Weakly Guided Adaptation for Robust Time Series Forecasting

Cheng, Yunyao; Chen, Peng; Guo, Chenjuan; Zhao, Kai; Wen, Qingsong; Yang, Bin; Jensen, Christian S.

Published in:
Proceedings of the VLDB Endowment

DOI (link to publication from Publisher):
10.14778/3636218.3636231

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

Weakly Guided Adaptation for Robust Time Series Forecasting

Yunyao Cheng
Aalborg University
yunyaoc@cs.aau.dk

Peng Chen
East China Normal University
pchen@stu.ecnu.edu.cn

Chenjuan Guo*
East China Normal University
cjguo@dase.ecnu.edu.cn

Kai Zhao
Aalborg University
kaiz@cs.aau.dk

Qingsong Wen
Alibaba Group
qingsongedu@gmail.com

Bin Yang
East China Normal University
byang@dase.ecnu.edu.cn

Christian S. Jensen
Aalborg University
csj@cs.aau.dk

ABSTRACT

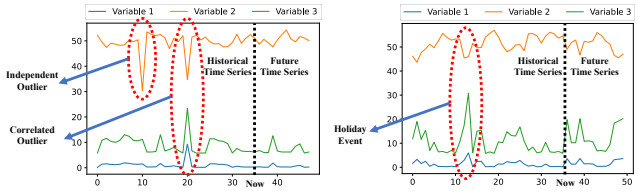
Robust multivariate time series forecasting is crucial in many cyber-physical and Internet of Things applications. Existing state-of-the-art robust forecasting models decompose time series into independent functions covering trends and periodicities. However, these independent functions fail to capture correlations among multiple time series, thereby reducing prediction accuracy. Moreover, existing robust forecasting models treat certain abrupt but normal changes, e.g., caused by holidays, as outliers because they occur infrequently and have data distributions that resemble those of outliers. This exacerbates model bias and reduces prediction accuracy. This paper aims to capture correlations across multiple time series and abrupt but normal changes, thereby improving prediction accuracy. We employ weak labels to partition the dataset into source and target domains. Then, we propose the Domain Adversarial Robust Forecaster (DARF). This forecasting model is based on adversarial domain adaptation and includes two novel modules: Correlated Robust Forecaster (CORF) and Domain Critic. Specifically, CORF constitutes an encoder-decoder framework proficient at robust multivariate time series forecasting, and Domain Critic works to reduce data bias. Extensive experiments and discussions show that DARF is capable of state-of-the-art forecasting accuracy.

PVLDB Reference Format:
Yunyao Cheng, Peng Chen, Chenjuan Guo, Kai Zhao,Qingsong Wen, Bin Yang, Christian S. Jensen Weakly Guided Adaptation for Robust Time Series Forecasting. PVLDB, 17(4): 766 - 779, 2023.
doi:10.14778/3636218.3636231

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/YunyaoCheng/DARF.

1 INTRODUCTION

The ongoing, rapid deployment of the Internet of Things and of cyber-physical and cloud monitoring systems results in the generation of massive volumes of time series data that hold the potential to enable a broad range of applications [9, 11, 12, 23, 36, 50]. A multivariate time series is a collection of correlated, time-aligned time series. For instance, in power transmission, an increase in the temperature of metal conductors is accompanied by an elevation in their electrical resistance. Thus, variables in a multivariate time series are often inherently correlated. Empowering a model to capture correlations across multiple time series can significantly enhance prediction accuracy for each variable, which in turn benefits many applications. Fig. 1 depicts a real-world multivariate time series composed of three variables. A forecasting model is trained on historical time series and is then employed to predict future values, demarcated by the black dashed line.



(a) Time Series with Outliers. (b) Time Series with Events.

Figure 1: Illustration of time series with outliers, events, and abrupt changes.

The prediction accuracy is influenced by factors such as the model's ability to capture correlations and complex temporal dynamics and its robustness to outliers. Specifically, the multivariate time series in Figure 1(a) has independent and correlated outliers. Independent outliers are outliers that appear in a single time series at a time point due to, e.g., machine malfunction of one machine in a cluster, while correlated outliers are outliers that appear across multiple time series at a time point. In practical scenarios, temporal patterns with limited samples, e.g., resulting from holidays, human behavior, or other predictable yet infrequent events, may resemble outliers; see Fig. 1(b). This resemblance can cause a forecasting model to mistake temporal dynamics due to regular events for outliers, causing the model to disregard infrequent but important temporal features.

This paper studies the problem of robust forecasting for multivariate time series, i.e., how to predict future time series based on historical data while being robust to outliers in the historical data that would otherwise reduce prediction accuracy. It is crucial to ensure that event features are not treated as outliers. By developing a

*: Corresponding author.

Proceedings of the VLDB Endowment, Vol. 17, No. 4 ISSN 2150-8097.
doi:10.14778/3636218.3636231

model with these capabilities, we enable more accurate forecasting of future time series.

State-of-the-art methods for robust multivariate time series forecasting often utilize seasonal-trend decomposition (STD) [34, 39, 48, 52]. These methods usually decompose input time series into multiple trend and seasonality functions, effectively hypothesizing that the time series consists of trend and seasonality backbones. A backbone encompasses prior knowledge of both trends and seasonal patterns, thereby enhancing robustness to overfitting caused by outliers.

Although existing solutions for robust time series forecasting represent important progress, unresolved problems remain that significantly influence the accuracy of robust forecasting.

Challenge 1: It is challenging for existing forecasting methods to handle multivariate time series with outliers. Seasonal-trend decomposition-based methods fail to capture temporal dynamics among multivariate time series because these methods decompose each time series or variable independently. Graph neural networks (GNN) [30, 42, 43], a prevalent approach to leveraging correlations in multivariate time series forecasting, suffer from limited robustness. Specifically, when multiple time series exhibit outliers simultaneously, a GNN, serving as a propagator of correlations, intensifies the biases induced by these outliers. This phenomenon, as shown in Fig. 1(a), is referred to as a "correlated outlier."

Challenge 2: Distinguishing event features from outliers can be challenging. Existing methods, including STD- and GNN-based approaches, employ statistical machine-learning frameworks. These methods are data-driven and rely on large numbers of time series samples to capture features. Consequently, these models struggle to capture event features when the numbers of samples that belong to events are limited. Additionally, since the patterns of events and outliers are similar, it is generally difficult for these models to accurately fit event features while disregarding outliers.

To overcome these challenges, we propose an adversarial domain adaptation framework called Domain Adversarial Robust Forecaster (DARF). More specifically, DARF addresses the two challenges as follows.

Addressing challenge 1: To ensure that DARF can capture correlations across multiple time series while remaining robust to outliers, we equip DARF with a correlated robust forecaster (CORF), which is an asymmetric encoder-decoder framework. CORF's encoder is a graph convolutional network that is designed to extract correlations among multiple time series and produce correlated features. CORF's decoder is a seasonal-trend decomposition-based component that decomposes the correlated features into seasonal and trend backbones to achieve robustness to outliers.

Addressing challenge 2: Distinguishing event features from outliers is crucial, as it ensures model robustness while enhancing prediction accuracy. However, labeling individual outliers is time-consuming and expensive. To overcome this, we integrate weak supervision into DARF, employing readily available event labels, such as public holidays. We assign time series without weak labels to a source domain and assign those with weak labels to a target domain. Then, the adversarial training ensures that the model captures event features effectively. By minimizing the distribution divergence between the source and target domains, the model aligns the distribution of time series where events occurred with that of time series without events. This approach enables the model to simultaneously capture event features and retain robustness to outliers.

In summary, it is crucial to enable robust forecasting that can capture correlations in multivariate time series. Moreover, capturing event features is essential both theoretically and practically to improve prediction accuracy. We make three main contributions:

- We propose a novel asymmetric encoder-decoder framework, called CORF, that comprises a GNN-based encoder designed to capture correlations among multiple variables and an STD-based decoder designed to ensure robustness.
- We introduce a novel adversarial domain adaptation framework to handle event features, leveraging weakly labeled data to distinguish source and target domains.
- We conduct extensive experiments on public real-world data sets that demonstrate state-of-the-art prediction accuracy. The experimental results show that the proposed DARF can effectively capture event information and improves robustness to outliers.

The remainder of the paper is organized as follows. Section 2 covers preliminaries. Section 3 provides details the methodology. Section 4 reports on the experimental study. Section 5 reviews related works and Section 6 concludes.

2 PRELIMINARIES

2.1 Multivariate Time Series Forecasting

We denote an observation in a multivariate time series by $\mathbf{x}_t \in \mathbb{R}^N$, where t is a timestamp and N indicates the number of variables in the observation. A model $\mathcal{F}$ uses a historical series $\langle \mathbf{x}_{t-H+1}, \mathbf{x}_{t-H+2}, \ldots, \mathbf{x}_t \rangle$ to forecast a future series $\langle \hat{\mathbf{x}}_{t+1}, \hat{\mathbf{x}}_{t+2}, \ldots, \hat{\mathbf{x}}_{t+F} \rangle$, where H and F are the length of historical and future series, respectively. The forecasting procedure is formulated as follows.

$$\mathcal{F}_{\Phi}(\mathbf{x}_{t-H+1}, \mathbf{x}_{t-H+2}, ..., \mathbf{x}_t) = (\hat{\mathbf{x}}_{t+1}, \hat{\mathbf{x}}_{t+2}, ..., \hat{\mathbf{x}}_{t+F}), \tag{1}$$

where Φ represents the parameters learned from training.

2.2 Weak Labels

Weak labels, which can often be acquired in an easy and cost-effective manner, serve to provide extra, pertinent information with the purpose of improving forecasting performance. For time series data, weak labels often relate to semantics associated with timestamps. For example, a weak label may indicate whether a timestamp is associated with an event such as a holiday, peak hours, or a sporting event. A raw dataset $\mathcal{D} = \{(\mathbf{H}_i, \mathbf{F}_i)\}_{i=1}^{I}$ contains I multivariate time series. Here, for the i-th time series, $\mathbf{H}_i = \langle \mathbf{x}_{i,t-H+1}, \mathbf{x}_{i,t-H+2}, \ldots, \mathbf{x}_{i,t} \rangle$ denotes the historical series, while $\mathbf{F}_i = \langle \mathbf{x}_{i,t+1}, \mathbf{x}_{i,t+2}, \ldots, \mathbf{x}_{i,t+F} \rangle$ represents the ground truth of the future series. Weak labels are used to divide $\mathcal{D}$ into a source domain, $\mathcal{D}_{\mathcal{S}} = \{(\mathbf{H}_i, \mathbf{F}_i)\}_{i=1}^{I_{\mathcal{S}}}$, where the corresponding weak labels indicate that no events occur, and a target domain, $\mathcal{D}_{\mathcal{T}}\{(\mathbf{H}_j, \mathbf{F}_j)\}_{j=1}^{I_{\mathcal{T}}}$, where the corresponding weak labels indicate the occurrences of events. For example, when we use weak labels indicating holidays, the source domain contains the time series from non-holidays, whereas the target domain contains the time series from holidays. We have $I_{\mathcal{S}} + I_{\mathcal{T}} = I$, and we often have $I_{\mathcal{S}} > I_{\mathcal{T}}$, i.e., the source domain has

more samples than does the target domain. The proposed model is trained by the divided source and target domains.

2.3 Self-adaptive Graph Learning

A graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ is comprised of a set of nodes $\mathcal{V}$, a set of edges $\mathcal{E}$, and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ of weights that capture the correlations among the N nodes. If $v_i, v_j \in \mathcal{V}$ and $(v_i, v_j) \in \mathcal{E}$, the non-zero value of $\mathbf{A}_{i,j}$ denotes the weight of edge (v_i, v_j); otherwise, the value is zero. In time series forecasting, the nodes represent variables, and $\mathbf{A}$ captures the correlations among these variables. Instead of using a traditional static graph with a structure predefined by domain experts, we employ a self-adaptive graph learning component [42]. This is because the correlations among variables in multivariate time series are often hidden. The self-adaptive adjacency matrix $\mathbf{A}$ is learned during training. The learning procedure is defined as follows.

$$\begin{aligned} \mathbf{M}_1 &= \tanh(\alpha \mathbf{E}_1 \mathbf{\Theta}_{emb1}) \\ \mathbf{M}_2 &= \tanh(\alpha \mathbf{E}_2 \mathbf{\Theta}_{emb2}) \\ \mathbf{A} &= \mathrm{ReLU}(\tanh(\mathbf{M}_1 \mathbf{M}_2^{\mathrm{T}} - \mathbf{M}_2 \mathbf{M}_1^{\mathrm{T}})), \end{aligned} \tag{2}$$

where $\mathbf{E}_1$ and $\mathbf{E}_2$ denote randomly initialized variable embeddings, $\mathbf{\Theta}_{emb1}$ and $\mathbf{\Theta}_{emb2}$ are learnable parameters, $\tanh(\cdot)$ is the hyperbolic tangent function, $\mathrm{ReLU}(\cdot)$ is the rectified linear unit activation function.

3 METHODOLOGY

3.1 Framework of DARF

In this section, we provide the details of the DARF framework. Furthermore, we explain how DARF utilizes weak labels to establish the source and target domains, and how it constructs an adversarial domain adaptation for time series forecasting.

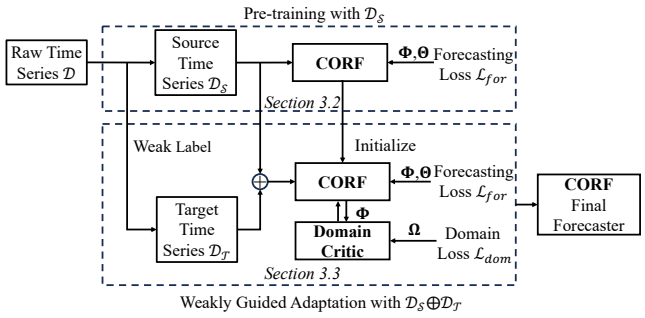


Figure 2: The DARF Framework. DARF employs weak labels to partition the source and target domains. It then pre-trains CORF using the source domain, followed by adversarial training of DARF using both domains. The trained CORF serves as the final forecaster.

An overview of DARF is shown in Fig. 2. We first utilize weak labels to divide raw dataset $\mathcal{D} = \{(\mathbf{H}_i, \mathbf{F}_i)\}_{i=1}^{I}$ into source domain $\mathcal{D}_\mathcal{S} = \{(\mathbf{H}_i, \mathbf{F}_i)\}_{i=1}^{I_\mathcal{S}}$ and target domain $\mathcal{D}_\mathcal{T}\{(\mathbf{H}_j, \mathbf{F}_j)\}_{j=1}^{I_\mathcal{T}}$. In this study, we assign any time series that includes holiday labels to the target domain and assign the remaining time series to the source domain.

Then, DARF uses the source domain $\mathcal{D}_\mathcal{S}$ to pre-train its encoder-decoder framework, CORF. Prior to adversarial training, we pre-train the model using source domain $\mathcal{D}_\mathcal{S}$ to establish the distribution of model parameters within the latent space. This facilitates the convergence of model parameters to the distribution of the target domain $\mathcal{D}_\mathcal{T}$ during adversarial training. At this point, the goal of CORF is to minimize the forecasting loss, $\mathcal{L}_{for}$, which is defined as follows.

$$\mathcal{L}_{for}(\mathcal{D}_\mathcal{S}; \mathcal{G}, \mathcal{F}) = \frac{1}{I_\mathcal{S}} \sum_{i=1}^{I_\mathcal{S}} l_{for}(\mathcal{F}(\mathcal{G}(\mathbf{H}_i)), \mathbf{F}_i), \tag{3}$$

where $\mathcal{G}(\cdot)$ represents the CORF encoder, $\mathcal{F}(\cdot)$ indicates the CORF decoder, and $l_{for}(\cdot)$ denotes the loss function for the CORF decoder. Here, the CORF encoder, a GNN, is leveraged to capture correlations, while the CORF decoder, a forecasting model grounded in STN, ensures model robustness. The parameters being optimized are $\mathbf{\Theta}$ and $\mathbf{\Phi}$, which belong to the CORF encoder and the CORF decoder, respectively. The detailed procedure is covered in Section 3.2.

Once CORF completes pre-training, we initiate the weakly guided adaptation for both the source and target domains. As illustrated in Fig. 2, the symbol $\oplus$ denotes the merging of sampling from the source and target domains, drawing samples from both source and target domains simultaneously for training on an equal scale. For instance, if DARF draws ten samples from the source domain, it will extract ten samples from the target domain. The rationale is that there are fewer in the target domain than in the source domain. By balancing the sampling scale, the model balances the loss between the target and source domains during training, thereby enhancing the model's capacity to capture event features. Subsequently, the learned parameters $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are utilized to initialize CORF's parameters. Given our adoption of an adversarial learning strategy, the CORF encoder serves as the generator within the adversarial network, while the Domain Critic acts as the discriminator. The Domain Critic is a neural network designed to reduce the distribution divergence between the source and target domains. Thus, the Domain Critic tries to maximize the domain loss $\mathcal{L}_{dom}$ to reduce the distance between the source and target domains. When the domain loss is larger, it becomes more difficult for the Domain Critic to distinguish whether a feature belongs to the source or the target domain, indicating that their distributions are getting closer. The domain loss $\mathcal{L}_{dom}$ is defined as follows.

$$\begin{aligned} &\mathcal{L}_{dom}(\mathcal{D}_\mathcal{S}, \mathcal{D}_\mathcal{T}; C, \mathcal{G}) = \\ &\qquad l_{dom}\left(\frac{1}{I_\mathcal{S}} \sum_{i=1}^{I_\mathcal{S}} C(\mathcal{G}(\mathbf{H}_i)), \frac{1}{I_\mathcal{T}} \sum_{j=1}^{I_\mathcal{T}} C(\mathcal{G}(\mathbf{H}_j))\right), \end{aligned} \tag{4}$$

where $C(\cdot)$ denotes the Domain Critic and $l_{dom}(\cdot)$ represents the metric to determine the distribution divergence. A set of parameters being optimized are $\mathbf{\Omega}$ which belong to the Domain Critic.

The adversarial manner necessitates alternating training between the CORF encoder and the Domain Critic. Therefore, the comprehensive optimization objective of weakly guided adaptation

is defined as follows.

$$\begin{aligned}\min_{\Theta,\Phi}\max_{\Omega}\mathcal{L}(\mathcal{D};\mathcal{G},\mathcal{F}) = &\min_{\Theta,\Phi}\max_{\Omega}\mathcal{L}_{for}(\mathcal{D}_{\mathcal{S}};\mathcal{G},\mathcal{F})\\ &+\mathcal{L}_{for}(\mathcal{D}_{\mathcal{T}};\mathcal{G},\mathcal{F})\\ &-\mathcal{L}_{dom}(\mathcal{D}_{\mathcal{S}},\mathcal{D}_{\mathcal{T}};C,\mathcal{G})\end{aligned} \tag{5}$$

Here, we implement adversarial domain adaptation in time series forecasting by minimizing the forecasting loss $\mathcal{L}_{for}$ and maximizing the domain loss $\mathcal{L}_{dom}$. The detailed procedure is presented in Section 3.3.

DARF repeatedly carries out the weakly guided adaptation until Θ, Φ, and Ω converge. It retains the model structure of CORF along with its encoder and decoder parameters, Θ and Φ. At this moment, CORF serves as the final forecaster.

3.2 CORF Pre-training

In this section, we delve into the principles of CORF, elucidating how its encoder captures correlations within multivariate time series, and how its decoder decomposes time series to ensure robustness to outliers. To lay the groundwork for the weakly guided adaptation in reducing the distribution divergence between source and target domains, we detail how to pre-train CORF on the source domain, facilitating the convergence of CORF parameters in the latent space.

3.2.1 CORF Encoder. Correlation plays a pivotal role in multivariate time series forecasting, as it uncovers relationships and dependencies among various time series. However, traditional robust forecasting frameworks primarily concentrate on enhancing the model's robustness to outliers via decomposing time series individually, which hinders their ability to effectively utilize the correlations.

As illustrated in Fig. 3, the CORF encoder processes the input historical time series $\mathbf{H}_i = \langle \mathbf{x}_{i,t-H+1}, \mathbf{x}_{i,t-H+2}, ..., \mathbf{x}_{i,t}\rangle$, and generates correlated features $\mathbf{G}_i = \langle \mathbf{g}_{i,1}, \mathbf{g}_{i,2}, ..., \mathbf{g}_{i,H}\rangle$. To capture short-term temporal dynamics, two gated 1-D convolutional neural networks (CNN-1 and CNN-2) [18] are utilized, which in turn assist the graph convolutional network in propagating information among neighboring time series via the learned adjacency matrix $\mathbf{A} = \sum_{k=1}^{K}\mathbf{A}^k$. This approach facilitates the extraction of correlations and the transformation of raw time series into representations within the latent space.

The CORF encoder employs a diffusion convolution layer [20] to establish these correlations, resulting in correlated features $\mathbf{G}_i$. The diffusion processes delineate the manner in which information propagates throughout the graph structure, capturing long-range dependencies and higher-order relationships among multiple time series. It is formulated as follows.

$$\mathbf{G}_i = \mathrm{MLP}(\sum_{k=1}^{K}\mathbf{A}^k\mathbf{X}'_i\mathbf{\Theta}_{dcl}), \tag{6}$$

where $\mathbf{G}_i$ denotes the output correlated features, $\mathbf{A}^k$ represents the k-th learned self-adaptive adjacency matrix obtained through Eq. 2, with K signifying the diffusion steps. $\mathbf{X}'_i$ refers to the input signals, while $\mathbf{\Theta}_{dcl}$ indicates the learned matrix. The $\mathrm{MLP}(\cdot)$ represents a 3-layer perceptron utilized to reshape the representation, ensuring the alignment of input features for the CORF decoder.

The input signal of diffusion convolution layer $\mathbf{X}'_i$ is generated by the gated 1-D CNNs, which are employed to construct representations containing short-term temporal information. The efficacy of the gating mechanism has been demonstrated to be advantageous in capturing temporal features [10]. The gated CNNs take input $\mathbf{H}_i = \langle 0, \mathbf{x}_{i,t-h+1}, \mathbf{x}_{i,t-h+2}, ..., \mathbf{x}_{i,t}\rangle$, where the padded 0 corresponds to PAD as illustrated in Fig. 3. The output of these gated CNNs, denoted as $\mathbf{X}'_i$, is formulated as follows.

$$\mathbf{X}'_i = \tanh(\mathbf{\Theta}_{cnn1}\mathbf{X}_i + bias_1) \odot \sigma(\mathbf{\Theta}_{cnn2}\mathbf{X}_i + bias_2), \tag{7}$$

where $\tanh(\cdot)$ denotes the hyperbolic tangent function, $\sigma(\cdot)$ represents the sigmoid function, $\odot$ indicates the Hadamard Product, and $\mathbf{\Theta}_{cnn1}$, $\mathbf{\Theta}_{cnn2}$, $bias_1$, and $bias_2$ correspond to learned parameters.

Here, we obtain the output of the CORF encoder, correlated features $\mathbf{G}_i$, which blend features from multiple time series. Because $\mathbf{G}_i$ passes through a diffusion convolution layer, which propagates features of multiple time series through adjacency matrices $\mathbf{A}$.

3.2.2 CORF Decoder. The CORF decoder is tasked with robustly predicting future time series. We design the decoder with two primary considerations in mind: capturing complex temporal dynamics effectively and decomposing seasonal-trend backbones. To achieve these objectives effectively, we design a two-level structure that encompasses both intra-block and inter-block components. For simplicity, the decoder depicted in Fig. 3 comprises a single trend and seasonal block. However, we accommodate a customizable number of blocks to cater to diverse types of time series with multiple trend and seasonal patterns.

The CORF decoder takes correlated features $\mathbf{G}_i$ as input and generates forecasting series $\hat{\mathbf{F}}_i = \langle \hat{\mathbf{x}}_{i,t+1}, \hat{\mathbf{x}}_{i,t+2}, ..., \hat{\mathbf{x}}_{i,t+F}\rangle$. The CORF decoder is a residual neural network that aggregates the outputs from each block. Consequently, the outputs of the trend and seasonal blocks are independently decomposed functions with fixed analytic forms, exhibiting robustness to outliers.

From the intra-block perspective, we design two distinct trend and seasonal blocks to capture the trend and seasonal temporal dynamics. For instance, the trend block receives correlated features, denoted as $\mathbf{G}_{tre}$. As shown in Fig. 3, the trend block is the first block, resulting in $\mathbf{G}_{tre}$ being equal to $\mathbf{G}_i$. The trend block generates two polynomials: the forecast polynomial $\hat{\mathbf{F}}_{tre}$, tasked with predicting future trend patterns, and the backcast polynomial $\hat{\mathbf{G}}_{tre}$, which concentrates on reconstructing historical trend patterns. Specifically, the intra-block components utilize gated temporal convolutional networks (TCN-1 and TCN-2) [42, 43], which have been shown to exhibit advantages such as capturing long-range dependencies and providing robustness. The output of gated temporal convolutional networks $\mathbf{G}'_{tre}$ in the trend block is defined as follows.

$$\mathbf{G}'_{tre} = \tanh(\mathbf{\Phi}_{tcn1}\mathbf{G}_{tre} + bias_3) \odot \sigma(\mathbf{\Phi}_{tcn2}\mathbf{G}_{tre} + bias_4), \tag{8}$$

where $\mathbf{\Phi}_{tcn1}$, $\mathbf{\Phi}_{tcn2}$, $bias_3$ and $bias_4$ are the learned parameters. Subsequently, the backcast and forecast layers receive $\mathbf{G}'_{tre}$ as input and generate the corresponding reconstructed and forecasted polynomials, respectively. The forecast trend polynomial $\hat{\mathbf{F}}_{tre}$ is defined as follows.

$$\begin{aligned}\boldsymbol{\tau} &= \mathrm{FC}(\mathbf{\Phi}_{for1}\mathbf{G}'_{tre})\\ \hat{\mathbf{F}}_{tre} &= \sum_{p=0}^{P}\boldsymbol{\tau}_p\mathbf{f}^p,\end{aligned} \tag{9}$$

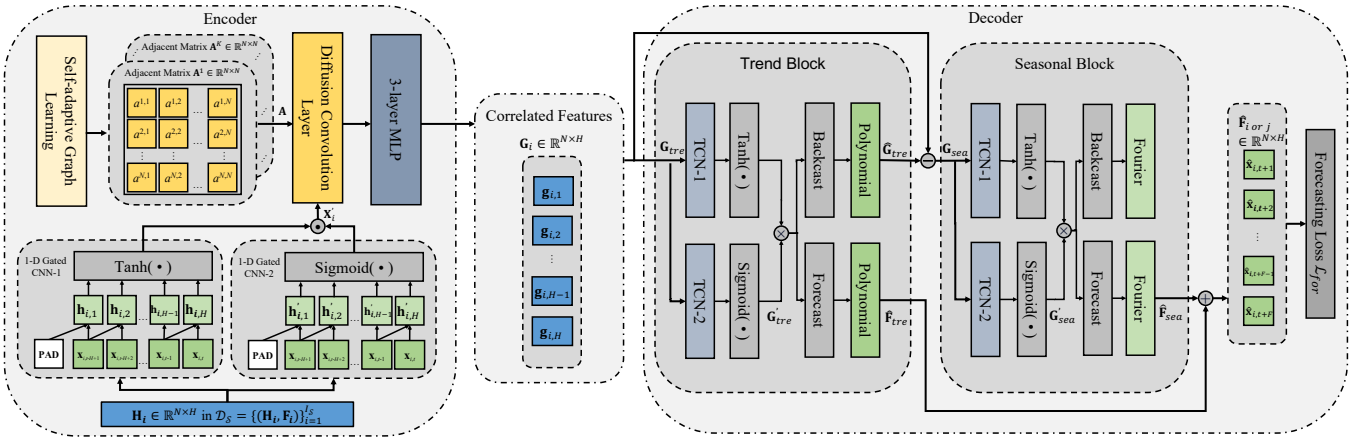


Figure 3: Overview of CORF. The CORF encoder captures correlations among multiple time series and outputs correlated features, while the CORF decoder detects temporal dynamics in the time series and outputs decomposed seasonality and trend to achieve strong robustness to outliers.

where $\mathrm{FC}(\cdot)$ represents the forecast layer, a fully connected layer featuring the learnable parameter $\mathbf{\Phi}_{for1}$. The vector $\mathbf{f} = [0, 1, ..., F-1]/H$ denotes the reconstructed observations. Additionally, $\boldsymbol{\tau}$ signifies the learned parameters of the forecast polynomial, while P indicates a hyper-parameter that controls the highest power. Similarly, the backcast trend polynomial $\hat{\mathbf{G}}_{tre}$ can be obtained using the same approach. The gated TCNs process applied in the seasonal block is consistent with the one employed in the trend block. Within the seasonal block, the gated TCNs accept $\mathbf{G}_{sea}$ as input and generate $\mathbf{G}'_{sea}$ as output. The main distinction is that the reconstructed and forecasted sequences in the seasonal block are based on Fourier Series. The forecast Fourier series $\hat{\mathbf{F}}_{sea}$ is defined as follows.

$$\begin{aligned} \boldsymbol{\epsilon} &= \mathrm{FC}(\mathbf{\Phi}_{for2}\mathbf{G}'_{sea}) \\ \hat{\mathbf{F}}_{sea} &= \sum_{p=0}^{\lfloor F/2-1 \rfloor} (\boldsymbol{\epsilon}_p \cos(2\pi p\mathbf{f}) + \boldsymbol{\epsilon}_{\lfloor p+F/2 \rfloor} \sin(2\pi p\mathbf{f})), \end{aligned} \tag{10}$$

where $\mathbf{\Phi}_{for2}$ denotes the learned parameters of the fully connected layer and $\boldsymbol{\epsilon}$ represents the learned parameters of the forecast Fourier series. Each module generates forecast series by producing functions with fixed analytic forms, in which $\mathbf{f}$ serves as independent variables. The decoder optimizes these function parameters, $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$, while maintaining the form of the output functions unchanged.

From the inter-block perspective, the decoder employs a residual connection [52] between the trend and seasonal blocks. We incorporate residual connections to serve multiple purposes: besides mitigating the vanishing gradient issue and expediting convergence, their primary function is to establish independent seasonal and trend time series backbones. This, in turn, bolsters the model's robustness in dealing with outliers and boosts its overall performance. Each block receives as input the difference between the previous block's input and the output of that block's backcast. For instance, as depicted in Fig. 3, the seasonal block's input is determined by the difference between the trend block's input and the output of the reconstructed polynomial from the trend block's backcast. The input of the seasonal block $\mathbf{G}_{sea}$ can be defined as follows.

$$\mathbf{G}_{sea} = \mathbf{G}_{tre} - \hat{\mathbf{G}}_{tre} \tag{11}$$

The reconstructed outputs from these blocks are merged to generate the forecasted future time series. The forecasting series, denoted by $\hat{\mathbf{F}}$, is defined as follows.

$$\hat{\mathbf{F}} = \hat{\mathbf{F}}_{tre} + \hat{\mathbf{F}}_{sea}, \tag{12}$$

where the forecasting series $\hat{\mathbf{F}}$ is a combination of trend and seasonal series, thus rendering the forecasting and optimization procedures as independent processes.

Algorithm 1 CORF Pre-training

Input: Source data $\mathcal{D}_{\mathcal{S}}$; historical and forecasting horizon H, F; highest power P; batch size b; learning rate of CORF η_{for}
Output: Learned parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$ of the encoder and decoder

Initialisation: Initialize the parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$ randomly
while parameters $\mathbf{\Theta}$ and $\mathbf{\Phi}$ are not converge **do**
  Load batch of time series $\{\mathbf{H}_i, \mathbf{F}_i\}_{i=1}^{b}$ from $\mathcal{D}_{\mathcal{S}}$
  for $iteration = 0$ to $\lfloor I_{\mathcal{S}}/b \rfloor$ **do**
    $\mathbf{G}_i \leftarrow \mathcal{G}(\mathbf{H}_i)$
    $\hat{\mathbf{F}}_i \leftarrow \mathcal{F}(\mathbf{G}_i)$
    $\mathbf{\Phi} \leftarrow \mathbf{\Phi} - \eta_{for} \nabla_{\mathbf{\Phi}} \mathcal{L}_{for}(\hat{\mathbf{F}}_i, \mathbf{F}_i)$
    $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \eta_{for} \nabla_{\mathbf{\Theta}} \mathcal{L}_{for}(\hat{\mathbf{F}}_i, \mathbf{F}_i)$
  end for
end while

3.2.3 Pre-training on the source domain. As illustrated in Alg. 1, we employ the source domain to train the encoder-decoder framework CORF. The objective is to ascertain the position of the source domain within the latent space, thereby enabling the transfer of the target domain into this defined space to reduce the distribution divergence between the source and the target. Consequently, the optimization goal is to minimize the forecasting loss, denoted as

$\mathcal{L}_{for}$, defined as follows.

$$\min_{\Theta,\Phi} \mathcal{L}_{for}(\hat{\mathbf{F}}_i, \mathbf{F}_i) = \min_{\Theta,\Phi} \frac{1}{I_{\mathcal{S}}} \sum_{\mathbf{H}_i \in \mathcal{D}_{\mathcal{S}}} ||\mathcal{F}_{\Phi}(\mathcal{G}_{\Theta}(\mathbf{H}_i)) - \mathbf{F}_i||_1, \quad (13)$$

where $\Theta = \{\alpha, \Theta_{emb1}, \Theta_{emb1}, \Theta_{dcl}, \Theta_{cnn1}, \Theta_{cnn1}\}$ signifies the collection of learnable parameters associated with the encoder, while $\Phi = \{\Phi_{tcn1}, \Phi_{tcn2}, \Phi_{for1}, \Phi_{for2}, \boldsymbol{\tau}, \boldsymbol{\epsilon}\}$ denotes the corresponding set for the decoder.

3.3 Weakly Guided Adaptation

Once the distribution of the source domain in the latent space is determined, we use adversarial training to project the target domain into the same latent space, aiming to minimize the distribution divergence with the source domain. This strategy enables the CORF decoder to better fit the event features within the target domain, thereby boosting prediction accuracy. We first elucidate the operation of the Domain Critic and then detail the optimization process of weakly guided adaptation.

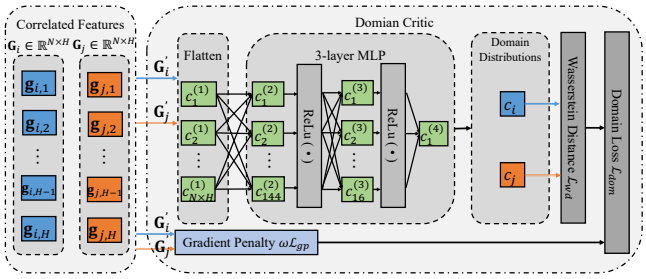


Figure 4: Overview of Domain Critic. The blue, orange, and black arrows denote data flows from the source, target, and combined domains, respectively.

3.3.1 Domain Critic. The distribution divergence in marginal probability distributions between the source and target domain, coupled with the sparsity of samples in the target domain, constitute critical factors that impede the model's capacity to fit the data accurately. Domain Critic serves as a discriminator within adversarial domain adaptation frameworks, with the primary objective of minimizing distribution divergence.

As illustrated in Fig. 4, the Domain Critic utilizes the Wasserstein metric, also known as the Wasserstein distance [26], which aims to minimize the distance between the correlated feature distributions $\mathbf{G}_i$ and $\mathbf{G}_j$ to assist the model in capturing event features. Here, $\mathbf{G}i$ and $\mathbf{G}j$ represent the outputs derived from the source domain $\mathcal{D}_{\mathcal{S}}$ and the target domain $\mathcal{D}_{\mathcal{T}}$ processed through the CORF encoder, respectively. Domain Critic accepts the correlated features $\mathbf{G}_i$, $\mathbf{G}_j$ as input and generates corresponding domain distributions $\hat{c}_i$, $\hat{c}_j$ for optimizing parameters in CORF encoder and Domain Critic. A smaller distance between the source and target domains signifies a more similar distribution of $\mathbf{G}_i$ and $\mathbf{G}_j$ within the latent space. Consequently, CORF decoder is capable of capturing features in the target domain even when the samples in the target domain are limited, as the CORF decoder is fully trained using the source domain, and the distribution of the target domain resembles the distribution of the source domain.

Initially, as shown in Fig. 4, the correlated features $\mathbf{G}_i$, $\mathbf{G}_j$ are processed through a flattened layer and a three-layer multilayer perceptron (MLP) to generate the domain distributions $\hat{c}_i$ and $\hat{c}_j$, defined as follows.

$$\hat{c}_i = \text{MLP}(\Omega_{mlp} \cdot \text{Flatten}(\mathbf{G}_i')), \quad (14)$$

where Ω_{mlp} denotes the learned parameters for the 3-layer MLP. The generation process for the domain distribution $\hat{c}_j$ follows the same procedure as that for $\hat{c}_i$.

Then, to achieve our optimization objective $\mathcal{L}_{wd}$, we compute the Wasserstein distance between the two distributions, $\hat{c}_i$ and $\hat{c}_j$. As directly calculating the integral presents challenges, we instead determine the dual representation. According to the Monge-Rubinstein theorem, the dual representation of the Wasserstein distance [26] W_1 can be expressed as follows.

$$W_1(\mathcal{X}_{\mathcal{S}}, \mathcal{X}_{\mathcal{T}}) = \sup_{||C||_L \leq 1} (\mathbb{E}_{\mathbf{G}_i \sim \mathcal{X}_{\mathcal{S}}}[C(\mathbf{G}_i)] - \mathbb{E}_{\mathbf{G}_j \sim \mathcal{X}_{\mathcal{T}}}[C(\mathbf{G}_j)]), \quad (15)$$

where $C(\cdot)$ represents the Domain Critic function that transforms correlated features $\mathbf{G}_i$ and $\mathbf{G}_j$ into domain distributions $\hat{c}_i$ and $\hat{c}_j$. Moreover, $\mathcal{X}_{\mathcal{S}}$ and $\mathcal{X}_{\mathcal{T}}$ signify one metric space of correlated features, while $\sup(\cdot)$ denotes the supremum of a set, subject to the following Lipschitz semi-norm:

$$||C||_L = \sup \left\| \frac{C(\mathbf{G}_i) - C(\mathbf{G}_j)}{d(\mathbf{G}_i, \mathbf{G}_j)} \right\|_1, \quad (16)$$

where function $d(\mathbf{G}_i, \mathbf{G}_j)$ denotes the distance between samples $\mathbf{G}_i$ and $\mathbf{G}_j$. The parameters of the Domain Critic function, C, adhere to the 1-Lipschitz constraint, which enables us to approximate the maximization of the loss, $\mathcal{L}_{wd}$, defined as follows.

$$\begin{aligned} \mathcal{L}_{wd}(\hat{c}_i, \hat{c}_j) &= \frac{1}{I_{\mathcal{S}}} \sum_{\mathbf{H}_i \in \mathcal{D}_{\mathcal{S}}} C(\mathcal{G}(\mathbf{H}_i)) - \frac{1}{I_{\mathcal{T}}} \sum_{\mathbf{H}_j \in \mathcal{D}_{\mathcal{T}}} C(\mathcal{G}(\mathbf{H}_j)) \\ &= \frac{1}{I_{\mathcal{S}}} \sum_{\mathbf{G}_i \in \mathcal{X}_{\mathcal{S}}} C(\mathbf{G}_i) - \frac{1}{I_{\mathcal{T}}} \sum_{\mathbf{G}_j \in \mathcal{X}_{\mathcal{T}}} C(\mathbf{G}_j) \end{aligned} \quad (17)$$

Upon obtaining the optimization objective function, $\mathcal{L}_{wd}$, it is crucial to ensure that the final domain loss function, $\mathcal{L}_{dom}$, adheres to the 1-Lipschitz constraint. Weight clipping, which restricts the model's parameters within a fixed range, can ensure the 1-Lipschitz constraint, but leads to gradient vanishing or exploding in model fitting. Inspired by the research [13], we incorporate a gradient penalty, $\mathcal{L}_{gp}$, as defined by the formula:

$$\mathcal{L}_{gp}(\mathbf{G}_i, \mathbf{G}_j) = \frac{1}{I_{gp}} \sum_{\hat{\mathbf{G}} \in \hat{\mathcal{X}}} (||\nabla_{\hat{\mathbf{G}}} C(\hat{\mathbf{G}})||_2 - 1)^2, \quad (18)$$

where $\hat{\mathbf{G}}$ signifies a random sample taken along the line connected the correlated feature pair $\mathbf{G}_i$ and $\mathbf{G}j$. The set of potential samples, $\hat{\mathcal{X}}$, comprises all possible instances of $\hat{\mathbf{G}}$. Furthermore, I_{gp} denotes the total number of samples. Employing this sampling method effectively enforces the 1-Lipschitz constraint and simultaneously tackles issues related to gradients vanishing and exploding.

3.3.2 Weekly guided adaptation on the source and target domain. As depicted in Alg. 2, we simultaneously and uniformly sample from both source and target domains generated from the weakly labeled process. We load the parameters Θ and Φ, obtained during the pre-training stage, into CORF. Next, we train the Domain Critic, and

Algorithm 2 DARF Weakly Guided Adaptation

Input: Source and target data $\mathcal{D}_{\mathcal{S}}$, $\mathcal{D}_{\mathcal{T}}$; historical and forecasting horizon H, F; highest power P; weight parameters ω,λ; batch size b; learning rate for CORF η_{for}, and for Domain Critic η_{dom}

Output: Learned parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$ of the encoder and decoder

Initialisation: Utilize the parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$ learned during pre-training; initialize parameter of Domain Critic Ω randomly
while parameters $\mathbf{\Theta}$, $\mathbf{\Phi}$, and Ω are not converge **do**
  Load batch of time series $\{\mathbf{H}_i, \mathbf{F}_i\}_{i=1}^{b}$ and $\{\mathbf{H}_j, \mathbf{F}_j\}_{j=1}^{b}$ from $\mathcal{D}_{\mathcal{S}}$ and $\mathcal{D}_{\mathcal{T}}$
  for $iteration = 0$ to $\max(\lfloor I_{\mathcal{S}}/b \rfloor, \lfloor I_{\mathcal{T}}/b \rfloor)$ **do**
    $\mathbf{G}_i \leftarrow \mathcal{G}(\mathbf{H}_i)$, $\mathbf{G}_j \leftarrow \mathcal{G}(\mathbf{H}_j)$
    $\hat{c}_i \leftarrow C(\mathbf{G}_i)$, $\hat{c}_j \leftarrow C(\mathbf{G}_j)$
    $\Omega \leftarrow \Omega - \eta_{dom} \nabla_{\Omega} [\mathcal{L}_{wd}(\hat{c}_i, \hat{c}_j) - \omega \mathcal{L}_{gp}(\mathbf{G}_i, \mathbf{G}_j)]$
  end for
  for $iteration = 0$ to $\max(\lfloor I_{\mathcal{S}}/b \rfloor, \lfloor I_{\mathcal{T}}/b \rfloor)$ **do**
    $\mathbf{G}_i \leftarrow \mathcal{G}(\mathbf{H}_i)$, $\mathbf{G}_j \leftarrow \mathcal{G}(\mathbf{H}_j)$
    $\hat{\mathbf{F}}_i \leftarrow \mathcal{F}(\mathbf{G}_i)$, $\hat{\mathbf{F}}_j \leftarrow \mathcal{F}(\mathbf{G}_j)$
    $\hat{c}_i \leftarrow C(\mathbf{G}_i)$, $\hat{c}_j \leftarrow C(\mathbf{G}_j)$
    $\mathbf{\Phi} \leftarrow \mathbf{\Phi} - \eta_{for} \nabla_{\mathbf{\Phi}} [\mathcal{L}_{for}(\hat{\mathbf{F}}_i, \mathbf{F}_i) + \mathcal{L}_{for}(\hat{\mathbf{F}}_j, \mathbf{F}_j)]$
    $\mathbf{\Theta} \leftarrow \mathbf{\Theta} - \eta_{for} \nabla_{\mathbf{\Theta}} [\mathcal{L}_{for}(\hat{\mathbf{F}}_i, \mathbf{F}_i) + \mathcal{L}_{for}(\hat{\mathbf{F}}_j, \mathbf{F}_j) + \lambda \mathcal{L}_{wd}(\hat{c}_i, \hat{c}_j)]$
  end for
end while

then jointly train the encoder and decoder of CORF. In generative adversarial learning, it is essential to alternate training between the generator and discriminator (corresponding to the CORF encoder and Domain Critic) to maintain a balanced learning process. As a result, we initially fix the CORF encoder and focus on training the Domain Critic. The optimization goal is to maximize the domain loss, given by $\mathcal{L}_{dom}$ and defined as follows.

$$\max_{\Omega} \mathcal{L}_{dom} = \max_{\Omega} \mathcal{L}_{wd}(\hat{c}_i, \hat{c}_j) - \omega \mathcal{L}_{gp}(\mathbf{G}_i, \mathbf{G}_j), \tag{19}$$

where $\Omega = \{\Omega_{mlp}\}$ signifies the collection of learnable parameters associated with Domain Critic, and ω denotes the weight parameter that balances the importance of the gradient penalty $\mathcal{L}_{gp}$. Subsequently, we fix the Domain Critic and continue with the training of both the encoder and the decoder of CORF. The objective function is defined as follows.

$$\min_{\mathbf{\Theta}, \mathbf{\Phi}} ((\mathcal{L}_{for}(\hat{\mathbf{F}}_i, \mathbf{F}_i) + \mathcal{L}_{for}(\hat{\mathbf{F}}_j, \mathbf{F}_j)) + \lambda \mathcal{L}_{wd}(\hat{c}_i, \hat{c}_j)), \tag{20}$$

where λ denotes the parameter that balances the importance of $\mathcal{L}_{wd}$ and the forecasting loss $\mathcal{L}_{for}$. By employing an alternating training approach, the CORF encoder is afforded the opportunity to adapt and enhance its outputs in response to the current performance of the Domain Critc, while the Domain Critic simultaneously learns to adjust to the evolving outputs generated by the CORF encoder.

4 EXPERIMENTAL STUDY

4.1 Experimental Setup

4.1.1 Datasets and Evaluation Metrics. We perform experimental studies on three benchmark datasets.

Electricity: The Electricity dataset [22], alternatively referred to as the Individual Household Electric Power Consumption Data Set, is accessible via the UCI repository.

ETTh2: The ETTh2 dataset [51] is a dataset that records the temperature and other associated variables of electricity transformers.

Traffic: The Traffic dataset [19], published by the California Department of Transportation, captures traffic congestion data on the San Francisco Bay area freeways through multiple sensors at different locations.

Table 1: Dataset related statistics.

Dataset	N	Length	Split Ratio	H	F	Event
Electricity	4	2075259	7:1:2	6∼24	6∼24	8.89%
ETTh2	7	17420	7:1:2	6∼24	6∼24	7.34%
Traffic	862	17544	7:1:2	6∼24	6∼24	8.61%

Tab. 1 provides summary statistics related to the three datasets, where N denotes the number of variables, "Length" represents the number of timestamps, "Split Ratio" indicates the train-validation-test distribution for the entire dataset, "Event" signifies the ratio of the number of timestamps assigned weak labels to the total number of timestamps, and H and F denote the lengths of the historical and forecasting horizons, respectively.

We adhere strictly to the scheme proposed in two previous studies [34, 37] and introduce three types of outliers—spike, dip, and abrupt trend changes—into the three real-world datasets. Spike outliers refer to sudden, significant increases in the time series, such as traffic congestion caused by an accident. Dip outliers indicate sudden, substantial decreases in the time series, such as missing data resulting in null values. Finally, abrupt trend changes refer to sudden changes in the time series trend, such as a power outage causing power consumption to remain at zero for a duration of time.

We utilize local public holidays, including the day before, the day of, and the day after a holiday, as weak labels to establish the target domain. This is motivated by the observation that the feature distribution exhibits similarity during times adjacent to holidays.

Metrics: We employ the well-established time series forecasting metrics, mean absolute error (MAE) and mean absolute percentage error (MAPE) [40, 41], to assess the prediction accuracy of DARF and other baselines. Lower values for these metrics indicate higher prediction accuracy.

4.1.2 Baselines. We chose two categories of baselines: one group consists of methods based on graph neural networks, while the other relies on seasonal-trend decomposition approaches. In the graph neural network-based methods, we consider MTGNN [42] and STEP [30]. (1) MTGNN combines the advantages of graph convolutional modules and temporal convolution modules. The graph convolutional module is employed to establish short-term correlations among multiple time series, while the temporal convolution module captures the temporal dynamics of the time series, making it particularly adept at predicting short time series. (2) STEP is a pre-trained model that integrates transformer and graph structures. The transformer offers significant advantages for long time series prediction, and therefore, the pre-trained transformer provides more long-term dependencies for subsequent graph structures to enhance prediction accuracy. In seasonal-trend decomposition-based methods, we consider RobustSTL [34], CoST [39], FEDFormer [52], and

Dlinear [48]. (3) RobustSTL is a statistical time series decomposition model. Therefore, we first use a temporal convolutional network to predict the time series, and then apply RobustSTL to decompose the forecasting sequences to enhance the model's ability to be robust to outliers. (4) CoST utilizes a trend feature disentangler to extract trend features, then employs a fast Fourier transform to extract the periodic features. Lastly, a contrastive learning component is used to learn the decomposed parts separately. (5) FEDFormer is a transformer-based model. It first decomposes the time series into trend and periodic components, and then connects the extracted features using frequency-enhanced attention to improve the model's prediction accuracy for long sequences. (6) Dlinear initially uses FEDFormer's seasonal-trend decomposition module to decompose the time series, and then adds a linear layer to learn the decomposed features for prediction.

4.1.3 Implementation Details. All experiments are conducted on a server running Linux 18.04 with an Intel Xeon W-2155 CPU @ 3.30GHz and one Tesla V100 GPU with 30GB memory.

We utilize grid search coupled with a greedy strategy to identify the optimal hyperparameters from the aforementioned methods. Furthermore, we also carefully tune the hyperparameters based on the recommendations of baselines. We utilize the Adam optimizer with a learning rate for η_{for} and η_{dom} within the range $[1e^{-4}, 1e^{-2}]$. The batch size b is selected from the set $\{32, 64, 128, 256, 512\}$. The number of trend and seasonal blocks are chosen from the options $\{1, 2, 3, 4, 5\}$. The value for the highest power P is derived from the set $\{1, 2, 3, 4, 5\}$. Additionally, weight parameters ω and λ are chosen from the set $\{0.1, 0.5, 1, 2, 10\}$.

4.2 Comparison of Forecasting Accuracy

We conduct experiments on the aforementioned baselines and DARF, using benchmark datasets. The results are presented in Tab. 2, where the best results are highlighted in bold, and the second-best are underlined. In this section, we set H and F to the same values. As shown in Tab. 2, DARF demonstrates state-of-the-art prediction accuracy across almost all datasets and for all forecasting horizon F in comparison to the other baselines.

The graph neural network-based methods, MTGNN and STEP, exhibit better performance when the forecasting horizon is at most 12, while their prediction accuracy weakens when the horizon exceeds 12. This can be attributed to the construction of a static graph in MTGNN, leading to correlations that do not change with temporal dynamics. STEP, on the other hand, employs a pre-train transformer-based encoder to provide long-term temporal dynamics to the graph structure, resulting in better performance compared to MTGNN when the forecasting horizon exceeds 12. However, it only mitigates the static impact of the correlations. Consequently, as the forecasting horizon increases, the prediction accuracy of graph neural network-based methods tends to decline.

Methods based on seasonal-trend decomposition, such as FEDformer, NLinear, CoST, and RobustSTL, generally exhibit weaker performance compared to graph neural network-based methods when the forecasting horizon is at most 12, but demonstrate better performance when the horizon exceeds 12. This is because seasonal-trend decomposition-based methods are not impacted by static graphs and can capture long-term features. Specifically, the prediction accuracy of FEDformer and NLinear increases with the rise in the forecasting horizon. Thus, FEDformer and NLinear are robust for long time series. Furthermore, the prediction accuracy of CoST and RobustSTL is insensitive to changes in the forecasting horizon. However, due to their simple structures, they are unable to capture complex temporal dynamics in time series, which prevents them from achieving high prediction accuracy.

DARF demonstrates stable prediction accuracy across both long and short forecasting horizons. The CORF decoder employs a decomposition strategy, enabling it to capture long-term features of time series, and hence, it performs well with long time series. Moreover, the CORF decoder utilizes a residual network structure instead of a stack of multi-layer neural networks. This accelerates the convergence speed of model parameters and reduces the risk of overfitting on short time series, therefore ensuring good prediction accuracy on a short forecasting horizon. In addition, CORF's encoder enhances its ability to capture correlations between multiple time series, which also contributes to improving the model's prediction accuracy. A more detailed explanation of the impact of correlations and event features on robust forecasting will be provided in the ablation study.

4.3 Ablation Study

To assess the impact of correlations, the seasonal-trend decomposition framework, and event features on robust forecasting, we conduct an ablation study, as illustrated in Tab. 3.

We establish two baselines, DARF w/o CORF-encoder-1 and DARF w/o CORF-encoder-2, to scrutinize the effect of correlations on robust forecasting. In DARF w/o CORF-encoder-1, we substitute the CORF encoder with a temporal convolutional network, and in DARF w/o CORF-encoder-2, we replace the CORF encoder with self-attention [32]. The results indicate a decrease in prediction accuracy in both setups compared to the original DARF, suggesting a significant contribution of correlations to prediction accuracy. Consequently, the CORF encoder enhances DARF's proficiency in capturing correlations within time series data.

To ascertain the influence of the seasonal-trend decomposition strategy on robust forecasting, we establish three baselines: DARF w/o CORF-decoder-1, DARF w/o CORF-decoder-2, and DARF w/o CORF-decoder-3. These variants replace the CORF decoder with MTGNN, a temporal convolutional network, and self-attention, respectively, which are skilled in capturing correlations, short-term temporal dynamics, and long-term temporal dynamics. Nevertheless, each of these variants shows a drop in prediction accuracy compared to DARF, due to their inability to robustly handle outliers. This highlights that the CORF decoder equips DARF with the capacity to withstand perturbations of outliers.

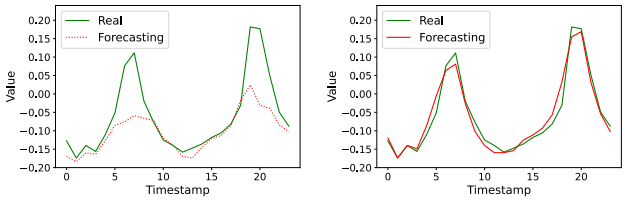
Lastly, we institute the baseline DARF w/o Domain Critic to ascertain the contribution of the adversarial domain adaptation framework to DARF's capacity to capture event features. DARF w/o Domain Critic eliminates the domain critic component of DARF and mixes the data from both the source and target domains for training. Compared to DARF, DARF w/o Domain Critic exhibits a decline in prediction accuracy, as it faces difficulties in capturing limited event features. Additionally, we selected a time series encompassing holidays randomly to showcase the effect of Domain

Table 2: Forecasting comparison of different algorithms. The best result is in boldface and the second best is underlined.

	Dataset	Electricity			Traffic			ETTh2		
Method	F	6	12	24	6	12	24	6	12	24
MTGNN	MAE	0.6341	0.6589	0.6726	0.4301	0.4683	0.6080	0.3568	0.3745	0.3858
	MAPE	0.1568	0.1638	0.1677	0.0993	0.1010	0.1449	0.1452	0.1576	0.1643
STEP	MAE	0.6353	0.6513	0.6685	0.4274	0.4715	0.5775	0.2980	0.3198	0.3215
	MAPE	0.1496	0.1538	0.1587	0.0996	0.1082	0.1389	0.1074	0.1294	**0.1218**
Dlinear	MAE	0.6788	0.6588	0.6383	0.6120	0.5877	0.5313	0.6346	0.4352	0.3305
	MAPE	0.1657	0.1647	0.1599	0.1461	0.1391	0.1228	0.2703	0.1849	0.1408
FEDformer	MAE	0.6703	0.6342	0.6340	0.5062	0.4865	0.4701	0.3519	0.3331	0.3077
	MAPE	0.1638	0.1552	0.1532	0.1161	0.1115	0.1054	0.1304	0.1276	0.1271
CoST	MAE	0.6582	0.6523	0.6561	0.4588	0.4698	0.4742	0.3701	0.3679	0.3808
	MAPE	0.1628	0.1613	0.1631	0.1047	0.1070	0.1076	0.1468	0.1428	0.1553
RobustSTL	MAE	0.6501	0.6780	0.6661	0.6508	0.7239	0.6995	0.2696	0.3499	0.3116
	MAPE	0.1566	0.1635	0.1617	0.1514	0.1665	0.1631	0.1077	0.1512	0.1596
DARF	MAE	**0.6086**	**0.6063**	**0.5914**	**0.4135**	**0.3998**	**0.3676**	**0.2284**	**0.3091**	**0.2564**
	MAPE	**0.1463**	**0.1458**	**0.1415**	**0.0891**	**0.0862**	**0.0799**	**0.0798**	**0.1177**	0.1231

Table 3: Ablation study of the proposed DARF algorithm. The best result is in boldface.

	Dataset	Electricity			Traffic			ETTh2		
Method	F	6	12	24	6	12	24	6	12	24
DARF w/o CORF-encoder-1	MAE	0.6487	0.6317	0.6315	0.4661	0.4575	0.432	0.3161	0.3452	0.3145
	MAPE	0.1564	0.1553	0.1548	0.1125	0.1078	0.0964	0.1225	0.1338	0.1419
DARF w/o CORF-encoder-2	MAE	0.6422	0.6294	0.6268	0.4683	0.4521	0.4308	0.3086	0.3327	0.3083
	MAPE	0.1532	0.1520	0.1516	0.1104	0.1042	0.0927	0.1196	0.1264	0.1385
DARF w/o CORF-decoder-1	MAE	0.6254	0.6435	0.6581	0.4228	0.4451	0.5537	0.3292	0.3375	0.3424
	MAPE	0.1532	0.1540	0.1625	0.0952	0.0980	0.1321	0.1284	0.1305	0.1457
DARF w/o CORF-decoder-2	MAE	0.6532	0.6714	0.6953	0.5075	0.5248	0.5828	0.3285	0.3563	0.3471
	MAPE	0.1642	0.1689	0.1704	0.1204	0.1253	0.1420	0.1255	0.1372	0.1483
DARF w/o CORF-decoder-3	MAE	0.6424	0.6315	0.6284	0.4837	0.4711	0.4662	0.3126	0.3281	0.3019
	MAPE	0.1588	0.1534	0.1515	0.1127	0.1106	0.1032	0.1054	0.1272	0.1295
DARF w/o Domain Critic	MAE	0.6395	0.6305	0.6230	0.4351	0.4217	0.3899	0.2414	0.3323	0.2742
	MAPE	0.1538	0.1526	0.1502	0.0941	0.0892	0.0835	0.0884	0.1235	0.1359
DARF	MAE	**0.6086**	**0.6063**	**0.5914**	**0.4135**	**0.3998**	**0.3676**	**0.2284**	**0.3091**	**0.2564**
	MAPE	**0.1463**	**0.1458**	**0.1415**	**0.0891**	**0.0862**	**0.0799**	**0.0798**	**0.1177**	**0.1231**



(a) DARF w/o Domain Critic. (b) DARF with Domain Critic.

Figure 5: The effect of Domain Critic on a Holiday Series.

Critic. As illustrated in the Fig. 5, the green curve represents the real time series, while the red curve denotes the forecasting series. The experimental results suggest that the adversarial domain adaptation framework enhances DARF's ability to seize event features.

4.4 Incremental Study of Outliers and Events

In order to investigate the impact of the ratio of outliers and events on the robustness of DARF, we perform an incremental study of outliers and events. Specifically, in the study of outliers, we consider the influence on model prediction accuracy of outliers that are independently distributed for each time series or variable, as well as correlated outliers that occur concurrently across each time series or variable. We perform experiments on Electricity. We set the percentages of 1%, 3%, 5%, and 10%. These values thus indicate the percentages of observations in the training set that are replaced by outliers, with both the historical and the forecasting horizon set to 12. Furthermore, for the study of events, we perform experiments, where the ratio of the number of weak labels used for training to the total number of weak labels, denoted as $\mathcal{P}$, is adjusted to be 0%, 25%, 50%, 75%, and 100%.

4.4.1 Study of Independent Outliers. The results in Tab. 4 show that as the proportion of outliers increases, the prediction accuracy of both DARF and the baselines decrease. This implies that the larger the proportion of outliers in the training set, the larger the disturbance to the forecasting models leading to a reduction in their prediction accuracy. We also observe that when the proportion of outliers is below 10%, seasonal-trend decomposition-based methods demonstrate higher robustness to outliers than graph neural network-based methods. However, when the proportion of outliers reaches 10%, the graph neural network-based methods exhibit the best robustness. Indeed, the traditional seasonal-trend decomposition-based methods undergo a robustness degradation when the proportion of outliers is high. We find that with the increase in outliers, uniformly distributed outliers are misinterpreted as periodic patterns by the seasonal module, reducing the model's prediction performance. Nevertheless, graph neural network-based methods maintain robustness even at high outlier proportions. We

Table 4: Results for Incremental Independent Outliers. The best result is in boldface and the second best is underlined.

Method	MTGNN		STEP		Dlinear		FEDformer		CoST		RobustSTL		**DARF**	
Ratio (%)	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
1	0.638	0.1578	0.6259	<u>0.1482</u>	0.6493	0.1623	0.6232	0.1523	<u>0.6157</u>	0.1507	0.6513	0.1581	**0.6012**	**0.1421**
3	0.6589	0.1638	0.6513	0.1538	0.6588	0.1647	<u>0.6342</u>	0.1552	0.6523	0.1613	0.6780	0.1635	**0.6063**	**0.1458**
5	0.6716	0.1675	0.6541	0.1592	0.6640	0.1658	<u>0.6327</u>	<u>0.1550</u>	0.6721	0.1663	0.6953	0.1783	**0.6086**	**0.1462**
10	0.6787	0.1720	<u>0.6614</u>	<u>0.1624</u>	0.6624	0.1661	0.6766	0.1677	0.7200	0.1726	0.6813	0.1679	**0.6088**	**0.1471**

Table 5: Results of Incremental Correlated Outliers. The best result is in boldface and the second best is underlined.

Method	MTGNN		STEP		Dlinear		FEDformer		CoST		RobustSTL		**DARF**	
Ratio (%)	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE	MAE	MAPE
1	0.6604	0.1643	0.6529	0.1550	0.6823	0.1703	0.6222	0.1531	<u>0.6209</u>	<u>0.1504</u>	0.6578	0.1620	**0.6080**	**0.1459**
3	0.6717	0.1674	0.6779	0.1604	0.6872	0.1718	<u>0.6425</u>	0.1592	0.6542	<u>0.1563</u>	0.6588	0.1564	**0.6125**	**0.1470**
5	0.6898	0.1723	0.6792	0.1679	0.6914	0.1727	<u>0.6352</u>	<u>0.1557</u>	0.6836	0.1608	0.6834	0.1678	**0.6143**	**0.1472**
10	0.7001	0.1775	0.6845	<u>0.1705</u>	0.7012	0.1733	0.6835	0.1710	0.7372	0.1771	<u>0.6825</u>	0.1774	**0.6139**	**0.1482**

find that when an observation of a variable at a certain timestamp becomes an outlier, other variables' observations at the same timestamp have a high probability of being normal. The graph can propagate correlations among variables, thereby enhancing the influence of normal observations and diminishing the impact of outliers. CORF integrates the strengths of both, with its encoder assuring the propagation of correlations among variables, while the decoder improves the model's robustness to outliers through its seasonal-trend decomposition mechanism.

4.4.2 Study of Correlated Outliers. The results in Tab. 5 show that with an increase in the proportion of correlated outliers, the prediction accuracy of both DARF and the baselines decline, with a more substantial decline than in the case of independent outliers. Thus correlated outliers exert a more profound disruptive effect on forecasting performance. Additionally, we find that for correlated outliers, the prediction accuracy of the graph neural network-based methods lags behind that of independent outliers. This can be attributed to the limited ability of the graphs to mitigate the influence of outliers through information propagation when outliers appear simultaneously at a particular timestamp across all variables. Under such circumstances, graph neural network-based methods capture outlier information and update model parameters accordingly, leading to incorrect features being learned. Conversely, seasonal-trend decomposition-based methods are less influenced by correlated outliers, and their decline in prediction accuracy is less pronounced than for graph neural networks. The reason is the fact that seasonal-trend decomposition methods are univariate models that remain insensitive to whether outliers occur simultaneously. In particular, CORF's decoder structure applies seasonal-trend decomposition independently to each variable, thus rendering it unaffected by correlated outliers.

Table 6: Prediction accuracy for different $\mathcal{P}$.

Method	$\mathcal{P}$	0%	25%	50%	75%	100%
Electricity	MAE	0.6305	0.6141	0.6111	0.6089	0.6063
	MAPE	0.1526	0.1492	0.1487	0.1462	0.1458
Traffic	MAE	0.4217	0.4102	0.4082	0.4024	0.3998
	MAPE	0.0892	0.0884	0.0871	0.0867	0.0862
ETTh2	MAE	0.3323	0.3207	0.3159	0.3112	0.3091
	MAPE	0.1235	0.1204	0.1193	0.1182	0.1177

4.4.3 Study of Events. As illustrated in Table 6, the prediction accuracy of DARF exhibits an upward trend across all three datasets with an increase in the ratio $\mathcal{P}$. This can be attributed to the fact that a larger $\mathcal{P}$ allows DARF to access more information pertaining to abrupt but normal changes. The model benefits from the more comprehensive training, enabling it to capture distinctive features.

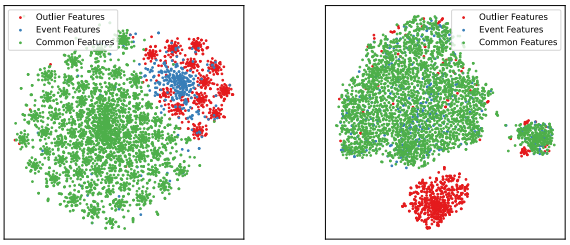
4.5 Visualization

We proceed to provide a visualization of DARF's functioning, covering both a correlated feature distribution analysis and an interpretability analysis. These analyses offer insight into the efficacy of the adversarial domain adaptation framework that captures event features and elucidates how DARF interprets the forecasting time series.

4.5.1 Distribution Analysis. In this experiment, the time series are classified into common series, event series, and outlier series. Here, an outlier series refers to any series that contains outliers. Next, the common series originate from the source domain, while the event series stem from the target domain. These time series are fed into the CORF encoder, yielding their respective correlated features. DARF controls the distribution of these correlated features at this stage. By visualizing the distribution of these correlated features before and after the training, we can observe whether DARF successfully captures the event features and discriminates them from outlier features.

The t-Distributed Stochastic Neighbor Embedding (t-SNE) is a technique for visualizing high-dimensional data [31]. The higher similarity between distributions results in closer proximity of points in a visualization.

Fig. 6 presents the distribution of correlated features visualized using t-SNE, where green dots represent common features, blue dots represent event features, and red dots represent outlier features. Fig. 6(a) shows the distribution of the correlated features before DARF training. Here, the distribution of event features resembles that of outlier features closely. There is a significant overlap in the distribution of event and outlier features, indicating higher similarity in their distributions. Conversely, Fig. 6(b) depicts the distribution of correlated features after DARF training. The distribution of the event features exhibits a significant overlap with the distribution of common features, while being distant from the



(a) Before DARF Training. (b) After DARF Training.

Figure 6: Distributions of correlated features.

outlier features. The visualization thus shows how DARF can successfully capture event features while being robust to outliers.

4.5.2 Interpretability Analysis. An approach to interpreting forecasted time series is to derive analytical forms for the function of forecasted time series. The CORF decoder decomposes time series into multiple trend and seasonal functions related to timestamps with analytic forms. A random case from the Electricity dataset is selected to illustrate how DARF interprets the forecasting series. We set the number of seasonal and trend blocks to 2 each. In Fig. 7(a), the real curve and DARF's forecasting curve are depicted in green and red, respectively. Fig. 7(b)–(e) display the curves of the output at the residual connections of each block in the CORF decoder, which consist of two periodic functions and two trend functions. The sum of these functions constitutes the green curve in Fig. 7(a). After DARF training, the learned parameters $\boldsymbol{\tau}$ and $\boldsymbol{\epsilon}$ are retained to shape the analytic forms for the trend and seasonal functions.

4.6 Model Scalability

We proceed to assess the scalability of DARF on the Electricity dataset. We set the forecasting horizon F to 12 and vary the historical horizon H among 6, 12, 24, 48, and 96. We then observe the effects on DARF's training time, GPU memory use, and prediction accuracy.

Figs. 8(a) and 8(b) show that as the historical horizon H increases, both the training time and GPU memory use of DARF increase. Among the models covered, Dlinear performs best in terms of training time and GPU memory use. Notably, as shown in Figs. 8(c) and 8(d), the prediction accuracy plateaus when H reaches and surpasses 48. To balance the utilization of computational resources with prediction accuracy, we suggest that the historical horizon H is set to 1–2 times the forecasting horizon F.

The theoretical time and space complexities of the two components of DARF, i.e., CORF and Domain Critic, are $O(HN^2)$, and $O(HN)$, respectively, where N is the number of variables. Therefore, the overall time and space complexity of DARF is $O(HN^2)$.

4.7 Parameter Sensitivity

We conduct a sensitivity analysis of four key parameters in DARF on the Electricity dataset: the number of blocks, the highest power P, and the weights ω and λ. The forecasting horizon F and the historical horizon H are both set to 12.

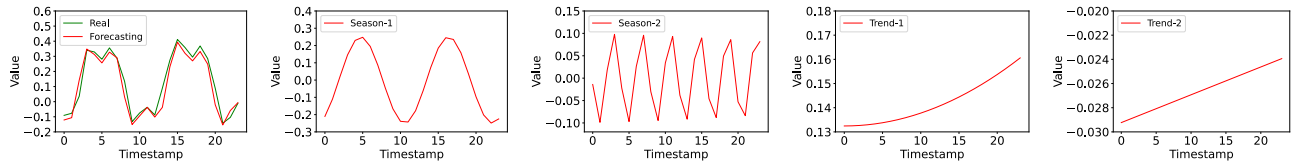
As depicted in Fig. 9(a), DARF achieves the best forecasting accuracy with 2 blocks. The number of blocks corresponds to the number of forecasting functions. When set to 2, the forecasting function comprises two trend functions and two seasonal functions. In addition, DARF's forecasting accuracy peaks at $P = 4$. As P is the highest degree of the polynomial in the trend function and the highest frequency in the seasonal function, higher P values suggest a more complex function, which increase the risk of overfitting. Fig. 9(b) indicates that DARF performs best when ω equals 0.5. ω signifies the weight of the penalty term in the domain loss, a higher ω enhances the generalization ability of the Domain Critic, but reduces its capability to minimize the distribution divergence between the source and target domains. Finally, DARF has the best performance when λ equals 1. As λ denotes the weight of domain loss in the overall loss, a higher λ improves DARF's ability to capture event features but decreases its ability to capture temporal dynamics.

5 RELATED WORK

Robust Forecasting: Robust time series forecasting aims to develop models that can handle data outliers and variability while still providing accurate forecasts. Seasonal-trend decomposition and graph neural networks attempt to address this problem from two different perspectives.

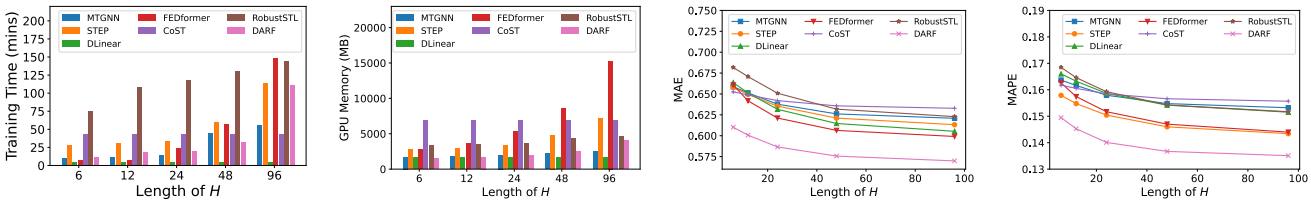
The original seasonal-trend decomposition is based on statistical principles [34, 35, 37, 44] and can decompose time series into multiple seasonal and trend functions. Two studies [34, 37] are the first to explore the theoretical feasibility of this method and its potential in time series forecasting. Two other studies [35, 44] utilize multiscale and maximal overlap discrete wavelet transform methods to decompose time series with multiple seasonalities into multiple independent periodic functions, thus improving decomposition accuracy. In addition, deep learning-based studies [4, 24, 39, 52] incorporate the idea of seasonal-trend decomposition into the deep learning framework. These methods have stronger generalization abilities and are better at addressing overfitting, allowing them to more accurately predict outcomes on data that is not seen during training. One study [39] designs trend and seasonal losses using contrastive learning and optimizes them separately to achieve the goal of seasonal-trend decomposition. Then, two studies [4, 52] combine the transformer framework and Fourier transformation to decompose time series and capture a global view.

Graph neural networks are used commonly in time series forecasting because they can capture correlations among multiple time series. Early graph neural networks [20, 46] are applied to traffic flow prediction using static graph structures to capture explicit spatial correlations among roads. Subsequent research proposes a graph learning module [42, 43] to capture implicit spatial correlations in datasets used for traffic forecasting. The most recent studies [5, 29, 47] extend this graph learning to time series forecasting, proposing explicit and implicit correlations for time series that represent both learned and known time series correlations, respectively.



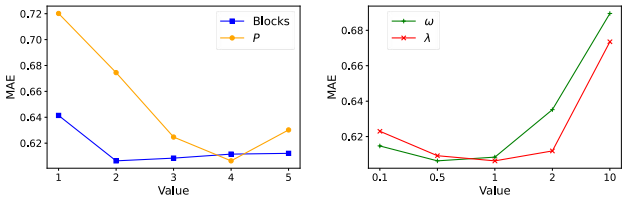
(a) The Real vs. Forecasting. (b) The 1st Seasonality. (c) The 2nd Seasonality. (d) The 1st Trend. (e) The 2nd Trend.

Figure 7: Interpretation of forecasted time series.



(a) Training Time Changing with H. (b) GPU Memory Changing with H. (c) MAE Changing with H. (d) MAPE Changing with H.

Figure 8: Scalability of DARF.



(a) MAE Varies with Blocks and P. (b) MAE Varies with ω and λ.

Figure 9: Parameter sensitivity of DARF.

Domain Adaptation: Domain adaptation was first introduced in computer vision [53], natural language processing [21], and classification tasks [25, 28, 38] to contend with distinct data distributions across various domains. In time series forecasting research, one study [14] develops a transfer learning framework that transfers domain-invariant feature representations from a pre-trained stacked deep residual network to the target domains to assist in predicting each target time series. A model [16] improves its performance on the target domain by sharing attention across multiple source and target domains. Additionally, a study [2] proposes a sparse associative structure alignment model for domain adaptation that relaxes the discovery of causal structure to a sparse associative structure to learn the domain-invariant representations. DARF is the first to investigate adversarial domain adaptation for the same dataset but with different distributions of patterns, specifically domain adaptation between common time series and event time series.

Weekly Supervised Learning: Weakly supervised learning is a machine learning approach that utilizes training data with incomplete, noisy, or imprecise labels, which is known as weakly labeled data. Early works [1, 15, 33] on weakly labeled learning occur commonly in computer vision. In the field of time series, a study [49] proposes a framework for time series classification on sensors data by extracting burst-based features from raw time series and automatically extending fuzzy time points to appropriate subsequences containing sufficient information. A study [45] explores the application of weakly labeled data in time series prediction, giving more granular weak labels to different patterns of time series to improve prediction accuracy. DARF is the first to apply weakly labeled learning to robust forecasting and proposes a framework that uses domain adaptation based on the characteristics of multivariate time series data.

6 CONCLUSIONS

We propose a weakly guided adaptation model, DARF, designed to enable robust forecasting of multivariate time series. To achieve robust forecasting of multivariate time series, we equip DARF with a correlated robust forecaster, whose encoder is able to capture multivariate correlations, thereby enhancing prediction accuracy. Additionally, the forecaster's decoder decomposes time series into a combination of trend and seasonal functions to achieve robustness. DARF employs adversarial domain adaptation to reduce distribution divergence between source and target domains, thereby capturing event features and further improving the prediction accuracy for time series that contain events. Looking ahead, it is promising to extend DARF to encompass foundation models, lightweight time series models [3, 27], traffic forecasting models [6–8], and to explore in greater depth the interpretability of neural networks [17].

ACKNOWLEDGMENTS

This work was partially supported by Independent Research Fund Denmark under agreements 8022-00246B and 8048-00038B and the Villum Fonden under agreement 40567.

REFERENCES

[1] Seungryul Baek, Kwang In Kim, and Tae-Kyun Kim. 2020. Weakly-Supervised Domain Adaptation via GAN and Mesh Model for Estimating 3D Hand Poses Interacting Objects. In *CVPR*. 6120–6130. https://doi.org/10.1109/CVPR42600.2020.00616

[2] Ruichu Cai, Jiawei Chen, Zijian Li, Wei Chen, Keli Zhang, Junjian Ye, Zhuozhang Li, Xiaoyan Yang, and Zhenjie Zhang. 2021. Time Series Domain Adaptation via Sparse Associative Structure Alignment. In *AAAI*. 6859–6867. https://ojs.aaai.org/index.php/AAAI/article/view/16846

[3] David Campos, Miao Zhang, Bin Yang, Tung Kieu, Chenjuan Guo, and Christian S. Jensen. 2023. LightTS: Lightweight Time Series Classification with Adaptive Ensemble Distillation. *Proc. ACM Manag. Data* 1, 2 (2023), 171:1–171:27.

[4] Weiqi Chen, Wenwei Wang, Bingqing Peng, Qingsong Wen, Tian Zhou, and Liang Sun. 2022. Learning to Rotate: Quaternion Transformer for Complicated Periodical Time Series Forecasting. In *KDD*. 146–156. https://doi.org/10.1145/3534678.3539234

[5] Yuzhou Chen, Ignacio Segovia-Dominguez, and Yulia R. Gel. 2021. Z-GCNETs: Time Zigzags at Graph Convolutional Networks for Time Series Forecasting. In *ICML (Proceedings of Machine Learning Research)*, Vol. 139. 1684–1694. http://proceedings.mlr.press/v139/chen21o.html

[6] Razvan-Gabriel Cirstea, Bin Yang, Chenjuan Guo, Tung Kieu, and Shirui Pan. 2022. Towards Spatio-Temporal Aware Traffic Time Series Forecasting. In *ICDE*. 2900–2913.

[7] Razvan-Gabriel Cirstea, Tung Kieu, Chenjuan Guo, Bin Yang, and Sinno Jialin Pan. 2021. EnhanceNet: Plugin Neural Networks for Enhancing Correlated Time Series Forecasting.. In *ICDE*. 1739–1750.

[8] Razvan-Gabriel Cirstea, Bin Yang, and Chenjuan Guo. 2019. Graph Attention Recurrent Neural Networks for Correlated Time Series Forecasting.. In *MileTS19@KDD*.

[9] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. 2019. Anomaly detection for IoT time-series data: A survey. *IEEE Internet of Things Journal* 7, 7 (2019), 6481–6494.

[10] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language Modeling with Gated Convolutional Networks. In *ICML (Proceedings of Machine Learning Research)*, Vol. 70. 933–941. http://proceedings.mlr.press/v70/dauphin17a.html

[11] Christos Faloutsos, Jan Gasthaus, Tim Januschowski, and Yuyang Wang. 2018. Forecasting big time series: old and new. *Proceedings of the VLDB Endowment* 11, 12 (2018), 2102–2105.

[12] Jingkun Gao, Xiaomin Song, Qingsong Wen, Pichao Wang, Liang Sun, and Huan Xu. 2020. RobustTAD: Robust time series anomaly detection via decomposition and convolutional neural networks. *KDD-MileTS* (2020).

[13] Xin Gao, Fang Deng, and Xianghu Yue. 2020. Data augmentation in fault diagnosis based on the Wasserstein generative adversarial network with gradient penalty. *Neurocomputing* 396 (2020), 487–494. https://doi.org/10.1016/j.neucom.2018.10.109

[14] Hailin Hu, MingJian Tang, and Chengcheng Bai. 2020. DATSING: Data Augmented Time Series Forecasting with Adversarial Domain Adaptation. In *CIKM*. 2061–2064. https://doi.org/10.1145/3340531.3412155

[15] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Cross-Domain Weakly-Supervised Object Detection Through Progressive Domain Adaptation. In *CVPR*. 5001–5009. https://doi.org/10.1109/CVPR.2018.00525

[16] Xiaoyong Jin, Youngsuk Park, Danielle C. Maddix, Hao Wang, and Yuyang Wang. 2022. Domain Adaptation for Time Series Forecasting via Attention Sharing. In *ICML*, Vol. 162. 10280–10297. https://proceedings.mlr.press/v162/jin22d.html

[17] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S. Jensen, Yan Zhao, Feiteng Huang, and Kai Zheng. 2022. Robust and Explainable Autoencoders for Unsupervised Time Series Outlier Detection. In *ICDE*. 3038–3050.

[18] Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J. Inman. 2019. 1D Convolutional Neural Networks and Applications: A Survey. *CoRR* abs/1905.03554 (2019). arXiv:1905.03554 http://arxiv.org/abs/1905.03554

[19] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *SIGIR*. 95–104. https://doi.org/10.1145/3209978.3210006

[20] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. 2018. Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In *ICLR*. https://openreview.net/forum?id=SJiHXGWAZ

[21] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Improving Multi-Task Deep Neural Networks via Knowledge Distillation for Natural Language Understanding. *CoRR* abs/1904.09482 (2019). arXiv:1904.09482 http://arxiv.org/abs/1904.09482

[22] Alexander Loginov, Malcolm I. Heywood, and Garnett Carl Wilson. 2016. Benchmarking a coevolutionary streaming classifier under the individual household electric power consumption dataset. In *IJCNN*. 2834–2841. https://doi.org/10.1109/IJCNN.2016.7727557

[23] Hao Miao, Yan Zhao, Chenjuan Guo, Bin Yang, Zheng Kai, Feiteng Huang, Jiandong Xie, and Christian S. Jensen. 2024. A Unified Replay-based Continuous Learning Framework for Spatio-Temporal Prediction on Streaming Data. *ICDE* (2024).

[24] Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. 2020. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *ICLR*. https://openreview.net/forum?id=r1ecqn4YwB

[25] Felix Ott, David Rügamer, Lucas Heublein, Bernd Bischl, and Christopher Mutschler. 2022. Domain Adaptation for Time-Series Classification to Mitigate Covariate Shift. In *MM*. 5934–5943. https://doi.org/10.1145/3503161.3548167

[26] Victor M Panaretos and Yoav Zemel. 2019. Statistical aspects of Wasserstein distances. *Annual review of statistics and its application* 6 (2019), 405–431.

[27] Simon Aagaard Pedersen, Bin Yang, and Christian S. Jensen. 2020. Anytime Stochastic Routing with Hybrid Learning. *Proc. VLDB Endow.* 13, 9 (2020), 1555–1567.

[28] Sanjay Purushotham, Wilka Carvalho, Tanachat Nilanon, and Yan Liu. 2017. Variational Recurrent Adversarial Deep Domain Adaptation. In *ICLR*. https://openreview.net/forum?id=rk9eAFcxg

[29] Chao Shang, Jie Chen, and Jinbo Bi. 2021. Discrete Graph Structure Learning for Forecasting Multiple Time Series. In *ICLR*. https://openreview.net/forum?id=WEHSlH5mOk

[30] Zezhi Shao, Zhao Zhang, Fei Wang, and Yongjun Xu. 2022. Pre-training Enhanced Spatial-temporal Graph Neural Network for Multivariate Time Series Forecasting. In *KDD*. 1567–1577. https://doi.org/10.1145/3534678.3539396

[31] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.

[32] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Annual Conference on Neural Information Processing Systems*. 5998–6008. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[33] Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge J. Belongie. 2017. Learning from Noisy Large-Scale Datasets with Minimal Supervision. In *CVPR*. 6575–6583. https://doi.org/10.1109/CVPR.2017.696

[34] Qingsong Wen, Jingkun Gao, Xiaomin Song, Liang Sun, Huan Xu, and Shenghuo Zhu. 2019. RobustSTL: A Robust Seasonal-Trend Decomposition Algorithm for Long Time Series. In *AAAI*. 5409–5416.

[35] Qingsong Wen, Kai He, Liang Sun, Yingying Zhang, Min Ke, and Huan Xu. 2021. RobustPeriod: Robust Time-Frequency Mining for Multiple Periodicity Detection. In *SIGMOD*. 2328–2337.

[36] Qingsong Wen, Linxiao Yang, Tian Zhou, and Liang Sun. 2022. Robust time series analysis and applications: An industrial perspective. In *KDD*. 4836–4837.

[37] Qingsong Wen, Zhe Zhang, Yan Li, and Liang Sun. 2020. Fast RobustSTL: Efficient and Robust Seasonal-Trend Decomposition for Time Series with Complex Patterns. In *KDD*. 2203–2213.

[38] Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. 2020. Multi-Source Deep Domain Adaptation with Weak Supervision for Time-Series Sensor Data. In *KDD*. 1768–1778. https://doi.org/10.1145/3394486.3403228

[39] Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven C. H. Hoi. 2022. CoST: Contrastive Learning of Disentangled Seasonal-Trend Representations for Time Series Forecasting. In *ICLR*. https://openreview.net/forum?id=PilZY3omXV2

[40] Xinle Wu, Dalin Zhang, Chenjuan Guo, Chaoyang He, Bin Yang, and Christian S. Jensen. 2021. AutoCTS: Automated Correlated Time Series Forecasting. *Proc. VLDB Endow.* 15, 4 (2021), 971–983. https://doi.org/10.14778/3503585.3503604

[41] Xinle Wu, Dalin Zhang, Miao Zhang, Chenjuan Guo, Bin Yang, and Christian S. Jensen. 2023. AutoCTS+: Joint Neural Architecture and Hyperparameter Search for Correlated Time Series Forecasting. *Proc. ACM Manag. Data* 1, 1 (2023), 97:1–97:26.

[42] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In *KDD*. 753–763. https://doi.org/10.1145/3394486.3403118

[43] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. 2019. Graph WaveNet for Deep Spatial-Temporal Graph Modeling. In *IJCAI*. 1907–1913. https://doi.org/10.24963/ijcai.2019/264

[44] Linxiao Yang, Qingsong Wen, Bo Yang, and Liang Sun. 2021. A Robust and Efficient Multi-Scale Seasonal-Trend Decomposition. In *ICASSP*. 5085–5089. https://doi.org/10.1109/ICASSP39728.2021.9413939

[45] Chin-Chia Michael Yeh, Nickolas Kavantzas, and Eamonn J. Keogh. 2017. Matrix Profile IV: Using Weakly Labeled Time Series to Predict Outcomes. *Proc. VLDB Endow.* 10, 12 (2017), 1802–1812. https://doi.org/10.14778/3137765.3137784

[46] Bing Yu, Haoteng Yin, and Zhanxing Zhu. 2018. Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. In *IJCAI*. 3634–3640. https://doi.org/10.24963/ijcai.2018/505

[47] Hongyuan Yu, Ting Li, Weichen Yu, Jianguo Li, Yan Huang, Liang Wang, and Alex X. Liu. 2022. Regularized Graph Structure Learning with Semantic Knowledge for Multi-variates Time-Series Forecasting. In *IJCAI*. 2362–2368. https://doi.org/10.24963/ijcai.2022/328

[48] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2022. Are Transformers Effective for Time Series Forecasting? *CoRR* abs/2205.13504 (2022). https://doi.org/10.48550/arXiv.2205.13504 arXiv:2205.13504

[49] Hanbo Zhang, Yawen Wang, Peng Wang, and Wei Wang. 2017. Burst-Based Event Classification on Weakly Labeled Time Series Data of Sensors. In *BigData Congress*. 452–459. https://doi.org/10.1109/BigDataCongress.2017.66

[50] Kai Zhao, Chenjuan Guo, Yunyao Cheng, Miao Zhang, Peng Han, and Bin Yang. 2024. Multiple Time Series Forecasting with Dynamic Graph Modeling. *Proc. VLDB Endow.* (2024).

[51] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *AAAI*. 11106–11115. https://ojs.aaai.org/index.php/AAAI/article/view/17325

[52] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *ICML (Proceedings of Machine Learning Research)*, Vol. 162. 27268–27286. https://proceedings.mlr.press/v162/zhou22g.html

[53] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-training. In *ECCV (Lecture Notes in Computer Science)*, Vol. 11207. 297–313. https://doi.org/10.1007/978-3-030-01219-9_18