MLP and is optimized for all hyper parameters, whereas our models are optimized on a subset of hyperparameters.

CauseMe. Table 5.3 presents results from a subset of the CauseMe benchmark. The results show that TAMCaD and the modifications applied to NAVAR significantly underperform compared to the baseline models. Several factors could contribute to this result. First, the training duration and the scope of hyperparameter tuning might not have been sufficient to fully optimize these models. Moreover, a primary issue is the focus on minimizing regression test loss on the test set, which may not always correspond to an effective causal model. Additionally, the regularization parameter appears inadequately constrained; the optimal model favored a λ value as low as 1e-3, whereas a higher value would likely be more appropriate. Notably, TAMCaD consistently records lower loss on the 30% test set compared to NAVAR during training, yet it achieves a lower AUROC. This suggests that, as expected, TAMCaD's attention mechanism operates differently from NAVAR's contributions, but may not be as interpretable as expected. This highlights the importance of considering model complexity and causal interpretation and underscores the necessity for further investigation in this field.

DREAM3. Table 5.4 illustrates the performance on the DREAM3 gene expression dataset. Assessing TAMCaD on this dataset provides more insights into the effectiveness of our method when dealing

	DREAM3				
	E.Coli 1	E.Coli 2	Yeast 1	Yeast 2	Yeast 3
TAMCaD	0.57	0.59	0.53	0.52	0.53
MLP-NAVAR	0.696	0.649	0.681	**0.601**	0.594
LSTM-NAVAR	**0.715**	**0.682**	**0.695**	0.599	**0.597**
cMLP	0.644	0.568	0.585	0.506	0.528
cLSTM	0.629	0.609	0.579	0.519	0.555
TCDF	0.614	0.647	0.581	0.556	0.557
SRU	0.657	0.666	0.617	0.575	0.550
eSRU	0.660	0.629	0.627	0.557	0.550
SELVAR	0.551	0.536	0.556	0.516	0.534
SLARAC	0.580	0.509	0.526	0.503	0.494

Table 5.4: AUROC scores for the DREAM3 gene expression dataset.

with systems that include a large number of (interaction) variables. TAMCaD achieves an AUROC score of 0.55 on average, demonstrating moderate results and suggesting potential applicability in complex biological data scenarios. Remarkably, this performance was obtained without optimization of hyperparameters, potentially achieving better results when optimized. Differences between TAMCaD and NAVAR are particularly evident, not only in the distinct causal matrices they generate, as highlighted in Figure 5.12, but also in their learning dynamics. TAMCaD identifies causal links very early in training, with its AUROC score tending to decrease as the model integrates more information through all attentions. In contrast, NAVAR shows a more consistent improvement in performance over time, benefiting from prolonged training and effective regularization.

Significant variability in results was observed across repeated experiments using identical hyperparameters, indicating a sensitivity to initial model parameter settings. To address the rapid convergence to local optima in TAMCaD, a lower learning rate is preferred. Moreover, we were unable to replicate the original NAVAR scores, potentially due to methodological differences. The highest scores reported in the literature were achieved after hyperparameter optimization, whereas our results are based on the average scores over multiple runs.

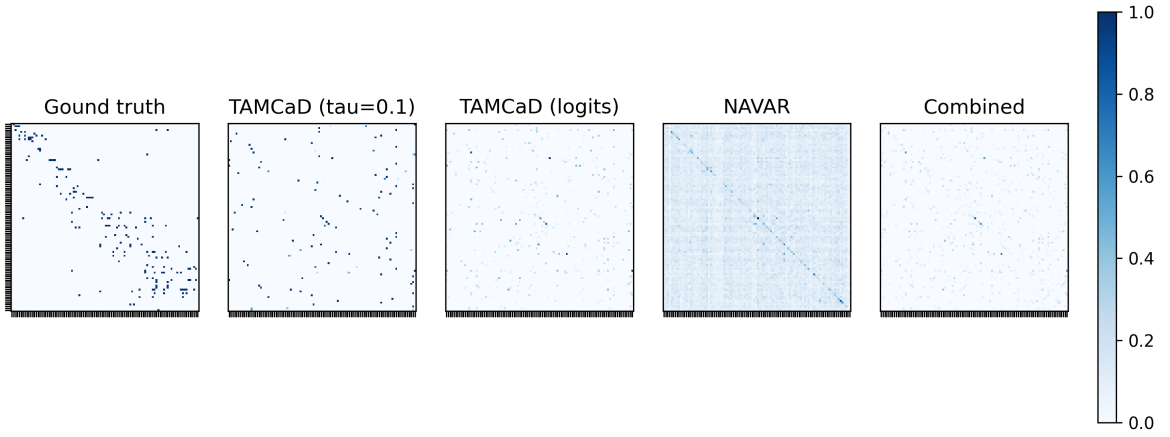

Figure 5.12: Predicted causal structures for DREAM3-Yeast-1.

Benchmark data vs. model complexity. As discussed in Section 5.1, it is beneficial to align model complexity with the complexity of benchmark data for optimal performance. For example, the CauseMe and DREAM3 datasets provide only about 300 and 21 time-steps per variable, respectively. However, as shown by [1], given this CauseMe data of a small system of variables with generally simple functional relationships, it is still possible to accurately learn and identify causal relationships. Therefore, the performance of our models is not attributable to insufficient data, which is often the case for training large, complex deep learning models such as TCNs. Consequently, the complexity of the model hinders its ability to generalize causal connections, limiting their applicability in real-world scenarios. Our approach may be too ambitious given the current data constraints. It highlights the need for either more regularized model designs or larger, more intricate real-world datasets to prove our model's usability. For example, a high-frequency sampled time series with well-established causes in the distant past would benefit from a model capable of capturing long-range temporal dependencies.

6 Future Work

Interpretability of Predicted Causal Structures. Future research could focus on the interpretability of causal structures generated by various methods. For example, NAVAR produces a causal structure based on quantitative contributions reflecting the strength of causal links. In contrast, TAMCaD uses its attention scores, reflecting intricate dynamics between variables. However, it is important to recognize that the causal links reconstructed from observational data are not definitively causal. At best, they can provide a strong basis to presume a causal relationship, which can be further explored through real-world experiments involving interventions. Exploring methods to align and compare predicted structures across different methods may forward the field of reliable causal discovery. In this process, methods might even be combined to produce more robust results. For the TAMCaD framework, we also propose developing a method to effectively regulate sparsity within attention matrices. In our experiments, we attempted to implement regularization to enforce higher entropy across attentions. However, this approach posed challenges during the learning process, as the learned matrix was unable to deviate from the optimal state found during its early training stage. While a denser matrix can still achieve optimal AUROC scores due to the correct identification of the most significant attention scores as causal links, such matrices might prove less beneficial in practical settings. This is because they could also include a high number of falsely identified causal connections. Therefore, addressing this issue involves not only optimizing the causal matrix for sparsity, but also ensuring the accurate representation of uncertainty scores.

Studying Different Structures of Causal Relationships. In this study, we applied our methods on synthetic data and simple causal model with variables that only have causal links in a triangle-like structure. In future work, it would be interesting to study how these methods performs on different structures, such as a long chain of non-cyclic variables. This would provide additional insight into the capabilities and limitations of these methods.

Alternative Approaches to Capture interactive relationships. An alternative approach to accurately predict non-additive relationships involves using a method that allows for multivariate inputs. One approach could be to use a modified version of NAVAR that randomly distributes variables across two networks instead of N networks. This will significantly improve computation time to train one model, particularly when there are many variables. The method would calculate the causal links between two subsets of variables with respect to all other variables. By training multiple networks using different variable selections, the resulting causal matrices can be combined into one final causal matrix. The number of subsets increases exponentially with the number of variables, making it impractical to test all possible combinations. However, measures can be implemented to suggest candidate subsets during training. The hypothesis becomes that the average causal links between individual variables will converge after a few training steps. This approach could lead to more accurate intermediate causal links as more information is provided to the model, making the final causal matrix more accurate as well.

Instantaneous Attention for Capturing Instantaneous Relationships. To deal with instantaneous relationships, data from $t+1$ is often included in the input as well [3]. This can efficiently be incorporated into our attention-based approach by allowing the variables (query in the dot-product attention approach) to interact with the embeddings at $t+1$. It is important to ensure that the attentions at time step $t+1$ are masked for the variable being predicted to prevent variables from attending to their future states, which would violate the principles of the auto-regressive model. To avoid overhead in our implementation, we omitted this approach from our implementation and recommend exploring it in future work.

Optimizing Sparse Attention for Interpretable Causal Discovery. Our observations revealed that the resulting causal matrix from our attention-based approach tends to be overly dense rather than sparse. To address this, we recommend that future research explore methods for inducing sparsity in attention mechanisms. This could potentially improve the model's ability to selectively focus on more relevant features during training. Without specific constraints, as in our approach, models tend to initially learn varied attentions. However, with extended training, these attentions often converge, becoming uniformly distributed as the model tries to include as much information as possible. Possible approaches may include alterations to the softmax function, such as SparseMax [63], or regularization techniques minimizing entropy [64]. Developing techniques that encourage sparse attention could help maintain meaningful attentions throughout the training process, which potentially improves the interpretability and effectiveness of the causal discovery.

Beyond Softmax: Alternatives for scoring Attentions. Attention mechanisms typically use the softmax function to convert attention scores into a normalized probability distribution. However, identifying and quantifying causal relationships with a softmax function presents some limitations. First, the softmax function forces each variable to attend to other variables, even for variables that lack incoming connections. Additionally, softmax calculates a class prediction relative to others, complicating the comparison of attention scores across multiple instances. Second, a model can converge to a local optimum, limiting the learning of new causal connections as training progresses. As discussed in Section 3.1.5, attention scores can vary between experiments while yielding the same accuracy. This suggests that the scores may not accurately reflect the strength of connections. Therefore, implementing regularization over attention or contribution scores could be beneficial (e.g., NAVAR regularization), as it would ensure consistent results over different experiments. Several modifications and alternatives to softmax have been proposed to address these limitations or to achieve specific desired properties. SparseMax offers the advantage of producing sparse attention weights, allowing it to assign zero weights to certain inputs [63]. This is particularly useful in interpretability and scenarios requiring focused attention on a subset of inputs. It operates by projecting input logits onto a simplex, achieved by subtracting a uniform value from the logits. However, its computational inefficiency with large input dimensions due to the need to sort the logits poses a problem. Softmax-1 is designed to address the issue of the disproportionate attention to irrelevant characters in a text by language models, due to the softmax function forcing each embedding to attend to other embeddings. It ensures that the sum of attention weights is one or less by including a 1 in the denominator, allowing for zero attention weights. The Gumbel-Softmax is a distribution that can be smoothly annealed into a categorical distribution [65]. It introduces a positive Gumbel noise to the input logits before applying the softmax. While dropout blocks the gradients though certain logits, the Gumbel-Softmax approach acts as a drop-in and enables the flow of gradients through all logits. This approach might prove useful in preventing the model from becoming trapped in local minima during training. Then, 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. Lastly, excluding the use of an activation function could

be investigated, where the logits are directly used as scalars. Our experiments did not yield consistent results across different experiment settings. However, Gumbel-Softmax performed generally well across most experiments. This may be due to the stochastic nature, preventing the model to get stuck in a local optimum.

Attentions in a Temporal Causal Context. One potential limitation of our approach arises from the consistent causal context across all time steps. Having a fixed causal structure may limit the adaptability of attention mechanisms to specific inputs. In an ideal scenario, attentions would be input-dependent, responding to changes in the data at each time step. When dealing with a static structure, we propose the use of a singular, parameterized attention matrix to enhance computational efficiency. Future research may further research the implications of using attentions in a context where attentions may be imbalanced across time, and could explore how the use of attentions can be optimized in the context of causal discovery. Furthermore, our research has exclusively focused on applying attentions over variables. Extending this to include attention along the temporal axis could be beneficial for accurately identifying the number of lags for each variable. However, our current approach in this poses a challenge, as the embedding of a variable integrates its historical data, rendering the application of attention over these historical embeddings impractical. We encourage further exploration of methods focusing on identifying the correct number of lags.

Refining Soft-AUROC for Consistent Uncertainty Evaluation In this work, we introduced the Soft-AUROC scoring method, which evaluates models based on both predictions and their uncertainty scores. However, we encountered a significant problem: different models may produce uncertainty scores with varying magnitudes of variance, which can invalidate comparisons between these scores. To overcome this issue, we propose to normalize the uncertainty scores. Then, by scaling these scores to optimize the Soft-AUROC metric, we can ensure a comparable assessment across different models.

7 Conclusion

In this study, we identified several challenges inherent in temporal causal discovery methods using deep learning, which have not been fully addressed by previous research. These challenges stem from the characteristics of the causal data, the methodologies employed, and the interpretation of the causal matrices. Here, we synthesize key insights from various experiments conducted to assess the performance and applicability of our proposed models for both synthetic and real-world data scenarios.

First, we experimented with the use of a TCN to expand the receptive field for capturing long-range, complex relationships. The experiments demonstrated that model complexity can influence both the memorization capabilities and the ability to learn causal relationships. However, higher complexity models, while overfitting to noise, are still able to generalize causal relationships. Therefore, despite theoretical advantages, a simpler, low-complexity approach, such as the MLP-NAVAR, often outperforms more complex models incorporating a TCN. This observation also underscores the need for better regularization in complex models and suggests that the causal relationships in our datasets might be sufficiently simple to be captured with a minimal number of neural network layers. We observed that the number of hidden dimensions is the largest factor here. Although a TCN could be advantageous in scenarios with a very large number of lags, in our moderate lag settings, it did not exhibit improved performance and required longer training times due to issues like vanishing gradients.

Furthermore, we introduced an attention-based approach alongside NAVAR's traditional contribution-based method, named TAMCaD, aiming to improve the model's ability to capture potential interactions across variables in the causal system. This shifts the causal discovery process from a post-hoc interpretation of quantitative contributions in an additive model to a dynamic learning approach during training, using attention scores. Our findings further suggest that additive models are adept at identifying the most evident relationships, possibly due to the inherent regularization from their additive nature, which currently makes them more robust than our proposed attention-based method. Nonetheless, our findings also suggest that the attention-based approach holds promise for improving temporal causal discovery. When disregarding the constructed causal matrix, this method yields superior results in training loss and noise-adjusted regression loss compared to the additive model, despite having an equivalent number of parameters. This improvement suggests that the model benefits from integrating data across multiple variables before making predictions, which could enhance its ability to predict interaction relationships and account for history-dependent influences. However, the alignment between the interpretability of attention scores and causal discovery performance requires further investigation.

Additionally, we integrated various uncertainty quantification methods within NAVAR's contribution mechanism and TAMCaD's attention mechanisms. These methods aim to measure the uncertainty associated with each inferred causal link. Our analysis indicates that ensemble models, due to their variability across produced causal matrices, outperform single models. Moreover, it is indeed possible to estimate predictive uncertainty for these causal links. Nonetheless, further research is needed to elucidate the factors that contribute to a model's uncertainty regarding specific causal and non-causal links.

When tested against real-world datasets like CauseMe and DREAM3, the models demonstrated varying degrees of success. Both TAMCaD and NAVAR incorporating a TCN underperformed compared to the baseline model. While TAMCaD demonstrated a lower loss on these benchmarks, its AUROC was

disappointingly low, indicating problems with the interpretability of the attention scores and, with that, its causal modeling capability.

Finally, we propose potential advancements and future research directions aimed at improving the reliability, interpretability, and overall effectiveness of temporal causal discovery methods that employ deep learning. We suggest future work explore the interplay between causal attentions and causal contributions, aiming to combine their strengths for more robust causal discovery. We also hope that this work will serve as a foundation for subsequent studies aiming to advance these methods.

Bibliography

[1] 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. iii, 3, 6, 8, 16, 20, 22, 27, 29, 56

[2] Matias Valdenegro-Toro and Daniel Saromo Mori. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1508–1516. IEEE, 2022. iii, 13, 14, 31, 33, 34, 54

[3] Meike Nauta, Doina Bucur, and Christin Seifert. Causal discovery with attention-based convolutional neural networks. *Machine Learning and Knowledge Extraction*, 1(1):19, 2019. 4, 17, 21, 22, 27, 30, 58

[4] 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. 4, 6, 16, 18, 27, 29

[5] Judea Pearl. *Causality*. Cambridge university press, 2009. 5, 7, 21, 22

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

[7] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. 5

[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, 7, 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, 6, 17

[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, 6

[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. 6, 16

[15] 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. 6

[16] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020. 6

[17] 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

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

[19] 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

[20] 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

[21] 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

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

[23] 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. 8

[24] 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. 8

[25] 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

[26] 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

[27] Pradeep Hewage, Ardhendu Behera, Marcello Trovati, Ella Pereira, Morteza Ghahremani, Francesco Palmieri, and Yonghuai Liu. Temporal convolutional neural (tcn) network for an effective weather forecasting using time-series data from the local weather station. *Soft Computing*, 24:16453–16482, 2020. 8

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

[29] 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

[30] 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

[31] 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. 9

[32] Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer, 2003. 12

[33] Matthias Seeger. Gaussian processes for machine learning. *International journal of neural systems*, 14(02):69–106, 2004. 12

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 12

[35] 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. 12

[36] Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*, 2022. 13, 32

[37] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018. 13

[38] 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. 13

[39] Alexander Amini, Wilko Schwarting, Ava Soleimany, and Daniela Rus. Deep evidential regression. *Advances in Neural Information Processing Systems*, 33:14927–14937, 2020. 14

[40] 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. 14

[41] Jay Heo, Hae Beom Lee, Saehoon Kim, Juho Lee, Kwang Joon Kim, Eunho Yang, and Sung Ju Hwang. Uncertainty-aware attention for reliable interpretation and prediction. *Advances in neural information processing systems*, 31, 2018. 14, 33

[42] Clive WJ Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society*, pages 424–438, 1969. 15

[43] Trevor J Hastie. Generalized additive models. In *Statistical models in S*, pages 249–307. Routledge, 2017. 15

[44] Michele Chambers and Thomas W Dinsmore. *Advanced analytics methodologies: Driving business value with analytics*. Pearson Education, 2014. 16

[45] Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019. 18

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

[47] 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, 37

[48] 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. 20

[49] Muhammad Saqib Sohail, Raymond HY Louie, Zhenchen Hong, John P Barton, and Matthew R McKay. Inferring epistasis from genetic time-series data. *Molecular biology and evolution*, 39(10):msac199, 2022. 20

[50] Luan Lin, Quan Chen, Jeanne P Hirsch, Seungyeul Yoo, Kayee Yeung, Roger E Bumgarner, Zhidong Tu, Eric E Schadt, and Jun Zhu. Temporal genetic association and temporal genetic causality methods for dissecting complex networks. *Nature Communications*, 9(1):3980, 2018. 20

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

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

[53] 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. 22

[54] Brady Neal. Introduction to causal inference from a machine learning perspective. *Course Lecture Notes (draft)*, 2020. 22

[55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 27

[56] 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. 27

[57] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 33

[58] Robert J Prill, Daniel Marbach, Julio Saez-Rodriguez, Peter K Sorger, Leonidas G Alexopoulos, Xiaowei Xue, Neil D Clarke, Gregoire Altan-Bonnet, and Gustavo Stolovitzky. Towards a rigorous assessment of systems biology models: the dream3 challenges. *PloS one*, 5(2):e9202, 2010. 37

[59] J Muñoz-Marí, G Mateo, J Runge, and G Camps-Valls. Causeme: an online system for benchmarking causal discovery methods, 2020. 38

[60] 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. 40

[61] Christoph Käding and Jakob Runge. Distinguishing cause and effect in bivariate structural causal models: A systematic investigation. *Journal of Machine Learning Research*, 24(278):1–144, 2023. 51

[62] Mikhail Figurnov, Shakir Mohamed, and Andriy Mnih. Implicit reparameterization gradients. *Advances in neural information processing systems*, 31, 2018. 54

[63] 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. 58

[64] Ori Press, Ravid Shwartz-Ziv, Yann LeCun, and Matthias Bethge. The entropy enigma: Success and failure of entropy minimization. *arXiv preprint arXiv:2405.05012*, 2024. 58

[65] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016. 58