CAUSAL TRIPLE ATTENTION TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Time series forecasting has historically been a key area of academic research and industrial applications. In multi-horizon and multi-series forecasting tasks, accurately capturing the local information in a sequence and effectively sharing global information across different sequences are very challenging, due to the complex dependencies over time in a long sequence and the heterogeneous nature across multiple time series. In this paper, from the perspective of causal inference, we give a theoretical analysis of these difficulties and establish a causal graph to identify the confounding relationship that generates harmful bias and misleads the time series model to capture the spurious correlations. We propose a causal triple attention time series forecasting model with three interpretable attention modules, which leverages the front-door adjustment to remove the confounding effect and help the model effectively utilize the local and global temporal information. We evaluate the performance of our model on four benchmark datasets and the results demonstrate the superiority over the state-of-the-art methods.

1 INTRODUCTION

Multi-horizon and multi-series time series forecasting has become a very intensive field of research. Compared to one-step-ahead predictions, multi-horizon forecasts provide the estimates for multiple future time points, enabling better decision making beforehand. Besides, multi-series forecasting from related time series not only provides richer information by utilizing inter-relationships across all time series but also alleviates the labor-intensive feature engineering and model design required for each time series. However, due to the native complex dependencies over time in a long sequence and the heterogeneous nature across multiple time series, the multi-horizon and multi-series time series forecasting has been always facing two major challenges: (1) how to leverage the local knowledge lying in a long sequence, and (2) how to effectively take advantage of the global knowledge extracted from multiple related time series.

Recent deep learning methods (Salinas et al., 2019; Rangapuram et al., 2018; Wen et al., 2017) based on recurrent and convolutional neural networks provide a data-driven manner to deal with time series forecasting tasks and achieve great accuracy in most application fields. Due to complex dependencies over time of recurrent networks and the limits of convolutional filters, these methods have difficulties in modeling long-term and complex relations in the time series data. Considering the dependencies of each time point in a sequence, attention-based methods (Fan et al., 2019; Li et al., 2019) are proposed by assigning different importance to the different time points. In these models, the local dependencies are effectively utilized for the prediction, but the global information of the relationship among different series is still unexplainable. Matrix factorization methods (Yu et al., 2016) and Bayesian methods that share information via hierarchical priors (Chapados, 2014) are used to learn multiple related time series by leveraging hierarchical structure (Hyndman et al., 2011). However, how to extract and share the right global information across different time series is still not fully exploited.

In this paper, we approach these two challenges from a new perspective, causal inference. Based on the Structural Causal Model (Pearl et al., 2016; Pearl & Mackenzie, 2018), the multi-horizon and multi-series forecasting tasks can be abstracted into a causal intervention problem with unobserved confounders. We refer to the confounder as common sense inferred from the time series data that can be seen as the summarized knowledge for a certain part of series, e.g., "a part of products sell extraordinarily well at a certain season", "The sales of new products continue to grow in a short-range due to new launches", and so on. However, these common senses usually are only applicable for part of the time points. The goal of such causal models is to remove the confounding

effect caused by unrelated common senses and focus on the mediator that is the predictive local and global information. Intuitively, such a philosophy of causal intervention can help us clarify how to extract and share the right local and global dependencies with the related time series. However, the common sense in time series tasks is usually hard to quantify. We cannot directly observe and stratify the confounder, and thus are unable to design causal intervention by deploying the simplest backdoor adjustment. Alternatively, we adopt front-door adjustment (Pearl, 1995) that does not require any knowledge on the confounder. Besides, the front-door adjustment can provide a more comprehensible way to understand the mediator, that is how the local and global information is utilized.

In conclusion, we design a **C**ausal **T**riple a**T**tention **T**ime series forecasting model (CTTT) based on a deep encoder-decoder recurrent architecture, which uses temporal and pattern attentions to accomplish front-door adjustment. We provide an intuitive understanding and causal theoretical proofs for leveraging these two attentions to complete the front-door adjustment. Then, the temporal and pattern attentions can theoretically shed light on how the local and global knowledge is effectively extracted from the data and how the right knowledge is accurately utilized to benefit the prediction of different series. Finally, we adopt transformer attention to associate the decoder with the encoder sequence to determine which parts of the encoder are more engaged for the decoder prediction and further improve forecast accuracy. CTTT utilizes these three attention modules to accomplish the front-door adjustment and help the model capture useful local and global knowledge without abusing unrelated information.

2 Related Work

Time Series Forecasting. While traditional statistical methods focus on parametric models, i.e., ARIMA models (Box & Jenkins, 1968), autoregressive (AR) (Box et al., 2015), exponential smoothing (Gardner Jr, 1985; Winters, 1960), and structural time series models (Harvey, 1990), current deep learning methods provide a data-driven manner to deal with time series forecasting tasks (Ahmed et al., 2010). With increasing data availability and computing power, deep learning methods have showed the promising results, such as sequence-to-sequence deep RNNs architecture (Zhang et al., 2019; Salinas et al., 2020), deep state space models (Rangapuram et al., 2018), temporal convolutional network (Bai et al., 2018), deep transformer models (Wu et al., 2020). In contrast to these work, our model integrates the causal inference and proposes a interpretable high-performance time series forecasting model. It provides insightful explanations about how to utilize the local and global knowledge implicit in long sequence and different multiple time series.

Causal Inference. Causal inference (Pearl, 2000; Rubin, 2005) has been an attractive research topic for a long time as it provides an effective way to uncover causal relationships in real-world problems. Nowadays, the combination of the incisive ideas in the causal inference and various deep learning model can help improve existing methodologies in a wide range of fields (Yao et al., 2020), such as treatment effect estimation with observational data (Li & Fu, 2017; Yao et al., 2018; Chu et al., 2020b), causality analysis of graph networked data (Chu et al., 2021), continual learning (Hu et al., 2021; Chu et al., 2020a), natural language processing task (Yang et al., 2021; Niu et al., 2021; Abbasnejad et al., 2020), few-shot learning (Yue et al., 2020; 2021), domain adaptation (Bengio et al., 2019). In this work, we aim to incorporate causal inference into time series forecasting task. In general, the backdoor adjustment is the most direct causal inference method to eliminate the confounding effect by splitting the detailed confounder into various strata. However, the confounder is unobservable in time series forecasting scenario. Therefore, we exploit the front-door adjustment (Pearl, 1995) to mitigate the dataset bias, which is a fundamental causal inference technique for deconfounding the unobserved confounder.

3 Background

Our **C**ausal **T**riple a**T**tention **T**ime series forecasting model (CTTT) is based on one deep encoder-decoder recurrent architecture with triple interpretable attentions for multi-horizon and multi-series forecasting, which removes the invisible confounding relationship existed in multi-series data. This confounding relationship can cause harmful bias that misleads the time series model to focus on the spurious correlations in data and thus reduce prediction accuracy. In this section, we will first present the problem statement and analyze the causality involved in the time series forecasting task.

3.1 PROBLEM STATEMENT

Our purpose is to predict the multiple future target values for multiple time series. Denoting the target value of time series i at time t by $y_{i,t}$, our goal is to model the conditional distribution

$$P(\mathbf{y}_{i,t_0:T}|\mathbf{y}_{i,1:t_0-1}, \mathbf{x}_{i,1:T}), \tag{1}$$

where $\mathbf{y}_{i,t_0:T} = \left\{y_{i,t_0}, y_{i,t_0+1}, \ldots, y_{i,T}\right\}$ denotes the target values of future time from time point t_0 for series i and $\mathbf{y}_{i,1:t_0-1} = \left\{y_{i,1}, \ldots, y_{i,t_0-2}, y_{i,t_0-1}\right\}$ denotes the target values of past time before time point t_0 for time series i. t_0 denotes the time point from which we assume $y_{i,t}$ to be unknown. Besides, $\mathbf{x}_{i,1:T} \in \mathbb{R}^m$ are covariates that contain *observed covariates* and *known covariates*. The observed covariates are only available in the past and are unknown beforehand. Known covariates can be predetermined and they are known for all time points. The covariates $\mathbf{x}_{i,1:T}$ can be series-dependent, time-dependent, or both. If some covariates do not depend on time, they are repeated along the time dimension. The information about absolute time and series are only available to the model through covariates by time parsing and series embedding. Besides, additional information about the series or time can be added into the covariate vectors, e.g., features about series item, variables predictive of outcome, and special time points (festivals or holidays). Due to the complex dependencies over long time and vanishing gradients problem of recurrent network, we adopt the rolling window procedure to split all of the series and we keep the total length T for each window, including conditioning window from 1 to $t_0 - 1$ and prediction window from t_0 to T.

Due to the rolling window procedure, we totally obtain n windows and mix them together. Our model opts for using a sequence-to-sequence setup, including one encoder network for the conditioning window and one decoder network for the prediction window. Information about the observations in the conditioning window is transferred to the prediction window by the encoder-decoder framework. We apply our model to each window. During the training stage, both conditioning and prediction windows have to lie in the past so that the $y_{i,t}$ are observed, but during the prediction stage $y_{i,t}$ is only available in the conditioning window. Note that the time index t is relative, i.e. $t = 1$ can correspond to a different actual time point for each i.

3.2 INTUITIVE UNDERSTANDING OF CAUSAL TRIPLE ATTENTION

The core of our CTTT model is the combination of three attention modules, i.e., temporal attention, pattern attention, and transformer attention. Prior to giving the theoretical support, we first provide the intuitive understanding of each attention module.

Temporal Attention. Similar to the self attention of each sentence in BERT (Devlin et al., 2018), to explore the dependencies of each time point and reveal the trend in each time series window, we apply the temporal attention to each series window relating different positions of a single window. The attention mechanism assigns different importance to the different time points of the input window and gives more attention to the more relevant time points.

Pattern Attention. Due to the heterogeneous nature across multiple time series, sharing information across all time series is difficult to accomplish in practice. Worse than that, it may bring extra bias to data, resulting in the reduction of prediction accuracy. Therefore, to effectively capture the shared information across all time series without abusing the extracted global information to the unrelated or inapplicable windows, we apply one pattern attention to all windows, so that the more informative windows are given larger weights for the sake of more pattern attention. Therefore, each window can only absorb valuable information for itself, avoiding being misled by irrelevant information.

Transformer Attention. Another challenge with recurrent neural networks is that learning long sequences can be difficult due to complex dependencies over time and vanishing gradients (Chang et al., 2017). The sequence-to-sequence model sequentially links two RNNs, i.e., an encoder and a decoder, through the last encoder cell state. This can be limiting as it forms a potential bottleneck between the encoder and decoder. Furthermore, earlier inputs have to pass through several layers to reach the decoder (Wu et al., 2020). The transformer attention is utilized to associate the decoder with the encoder sequence to determine which parts of the encoder are more engaged for the decoder prediction and thus further improve forecast accuracy.

3.3 CAUSALITY ANALYSIS

Based on the Structural Causal Model (Pearl et al., 2016; Pearl & Mackenzie, 2018), we provide the theoretical supports for the temporal attention and pattern attention modules. The predicted target values $\mathbf{y}_{i,t_0:T}$ in the prediction window are conditioned by the combination of known target

values $\mathbf{y}_{i,1:t_0-1}$ in condition window and covariates $\mathbf{x}_{i,1:T}$ in condition and prediction windows. For convenience, we use r_i to denote this combination of all inputs in i-th window. In fact, not all of the information (all time points, known target values, and covariates) are useful for the prediction of target values $\mathbf{y}_{i,t_0:T}$. Instead of directly relationship $R \rightarrow Y$, there exists one mediator M, which refers to the knowledge extracted from original input R and used for the prediction of target values Y, i.e., $R \rightarrow M \rightarrow Y$.

Besides, the heterogeneous nature across different time series brings bias into the dataset. The dataset bias is essentially caused by the confounder C that makes input R and target values Y correlated via C indirectly. In this case, we refer to the confounder C as common sense inferred from the data, e.g., "high-velocity items can exhibit qualitatively different behavior than low-velocity items", "a part of products sell extraordinarily well at a certain festival", "The sales of new products continue to grow in a short-range due to new launches", and so on. However, these common senses are not applicable for all series windows, so that this confounding relationship may cause harmful bias that misleads the time series model to focus on the spurious correlations in data and thus reduce prediction accuracy, e.g., if one window conforms to this extracted common sense, it will enjoy the great benefit; if not, the prediction accuracy of this window will be compromised by this spurious knowledge. In conclusion, we present this causal graph in Figure 1. $R \rightarrow M$ denotes the hidden knowledge extracted from the input; $C \rightarrow R$ denotes that real scenarios are generated by common sense; $M \rightarrow Y$ denotes the prediction based on the predictive knowledge inferred from input observations. In addition, this Y is also influenced by common sense C.

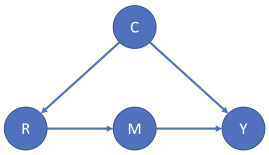

Figure 1: Causal relationship in the multi-series forecasting dataset.

Besides the legitimate causal path from input R via mediator M to Y, the "backdoor" path $R \leftarrow C \rightarrow M \rightarrow Y$ also contributes an effect to Y via confounder C, which will induce spurious correlation between R and Y. Therefore, if we directly train the model based on the correlation $P(Y|R)$ without intervention on confounders, no matter how large the amount of training data is, the model can never identify the true causal effect from R to Y (Pearl, 2000; Rubin, 2005). To remove the confounding relationship between R and Y, we should block $R \leftarrow C \rightarrow Y$ to get the causal effect between R and Y. The backdoor adjustment is the most direct method to eliminate the spurious correlation by approximating the "physical intervention" (Pearl & Mackenzie, 2018; Yang et al., 2021). To use the backdoor adjustment, we need to know the details of the confounder for splitting it into various strata. However, in time series tasks, we have no idea about what common sense constructs the confounders in the dataset, thus we are unable to deploy the backdoor adjustment. Alternatively, we adopt front-door adjustment that does not require any knowledge on the confounder. Besides, the front-door adjustment can provide a more comprehensible way to understand the mediator, that is how the local and global information is utilized.

Therefore, instead of the likelihood $P(Y|R)$, we use the causal intervention $P(Y|do(R))$ (Pearl, 1995) for the time series forecasting to get the true causal relationship between R and Y. The front-door adjustment calculates $P(Y|do(R))$ along with the front-door path $R \rightarrow M \rightarrow Y$, which is constructed from two partially causal effects $P(M|do(R))$ and $P(Y|do(M))$, i.e., $P(Y|do(R)) = \sum_m P(M = m|do(R))P(Y|do(M = m))$.

Similarly, to calculate $P(M = m|do(R))$, we should block the backdoor path $R \leftarrow C \rightarrow Y \leftarrow M$ between R and M. We can observe there is a collider ($C \rightarrow Y \leftarrow M$) in this backdoor path. The result of having a collider in the path is that the collider blocks the association between the variables that influence it (Pearl, 1995). Thus, the collider does not generate an unconditional association between the variables that determine it. Therefore, this path is naturally blocked and we have $P(M = m|do(R)) = P(M = m|R)$.

For $P(Y|do(M))$, we need to block the backdoor path $M \leftarrow R \leftarrow C \rightarrow Y$ between M and Y. Since we do not know the details about the confounder C, thus we have to block this path by intervening R, i.e., $P(Y|do(M = m)) = \sum_r P(Y|M = m, R = r)P(R = r)$. Finally, we can get:

$$P(Y|do(R)) = \sum\nolimits_m P(M = m|R) \sum\nolimits_r P(R = r)[P(Y|M = m, R = r)]. \tag{2}$$

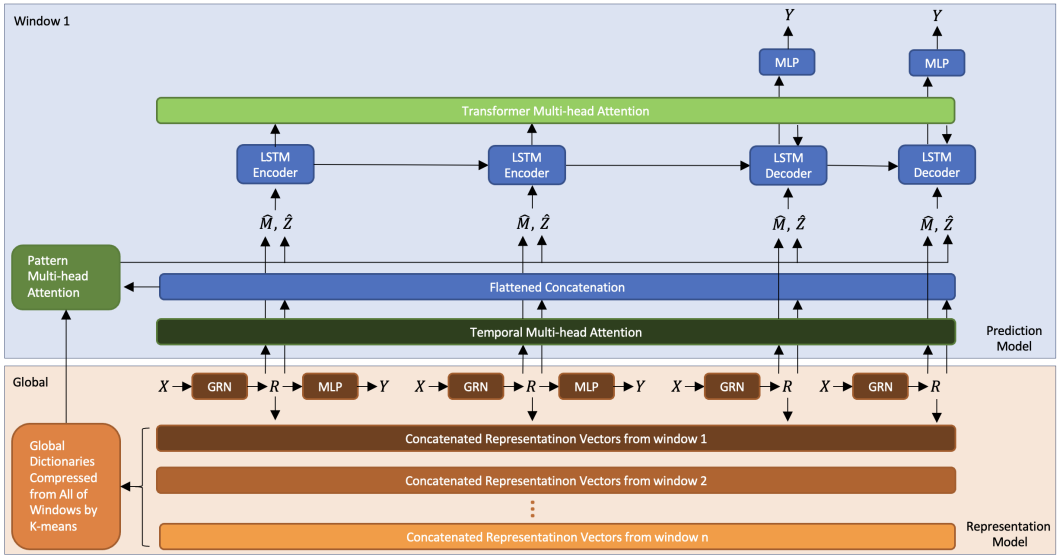

Figure 2: Our causal triple attention time series forecasting model (CTTT) contains two parts, i.e., the representation model and the prediction model. The representation model is used for learning the representation vector for each time point, which utilizes gated residual network to select relevant features and gated linear units to suppress unnecessary information. The prediction model is a encoder-decoder recurrent network with LSTM cells to predict the target values based on the representation vectors learned from the representation model. Three attention modules are deployed to help model capture the local and global information and mitigate the confounding effect.

4 Our Proposed Framework

4.1 Representation Model

As shown in Fig. 2, Our CTTT consists of two main components, i.e., representation model and prediction model. In the following, we present the details of each component.

Most real-world time series datasets contain features with less predictive content, thus variable selection is necessary to help with model performance. Inspired by the variable selection network in Lim et al. (2021), we propose a representation model, which is independent of the following prediction model and is trained before the training of the prediction model. The covariates X are input into the gated residual network (GRN) with gated linear units (GLUs) to generate the representation vectors R. To make the representation vectors rich with more predictive information, we put them in one supervised learning of target value y in the conditioning window. The purpose of this model is to obtain the representation vector for each time point, which will be used in the prediction model.

This representation model is necessary in two ways. First, it is trained by predicting observed target values $\mathbf{y}_{i,t_0:T}$, so that we can get the representation vectors $\mathbf{r}_{i,1:T}$ that include the information predictive of the target value. Second, it can provide insights into which variables are most significant for the target prediction and also remove any unnecessary noisy inputs which could negatively impact the performance (Lim et al., 2021).

We use entity embeddings for series item and categorical variables, and linear transformations for continuous variables, so that m covariates and one series item are transformed into $m+1$ d-dimensional vectors $\boldsymbol{e}_{j,t}^{(k)} \in \mathbb{R}^d$, which denotes the k-th transformed input at time t for window j. Let $\boldsymbol{\xi}_{j,t}$ be the concatenation of flattened transformed inputs $\boldsymbol{e}_{j,t}^{(1)}, \ldots, \boldsymbol{e}_{j,t}^{(m+1)}$. Variable selection weights $\boldsymbol{v}_{j,t}$ are generated by feeding $\boldsymbol{\xi}_{j,t}$ through a GRN, followed by a Softmax layer, i.e., $\boldsymbol{v}_{j,t} = \text{Softmax}\left(\text{GRN}_v(\boldsymbol{\xi}_{j,t})\right)$. Except for the GRN_v for the weights, the transformed input has its own GRN, i.e., $\tilde{\boldsymbol{e}}_{j,t}^{(k)} = \text{GRN}_{e^{(k)}}\left(\boldsymbol{e}_{j,t}^{(k)}\right)$, where $k = 1, \ldots, m+1$ and $\tilde{\boldsymbol{e}}_{j,t}^{(k)}$ is the filtered transformed input. GRN_v and $\text{GRN}_{e^{(k)}}$ are shared across all time points t and all windows j. The representa-

tion vector $\boldsymbol{r}_{j,t}$ are obtained by weighted sum of filtered transformed inputs $\tilde{\boldsymbol{e}}_{j,t}^{(k)}$ and their variable selection weights $\boldsymbol{v}_{j,t}$, i.e., $\boldsymbol{r}_{j,t} = \sum_{k=1}^{m+1} \boldsymbol{v}_{j,t}^{(k)} \tilde{\boldsymbol{e}}_{j,t}^{(k)}$, where $\boldsymbol{v}_{j,t}^{(k)}$ is the k-th element of vector $\boldsymbol{v}_{j,t}$.

In this representation model, we note that the *known covariates* are input into both the conditioning window and the prediction window, which are known all time points. If there are observed covariates in the dataset, which are only available in the past and are unknown beforehand, we only input them into the conditioning window. Because each covariate has its own GRN and the final representation $\boldsymbol{r}_{j,t}$ is calculated by weighted sum (the dimension is unchanged), we only need to rescale the variable selection weights $\boldsymbol{v}_{j,t}$ in the prediction window to adapt to the absence of observed covariates. Therefore, there's no limit to the type of covariates in our model.

4.2 PREDICTION MODEL

According to the causality analysis for the imbalanced time series data, we are introducing how to utilize the temporal and pattern attention modules to accomplish this front-door adjustment (Eq. (2)) in a deep framework. We can parameterize the predictive distribution $P(Y|M,R)$ as a network $g(\cdot)$, which is one encoder-decoder recurrent neural network with LSTM cell, i.e., $P(Y|M,R) = g(M,R)$. Besides, we need to sample R, i.e., $\sum_r P(R=r)$ and M, i.e., $\sum_m P(M=m|R)$, and feed them into the network to complete $P(Y|do(R))$ according to expression of Eq. (2). Because the network forward-pass consumption for all of these samples is prohibitively expensive, we apply Normalized Weighted Geometric Mean (NWGM) approximation (Xu et al., 2015; Srivastava et al., 2014) to absorb the outer sampling into the feature level and thus only need to forward the "absorbed input" in the network for once (Yue et al., 2020; Hu et al., 2021; Yang et al., 2021). By NWGM approximation, $\sum_m P(M=m|R)$ and $\sum_r P(R=r)$ in Eq. (2) can be absorbed into the network:

$$P(Y|do(R)) \approx g(\hat{\boldsymbol{M}}, \hat{\boldsymbol{R}}), \text{ where } \hat{\boldsymbol{M}} = \sum\nolimits_m P(M=m|h(R))m, \ \hat{\boldsymbol{R}} = \sum\nolimits_r P(R=r|f(R))r, \quad (3)$$

where $h(\cdot)$ and $f(\cdot)$ denote query embedding functions which can transform the representation vectors R into two query sets.

Following the idea about the attention in Yang et al. (2021), the estimations $\hat{\boldsymbol{R}}$ and $\hat{\boldsymbol{M}}$ in Eq. (3) are classic attention network calculates. The nature of attention mechanism can be summarized as the common Q-K-V notation. Attention mechanism scales values $\boldsymbol{V}$ based on relationships between keys $\boldsymbol{K}$ and queries $\boldsymbol{Q}$ i.e., Attention$(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = A(\boldsymbol{Q},\boldsymbol{K})\boldsymbol{V}$, where $A(\cdot)$ is a normalization function. A common choice is scaled dot-product attention (Vaswani et al., 2017), i.e., $(\boldsymbol{Q},\boldsymbol{K}) =$ Softmax$(\boldsymbol{Q}\boldsymbol{K}^T/\sqrt{d_{attn}})$.

To improve the learning capacity of the standard attention mechanism, multi-head attention is proposed in (Vaswani et al., 2017), employing different heads for different representation subspaces:

$$\text{MultiHeadAttention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = \tilde{\boldsymbol{H}}\,\boldsymbol{W}_H, \quad (4)$$

$$\tilde{\boldsymbol{H}} = \frac{1}{H}\sum\nolimits_{h=1}^{H} \text{Attention}(\boldsymbol{Q}\,\boldsymbol{W}_Q^{(h)}, \boldsymbol{K}\,\boldsymbol{W}_K^{(h)}, \boldsymbol{V}\,\boldsymbol{W}_V^{(h)}), \quad (5)$$

where $h = 1,\ldots,H$ is the indicator of head, $\boldsymbol{W}_H$ is used for final linear mapping and $\boldsymbol{W}_K^{(h)}$, $\boldsymbol{W}_Q^{(h)}$, $\boldsymbol{W}_V^{(h)}$ are head-specific weights for keys, queries and values.

Specifically, the estimation of $\hat{\boldsymbol{M}}$ can be expressed as temporal attention, i.e., MultiHeadAttention($\boldsymbol{Q}_{Tem}$, $\boldsymbol{K}_{Tem}$, $\boldsymbol{V}_{Tem}$). In this case, all the $\boldsymbol{K}_{Tem}$ and $\boldsymbol{V}_{Tem}$ come from one window and they are the representation vector of each time point $\boldsymbol{r}_{j,1},\ldots,\boldsymbol{r}_{j,T}$. Because this is one self-attention, $\boldsymbol{Q}_{Tem}$ is $h(R)$ and also comes from the representation vector. For $\boldsymbol{A}_{Tem}(\boldsymbol{Q}_{Tem},\boldsymbol{K}_{Tem})$, each attention vector $\boldsymbol{a}_{Tem}$ is the network estimation of the probability $P(M=m|h(R))$. For the estimation $\hat{\boldsymbol{R}}$, it is a pattern attention i.e., MultiHeadAttention($\boldsymbol{Q}_{Pat}$, $\boldsymbol{K}_{Pat}$, $\boldsymbol{V}_{Pat}$), where $\boldsymbol{K}_{Pat}$ and $\boldsymbol{V}_{Pat}$ come from the other windows in the data, and $\boldsymbol{Q}_{Pat}$ comes from $f(R)$. In this case, $\boldsymbol{a}_{Pat}$ approximates $P(R=r|f(R))$. In the implementation, because it is impossible to calculate the pattern attention by using all windows in the data, we set $\boldsymbol{K}_{Pat}$ and $\boldsymbol{V}_{Pat}$ as the global dictionaries compressed from the whole dataset. This step also can help to summarize the information and remove the noise. We initialize this dictionary by using K-means over all the windows' representation vectors, i.e., $Concatenate[\boldsymbol{r}_{j,1}^T,\ldots,\boldsymbol{r}_{j,T}^T]$ $(j=1,\ldots,n)$, the concatenated flatten representation vectors of each time point in one window. In this way, $\boldsymbol{V}_{Pat}$

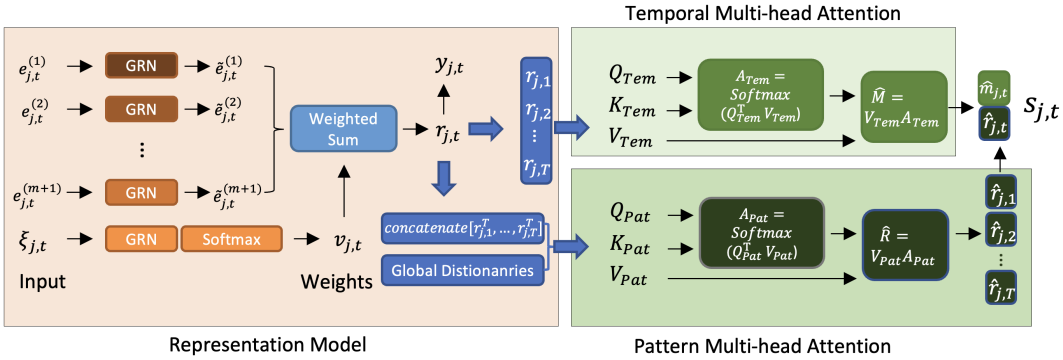

Figure 3: The transformed series item and covariates are input to learn the representation vectors and then to estimate the temporal and pattern attention.

and $\boldsymbol{V}_{Tem}$ stay in the same representation space, which guarantees that the estimations of temporal attention and pattern attention: $\hat{\boldsymbol{Z}}$ and $\hat{\boldsymbol{X}}$ in Eq. (3) have the same distribution.

To sum up, as shown in Figure 3, $\hat{\boldsymbol{m}}_{j,t}$ and $\hat{\boldsymbol{r}}_{j,t}$ are respectively estimated by temporal attention and pattern attention. Therefore, we can get new representation vectors space $\boldsymbol{S}$ after the front-door adjustment, i.e, $\boldsymbol{s}_{j,t} = Concatenate[\hat{\boldsymbol{m}}_{j,t}^T, \hat{\boldsymbol{r}}_{j,t}^T]^T$. Now, we can input the $\boldsymbol{S}$ into our encoder-decoder recurrent network g to estimate the $P(Y|do(R))$.

The simplest encoder-decoder model consists of two RNNs based on LSTMs, i.e., one for the encoder and the other for the decoder. Encoder RNN reads the source sentence and the final state is used as the initial state of the decoder RNN. The hope is that the final encoder state "encodes" all information about the source, and the decoder can generate the target sentence based on this vector. However, its performance degrades with long sentences because it cannot adequately encode a long sequence into the intermediate vector even with LSTM cells. Therefore, we add one transformer attention into the encoder-decoder model. At each decoder step, it decides which encoder parts are more important. In this setting, the encoder does not have to compress the whole source into a single vector, it takes all RNN states into account, instead of the last one of encoder.

4.3 Quantile Outputs

In line with previous work, CTTT also generates prediction intervals on top of point forecasts. This is achieved by the simultaneous prediction of various percentiles (e.g. 10^{th}, 50^{th} and 90^{th}) at each time step. Quantile forecasts are generated by one neural network z based on the output from the decoder part, i.e., $\hat{y}(q, j, t) = z(g(s_{j,t}))$, where q is the specified quantile. CTTT is trained by jointly minimizing the quantile loss (Wen et al., 2017), summed across all quantiles, windows, and time points in prediction window:

$$\mathcal{L} = \sum_{j=1}^{n} \sum_{q \in \mathcal{Q}} \sum_{t=t_0}^{T} \frac{QL\left(y_{j,t},\ \hat{y}(q,j,t),\ q\right)}{m\tau_{max}} \tag{6}$$

$$QL(y,\ \hat{y},\ q) = q(y - \hat{y})_+ + (1-q)(\hat{y} - y)_+, \tag{7}$$

where $\mathcal{Q}$ is the set of quantiles and $\mathcal{Q} = \{0.1, 0.5, 0.9\}$. $(.)_+ = \max(0, .)$. For out-of-sample test, we define Ω as the domain of test windows. We evaluate the normalized quantile losses and compare P50 and P90 risk for consistency with previous work (Salinas et al., 2019; Rangapuram et al., 2018; Li et al., 2019):

$$q\text{-Risk} = \frac{2 \sum_{j \in \Omega} \sum_{t=t_0}^{T} QL\left(y_{j,t},\ \hat{y}(q,j,t),\ q\right)}{\sum_{j \in \Omega} \sum_{t=t_0}^{T} |y_{j,t}|}. \tag{8}$$

5 Experiments

5.1 Datasets

In line with previous work (Salinas et al., 2019; Rangapuram et al., 2018; Li et al., 2019; Lim et al., 2021), we choose four commonly used benchmarks, i.e., Electricity, Traffic, Retail, and Volatility. The UCI Electricity Load Diagrams Dataset (**Electricity**) contains hourly time series of the electricity consumption of 370 customers (Yu et al., 2016; Salinas et al., 2019). The UCI PEM-SF Traffic

Dataset (**Traffic**) contains the hourly occupancy rate, between 0 and 1, of 440 SF Bay Area freeways. For the Electricity and Traffic datasets, we use the past week (i.e. 168 hours) to forecast over the next 24 hours. Favorite Grocery Sales Dataset (**Retail**) is from the Kaggle competition (Favorita, 2018), which combines metadata for different products and the stores. We forecast log product sales 30 days, using 90 days of past information. The OMI realized library (**Volatility**) (Heber et al., 2009) contains daily realized volatility values of 31 stock indices computed from intraday data, along with their daily returns. We consider forecasting over the next week using information over the past year. The detailed information about datasets is presented in Table 1. For each dataset, we partition all time series into 3 parts – a training set for learning, a validation set for hyperparameter tuning, and a test set for performance evaluation. Hyperparameter optimization is conducted via random search, using 60 iterations. Full search ranges for all hyperparameters are listed in Table 2.

Table 1: Information on Four Benchmarks.

Dataset Details	**Electricity**	**Traffic**	**Retail**	**Volatility**
Target Type	$\mathbb{R}$	[0, 1]	$\mathbb{R}$	$\mathbb{R}$
Num. Series	370	440	130k	41
Num. Samples	500k	500k	500k	100k
Con. Window Size	168	168	90	252
Pre. Window Size	24	24	30	5
Num. Variables	5	5	20	8

Table 2: Model Hyperparameters.

Hyperparameters	**Full Search Ranges**
Dropout Rate	0.1, 0.2, 0.3
Minibatch Size	64, 128, 256
Learning Rate	0.0001, 0.001, 0.01
Num. Head	1, 4
Num. LSTM Layers	2,3
Num. LSTM Nodes	30, 40
Representation Size	10, 20, 30, 40

5.2 Baseline Methods

We extensively compare our model to previous work for multi-series and multi-horizon forecasting, such as the classical forecasting methods ARIMA (Box & Jenkins, 1968) and ETS (Gardner Jr, 1985), the recent matrix factorization method TRMF (Yu et al., 2016), simple sequence-to-sequence models with global contexts (Seq2Seq), the multi-horizon quantile recurrent forecaster (MQRNN) (Wen et al., 2017), DeepAR (Salinas et al., 2019), DSSM (Rangapuram et al., 2018), the transformer-based architecture of (Li et al., 2019) with local convolutional processing, and temporal fusion transformers with interpretable attention and variable selection (TFT) (Lim et al., 2021). Because iterative models assume that all input covariates are known, we accommodate this by imputing unknown future inputs with their last available value. The results for previous work have been reproduced from Li et al. (2019); Lim et al. (2021) for consistency.

Table 3: P50 and P90 quantile losses on four real-world datasets. Lower q-Risk better.

Electricity	**ARIMA**	**ETS**	**TRMF**	**DeepAR**	**DSSM**
P50 losses	0.154	0.102	0.084	0.075	0.083
P90 losses	0.102	0.077	-	0.040	0.056
	ConvTrans	**Seq2Seq**	**MQRNN**	**TFT**	**CTTT**
P50 losses	0.059	0.067	0.077	0.055	**0.052**
P90 losses	0.034	0.036	0.036	0.027	**0.025**

Traffic	**ARIMA**	**ETS**	**TRMF**	**DeepAR**	**DSSM**
P50 losses	0.223	0.236	0.186	0.161	0.167
P90 losses	0.137	0.148	-	0.099	0.113
	ConvTrans	**Seq2Seq**	**MQRNN**	**TFT**	**CTTT**
P50 losses	0.122	0.105	0.117	0.095	**0.091**
P90 losses	0.081	0.075	0.082	0.070	**0.065**

Volatility	**DeepAR**	**CovTrans**	**Seq2Seq**	**MQRNN**	**TFT**	**CTTT**
P50 losses	0.050	0.047	0.042	0.042	0.039	**0.038**
P90 losses	0.024	0.024	0.021	0.021	0.020	**0.018**

Retail	**DeepAR**	**CovTrans**	**Seq2Seq**	**MQRNN**	**TFT**	**CTTT**
P50 losses	0.574	0.429	0.411	0.379	0.354	**0.347**
P90 losses	0.230	0.192	0.157	0.152	0.147	**0.139**

5.3 PERFORMANCE

Table 3 shows the performance of our model and baseline methods on the four datasets, i.e., Electricity, Traffic, Retail, and Volatility. We report the results of q-Risk defined in Eq. (8) on the test sets. CTTT achieves the best performance concerning P50 and P90 quantile losses in all four datasets. In fact, compared to other deep neural network models, our model has a similar composition: all are based on the sequence-to-sequence network, recurrent structures, and attention module. Compared with other state-of-the-art models, the accuracy improvement of our model has mainly benefited from the causal inference front-door adjustment to help the model effectively utilize the shared global knowledge along with the series and across different series.

To prove the usefulness of each attention module, we perform two ablation studies of CTTT. Because the temporal attention and pattern attention share the task of front-door adjustment, we remove them together and create the CTTT (w/o Front-door) instead, where the representation vectors learned from the representation model are directly input into the encoder-decoder recurrent network. The second ablation study is CTTT (w/o Trans) where the transformer attention is removed and there is only one original encoder-decoder network connecting via the last encoder cell state. As shown in Figure 4, the performance becomes poor after removing either the transformer attention or the temporal and pattern attention, compared to the original CTTT. Therefore, these three attention modules are essential components of our model. Besides, to visualize the importance of each variable, we present the variable selection weights defined in section 4.1. Figure 5 shows that only a subset of covariates is important for the prediction of the target value.

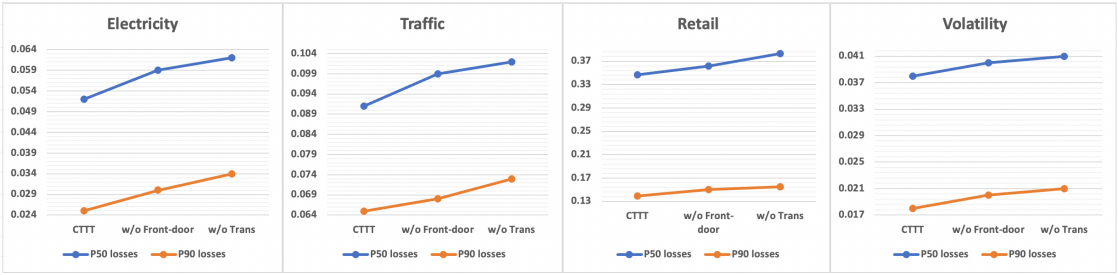

Figure 4: The results of ablation studies CTTT (w/o Front-door) and CTTT (w/o Trans).

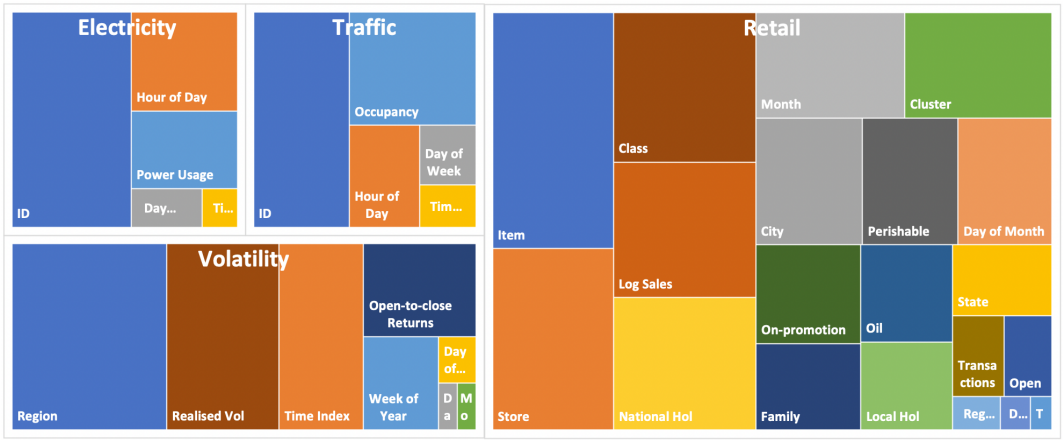

Figure 5: The importance of each variable in Electricity, Traffic, Volatility, and Retail datasets. The size of square represents the relative importance compared with other variables in the same dataset.

6 CONCLUSION

The proposed CTTT is one multi-horizon and multi-series forecasting model based on the deep encoder-decoder recurrent architecture with triple interpretable attention modules, i.e., temporal attention, pattern attention, and transformer attention. From the perspective of causal inference, we present the confounding relationship hidden in the complex time series data and utilize the attention mechanism to help the model capture the useful local and global knowledge without abusing unrelated information. Experimental results on four datasets show that CTTT is highly adaptable to complicated time series forecasting tasks and has significant forecasting performance improvements.

REFERENCES

Ehsan Abbasnejad, Damien Teney, Amin Parvaneh, Javen Shi, and Anton van den Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10044–10054, 2020.

Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews*, 29(5-6): 594–621, 2010.

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.

Yoshua Bengio, Tristan Deleu, Nasim Rahaman, Rosemary Ke, Sébastien Lachapelle, Olexa Bilaniuk, Anirudh Goyal, and Christopher Pal. A meta-transfer objective for learning to disentangle causal mechanisms. *arXiv preprint arXiv:1901.10912*, 2019.

George EP Box and Gwilym M Jenkins. Some recent advances in forecasting and control. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 17(2):91–109, 1968.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark Hasegawa-Johnson, and Thomas S Huang. Dilated recurrent neural networks. *arXiv preprint arXiv:1710.02224*, 2017.

Nicolas Chapados. Effective bayesian modeling of groups of related count time series. In *International conference on machine learning*, pp. 1395–1403. PMLR, 2014.

Zhixuan Chu, Stephen Rathbun, and Sheng Li. Continual lifelong causal effect inference with real world evidence. 2020a.

Zhixuan Chu, Stephen L Rathbun, and Sheng Li. Matching in selective and balanced representation space for treatment effects estimation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 205–214, 2020b.

Zhixuan Chu, Stephen L Rathbun, and Sheng Li. Graph infomax adversarial learning for treatment effect estimation with networked observational data. *arXiv preprint arXiv:2106.02881*, 2021.

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.

Chenyou Fan et al. Multi-horizon time series forecasting with temporal attention learning. In *KDD*, 2019.

Corporacion Favorita. Corporacion favorita grocery sales forecasting competition, 2018. URL `https://www.kaggle.com/c/favorita-grocery-sales-forecasting/`.

Everette S Gardner Jr. Exponential smoothing: The state of the art. *Journal of forecasting*, 4(1): 1–28, 1985.

Andrew C Harvey. Forecasting, structural time series models and the kalman filter. 1990.

Gerd Heber, Asger Lunde, Neil Shephard, and Kevin K. Sheppard. Oxford-man institute's realized library, 2009. URL `https://realized.oxford-man.ox.ac.uk/`.

Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning, 2021.

Rob J Hyndman, Roman A Ahmed, George Athanasopoulos, and Han Lin Shang. Optimal combination forecasts for hierarchical time series. *Computational statistics & data analysis*, 55(9): 2579–2589, 2011.

Sheng Li and Yun Fu. Matching on balanced nonlinear representations for treatment effects estimation. In *NIPS*, 2017.

Shiyang Li et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *NeurIPS*, 2019.

Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 2021.

Yulei Niu, Kaihua Tang, Hanwang Zhang, Zhiwu Lu, Xian-Sheng Hua, and Ji-Rong Wen. Counterfactual vqa: A cause-effect look at language bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12700–12710, 2021.

Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

Judea Pearl. *Causality: models, reasoning and inference*, volume 29. Springer, 2000.

Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018.

Judea Pearl, Madelyn Glymour, and Nicholas P Jewell. *Causal inference in statistics: A primer*. John Wiley & Sons, 2016.

Syama Sundar Rangapuram et al. Deep state space models for time series forecasting. In *NIPS*, 2018.

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

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2019. ISSN 0169-2070.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*. 2017.

Ruofeng Wen et al. A multi-horizon quantile recurrent forecaster. In *NIPS 2017 Time Series Workshop*, 2017.

Peter R Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep transformer models for time series forecasting: The influenza prevalence case. *arXiv preprint arXiv:2001.08317*, 2020.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pp. 2048–2057, 2015.

Xu Yang, Hanwang Zhang, Guojun Qi, and Jianfei Cai. Causal attention for vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9847–9857, 2021.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, 31, 2018.

Liuyi Yao, Zhixuan Chu, Sheng Li, Yaliang Li, Jing Gao, and Aidong Zhang. A survey on causal inference. *arXiv preprint arXiv:2002.02770*, 2020.

Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. In *NIPS*. 2016.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. *arXiv preprint arXiv:2009.13000*, 2020.

Zhongqi Yue, Tan Wang, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Counterfactual zero-shot and open-set visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15404–15414, 2021.

Yi-Fan Zhang, Peter J Thorburn, Wei Xiang, and Peter Fitch. Ssim—a deep learning approach for recovering missing time series sensor data. *IEEE Internet of Things Journal*, 6(4):6618–6628, 2019.