Caformer: Rethinking Time Series Analysis from Causal Perspective

Kexuan Zhang, Xiaobei Zou, Yang Tang

***Abstract*—Time series analysis is a vital task with broad applications in various domains. However, effectively capturing cross-dimension and cross-time dependencies in non-stationary time series poses significant challenges, particularly in the context of environmental factors. The spurious correlation induced by the environment confounds the causal relationships between cross-dimension and cross-time dependencies. In this paper, we introduce a novel framework called Caformer (Causal Transformer) for time series analysis from a causal perspective. Specifically, our framework comprises three components: Dynamic Learner, Environment Learner, and Dependency Learner. The Dynamic Learner unveils dynamic interactions among dimensions, the Environment Learner mitigates spurious correlations caused by environment with a back-door adjustment, and the Dependency Learner aims to infer robust interactions across both time and dimensions. Our Caformer demonstrates consistent state-of-the-art performance across five mainstream time series analysis tasks, including long- and short-term forecasting, imputation, classification, and anomaly detection, with proper interpretability.**

***Index Terms*—Time series, time series forecasting, causal intervention, back-door adjustment.**

I. Introduction

Time series analysis holds immense practical value in real-world fields, including finance [1], [2], climate science [3], and traffic management [4]. It has attained growing attention among researchers for its broad practical value [5]. However, with complex temporal dependencies and dimension interactions involved, the analysis of time series is challenging. Numerous efforts [1], [6] have been devoted to struggling with the modeling of dependencies. Within this spectrum, Transformer [7] and its variants have shown notable capabilities in capturing cross-time dependency [8]. Furthermore, there is a growing emphasis on interactions between cross-dimension and cross-time dependencies for a more comprehensive understanding [9].

However, such dependencies are dynamic and susceptible to external effects. As the inherent property of time series, non-stationary leads to continual changes in statistical characteristics and joint distributions, which can be manifested as dynamic variations in dependencies [10]. Fig. 1 demonstrates our understanding of the dynamic interactions within time series. The cross-dimension and cross-time dependencies can be derived successively from the time series. Cross-dimension dependency captures interactions among dimensions, while cross-time dependency represents the dynamic propagation and interactions among dimensions over time.

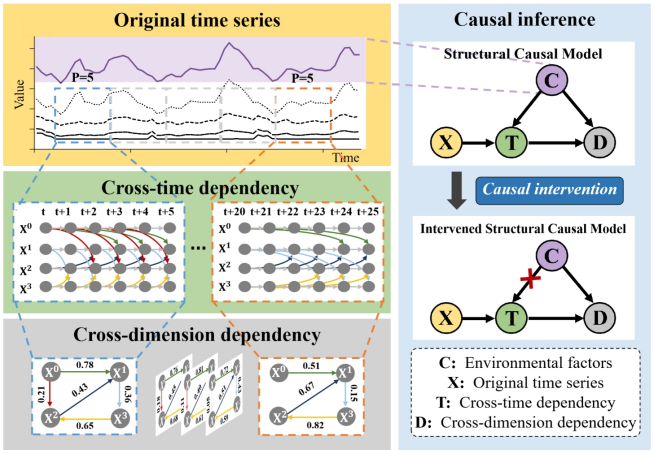

Fig. 1. Dynamic interactions within time series from a causal perspective. The original time series **X** comprises four variables, represented by various types of black lines, along with a purple line symbolizing environmental factors. The cross-time dependency is extracted from two time periods with the same patch size $P = 5$, while the cross-dimension dependency is inferred from the corresponding cross-time dependency. The structural causal model shows the causality between environmental factors, original time series, cross-dimension and cross-time dependencies.

It is worth noting that despite the significant strides in handling complex interactions in previous research [1], [11], a comprehensive exploration into the causal chains underlying these dynamic dependencies remains incomplete. With regard to cross-time dependency, prevailing approaches [11], [12] strive to embed data points from all dimensions at the same time step into a feature vector [9], aiming to uncover inner dependencies among different time steps. Nevertheless, it is crucial not to overlook the impact of external factors that propagate over time, as they may introduce biases. Concerning cross-dimension dependency, prior methods [1], [9] grapple with explicit capturing of dependencies from each individual dimension-specific embedding. However, the learned dependency is approached from a correlation perspective, ignoring the underlying causal graph, where the interaction strength within the graph can indeed vary.

To address the aforementioned challenges, we propose a general model termed Caformer (**Ca**usal Trans**former**) for time series analysis with a task-independent backbone and different heads for specific downstream tasks. The core idea is to obtain a deep understanding of the dynamics that character-

Kexuan Zhang, Xiaobei Zou and Yang Tang are with the Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China (e-mail: kexuanzhang123@gmail.com; xbeizou@gmail.com; yangtang@ecust.edu.cn).

This work was supported by National Natural Science Foundation of China (62233005, 62293502), the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017, Fundamental Research Funds for the Central Universities (222202317006) and Shanghai AI Lab.

ize causal relationships between cross-dimension dependency and cross-time dependency, especially in the presence of environmental factors. Specifically, the main contributions are as follows:

- We establish a structural causal model to explain the underlying mechanisms of interactions in time series and demonstrate the problem of dependency varying from a causal perspective. A novel framework called Caformer is proposed for more accurate and interpretable time-series analysis.
- The Dynamic Learner is proposed to uncover the dynamic underlying causal relations concerning cross-dimension dependency. The Environment Learner quantifies and stratifies the environmental factors with the back-door adjustment for a robust cross-time dependency under varying environments. The Dependency Learner is designed to infer the robust interactions between these two dependencies.
- Caformer achieves consistent state-of-the-art performance on multiple mainstream time series analysis tasks, demonstrating its excellent generalization ability for different tasks and its proper interpretability.

II. Related work

A. Time series analysis.

In recent years, there has been extensive research into time series analysis, with a particular focus on investigating cross-dimension and cross-time dependencies. Table I summarizes the distinctive characteristics of the existing time series methods. The RNN-based methods [13] capture cross-time dependency with the recurrent structure. The MLP-based methods [14], [15] encode the cross-time dependency with parameterized MLP layers. The TCN-based methods [16] obtain the cross-time dependency with the convolution kernels. Meanwhile, the Transformer-based methods [11], [17] embed data points from the same time step and learn the cross-time dependency with the well-designed attention mechanisms. Notably, there has been a recent surge of interest [1], [9] in cross-dimension dependency with an enhanced understanding of time series. Consequently, developing a comprehensive understanding of dependencies within time series data is crucial for effective time series analysis. In this paper, we address both cross-dimension and cross-time dependencies and refine the learning of such dependencies from a dynamic perspective.

B. Dynamics in time series.

The identification of deterministic dynamics within observed time series provides valuable insights into underlying physical processes, especially on shorter timescales [23]. Interactions among components are widespread in natural systems, contributing to the complexity of time series analysis. While significant efforts have been devoted to learning dynamical models of interactions using techniques like graph neural networks [24], [25] and attention mechanisms [26], it's noteworthy that dynamic relationships are often overlooked in time series analysis. Existing methods primarily focus on learning dependencies across multiple dimensions, neglecting explicit modeling of dynamic interactions. In this paper, we propose an approach that explicitly learns the dynamics within the time series to enhance the quality of learned dependencies.

TABLE I
Different characteristics of the popular Transformer-based models. CDD (**C**ross-**D**imension **D**ependency) and CTD (**C**ross-**T**ime **D**ependency) indicate whether the methods learn cross-dimension and cross-time dependencies, respectively. Dynamic shows the condition of the learning of dynamic interactions.

Model name	Task	CDD	CTD	Dynamic
[18]	Imputation	✗	✓	✗
[19]	Classification	✗	✓	✗
[20]	Classification	✓	✓	✗
[21]	Anomaly Detection	✓	✓	✗
[6]	Anomaly Detection	✗	✓	✗
[22]	Forecasting	✗	✓	✗
[11]	Forecasting	✗	✓	✗
[14]	Forecasting	✗	✓	✗
[9]	Forecasting	✓	✓	✗
[12]	Forecasting	✗	✓	✗
[1]	Forecasting	✓	✗	✗

C. Causal inference.

Causal inference plays a crucial role in uncovering the causal structure of systems and quantifying causal effects. It achieves this by integrating domain knowledge, machine models, and observational or interventional data [27]. Recognizing that "Correlation is not causation," recent advancements in deep learning models have incorporated causal inference to infer causal effects in various domains, including computer vision [28], natural language processing [29], and robotics [30]. Spurious correlation refers to situations where a misleading correlation between two variables arises due to the influence of a third causal variable [31]. These accidental correlations often result from factors such as sample selection bias, making it challenging to discern true causation. With the help of causal inference, such spurious bias can be eliminated, and the desired model effects can be disentangled. In our work, we leverage the structural causal model developed by Pearl [32] to analyze the relationships among the original time series, the cross-dimension dependency, the cross-time dependency, and the environmental factors.

III. Caformer

To comprehensively unveil cross-dimension and cross-time dependencies within time series, a novel framework called Caformer is proposed, depicted in Fig. 2.

Given $\mathbf{X} \in \mathbb{R}^{M \times L}$, representing the multivariate time series with a length of L and M recorded dimensions, each $\mathbf{X}_t^i$ can be expressed as follows:

$$\mathbf{X}_t^i := \mathrm{f}_i(\mathrm{pa}(\mathbf{X}_t^i), \boldsymbol{\eta}_t^i), \tag{1}$$

where $\mathrm{f}_i(\cdot)$ is the nonlinear functional causal mechanism determining the value of $\mathbf{X}_t^i$ based on its causal parents $\mathrm{pa}(\mathbf{X}_t^i)$ and the independent noise $\boldsymbol{\eta}_t^i$. The causal parents $\mathrm{pa}(\mathbf{X}_t^i)$ form a subset of $\{\mathbf{X}_t, \ldots, \mathbf{X}_{t-\tau_{\max}}\} \backslash \{\mathbf{X}_t^i\}$ with time lag $\tau_{\max} \geq 0$ [27], indicating all the causes of $\mathbf{X}_t^i$ except

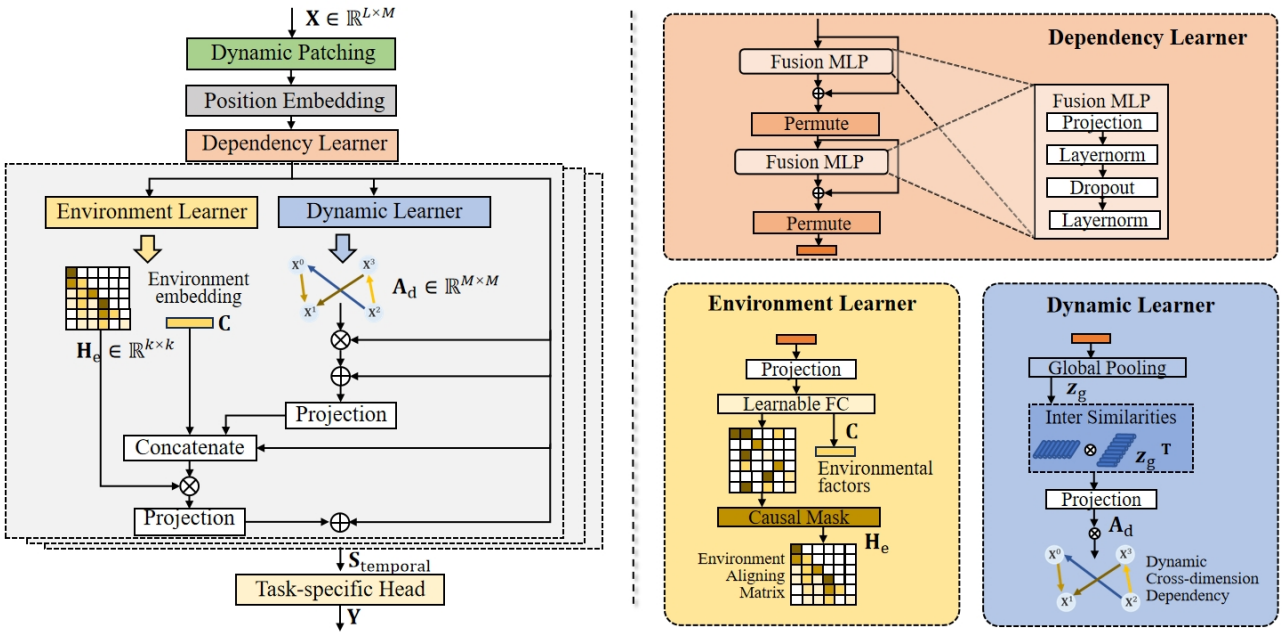

Fig. 2. The architecture of Caformer (**Ca**usal Trans**former**). It comprises three components: Dependency Learner, Environment Learner, and Dynamic Learner.

for itself. As direct causes, $\text{pa}(\mathbf{X}_t^i)$ propagates their effects on $\mathbf{X}_t^i$ with finite time lags of at most $\tau_{\max}$ time steps. The noise $\boldsymbol{\eta}_t^i$ encompasses the effects of all the external factors. Combining this insight with Fig. 1, it can be deduced that $\text{f}_i(\cdot)$ reflects the causal interactions among dimensions as the cross-dimension dependency, while $\text{pa}(\cdot)$ captures the propagation among dimensions from the temporal perspective, embodying cross-time dependency. Additionally, the influence of external factors $\boldsymbol{\eta}_t^i$ cannot be neglected. Their impact on time series can contribute to inevitable distribution variations and dynamic changes. Therefore, it is of great importance to have a comprehensive understanding of cross-dimension and cross-time dependencies to reveal the underlying causal mechanisms within time series.

A. Structural Causal Model

We construct a structural causal model [32] to elucidate the causal relationships among four variables: the original time series $\mathbf{X}$, the environmental factors $\mathbf{C}$, the cross-time dependency $\mathbf{T}$, and the cross-dimension dependency $\mathbf{D}$, in Fig. 1. The directed edges in the graph denote the causal relationships between nodes. The specifics of these cause-effect relationships are elaborated below:

$\mathbf{X} \rightarrow \mathbf{T} \rightarrow \mathbf{D}$. The cross-time dependency can be discerned from time series, whereas the cross-dimension dependency represents the dynamic interactions concealed within the cross-time dependency. These paths elucidate the inferential process of distinct dependencies within time series.

$\mathbf{T} \leftarrow \mathbf{C} \rightarrow \mathbf{D}$. Both the cross-time dependency $\mathbf{T}$ and the cross-dimension dependency $\mathbf{D}$ are susceptible to the influence of environmental factors. In the context of cross-time dependency, the environment introduces dynamic dependencies through non-stationary distributions. These distributions induce shifts in time lags at the time level, causing a delay or advancement in the point of action. Concerning cross-dimension dependency, the environment's impact on one dimension alters the direction and strength of interactions among all dimensions, leading to a fluctuating cross-dimension dependency.

However, the back-door path $\mathbf{T} \leftarrow \mathbf{C} \rightarrow \mathbf{D}$ between $\mathbf{T}$ and $\mathbf{D}$ can bring the spurious correlation when inferring $\mathbf{D}$ from $\mathbf{T}$. Changes in cross-time dependency may be erroneously interpreted as a shift in the strength of the action, introducing additional non-causal cross-dimension dependency due to variations in propagation through different time lags.

B. Causal Intervention via Back-door Adjustment

As stated above, the inference of the cross-dimension and cross-time dependencies can be confounded by the inevitable environmental factors. Fortunately, with the guidance of the back-door criterion [32], the edge $\mathbf{C} \rightarrow \mathbf{T}$ can be cut off to eliminate the confounding between $\mathbf{T}$ and $\mathbf{D}$. Given the stratified environmental factors, the back-door adjustment can be conducted as in Eq. (2).

$$\mathcal{P}(\mathbf{D}|do(\mathbf{T})) = \sum_{i}^{n} \mathcal{P}(\mathbf{D}|\mathbf{T}, \mathbf{c}_i)\mathcal{P}(\mathbf{c}_i), \tag{2}$$

where n indicates the number of the stratified environmental factors $\{\mathbf{c}_i\}_{i=1}^n$, and $do(\mathbf{T})$ denotes the do-calculus on $\mathbf{T}$, cutting off all the edges pointing to $\mathbf{T}$. It can be inferred from Eq. (2) that the cross-dimension dependency is calculated by incorporating each environmental factor $\mathbf{c}_i$ with the cross-time dependency and applying weights through $\mathcal{P}(\mathbf{c}_i)$.

Consequently, $\text{pa}(\cdot)$ in Eq. (1) is equivalent to the inference of $\mathcal{P}(\mathbf{T})$, while $\mathcal{P}(\mathbf{D})$ can be modified as $\mathcal{P}(\mathbf{D}|do(\mathbf{T}))$ concerning $\text{f}_i(\cdot)$ to eliminate spurious correlations induced by the environment. In this paper, the Dynamic Learner is proposed to reveal the underlying dynamic interactions among dimensions;

the Environment Learner is designed to quantify and stratify the environmental factors with back-door adjustments; and the Dependency Learner is introduced to grasp robust interactions among these two dependencies.

C. Dependency Learner and Dynamic Learner

Since the underlying causal relationships within time series are not constant over time due to non-stationarity, a fixed structure cannot conclusively represent the entire time series. However, individual time point in time series may be easily inferred from their neighbors and might not contain significant semantic information. Hence, we make the assumption that the causal relationships within a specific period of time remain stable and try to learn the dimension-independent embedding with the patches of series. The illustration of the patching process is shown in Fig. 3 (a).

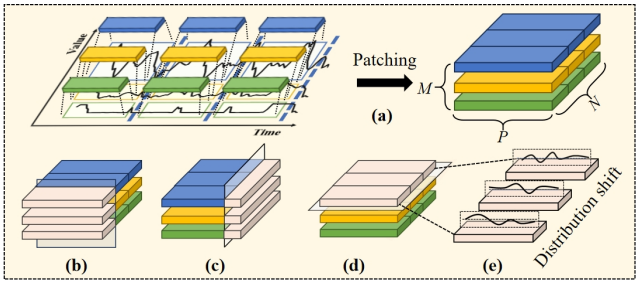

Fig. 3. The illustration of patching process and the dependencies among patched time series.

Dependency Learner. With the patched data $\mathbf{X}_{\mathrm{p}} \in \mathbb{R}^{M\times P\times N}$, where $N = \frac{L-P}{S}+2$ is the number of patches, and both P and S are hyper-parameters concerning patching, the obtained embedding can represent N stacked time patches with different underlying causal relationships. The Dependency Learner is proposed to model the interactions between cross-dimension and cross-time dependencies. As depicted in Fig. 3, these embeddings encompass various dependencies. Fig. 3 (b) indicates the cross-dimension dependency over a period of time; Fig. 3 (c) shows the cross-dimension dependency at the same time step; Fig. 3 (d) refers to the cross-time dependency among patches. Since the points of the same time step may basically represent completely different physical meanings recorded by inconsistent measurements [1], only the dependencies in Fig. 3 (b) and (d) are learned with Dependency Learner in Fig. 2. The in-patch normalization is also proposed to eliminate the distribution shift between patches as shown in Fig. 2 (e). Finally, the interactions between cross-dimension and cross-time dependencies $\mathbf{I}_{\mathrm{de}} \in \mathbb{R}^{N\times M\times E}$ are obtained, where E denotes the size of the embeddings.

Dynamic Learner. The Dynamic Learner is proposed to infer the underlying dynamics in different patch series for cross-dimension dependency. The global dimension information $\mathbf{z}_{\mathrm{g}} \in \mathbb{R}^{N\times M\times 1}$ is first obtained from the learned dependencies $\mathbf{I}_{\mathrm{de}}$ in Dependency Learner. The inter-similarity $\mathbf{A}_{\mathrm{d}} \in \mathbb{R}^{N\times M\times M}$ is then calculated for the expression of the dynamic interactions between dimensions.

$$\mathbf{z}_{\mathrm{g}} = \mathrm{FC}(\mathrm{GPooling}(\mathbf{I}_{\mathrm{de}})), \tag{3}$$

$$\mathbf{A}_{\mathrm{d}} = \mathrm{Norm}(\phi(\frac{\mathbf{z}_{\mathrm{g}}\mathbf{z}_{\mathrm{g}}^{\mathsf{T}}}{\sqrt{\alpha}})), \tag{4}$$

where $\mathrm{GPooling}(\cdot)$ denotes the pooling function, $\mathrm{FC}(\cdot)$ is the Fully Connected layer, $\phi(\cdot)$ is the Softmax function, $\mathrm{Norm}(\cdot)$ is the normalization function, and α is the given parameter for normalization. With $\mathbf{A}_{\mathrm{d}}$, the cross-dimension dependency $\mathbf{D} \in \mathbb{R}^{N\times M\times M}$ can be strengthened as follows:

$$\mathbf{D} = \mathbf{A}_{\mathrm{d}}\mathbf{I}_{\mathrm{de}}. \tag{5}$$

D. Environment Learner

As stated above, explicit modeling of environmental factors can promote the elimination of spurious correlations between cross-dimension and cross-time dependencies. The Environment Learner is proposed to quantify and stratify the environmental factors for both $\mathcal{P}(\mathbf{c}_i)$ and $\{\mathbf{c}_i\}_{i=1}^{n}$, as shown in Fig. 2. With the dependencies $\mathbf{I}_{\mathrm{de}}$ learned from Dependency Learner, the environmental factors $\mathbf{C}$ can be inferred as follows:

$$\mathbf{S}_{\mathrm{e}} = \mathrm{F}_{\mathrm{e}}(\mathbf{I}_{\mathrm{de}}), \tag{6}$$

$$\mathbf{C} = \boldsymbol{\alpha}_2(\mathrm{ReLU}(\boldsymbol{\alpha}_1(\mathbf{S}_{\mathrm{e}}) + \boldsymbol{\gamma}_1)) + \boldsymbol{\gamma}_2, \tag{7}$$

where $\mathrm{F}_{\mathrm{e}}(\cdot)$ denotes the modeling of the latent representation of environmental factors, $\boldsymbol{\alpha}_1$, $\boldsymbol{\alpha}_2$ and $\boldsymbol{\gamma}_1$, $\boldsymbol{\gamma}_2$ are learnable model parameters. $\mathrm{ReLU}(\cdot)$ indicates the ReLU function. After modeling $\{\mathbf{c}_i\}_{i=1}^{n}$, $\mathbf{H}_e \in \mathbb{R}^{k\times k}$ is used to denote $\mathcal{P}(\mathbf{c}_i)$, where k is the hyper-parameter determining the stratifying condition.

$$\mathbf{H}_{\mathrm{e}} = \mathrm{Norm}(\phi(\frac{\mathrm{Proj}(\mathbf{C})\mathrm{Proj}(\mathbf{C})^{\mathsf{T}}}{\sqrt{\beta}})), \tag{8}$$

where $\mathrm{Proj}(\cdot)$ maps the the environmental factors to k dimensions, and β is the predefined parameter for normalization. Instead of directly using $\mathbf{H}_{\mathrm{e}}$, a causal mask function $\mathrm{M}_{\mathrm{c}}(\cdot)$ is used to insure the potential embedding consistent with causality, since future value can't have an impact on history information. $\mathcal{P}(\mathbf{c}_i)$ can ultimately be expressed as $\mathbf{H}_{\mathrm{ce}} = \mathrm{M}_c(\mathbf{H}_{\mathrm{e}})$.

The environmental factors, the cross-time dependency, and the intervened cross-dimension dependency are concentrated together, strengthening the learned temporal features $\mathbf{S}_{\mathrm{temporal}}$ with fully learned dependencies.

$$\mathbf{T} = \mathrm{Norm}(\mathbf{H}_{\mathrm{ce}}\mathrm{FC}(\mathrm{Concat}(\mathbf{D}, \mathbf{C}, \mathbf{I}_{\mathrm{de}}))), \tag{9}$$

$$\mathbf{S}_{\mathrm{temporal}} = \mathbf{T} + \mathbf{I}_{\mathrm{de}}, \tag{10}$$

Finally, $\mathbf{S}_{\mathrm{temporal}}$ can be imported into any task-specific head, corresponding to different analysis tasks for the specific label $\mathbf{Y}$.

IV. Experiments

To validate the effectiveness of Caformer, comprehensive experiments are conducted on five popular time series analysis tasks, including long- and short-term forecasting, imputation, classification, and anomaly detection, utilizing benchmark

TABLE II
SUMMARY OF EXPERIMENT BENCHMARKS AND CORRESPONDING METRICS.

Tasks	Benchmarks	Metrics
Long-term forecasting	ETT (4 subsets), Electricity, Traffic, Weather, Exchange, ILI	MSE, MAE
Short-term forecasting	M4 (6 subsets)	SMAPE, MASE, OWA
Imputation	ETT (4 subsets), Electricity, Weather	MSE, MAE
Classification	UEA (10 subsets)	Classification Accuracy
Anomaly detection	SMD, MSL, SMAP, SWaT, PSM	Precision, Recall, F1-score

datasets. To ensure a fair comparison, we adhere to the well-established experimental setups and implementations of task-general baselines in [5].

Implementation Table II summarizes the benchmarks. All the experiments are implemented in PyTorch and conducted on a single NVIDIA GeForce RTX 3090 GPU with 24GB memory.

To ensure the generalization of the model, the number of layers, the size of the matrix, and the patch size are designed respectively, considering the different characteristics of the models and datasets. In order to provide a fair comparison, we exclusively assess the base models' capabilities while maintaining the identical input embedding and final projection layer across them. The modular architecture of learning a task-independent 'backbone' to capture the temporal features and different 'heads' for specific downstream tasks is adopted.

We compare our model with the latest models in the time series community. The baselines include Transformer-based models: iTransformer [1], PatchTST [12], Crossformer [9], FEDformer [17], and ETSformer [33]; Convolution-based models: TimesNet [5]; MLP-based models: DLinear [15], and LightTS [34].

Concerning the metrics, both the mean square error (MSE) and mean absolute error (MAE) are used as metrics in long-term forecasting tasks. For short-term forecasting, the symmetric mean absolute percentage error (SMAPE), mean absolute scaled error (MASE) and overall weighted average (OWA) are adopted as the metrics. In imputation tasks, both the mean square error (MSE) and mean absolute error (MAE) are introduced. For classification, we use the accuracy as metrics. Concerning the anomaly detection tasks, we adopt the F1-score, which is the harmonic mean of precision and recall. All the metrics are calculated as follows:

$$\text{MSE} = \frac{1}{T}\sum_{i=1}^{T}(\mathbf{X}_i - \hat{\mathbf{X}}_i)^2, \tag{11}$$

$$\text{MAE} = \frac{1}{T}\sum_{i=1}^{T}\left|\mathbf{X}_i - \hat{\mathbf{X}}_i\right|, \tag{12}$$

$$\text{SMAPE} = \frac{200}{T}\sum_{i=1}^{T}\frac{\left|\mathbf{X}_i - \hat{\mathbf{X}}_i\right|}{\left|\mathbf{X}_i\right| + \left|\hat{\mathbf{X}}_i\right|}, \tag{13}$$

$$\text{MAPE} = \frac{100}{T}\sum_{i=1}^{T}\frac{\left|\mathbf{X}_i - \hat{\mathbf{X}}_i\right|}{\left|\mathbf{X}_i\right|}, \tag{14}$$

$$\text{MASE} = \frac{1}{T}\sum_{i=1}^{T}\frac{\left|\mathbf{X}_i - \hat{\mathbf{X}}_i\right|}{\frac{1}{T-m}\sum_{j=m+1}^{T}\left|\mathbf{X}_j - \mathbf{X}_{j-m}\right|}, \tag{15}$$

$$\text{OWA} = \frac{1}{2}\left[\frac{\text{SMAPE}}{\text{SMAPE}_{\text{Naive2}}} + \frac{\text{MASE}}{\text{MASE}_{\text{Naive2}}}\right], \tag{16}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{17}$$

where $\mathbf{X}_i$ and $\hat{\mathbf{X}}_i$ are the ground truth and forecasting of the i-th time step in T future time steps, and m is the periodicity of the data. $\text{SMAPE}_{\text{Naive2}}$ and $\text{MASE}_{\text{Naive2}}$ are SMAPE and MASE of the baseline method provided in the M4 competition in [35].

A. Main Results

As shown in Fig. 4, our Caformer achieves consistent state-of-the-art performance on five mainstream analysis tasks. The details and summarized results of each task are shown in the following subsections.

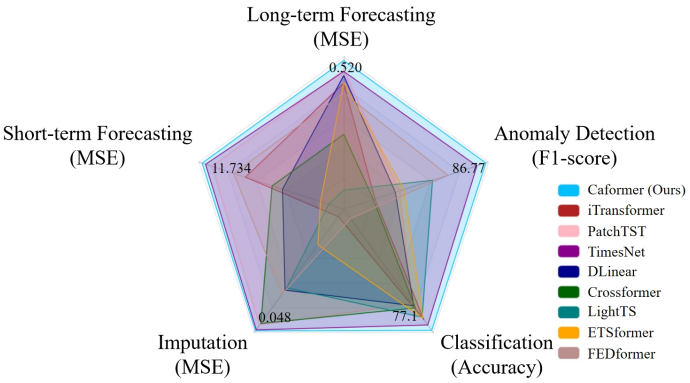

Fig. 4. Comparison between models on five mainstream time series analysis tasks. A larger area indicates a better generalization result of the method across tasks.

B. Long-term forecasting

Setups Long-term forecasting is a category of time series forecasting tasks characterized by an extensive forecasting horizon as output, demanding models with robust capabilities in long-term modeling. We conduct long-term forecasting experiments on 9 popular benchmarks, including ETT [22], Electricity[1], Weather[2], ILI[3], Traffic[4], and Exchange [36]. The MSE and MAE are used as the metrics, and the input lengths

[1]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014
[2]https://ncei.noaa.gov/data/local-climatological-data
[3]https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html
[4]http://pems.dot.ca.gov/

TABLE III
LONG-TERM FORECASTING TASK. THE LENGTHS OF THE INPUT SEQUENCES ARE SET AS 36 FOR ILI AND 96 FOR THE OTHERS. ALL THE RESULTS ARE THE AVERAGE OF FOUR PREDICTION LENGTHS, THAT IS {24, 36, 48, 60} FOR ILI AND {96, 192, 336, 720} FOR THE OTHERS. A LOWER MSE OR MAE INDICATES A BETTER PERFORMANCE. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND BEST ARE UNDERLINED.

Models		Ours		iTransformer (2024)		PatchTST (2023)		TimesNet (2023)		Dlinear (2023)		Crossformer (2023)		LightTS (2022)		ETSformer (2022)		FEDformer (2022)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.368**	**0.391**	0.394	0.409	0.377	0.396	0.384	0.402	0.386	0.400	0.420	0.439	0.424	0.432	0.494	0.479	0.376	0.419
	192	**0.418**	**0.422**	0.448	0.440	0.425	0.426	0.436	0.429	0.437	0.432	0.541	0.520	0.475	0.462	0.538	0.504	0.420	0.448
	336	**0.453**	**0.440**	0.492	0.465	0.461	0.448	0.491	0.469	0.481	0.459	0.722	0.648	0.518	0.488	0.574	0.521	0.459	0.465
	720	**0.456**	**0.458**	0.520	0.503	0.529	0.500	0.521	0.500	0.519	0.516	0.811	0.691	0.547	0.533	0.562	0.535	0.506	0.507
	Avg	**0.424**	**0.428**	0.464	0.454	0.448	0.443	0.458	0.450	0.456	0.452	0.623	0.574	0.491	0.479	0.542	0.510	0.440	0.460
ETTh2	96	**0.283**	**0.338**	0.297	0.349	0.310	0.353	0.340	0.374	0.333	0.387	0.745	0.584	0.397	0.437	0.340	0.391	0.358	0.397
	192	**0.365**	**0.390**	0.380	0.400	0.390	0.405	0.402	0.414	0.477	0.476	0.877	0.656	0.520	0.504	0.430	0.439	0.429	0.439
	336	**0.399**	**0.423**	0.428	0.432	0.430	0.434	0.452	0.452	0.594	0.541	1.043	0.731	0.626	0.559	0.485	0.479	0.496	0.487
	720	**0.420**	**0.443**	0.427	0.445	0.438	0.449	0.462	0.468	0.831	0.657	1.104	0.763	0.863	0.672	0.500	0.497	0.463	0.474
	Avg	**0.367**	**0.399**	0.383	0.407	0.392	0.410	0.414	0.427	0.559	0.515	0.942	0.684	0.602	0.543	0.439	0.452	0.437	0.449
ETTm1	96	**0.306**	**0.358**	0.343	0.378	0.332	0.369	0.338	0.375	0.345	0.372	0.370	0.404	0.374	0.400	0.375	0.398	0.379	0.419
	192	**0.347**	**0.380**	0.381	0.395	0.373	0.389	0.374	0.387	0.380	0.389	0.460	0.488	0.400	0.407	0.408	0.410	0.426	0.441
	336	**0.389**	**0.401**	0.419	0.418	0.408	0.411	0.410	0.411	0.413	0.413	0.637	0.607	0.438	0.438	0.435	0.428	0.445	0.459
	720	**0.452**	**0.434**	0.486	0.456	0.463	0.446	0.478	0.450	0.474	0.453	0.863	0.720	0.527	0.502	0.499	0.462	0.543	0.490
	Avg	**0.374**	**0.393**	0.407	0.411	0.394	0.404	0.400	0.406	0.403	0.407	0.582	0.555	0.435	0.437	0.429	0.425	0.448	0.452
ETTm2	96	**0.170**	**0.254**	0.183	0.268	0.177	0.259	0.187	0.267	0.193	0.292	0.270	0.372	0.209	0.308	0.189	0.280	0.203	0.287
	192	**0.231**	**0.283**	0.252	0.312	0.242	0.301	0.249	0.309	0.284	0.362	0.242	0.301	0.311	0.382	0.253	0.319	0.269	0.328
	336	**0.296**	**0.325**	0.313	0.350	0.303	0.343	0.321	0.351	0.369	0.427	0.303	0.343	0.442	0.466	0.314	0.357	0.325	0.366
	720	**0.405**	**0.402**	0.411	0.406	0.410	0.404	0.408	0.403	0.554	0.522	0.410	0.404	0.675	0.587	0.414	0.413	0.421	0.415
	Avg	**0.276**	**0.316**	0.290	0.334	0.283	0.327	0.291	0.333	0.350	0.401	0.306	0.355	0.409	0.436	0.293	0.342	0.305	0.349
Electricity	96	0.153	0.259	**0.148**	**0.240**	0.170	0.260	0.168	0.272	0.197	0.282	0.150	0.253	0.207	0.307	0.187	0.304	0.193	0.308
	192	**0.159**	**0.245**	0.162	0.253	0.187	0.276	0.184	0.289	0.196	0.285	0.167	0.267	0.165	0.264	0.199	0.315	0.201	0.315
	336	**0.171**	0.275	0.178	**0.269**	0.203	0.291	0.198	0.300	0.209	0.301	0.189	0.287	0.230	0.333	0.212	0.329	0.214	0.329
	720	**0.193**	**0.287**	0.225	0.317	0.245	0.325	0.220	0.320	0.245	0.333	0.256	0.337	0.265	0.360	0.233	0.345	0.246	0.355
	Avg	**0.169**	**0.267**	0.178	0.270	0.201	0.288	0.192	0.295	0.212	0.300	0.191	0.286	0.229	0.329	0.208	0.323	0.214	0.327
Weather	96	**0.168**	**0.210**	0.174	0.214	0.179	0.220	0.172	0.220	0.196	0.255	0.174	0.239	0.182	0.242	0.197	0.281	0.217	0.296
	192	**0.196**	**0.243**	0.221	0.254	0.222	0.257	0.219	0.261	0.237	0.296	0.221	0.287	0.227	0.287	0.237	0.312	0.276	0.336
	336	**0.261**	**0.283**	0.278	0.296	0.279	0.297	0.280	0.306	0.283	0.335	0.277	0.340	0.282	0.334	0.298	0.353	0.339	0.380
	720	0.350	**0.345**	0.358	0.349	0.357	0.351	0.365	0.359	**0.345**	0.381	0.371	0.410	0.352	0.386	0.352	0.288	0.403	0.428
	Avg	**0.244**	**0.270**	0.258	0.279	0.259	0.281	0.259	0.287	0.265	0.317	0.261	0.319	0.261	0.312	0.271	0.334	0.309	0.360
ILI	24	**2.003**	**0.915**	3.014	1.169	2.281	0.926	2.317	0.934	2.215	1.081	3.478	1.242	8.313	2.144	2.527	1.020	3.228	1.260
	36	2.010	0.923	2.991	1.172	2.344	0.938	1.972	**0.920**	**1.963**	0.963	4.268	1.399	6.631	1.902	2.615	1.007	2.679	1.080
	48	**1.996**	**0.910**	2.862	1.146	2.184	**0.910**	2.238	0.940	2.130	1.024	3.774	1.287	7.299	1.982	2.359	0.972	2.622	1.078
	60	**1.978**	**0.903**	2.955	1.175	2.092	0.921	2.027	0.928	2.368	1.096	4.009	1.335	7.283	1.985	2.487	1.016	2.857	1.157
	Avg	**1.997**	**0.913**	2.956	1.166	2.225	0.924	2.139	0.931	2.169	1.041	3.882	1.316	7.382	2.003	2.497	1.004	2.847	1.144
Traffic	96	0.473	0.285	**0.395**	**0.268**	0.544	0.360	0.593	0.321	0.650	0.396	0.522	0.290	0.615	0.391	0.607	0.392	0.587	0.366
	192	0.474	0.311	**0.417**	**0.276**	0.540	0.354	0.617	0.336	0.598	0.370	0.530	0.293	0.601	0.382	0.621	0.399	0.604	0.373
	336	0.495	0.300	**0.433**	**0.283**	0.552	0.360	0.629	0.336	0.605	0.373	0.558	0.305	0.613	0.386	0.622	0.396	0.621	0.383
	720	0.528	0.313	**0.467**	**0.302**	0.590	0.378	0.640	0.350	0.645	0.394	0.589	0.328	0.658	0.407	0.632	0.396	0.626	0.382
	Avg	0.493	0.302	**0.428**	**0.282**	0.557	0.363	0.620	0.336	0.625	0.383	0.550	0.304	0.622	0.392	0.621	0.396	0.610	0.376
Exchange	96	**0.079**	**0.197**	0.103	0.231	0.085	0.202	0.107	0.234	0.088	0.218	0.256	0.367	0.116	0.262	0.085	0.204	0.148	0.278
	192	**0.176**	**0.294**	0.189	0.313	0.180	0.301	0.226	0.334	**0.176**	0.315	0.469	0.508	0.182	0.303	0.271	0.380	0.604	0.373
	336	**0.297**	**0.401**	0.377	0.447	0.336	0.420	0.367	0.448	0.313	0.427	0.975	0.763	0.377	0.466	0.348	0.428	0.460	0.500
	720	**0.823**	**0.684**	0.870	0.705	0.881	0.710	0.964	0.746	0.839	0.695	1.618	1.028	0.831	0.699	1.025	0.774	1.195	0.841
	Avg	**0.343**	**0.394**	0.384	0.424	0.371	0.408	0.416	0.440	0.354	0.414	0.830	0.667	0.377	0.432	0.432	0.447	0.602	0.498

are fixed for a fair comparison.

Results As shown in Table III, Caformer shows great performance in long-term forecasting. With a thorough grasp of interactions within time series, Caformer proficiently learns the underlying causal relationships, effectively capturing features that contribute to predicting future trends.

C. Short-term forecasting

Setups We also conduct short-term forecasting, which has a relatively short horizon, on M4 dataset [35]. The M4 dataset encompasses 100,000 distinct time series collected at various frequencies, categorized into 6 subsets: yearly, quarterly, monthly, weekly, daily, and hourly. The input lengths are set at twice the prediction lengths. Three metrics, namely SMAPE, MASE, and OWA, are employed for evaluation.

Results Table IV demonstrates that our model is superior to other methods in most cases. Given that the time series in M4 dataset originate from diverse sources with varying properties, the remarkable performance of our Caformer underscores the robustness of the learned features.

TABLE IV
SHORT-TERM FORECASTING TASK. THE PREDICTION LENGTHS ARE IN $[6, 48]$. THE RESULTS ARE AVERAGED FROM SEVERAL DATASETS UNDER DIFFERENT SAMPLE INTERVALS. LOWER METRICS INDICATE BETTER PERFORMANCE. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND BEST ARE UNDERLINED.

Models		Ours	iTransformer (2024)	PatchTST (2023)	TimesNet (2023)	DLinear (2023)	Crossformer (2023)	LightTS (2022)	ETSformer (2022)	FEDformer (2022)
Yearly	SMAPE	**13.365**	14.321	13.550	13.387	16.965	13.392	14.247	18.009	13.728
	MASE	**2.895**	3.230	3.028	2.996	4.283	3.001	3.109	4.487	3.048
	OWA	**0.776**	0.845	0.796	0.786	0.781	0.787	0.827	1.115	0.803
Quarterly	SMAPE	**9.989**	10.764	10.195	10.100	12.145	16.317	11.364	13.376	10.792
	MASE	**1.171**	1.284	1.205	1.182	1.520	2.197	1.328	1.906	1.283
	OWA	**0.882**	0.956	0.902	0.890	1.106	1.542	1.000	1.302	0.958
Monthly	SMAPE	**12.612**	14.245	12.96	12.670	13.514	12.924	14.014	14.588	14.260
	MASE	**0.914**	1.110	0.968	0.933	1.037	0.966	1.053	1.368	1.102
	OWA	**0.861**	1.016	0.905	0.878	0.956	0.902	0.981	1.149	1.012
Others	SMAPE	**4.723**	5.780	5.005	4.891	6.709	5.493	15.880	7.267	4.954
	MASE	**3.105**	4.153	3.394	3.302	4.953	3.690	3.690	5.240	3.264
	OWA	**0.991**	1.263	1.062	1.035	1.487	1.160	3.474	1.591	1.036
W.Avg.	SMAPE	**11.734**	12.999	12.034	11.829	13.639	13.474	13.525	14.718	12.840
	MASE	**1.571**	1.792	1.620	1.585	2.095	1.866	2.111	2.408	1.701
	OWA	**0.842**	0.948	0.867	0.851	1.051	0.985	1.051	1.172	0.918

D. Imputation

Setups The absence of data significantly hampers the performance of downstream analysis tasks, and time series imputation is a common practice employed to address missing data resulting from malfunctions. We select ETT [22], Electricity[1], and Weather[2] as our benchmarks. The time points are randomly masked in the ratio of $\{12.5\%, 25\%, 37.5\%, 50\%\}$ to imitate different proportions of missing data. Both the MSE and MAE are used as metrics.

Results Table V shows the great performance of our model in imputation tasks. We argue that the insufficiency of data has a certain impact on the learning of cross-dimension and cross-time dependencies, thereby introducing undesirable biases. Thus, some outcomes in Table V are not particularly favorable. Addressing these challenges constitutes a focal point for our future research direction.

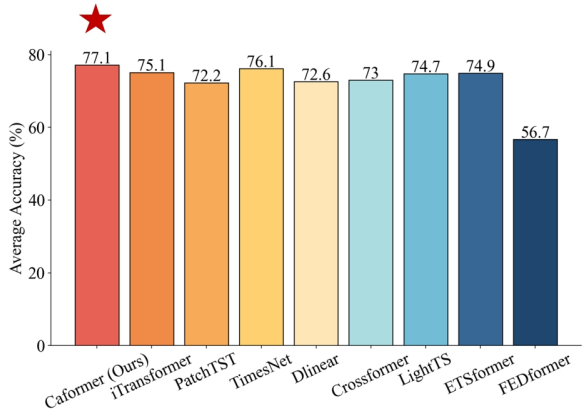

Fig. 5. Model comparison in classification. The results are averaged from 10 subsets of UEA.

E. Classification

Setups Time series classification aims to distinguish meaningful data patterns and effectively classify each time series. We conduct the experiments with 10 datasets from UEA Time Series Classification Archive [37]. The classification accuracy is used as the metric.

Results As shown in Fig. 5, Caformer achieves the best performance with an average accuracy of 77.1%. By emphasizing dynamic interactions within time series, our model excels in its ability to acquire high-level representations.

F. Anomaly detection

Setups Anomaly detection aims to uncover anomalies in time series. We select five datasets, including SMD [38], MSL [39], SMAP [39], SWaT [40], and PSM [41], as benchmarks.

Results Table VI demonstrates the excellent performance of Caformer in anomaly detection. The existence of anomalies introduces biases during inference, just as the problem in imputation tasks. Subsequent research will be undertaken to address this issue.

G. Ablation study and Hyper-parameter sensitivity

1) Ablation study: To examine the effectiveness of each component in Caformer, we perform ablation experiments with the following variants: a) **w/o Dep**, which omits Dependency Learner for interactions among dependencies. b) **w/o Env**, which excludes Dynamic Learner for environment information. c) **w/o Dyn**, which does not utilize Dynamic Learner for dynamic interactions related to cross-dimension dependency. The results are shown in Table VII.

2) Hyper-parameter sensitivity: To assess the hyperparameter sensitivity of Caformer, we perform experiments with varying model parameters, including the patch size and stride, the number of stacked blocks, and the shape of the aligning matrix. The detailed results are provided below.

Results of different patch sizes and strides. To verify the impact of the patch size and the stride, we perform experiments

TABLE V

IMPUTATION TASK. WE RANDOMLY MASK {12.5%, 25%, 37.5%, 50%} OF TIME POINTS IN THE LENGTH-96 TIME SERIES. THE RESULTS ARE AVERAGED FROM 4 DIFFERENT MASK RATIOS. A LOWER MSE OR MAE INDICATES A BETTER PERFORMANCE. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND BEST ARE UNDERLINED.

Models		Ours		iTransformer (2024)		PatchTST (2023)		TimesNet (2023)		DLinear (2023)		Crossformer (2023)		LightTS (2022)		ETSformer (2022)		FEDformer (2022)	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	12.5%	0.073	0.168	0.097	0.219	0.091	0.199	**0.057**	**0.159**	0.151	0.267	0.170	0.284	0.240	0.345	0.126	0.263	0.070	0.190
	25%	0.083	0.192	0.125	0.249	0.105	0.215	**0.069**	**0.178**	0.180	0.292	0.188	0.300	0.265	0.364	0.169	0.304	0.106	0.236
	37.5%	0.104	0.211	0.158	0.280	0.122	0.232	**0.084**	**0.196**	0.215	0.318	0.210	0.318	0.296	0.382	0.220	0.347	0.124	0.258
	50%	0.117	0.223	0.225	0.338	0.143	0.248	**0.102**	**0.215**	0.257	0.347	0.243	0.342	0.334	0.404	0.293	0.402	0.165	0.299
	Avg	0.094	0.198	0.151	0.272	0.115	0.224	**0.078**	**0.187**	0.201	0.306	0.203	0.311	0.284	0.373	0.202	0.329	0.117	0.246
ETTh2	12.5%	0.050	0.156	0.094	0.209	0.055	0.149	**0.040**	**0.130**	0.100	0.216	0.135	0.242	0.101	0.231	0.187	0.319	0.095	0.212
	25%	0.056	0.147	0.119	0.237	0.061	0.158	**0.046**	**0.141**	0.127	0.247	0.158	0.262	0.115	0.246	0.279	0.390	0.137	0.258
	37.5%	0.061	0.154	0.148	0.265	0.066	0.166	**0.052**	**0.151**	0.158	0.276	0.176	0.279	0.126	0.257	0.400	0.465	0.187	0.304
	50%	0.065	0.161	0.194	0.303	0.073	0.174	**0.060**	**0.162**	0.183	0.299	0.208	0.306	0.136	0.268	0.602	0.572	0.232	0.341
	Avg	0.058	0.155	0.139	0.254	0.064	0.162	**0.049**	**0.146**	0.142	0.259	0.169	0.272	0.119	0.250	0.367	0.436	0.163	0.279
ETTm1	12.5%	**0.013**	**0.081**	0.045	0.147	0.019	0.147	0.019	0.092	0.058	0.162	0.051	0.158	0.075	0.180	0.067	0.188	0.035	0.135
	25%	**0.018**	**0.091**	0.060	0.171	0.060	0.171	0.023	0.101	0.080	0.193	0.048	0.141	0.093	0.206	0.096	0.229	0.052	0.166
	37.5%	**0.022**	**0.099**	0.077	0.195	0.076	0.194	0.029	0.111	0.103	0.219	0.059	0.170	0.113	0.231	0.133	0.271	0.069	0.191
	50%	**0.028**	**0.114**	0.104	0.229	0.102	0.226	0.036	0.124	0.132	0.248	0.067	0.181	0.134	0.255	0.186	0.323	0.089	0.218
	Avg	**0.020**	**0.096**	0.072	0.186	0.064	0.185	0.027	0.107	0.093	0.206	0.056	0.163	0.104	0.218	0.120	0.253	0.062	0.177
ETTm2	12.5%	0.025	0.093	0.052	0.152	0.051	0.151	**0.018**	**0.080**	0.062	0.166	0.025	0.092	0.034	0.127	0.108	0.239	0.056	0.159
	25%	0.027	0.098	0.070	0.179	0.124	0.248	**0.020**	**0.085**	0.085	0.196	0.086	0.193	0.042	0.143	0.164	0.294	0.080	0.195
	37.5%	0.030	0.104	0.091	0.203	0.157	0.280	**0.023**	**0.091**	0.106	0.222	0.091	0.198	0.051	0.159	0.237	0.356	0.110	0.231
	50%	0.033	0.110	0.116	0.231	0.214	0.329	**0.026**	**0.098**	0.131	0.247	0.097	0.204	0.059	0.174	0.323	0.421	0.156	0.276
	Avg	0.029	0.101	0.082	0.191	0.137	0.252	**0.022**	**0.088**	0.096	0.208	0.075	0.172	0.046	0.151	0.208	0.327	0.101	0.215
Electricity	12.5%	**0.058**	**0.167**	0.073	0.190	0.072	0.189	0.085	0.202	0.092	0.214	0.079	0.199	0.102	0.229	0.196	0.321	0.107	0.237
	25%	**0.068**	**0.182**	0.090	0.214	0.090	0.203	0.089	0.206	0.118	0.247	0.092	0.217	0.121	0.252	0.207	0.332	0.120	0.251
	37.5%	**0.080**	**0.197**	0.106	0.234	0.106	0.234	0.094	0.213	0.144	0.276	0.103	0.230	0.141	0.273	0.219	0.344	0.136	0.266
	50%	**0.097**	**0.210**	0.126	0.256	0.127	0.257	0.100	0.221	0.175	0.305	0.115	0.244	0.160	0.293	0.235	0.357	0.158	0.284
	Avg	**0.076**	**0.189**	0.988	0.224	0.099	0.221	0.092	0.210	0.132	0.260	0.097	0.223	0.131	0.262	0.214	0.339	0.130	0.259
Weather	12.5%	**0.020**	**0.033**	0.038	0.086	0.029	0.049	0.025	0.045	0.039	0.084	0.039	0.095	0.047	0.101	0.057	0.141	0.041	0.107
	25%	**0.025**	**0.041**	0.046	0.105	0.031	0.053	0.029	0.052	0.048	0.103	0.042	0.102	0.052	0.111	0.065	0.155	0.064	0.163
	37.5%	**0.028**	**0.047**	0.056	0.123	0.034	0.058	0.031	0.057	0.057	0.121	0.045	0.106	0.058	0.121	0.081	0.180	0.107	0.229
	50%	**0.031**	**0.053**	0.069	0.143	0.039	0.066	0.034	0.062	0.066	0.134	0.047	0.112	0.065	0.133	0.102	0.207	0.183	0.312
	Avg	**0.026**	**0.043**	0.052	0.114	0.033	0.057	0.030	0.054	0.052	0.110	0.043	0.104	0.055	0.117	0.076	0.171	0.099	0.203

TABLE VI

ANOMALY DETECTION TASK. A HIGHER VALUE OF PRECISION, RECALL, AND F1-SCORE INDICATES A BETTER PERFORMANCE. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND BEST ARE UNDERLINED.

Models		Ours	iTransformer (2024)	PatchTST (2023)	TimesNet (2023)	DLinear (2023)	Crossformer (2023)	LightTS (2022)	ETSformer (2022)	FEDformer (2022)
SMD	Precision	90.21	76.28	87.5	88.66	83.62	87.44	87.10	87.44	87.95
	Recall	83.18	85.57	82.2	83.14	71.52	59.10	78.42	79.23	82.39
	F1-score	**86.55**	80.66	84.7	85.81	77.10	70.53	82.53	83.13	85.08
MSL	Precision	89.58	86.15	88.34	83.92	84.34	90.33	82.40	85.13	77.14
	Recall	78.56	62.63	70.92	86.42	85.42	72.78	75.78	84.93	80.07
	F1-score	83.71	72.53	78.68	**85.15**	84.88	80.69	78.95	85.03	78.57
SMAP	Precision	92.54	90.68	90.41	92.52	92.32	89.91	92.58	92.25	90.47
	Recall	62.45	52.77	55.61	58.29	55.41	55.42	55.27	55.75	58.10
	F1-score	**74.57**	66.71	68.86	71.52	69.26	68.57	69.21	69.50	70.76
SWAT	Precision	94.65	92.21	90.95	86.76	80.91	97.87	91.98	90.02	90.17
	Recall	89.36	93.08	79.74	97.32	95.30	84.44	94.72	80.36	96.42
	F1-score	91.93	92.64	84.98	91.74	87.52	90.66	**93.33**	84.91	93.19
PSM	Precision	98.82	97.96	98.51	98.19	98.28	98.64	98.37	99.31	97.31
	Recall	95.41	92.10	93.44	96.76	89.26	94.66	95.97	85.28	97.16
	F1-score	97.09	94.94	96.11	97.47	93.55	96.61	97.15	91.76	**97.23**
Average F1-score		**86.77**	81.49	82.66	86.34	82.46	81.41	84.23	82.87	84.97

TABLE VII
ABLATION OF DIFFERENT COMPONENTS IN CAFORMER IN LONG-TERM FORECASTING TASK. THE RESULTS ARE AVERAGED FROM DIFFERENT INPUT LENGTHS.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Electricity		Weather		ILI		Traffic		Exchange	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
w/o Dep	0.443	0.438	0.383	0.406	0.386	0.401	0.281	0.324	0.199	0.283	0.251	0.278	2.210	0.916	0.536	0.356	0.367	0.400
w/o Dyn	0.431	0.437	0.375	0.409	0.383	0.409	0.291	0.324	0.175	0.312	0.265	0.283	2.145	0.920	0.532	0.348	0.365	0.401
w/o Env	0.436	0.438	0.378	0.405	0.385	0.396	0.281	0.324	0.198	0.282	0.251	0.278	2.116	0.917	0.551	0.358	0.367	0.401
Caformer	**0.424**	**0.428**	**0.367**	**0.399**	**0.374**	**0.393**	**0.276**	**0.316**	**0.169**	**0.267**	**0.244**	**0.270**	**1.997**	**0.913**	**0.493**	**0.302**	**0.343**	**0.394**

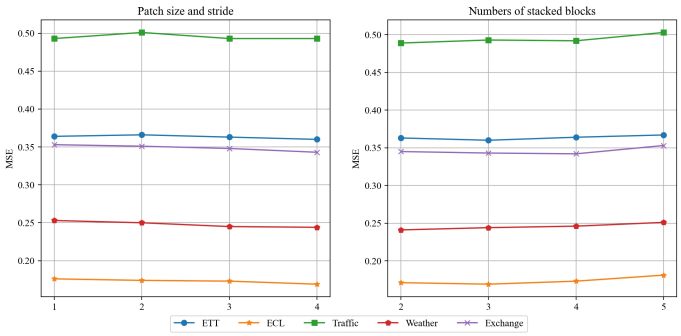

Fig. 6. Hyperparameter sensitivity with respect to the patch size and stride and the numbers of stacked block.

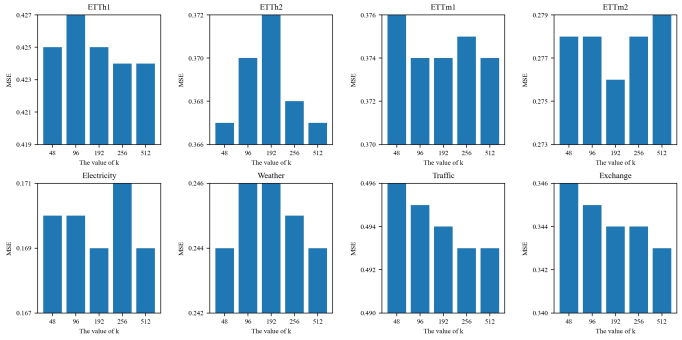

Fig. 7. Hyper-parameter sensitivity with respect to the shape of environment aligning matrix.

with four settings: (P = 4, S = 4), (P = 8, S = 4), (P = 8, S = 8), and (P = 16, S = 8). Results are shown in Fig. 6. It can be inferred that the performance is stable for different settings of the patch size and the stride, indicating the robustness of our model to these two parameters. Given that the optimal patch size and stride may vary across datasets, we conduct the setting as (P = 16, S = 8) in our model.

Results of different numbers of stacked blocks. To verify the model's sensitivity to different numbers of blocks, the numbers of stacked blocks are selected as 2, 3, 4, and 5 to make the comparison. Results can be found in Fig. 6. It can be inferred that the number of stacked blocks has an effect on performance, and it is not favored to be large concerning efficiency and performance. As a result, the number of stacked blocks is selected as 3 in our model.

Results of different shapes of aligning matrices. As stated above, $\mathbf{H}_e \in \mathbb{R}^{k\times k}$ is used as the environment aligning matrix to fuse the environmental factors, and its shape k is the pre-defined hyper-parameter determining the stratifying condition. To verify the impact of the shape k, we conduct experiments with $k \in \{48, 96, 192, 256, 512\}$. Results are shown in Fig. 7. It can be inferred that the performance doesn't vary significantly with different shapes of aligning matrix. This finding indicates the robustness of our model to this parameter. Moreover, it's worth noting that the shapes of aligning matrices may be influenced by the characteristics of the corresponding dataset. As a general recommendation, a shape of 256 for aligning matrices is considered a good choice for most datasets.

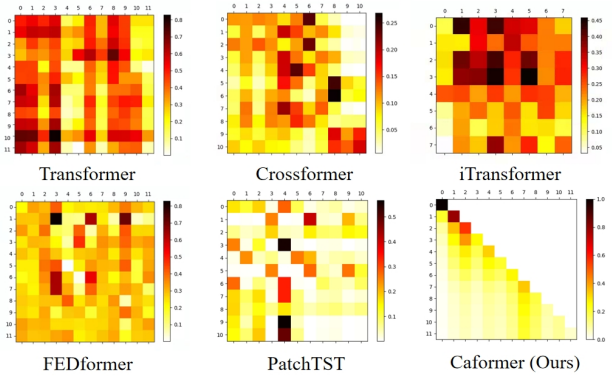

Fig. 8. Comparison of attention maps or alignment matrix obtained by different models. These visualizations of correlations within time series indicate models' capacity in capturing interactions.

H. Interpretability analysis

Transformer-based methods leverage the self-attention mechanism, facilitating the learning of long-range dependencies via query-key-value dot product attention. These dot product operations ascertain the significance of a particular token in relation to others, fostering information interactions and alignment between tokens. However, as illustrated in Fig. 8, attention maps inferred by prior methods often lack interpretability. This arises from the inherent nature of these methods, which learn alignment rules based on correlation, disregarding the fundamental causality. Indeed, self-attention mimics a content-based retrieval process, utilizing pairwise interactions grounded in statistical correlation. To enhance the comparability of individual methods in correlation learning,

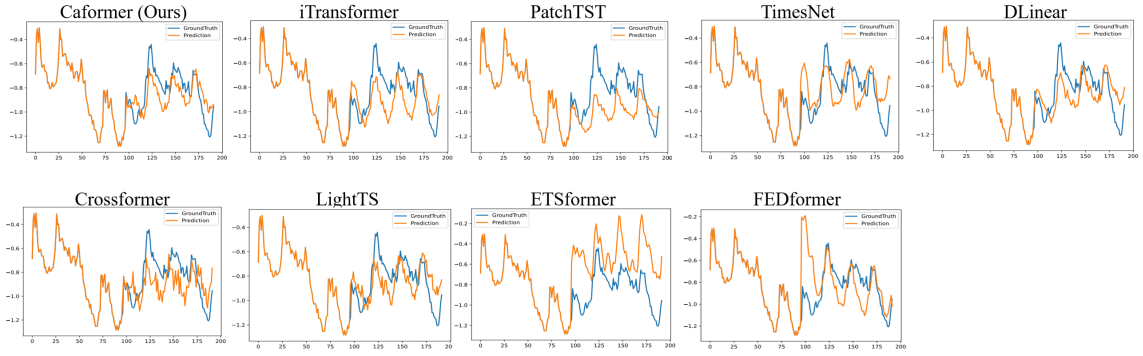

Fig. 9. The visualization of long term forecasting results with ETTh1 dataset by different models under the input-96-predict-96 setting. The blue lines stand for the ground truth and the orange lines stand for the prediction.

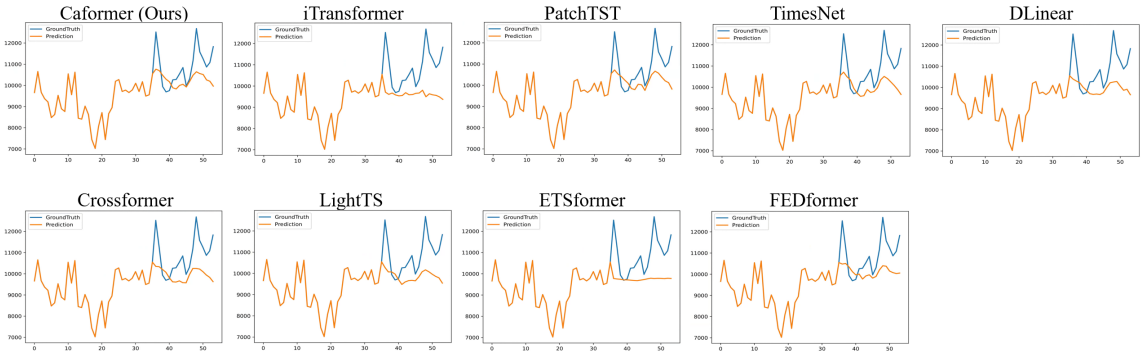

Fig. 10. The visualization of imputation results with Electricity dataset by different models under the 37.5% mask ratio setting. The blue lines stand for the ground truth and the orange lines stand for the prediction.

we standardize the alignment matrix to a value of 12 using the ETTh1 dataset. This is because Caformer computes the relationship between 12 time periods that can be considered to have the same underlying causal structure. For matrices exceeding dimensions of 12×12, we patch the original matrix and compute the mean value for each patch, representing the relationship between the region and its neighbors. As depicted in Fig. 8, Caformer exhibits the strongest correlation along the matrix diagonal, and this correlation is not uniformly 1. This consistency with causality elucidates the impact of external effects. Also, the time restriction map effectively guides learned alignments to adhere to Granger causality, ensuring causes precede their effects, resulting in a lower-triangular matrix.

I. Showcases

For a comprehensive comparison among different models, we present showcases for regression tasks, encompassing long-term forecasting as depicted in Fig. 9, short-term forecasting illustrated in Fig. 11, and imputation showcased in Fig. 10.

V. Conclusion and Future Works

In this paper, we address the challenges of learning cross-dimension and cross-time dependencies in the presence of the environment from a causal perspective. With a comprehensive understanding of the dynamic interactions within time series, our Caformer is able to eliminate the impact of the environment while obtaining causal relationships among cross-dimension and cross-time dependencies for robust temporal representations. Experimentally, our Caformer shows great generality and performance in five mainstream analysis tasks with proper interpretability. Future investigations will delve deeper into the interactions within the time series.

Appendix
Derivation of Back-door Adjustment

We use the back-door adjustment [32] for causal intervention. In this section, we first introduce the basic rules of do-calculus do$(\cdot)$, then we present the derivation of backdoor adjustment for the proposed causal graph in Fig. 3 based on these rules.

Rules of Do-calculus. Given an arbitrary causal directed acyclic graph G, there are three nodes respectively represented by X, Y, and Z. Particularly, $G_{\overline{X}}$ denotes the intervened causal graph with the removal of all incoming arrows leading to X, $G_{\underline{X}}$ denotes another intervened causal graph with the removal of all outgoing arrows from X. The lower cases x, y, and z are used to represent the respective values of nodes: $X = x$, $Y = y$, and $Z = z$. We illustrate the details of G, $G_{\overline{X}}$ and $G_{\underline{X}}$ in Fig. 12.

For any interventional distribution compatible with G, we

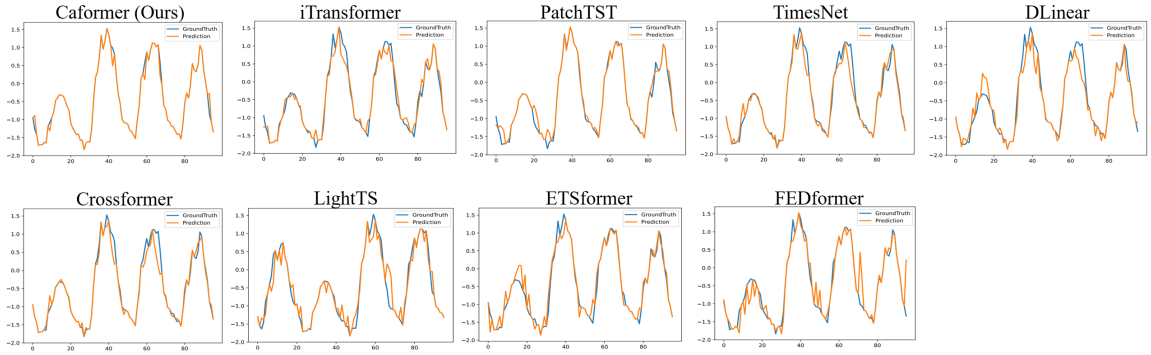

Fig. 11. The visualization of short term forecasting results with the M4 dataset by different models under the monthly setting. The blue lines stand for the ground truth and the orange lines stand for the prediction.

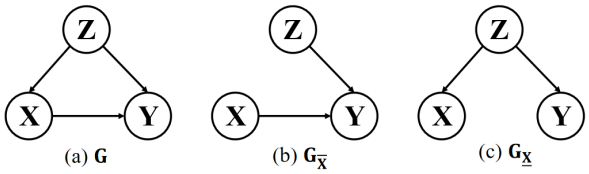

Fig. 12. Illustration of different causal directed acyclic graph

introduce the following three rules:

Rule 1. Insertion/deletion of observations:

$$P(y|\text{do}(x), z) = P(y|\text{do}(x)), \quad \text{if}(y \perp\!\!\!\perp z|x)_{G_{\overline{X}}}, \tag{18}$$

Rule 2. Action/observation exchange:

$$P(y|\text{do}(x), \text{do}(z)) = P(y|\text{do}(x), z), \quad \text{if}(y \perp\!\!\!\perp z|x)_{G_{\overline{X}\underline{Z}}}, \tag{19}$$

Rule 3. Insertion/deletion of actions:

$$P(y|\text{do}(x), \text{do}(z)) = P(y|\text{do}(x)), \quad \text{if}(y \perp\!\!\!\perp z|x)_{G_{\overline{XZ}}}, \tag{20}$$

where $(y \perp\!\!\!\perp z|x)_{G_{\overline{X}}}$ means that y and z are independent of each other given x in G. Based on these rules, we can derive the interventional distribution $P(Y|\text{do}(T))$ for our proposed causal graph.

$$P(T|\text{do}(X)) \tag{21}$$
$$= \sum_c P(T|\text{do}(X), C = c)P(C = c|\text{do}(X)) \tag{22}$$
$$= \sum_c P(T|\text{do}(X), C = c)P(C = c) \tag{23}$$
$$= \sum_c P(T|X, C = c)P(C = c) \tag{24}$$

where (A.4) follows the law of total probability, (A.5) is obtained via Rule 3 given cX in $G_{\overline{X}}$, and (A.6) can be inferred from Rule 2 which changes the intervention term into observation as $TX|c$ in $G_{\underline{X}}$.

By stratifying C into discrete components $C = \{c_i\}_{i=1}^n$, we can finally express $P(T|\text{do}(X))$ as follows:

$$P(T|\text{do}(X)) = \sum_i^n P(T|X, C = c_i)P(C = c_i) \tag{25}$$

REFERENCES

[1] Y. Liu, T. Hu, H. Zhang, H. Wu, S. Wang, L. Ma, and M. Long, "itransformer: Inverted transformers are effective for time series forecasting," *arXiv preprint arXiv:2310.06625*, 2023.

[2] N. Passalis, A. Tefas, J. Kanniainen, M. Gabbouj, and A. Iosifidis, "Deep adaptive input normalization for time series forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 9, pp. 3760–3765, 2020.

[3] W. Zheng and J. Hu, "Multivariate time series prediction based on temporal change information learning method," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 10, pp. 7034–7048, 2023.

[4] Z. Wu, D. Zheng, S. Pan, Q. Gan, G. Long, and G. Karypis, "Traversenet: Unifying space and time in message passing for traffic forecasting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 2, pp. 2003–2013, 2024.

[5] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," in *The Eleventh International Conference on Learning Representations*, 2023.

[6] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *International Conference on Learning Representations*, 2022.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[8] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun, "Transformers in time series: A survey," *arXiv preprint arXiv:2202.07125*, 2022.

[9] Y. Zhang and J. Yan, "Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting," in *The Eleventh International Conference on Learning Representations*, 2023.

[10] Y. Liu, H. Wu, J. Wang, and M. Long, "Non-stationary transformers: Exploring the stationarity in time series forecasting," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 9881–9893.

[11] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 22 419–22 430.

[12] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *The Eleventh International Conference on Learning Representations*, 2023.

[13] F. M. Bianchi, S. Scardapane, S. Lkse, and R. Jenssen, "Reservoir computing approaches for representation and classification of multivariate time series," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2169–2179, 2021.

[14] V. Ekambaram, A. Jati, N. Nguyen, P. Sinthong, and J. Kalagnanam, "Tsmixer: Lightweight mlp-mixer model for multivariate time series forecasting," *arXiv preprint arXiv:2306.09364*, 2023.

[15] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on Artificial Intelligence*, vol. 37, no. 9, June 2023, pp. 11 121–11 128.

[16] K. Zhang, C. Li, and Q. Yang, "Trid-mae: A generic pre-trained model for multivariate time series with missing values," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. New York, NY, USA: Association for Computing Machinery, 2023, p. 31643173.

[17] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the 39th International Conference on Machine Learning*, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. PMLR, July 2022, pp. 27 268–27 286.

[18] S. Yang, M. Dong, Y. Wang, and C. Xu, "Adversarial recurrent time series imputation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 1639–1650, 2023.

[19] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song, "Gated transformer networks for multivariate time series classification," *arXiv preprint arXiv:2103.14438*, 2021.

[20] R. Zuo, G. Li, B. Choi, S. S Bhowmick, D. N.-y. Mah, and G. L. Wong, "Svp-t: A shape-level variable-position transformer for multivariate time series classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, June 2023, pp. 11 497–11 505.

[21] Y. Zheng, H. Y. Koh, M. Jin, L. Chi, K. T. Phan, S. Pan, Y.-P. P. Chen, and W. Xiang, "Correlation-aware spatialtemporal graph learning for multivariate time-series anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[22] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," vol. 35, no. 12, May 2021, pp. 11 106–11 115.

[23] M. Barahona and C.-S. Poon, "Detection of nonlinear dynamics in short, noisy time series," *Nature*, vol. 381, no. 6579, pp. 215–217, 1996.

[24] M. Zečević, D. S. Dhami, P. Veličković, and K. Kersting, "Relating graph neural networks to structural causal models," *arXiv preprint arXiv:2109.04173*, 2021.

[25] S. Löwe, D. Madras, R. Zemel, and M. Welling, "Amortized causal discovery: Learning to infer causal graphs from time-series data," in *Proceedings of the First Conference on Causal Learning and Reasoning*, ser. Proceedings of Machine Learning Research, vol. 177. PMLR, April 2022, pp. 509–525.

[26] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf, "Recurrent independent mechanisms," in *International Conference on Learning Representations*, 2021.

[27] J. Runge, A. Gerhardus, G. Varando, V. Eyring, and G. Camps-Valls, "Causal inference for time series," *Nature Reviews Earth & Environment*, vol. 4, no. 7, pp. 487–505, July 2023.

[28] D. Zhang, H. Zhang, J. Tang, X.-S. Hua, and Q. Sun, "Causal intervention for weakly-supervised semantic segmentation," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 655–666.

[29] E. D. Abraham, K. D'Oosterlinck, A. Feder, Y. Gat, A. Geiger, C. Potts, R. Reichart, and Z. Wu, "Cebab: Estimating the causal effects of real-world concepts on nlp model behavior," in *Advances in Neural Information Processing Systems*, vol. 35. Curran Associates, Inc., 2022, pp. 17 582–17 596.

[30] L. Castri, S. Mghames, and N. Bellotto, "From continual learning to causal discovery in robotics," in *Proceedings of The First AAAI Bridge Program on Continual Causality*, ser. Proceedings of Machine Learning Research, vol. 208. PMLR, February 2023, pp. 85–91.

[31] B. D. Haig, "What is a spurious correlation?" *Understanding Statistics*, vol. 2, no. 2, pp. 125–132, 2003.

[32] J. Pearl, *Causality: Models, Reasoning and Inference*. New York: Cambridge University Press, 2000.

[33] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Etsformer: Exponential smoothing transformers for time-series forecasting," *arXiv preprint arXiv:2202.01381*, 2022.

[34] T. Zhang, Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, and J. Li, "Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures," *arXiv preprint arXiv:2207.01186*, 2022.

[35] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "The m4 competition: 100,000 time series and 61 forecasting methods," *International Journal of Forecasting*, vol. 36, no. 1, pp. 54–74, 2020, m4 Competition.

[36] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long- and short-term temporal patterns with deep neural networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. New York, NY, USA: Association for Computing Machinery, 2018, p. 95104.

[37] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The uea multivariate time series classification archive, 2018," *arXiv preprint arXiv:1811.00075*, 2018.

[38] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2019, p. 28282837.

[39] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2018, p. 387395.

[40] A. P. Mathur and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ics security," in *2016 International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE, 2016, pp. 31–36.

[41] A. Abdulaal, Z. Liu, and T. Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. New York, NY, USA: Association for Computing Machinery, 2021, p. 24852494.