Improving Model Generalization for Short-Term Customer Load Forecasting With Causal Inference

Zhenyi Wang, *Graduate Student Member, IEEE*, Hongcai Zhang, *Senior Member, IEEE*, Ruixiong Yang, and Yong Chen

***Abstract*—Short-term customer load forecasting is vital for the normal operation of power systems. Unfortunately, conventional machine learning-based forecasting methods are susceptible to generalization issues (e.g., the customer heterogeneity and distribution drift of load data), manifested in model performance degradation. In recent years, some studies have employed the advanced deep learning technology, such as online learning, to overcome the aforesaid problems. However, these methods can only alleviate the adverse impacts of generalization problems on model performance, because they are inherently built on unstable relationships (i.e., correlations). In this paper, we propose a novel causal inference-based method to improve the generalization for short-term customer load forecasting models. Specifically, we first investigate the causal relations between input features and the output in existing methods, and introduce the load characteristics as an extra model input to enhance the causality. Then, we closely inspect the causality in models by using the causal graph to distinguish the confounder, followed by employing the causal intervention with do-calculus to eliminate the spurious correlations caused by the confounder. Moreover, we propose a novel load forecasting framework with the load characteristic extraction, characteristic pool approximation and characteristic-injected model to realize the causal intervention in an efficient and fidelity way. Finally, the effectiveness and superiority of our proposed method are validated on a public dataset.**

***Index Terms*—Causal inference, model generalization, short-term load forecasting, spurious correlation, transformer.**

I. Introduction

IN PROMOTING the achievement of carbon neutrality, there is an increasingly urgent need for flexibility resources to eliminate the system fluctuations brought by the increasing integration of distributed energy resources. According to statistics [1], the residential sector accounts for over 20% of national energy consumption, harboring tremendous regulation potentials. Furthermore, with the extensive deployment of advanced metering infrastructures, it provides a solid foundation for data analytics to improve energy efficiency at the customer level. To this end, demand response emerges at a historic moment, which coordinates regulation resources from the demand side to maintain the system balance [2].

Accurate load forecasting is recognized as an indispensable part of implementing demand response programs, especially short-term customer load forecasting [3]. For aggregators, load forecasting facilitates the identification of suitable customer groups to participate in demand response programs, and the bidding in the market for maximizing revenue. For customers, forecasted loads contribute to scheduling future load consumption and evaluating regulation capacity for demand response [4]. Unlike the grid-level forecasting with relatively regular patterns, customer load forecasting is more challenging due to the highly volatile nature of individual loads [5].

Short-term customer load forecasting has been in the research hotspot for many years [3]. Broadly speaking, existing methods can be classified into two main groups: *statistical-based* and *machine learning-based* methods. The statistical-based methods leverage mathematical models to capture the relationship between input features and the output for load forecasting. For instance, Li et al. [6] adopted the least absolute shrinkage and selection operator to explore the sparsity in historical loads. Teeraratkul et al. [7] proposed a shape-based method with the dynamic time warping to forecast household customer loads. On the other hand, with the boom of deep learning, the machine learning-based methods has became the primary pillar in load forecasting. Particularly, numerous neural network models that are capable of sequence modeling load data with temporal dependencies, have been widely explored. For example, Li et al. [8] developed an improved load forecasting method with the long short-term memory neural network. Lin et al. [9] proposed a spatial-temporal load forecasting framework based on the graph neural network to capture the hidden spatial dependencies. Moreover, Zhou et al. [10] proposed a robust load forecasting model based on Bayesian learning. However, owing to the heterogeneity in load consumption habits, deep learning-based methods are prone to performance degradation, especially when applied to load forecasting for multiple customers [11].

To remedy the aforementioned issue, the intuitive way is to train individual forecasting models for each customer, known as the *local* model [12]. Apart from that, some researchers advocate the *global* model [13], which utilizes a single model to perform load forecasting for multiple different customers.

Manuscript received 8 April 2024; revised 10 July 2024; accepted 26 August 2024. Date of publication 30 August 2024; date of current version 26 December 2024. This work was supported in part by the Science and Technology Development Fund, Macau, SAR, under Grant 001/2024/SKL and Grant 0053/2022/AMJ, and in part by the Science and Technology Project of Guangdong Power Grid Company Ltd. under Grant GDKJXM20222420. Paper no. TSG-00586-2024. *(Corresponding author: Hongcai Zhang.)*

Zhenyi Wang and Hongcai Zhang are with the State Key Laboratory of Internet of Things for Smart City and the Department of Electrical and Computer Engineering, University of Macau, Macau, China (e-mail: hczhang@um.edu.mo).

Ruixiong Yang and Yong Chen are with the Zhuhai Power Supply Bureau, Guangdong Power Grid Company Ltd., Zhuhai 519000, Guangdong, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2024.3452490.

Digital Object Identifier 10.1109/TSG.2024.3452490

Furthermore, there has been an influx of researches that simultaneously consider the customers' habit deviation and model efficiency, as a compromise between local and global manners. For instance, Yang et al. [14] designed a novel multitask load forecasting framework with Bayesian deep learning and clustering-based pooling techniques. Qin et al. [15] proposed a novel load forecasting model for individual buildings, which integrates the federated learning, search technique, and personalization approach. However, although these methods consider load heterogeneity among customers, using static training data results in ignoring the data distribution variation over time [16]. What's worse, customer loads hold susceptibility to many factors in practice, which will aggravate the data distribution variation and lead to poor model performance [17].

To overcome the problem of distribution variation over time, some recent studies employ the dynamic approach to continuously learn new data to update models. Von Krannichfeldt et al. [16] developed a hybrid wind power forecasting method integrated with ensemble learning and online learning to exploit the most recent information. Li et al. [18] proposed a deep kernel method with deep soft spiking neural networks for residential load forecasting, which combines both offline and online learning schemes. Moreover, Yang and Youn [19] proposed a novel individual load forecasting method based on temporal data pooling to provide prediction with the most probable model. In addition to load forecasting, there are studies on building models with consistent performance in different environments [20]. In general, these approaches can be classified into three categories: data manipulation, learning strategy, and representation learning. However, these deep learning-based approaches are purely built on correlations between variables (i.e., input features and outputs) [21]. Specifically, these correlations only represent superficial relationships between variables under particular contexts (e.g., within training datasets), instead of essential laws (i.e., invariant mappings with probably physical significance) that models genuinely want to learn. In other words, correlations are unstable relationships between input features and outputs, because they may change with different circumstances [22]. In this way, the above existing approaches are still in the realm of instability, which has been identified as the culprit for the instability and low generalization of deep learning models [23]. Therefore, most existing techniques only simply mitigate the adverse effects owing to the data distribution discrepancy, rather than getting rid of the root cause (i.e., unstable relationships).

To address the aforementioned problems, some researchers make use of the causality between input features and outputs to build models, rather than correlations. Specifically, causality describes the intrinsic and universal dependencies between variables, which remain invariant across different circumstances [22]. In recent years, causality (or causal inference) has attracted considerable attention, especially in improving the model generalization and explainability [21], [22], [23]. However, to the best of our knowledge, there is scarcely any research in power systems that involves causal inference, even though a great deal of studies intersect with machine learning.

With the aim of bridging the aforementioned research gap, we propose a novel method based on causal inference to build generalization-improved short-term load forecasting models. Specifically, we first analyze input features of existing methods from the causality view, and innovatively design and introduce load characteristics as extra model input. Then, we resort to the causal graph to scrutinize the causality in models and identify the confounder that brings bad effects to model generalization. Next, we utilize the do-calculus and backdoor adjustment for causal intervention via observational data, which removes spurious correlation caused by confounder. Finally, we propose a novel load forecasting framework combined with the load characteristic extraction, characteristic pool approximation and characteristic-injected model, to implement the causal intervention. To the best of our knowledge, we are the first to utilize causal inference for load forecasting. Compared with the published literature, our contributions are threefold:

1) We create a new paradigm based on causal inference to construct generalization-improved models for short-term customer load forecasting. Different from existing methods, we build forecasting models by relying on causality rather than correlation, where the load characteristics are designed and injected as an extra model input. In addition, the causal intervention with do-calculus is applied to eliminate spurious correlations induced by the confounder, which guarantees the model's generalization.
2) We propose a novel load forecasting framework to realize the causal intervention using only observational data. A characteristic-injected model is designed to perform load forecasting with load characteristics, which are extracted from historical loads by a well-designed extraction task and model. Moreover, a characteristic pool approximation is developed to compute do-calculus efficiently.
3) We validate that our proposed method can improve model generalization from a data perspective, rather than designing complex models. Furthermore, compared with most existing model-specific methods, our proposed method is model-agnostic and can be applied to different load forecasting models to improve their model generalization.

The remainder of this paper is structured as follows. Section II describes the load forecasting task and generalization issues involved. Sections III and IV unveil full details of the proposed method. Section V validates the effectiveness and superiority of our proposed method. Section VI concludes this paper.

II. Problem Statement

We begin with the introduction of notations and definitions. Let $\mathcal{X}$ denote a feature space and $\mathcal{Y}$ an output space. Similarly, X and Y are random variables of feature and output, respectively. We denote the data sample with feature and output as $(\boldsymbol{x}, y)$ and the corresponding joint distribution as P_{XY}. Accordingly, the domain that is composed of data sampled from the joint distribution of feature and output, is denoted as

$\mathcal{D}$, i.e., $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n} \sim P_{XY}$, where $\boldsymbol{x} \in \mathcal{X}$, $y \in \mathcal{Y}$, and n the number of data samples. In this paper, each customer with loads and features constitutes a domain.

A. Load Forecasting

Load forecasting is a typical time-series forecasting issue, which predicts future values of electricity loads. The machine learning methods realize load forecasting by exploiting the relationship between input features and future loads [24]. Specifically, these features usually include target observations (i.e., historical loads) and exogenous factors (e.g., date and forecasted weather). Formally, this can be expressed as:

$$y = f(\boldsymbol{x}; \boldsymbol{\theta}), \tag{1}$$

where $f(\cdot; \boldsymbol{\theta})$ is the machine learning model with parameters $\boldsymbol{\theta}$. Here, $\boldsymbol{x}$ and y represent the input features and forecasted loads. Note that we focus on day-ahead forecasting in this paper, i.e., y contains the entire load profile for the next day.

The machine learning models aim to learn the general and predictive knowledge from training data, and then apply the well-trained model to new data. Therefore, the objective function of load forecasting models is formulated as:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim P_{XY}}\big[\ell(f(\boldsymbol{x}; \boldsymbol{\theta}), y)\big], \tag{2}$$

where ℓ is the loss function, which quantifies the discrepancy.

B. Generalization Issue

Machine learning methods usually require that the training and test data satisfy the assumption of being independently and identically distributed. However, this assumption does not always hold in reality, which leads to the model performance deterioration due to data distribution gaps [20]. It is commonly known as the model generalization issue. Particularly, this is more likely to happen with customer load data, which is highly heterogeneous and vulnerable to the environment [25], [26].

Suppose we have M customers for model training $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i^j, y_i^j)\}_{i=1}^{n_j} | j = 1, \ldots, M\}$, where n_j is the data amount of the j-th customer, and the joint probability distribution varies for each domain due to the load heterogeneity, i.e., $P_{XY}^i \neq P_{XY}^j, i \neq j$. In this paper, we broadly classify the model generalization issue into the following two scenarios, according to how the data distribution gap arises:

1) Unseen Customer: Typically, we can not acquire load data from all possible customers to train data-driven models, because it is expensive and even prohibitively impossible. Therefore, there will be the data distribution gaps between the training and other customers. A good load forecasting model is supposed to be capable of generalizing to customers who did not participate in model training, i.e., unseen domains $\mathcal{D}_{\text{unseen}} = \{(\boldsymbol{x}_i^j, y_i^j)\}_{i=1}^{n_j} | j = M+1, \ldots, M'\}$. Here, M' is the new number of customers considering the unseen customers, where the total number of unseen customers is $M' - M$.

2) Distribution Drift: Since electricity loads are vulnerable to time, weather and social behavior, their data distribution will inevitably vary. Specifically, customers' load consumption habits change over time in unforeseen ways, e.g., appliance upgrades and weather fluctuation. This will result in distribution gaps between the training and real-time data for the same customer, termed as *distribution drift* in this paper. Similarly, the load forecasting model should be able to generalize to customers with distribution drift, i.e., drift domains $\mathcal{D}_{\text{drift}} = \{(\boldsymbol{x}_i^j, y_i^j)\}_{n_j+1}^{n_j'} | j = 1, \ldots, M\}$. Here, n_j' denotes the new data amount of the j-th customer, which means the customer's increased data amount over time is $n_j' - n_j$.

C. Design Goal

Our goal is to build a generalizable load forecasting model by using training domains to realize a minimum forecasting error on all possible domains, which can be formulated as:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\in\mathcal{D}_{\text{all}}}\big[\ell(f(\boldsymbol{x}; \boldsymbol{\theta}), y)\big], \tag{3}$$

where $\mathcal{D}_{\text{all}}$ denotes all possible domains, i.e., $\mathcal{D}_{\text{all}} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{unseen}} \cup \mathcal{D}_{\text{drift}}$, and $\mathcal{D}_{\text{train}} \cap \mathcal{D}_{\text{unseen}} \cap \mathcal{D}_{\text{drift}} = \varnothing$.

III. Causal Inference-Based Paradigm for Model Generalization Improvement

In this section, we elaborate the proposed paradigm based on causal inference for model generalization improvement. We first explore the correlation and causality in existing models. Then, we raise load characteristics and inject them as model extra input, followed by the dissection from a causal view. Finally, we reveal the causal intervention with do-calculus.

A. Analysis of Correlation and Causality in Load Forecasting

To understand how data distribution gaps degrade load forecasting performance, we investigate existing machine learning-based methods, especially those based on deep learning. Since machine learning is built on statistical theory, most methods learn and exploit the statistical correlations between the input features and output for load forecasting [27]. However, these statistical correlations are unstable mapping relationships, in other words, highly dependent on training data. In this way, many machine learning-based methods have been shown to be successful when the test and training data come from the same distribution. Unfortunately, the distribution gap between them is often unavoidable in real applications, which is bound to result in instability and degeneration of model performance.

Unlike correlation, causality is a stable mapping because it portrays intrinsic (i.e., cause-and-effect) relationships between input features and outputs, rather than statistical relationships [28]. Theoretically, the causality is independent of the specific data, thus making it suitable for the data distribution gap problem. It should be noted causality is a special kind of correlation, so we can exploit it to improve the model generalization based on existing correlation-based load forecasting methods.

According to the existing studies, there are three main categories of input features involved in short-term load forecasting, as shown in Figure 1(a): *time* (e.g., hour and day-of-week), *weather* (e.g., temperature and humidity), and *electricity* (e.g., historical loads and related statistical values). For the first two

categories, their causal effects on future loads are apparent and intuitive. For example, date or temperature has a direct decisive impact on future load consumption but not vice versa. This indicates that time and weather features are the cause, while the future load is the corresponding effect. However, things take a turn for the last category, since it is unreasonable if we say historical loads can determine future loads. On the contrary, historical loads have some statistical relationships with future loads, which is correlation instead of causality. Therefore, the performance instability stems from these correlations.

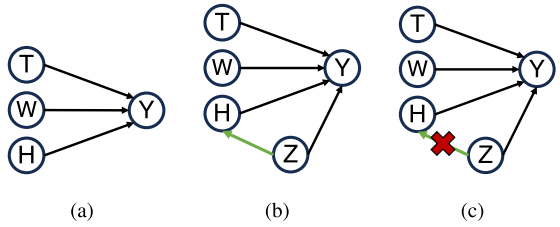

Fig. 1. Causal graphs of load forecasting models: (a) traditional methods; (b) inject load characteristics as an additional input; (c) apply causal intervention after injection to remove spurious correlation. T: time features, W: weather features, H: historical loads, Y: future loads, Z: load characteristics.

B. Causal View of Load Forecasting With Load Characteristics

Since the knowledge of electricity features is indispensable in load forecasting, we need new causal items to replace the correlation one (i.e., historical loads). Inspired by load patterns that provide qualitative insights into customer load habits, we expect to find quantitative content with similar functionality, which we call *load characteristics*. Specifically, load characteristics are implicit features of customers' load habits, which can depict their impact on electricity consumption loads. Unlike load patterns with qualitative descriptions, load characteristics can serve as model inputs for calculation because of their qualitative nature. In addition, since load characteristics can not be obtained directly in accordance with the available data, we design an extraction task and model to extract them from historical loads. With the great representation ability of neural networks in time-series data, load characteristics can be precisely distilled from corresponding historical loads by the extraction model via the extraction task (see details in Section IV-A). Once customers' load habits are grasped by load characteristics, it is natural to determine customers' future loads. Therefore, we believe that load characteristics have causal effects on future loads, rather than correlation.

Considering that load characteristics are derived from the historical loads, there is ineluctably information loss during extraction processes. It is important to mention that historical loads may contain other causal information beneficial to load forecasting except for load habits, e.g., socio-demographics. Furthermore, with the high computational capability of deep learning models, redundancy is better than deficiency in terms of input data [29]. Hence, we still retain historical loads as the model input, instead of discarding them directly. In other words, we introduce load characteristics as an additional input on the basis of input features in existing methods.

After injecting load characteristics, we resort to the causal graph [28] for qualitative analysis, which is a directed acyclic graph. Specifically, we scrutinize the causality in the model and build up a causal graph, as indicated in Figure 1(b), where each node denotes a type of input features. In particular,

- Node T denotes the time features, including the hour, date, and day-of-week of target future loads.
- Node W denotes the weather features, such as temperature and humidity data from the weather forecast.
- Node H denotes the historical loads, such as load records of seven days prior to the target day.
- Node Z denotes the load characteristics extracted from historical loads, i.e., $z = g(x)$, where $g(\cdot)$ is the extraction model. Note that the load characteristics extraction and load forecasting tasks are performed independently.
- Node Y denotes the future loads of the target day.

On the other hand, an edge in the causal graph describes a causation between variables, e.g., $T \rightarrow Y$ represents that T has a causal effect on Y. In particular,

- Edge $\{T, W, H\} \rightarrow Y$ represents that future loads Y are determined by three factors: time features T, weather features W, and historical loads H, which are widely used in existing load forecasting methods. To emphasize, we preserve the cause node H to avoid discarding the possible latent causal effects on Y in historical loads.
- Edge $Z \rightarrow \{H, Y\}$ represents that load characteristics Z can decide both the historical and future loads, because Z embodies the customer's load consumption habits. This is easy to comprehend: regardless of the past or future, customers' electricity consumption is determined by their load habits, and the difference is only whether it has already happened or not. That is why we inventively add a cause node Z to enhance the model generalization.

From the causal graph, we discover that load characteristics Z is a confounder [28] that affects both H and Y. This results in two causal paths starting from Z to Y, i.e., $Z \rightarrow Y$ and $Z \rightarrow H \rightarrow Y$. The first path is what we expected, while the second path goes beyond our initial intention. This is because we expect H provides additional causal effects on Y, rather than further propagating or amplifying causal effects of Z on Y. Moreover, the path $Z \rightarrow H \rightarrow Y$ reveals the effect of H on Y is influenced by the value of Z, i.e., conditional on Z.

Formally, for most existing load forecasting methods, we formulate them as the conditional probability $P(Y|T, W, H)$, and then derive it by the following steps:

$$\begin{aligned} P(Y|T,W,H) &\overset{(1)}{=} \sum_{z\in\mathcal{Z}} P(Y,z|T,W,H) \\ &\overset{(2)}{=} \sum_{z\in\mathcal{Z}} P(Y|T,W,H,z)\ P(z|T,W,H) \\ &\overset{(3)}{=} \sum_{z\in\mathcal{Z}} P(Y|T,W,H,z)\ P(z|H) \\ &\overset{(4)}{=} \sum_{z\in\mathcal{Z}} P(Y|T,W,H,z)\ P(H|z)\ P(z), \end{aligned} \tag{4}$$

where $\mathcal{Z}$ denotes the sample space of load characteristics. In particular, (1) follows the law of total probability; (2) and (4) obey the Bayes' theorem; and (3) holds because T and W are independent to Z according to the causal graph.

It is worth mentioning that there lays a special term $P(H|z)$ in Eq. (4). Suppose that historical loads h_1 and h_2 both correspond to the load characteristics z, where h_1 is common and h_2 is uncommon. In this way, given the identical value of time and weather features, h_1 will have more influence than h_2 on the forecasting model $P(Y|T, W, H)$, since $P(h_1|z)$ is larger than $P(h_2|z)$. In other words, most existing methods pay too much attention to common load data than it deserves, even though forecasting models are expected to treat every historical load data fairly and equally. Consequently, $P(Y|T, W, H)$ will favor the load forecasting under the common load consumption scenarios and be weak in uncommon ones. As a result, the data distribution gaps of historical loads are improperly amplified by $P(H|z)$, which results in the low model generalization.

C. Causal Intervention With Do-Calculus for Confounder

According to the causal theory [28], Z leads to a spurious correlation between H and Y, i.e., $H \leftarrow Z \rightarrow Y$. In this way, the total correlation between H and Y is made up of both 'favorable' causal correlation and 'harmful' spurious correlation. Moreover, this spurious correlation will weaken the stability of model performance [22], which is contrary to our goal. Therefore, $Z \rightarrow H$ needs to be eliminated in the load forecasting model since it brings the bad effect.

To remove the spurious correlation, we consider building a load forecasting model that is immune to the impact of $Z \rightarrow H$. Intuitively, if we can arbitrarily manipulate customers' load behaviors to randomly produce actual consumption loads (also called randomized experiments [28]), the historical loads H are free from load characteristics Z. Under this circumstance, H is completely controlled by our manipulation instead of other factors, which means that $Z \rightarrow H$ no longer exists and accordingly there is no confounder and spurious correlation. However, the feasibility of this approach is quite low, because no one has the authority to forcibly intervene in customers' load consumption behaviors, especially academic researchers. Hence, it is impossible to implement randomized experiments and then recollect intervened data for model training.

Thanks to the progress in causal science, we can get rid of the performing intervention issue by adopting the do-calculus [28], which achieves the same effects using observational data. Formally, $do(H)$ denotes removing the impact of H's parent nodes, i.e., cutting off the edge $Z \rightarrow H$ in Figure 1(b). Specifically, performing $do(H)$ blocks the effect of Z on H, as shown in Figure 1(c). In this way, we formulate our load forecasting model as $P(Y|T, W, do(H))$, and then derive it according to the backdoor adjustment formula [28], as follows:

$$\begin{aligned} P_G(Y|T, W, do(H)) &\overset{(1)}{=} P_{G'}(Y|T, W, H) \\ &\overset{(2)}{=} \sum_{z\in\mathcal{Z}} P_{G'}(Y|T, W, H, z)\ P_{G'}(z|T, W, H) \\ &\overset{(3)}{=} \sum_{z\in\mathcal{Z}} P_{G'}(Y|T, W, H, z)\ P_{G'}(z) \\ &\overset{(4)}{=} \sum_{z\in\mathcal{Z}} P_{G}(Y|T, W, H, z)\ P_{G}(z), \end{aligned} \tag{5}$$

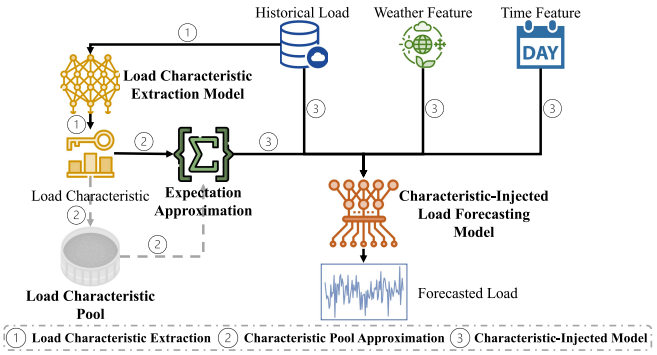

Fig. 2. The causal intervention-based load forecasting framework.

where G and G' represents the causal graphs in Figures 1(b) and 1(c), respectively. The only difference between G and G' is whether the causal intervention is performed. $P_{G'}$ denotes the probability evaluated on G'. For the sake of clarity, we will henceforth substitute P for P_G. To better comprehend, we also explain the derivation in Eq. (5) step by step:

- (1) is based on backdoor criterion, as $do(H)$ blocks the only backdoor path $H \leftarrow Z \rightarrow Y$;
- (2) obeys the law of total probability and Bayes' theorem;
- (2) is because node Z has no dependent variables;
- (4) holds since the causal mechanism $\{T, W, H, Z\} \rightarrow Y$ is consistent on both G and G', which is same for $P(z)$.

According to Eq. (5), causal intervention with do-calculus (i.e., $P(Y|T, W, do(H))$) can be equivalently realized with observational data by calculating $P(Y|T, W, H, z)$ and $P(z)$. Furthermore, compared with Eq. (4), the evil term $P(H|z)$ is removed, which ensures each sample of H has the equivalent weight. This makes sense because the load habit information is already represented by Z, so there is no need to favor one or a group of H as most existing methods. It is worth noting that our load forecasting model can have stable performance even when data distribution gaps exist. This is because the model has seen multiple possible values of Z during training stages, through the mathematical expectation of $P(Y|T, W, H, z)$.

IV. Causal Intervention-Based Load Forecasting Framework

According to the aforesaid theoretical analysis, the model generalization can be enhanced by injected load characteristics and causal intervention with do-calculus. From Eq. (5), we need to first extract the load characteristics Z from historical loads H, then perform the load forecasting $P(Y|T, W, H, z)$ with additional Z, and finally estimate $P(Y|T, W, do(H))$ based on $\sum_z P(Y|T, W, H, z)\ P(z)$. Therefore, we propose a causal intervention-based load forecasting framework to implement it, as shown in Figure 2. Specifically, there are mainly three parts in the framework: *load characteristic extraction*, *characteristic pool approximation* and *characteristic-injected model*, which

Algorithm 1: Causal Intervention Load Forecasting

Input : The model parameters θ, historical load dataset D_{load}, result dataset D_y, weather feature dataset $D_{weather}$, time feature dataset D_{time}, batch size B, training epoch number E, Adam algorithm parameters $\alpha_{Adam}, \beta_1, \beta_2$.

Output: The well-trained load forecasting model $f(\cdot;\theta)$

Initialization:
Train the characteristic extraction model $g(\cdot;\omega_1)$ as per Algorithm 2;
Construct the load characteristic pool P by sampling historical data from D_{load} and then extracting load characteristics with $g(\cdot;\omega_1)$;
Procedure:
for $e = 1, \ldots, E$ **do**
  for *each batch of training data* **do**
    Sample B historical loads $h \sim D_{load}$ and output results $y \sim D_y$, and select the corresponding weather and time features $w \sim D_{weather}, t \sim D_{time}$;
    for $i = 1, \ldots, B$ **do**
      Extract the load characteristic $z_i \leftarrow g(h_i;\omega_1)$;
      Find similar load characteristics of z_i from P, and build approximate set $\hat{\mathcal{Z}}_i' = \{z_i, z_i^1, \ldots, z_i^n, \ldots\}$;
      Calculate z_i's approximate form $\hat{z}_i \leftarrow \frac{1}{|\hat{\mathcal{Z}}_i'|}\sum_{j=0}^{|\hat{\mathcal{Z}}_i'|} z_i^j$;
      Perform load forecasting $\hat{y}_i \leftarrow f(t_i, w_i, h_i, \hat{z}_i;\theta)$;
      Calculate the model's loss $L^{(i)} \leftarrow \|\hat{y}_i - y_i\|^2$;
    end
    Update model's parameters based on Adam algorithm $\theta \leftarrow Adam(\nabla_\theta \frac{1}{B}\sum_{i=1}^{B} L^{(i)}, \alpha_{Adam}, \beta_1, \beta_2)$;
  end
end
return θ

will be expounded successively in the following. In addition, the implementation details of the proposed load forecasting framework are summarized in Algorithm 1.

A. Load Characteristic Extraction

Since customers' load habits can decide their load consumption, we expect load characteristics that represent load habits to expose electricity loads. Here, we design an extraction task to realize it, whose goal is to recover the corresponding load profiles via extracted load characteristics. In more detail, we believe that the extracted load characteristics hold the key information of historical loads, if recovered loads are close to real loads. Formally, the extraction task's objective is as:

$$\min \| \left(g'(z;\omega_2) - h \right) \|_2^2, \text{ where } z = g(h+\epsilon;\omega_1), \tag{6}$$

where $g(\cdot;\omega_1)$ and $g'(\cdot;\omega_2)$ denote the characteristic extraction and load recovery models; h and z represent the values of H and the corresponding Z. The noise ϵ prevents g and g' from degenerating into the linear mapping model.

To accomplish the extraction task, we exploit the encoder-decoder architecture [30], where the encoder distills load characteristics and the decoder accordingly restores load profiles. Moreover, we adopt Transformer model [30] to deal with load data with complex temporal dependencies. In crude terms, the Transformer can focus on relevant parts and ignore useless contents by virtue of the attention mechanism [30]. This lays the foundation for our extraction model to distill key valuable information from historical loads. In particular, we build our extraction model based on the implementation in [31], and the model training procedure is presented in Algorithm 2. In consequence, we omit the exhaustive description of the load characteristic extraction model architecture here.

Algorithm 2: Load Characteristic Extraction

Input : The characteristic extraction model parameters ω_1, load recover model parameters ω_2, historical load dataset D_{load}, batch size B, training epoch number E, Adam algorithm parameters $\alpha_{Adam}, \beta_1, \beta_2$.

Output : The trained characteristic extraction model $g(\cdot;\omega_1)$.

Procedure:
for $e = 1, \ldots, E$ **do**
  for *each batch of training data* **do**
    Sample B load profiles $h \sim D_{load}$ and noises $\epsilon \sim \mathcal{N}(0, I)$;
    Extract the load characteristics $z = g(h+\epsilon;\omega_1)$, and then recover the load profiles $g'(z;\omega_2)$;
    Calculate batch loss $L \leftarrow \frac{1}{B}\sum_B \|g'(z;\omega_2) - h\|$;
    Update two models' parameters based on Adam algorithm $\omega_i \leftarrow Adam(\nabla_{\omega_i} L, \alpha_{Adam}, \beta_1, \beta_2), i = 1, 2$;
  end
end
return ω_1

B. Characteristic Pool Approximation

Because the sample space of Z is theoretically infinite, the calculation of $\sum_{z\in\mathcal{Z}} P(Y|T, W, H, z)\ P(z)$ is intractable [32], especially for high heterogeneity of customer loads. To address this issue, we devise a characteristic pool for approximation, which consists load characteristics extracted from customers' historical loads. Therefore, we can estimate Eq. (5) as follows:

$$P(Y|T, W, do(H)) \approx \sum_{z\in\mathcal{Z}'} P(z) f(T, W, H, z), \tag{7}$$

where $\mathcal{Z}'$ is the load characteristics set from the characteristic pool, and f is the load forecasting model in Section IV-C.

However, Eq. (7) remains difficult to compute as there are a large number of possible values in $\mathcal{Z}'$. Therefore, we derive an efficient approximation to convert the outer summation into the calculation in model f. To be specific, given a historical load h and the corresponding load characteristics z, we use the clustering algorithm (e.g., KNN) to select similar values of z from the designed characteristic pool. Then, we construct the approximate set of $\mathcal{Z}'$ by these similar characteristics. Next, we calculate the average of Z based on the approximate set. Consequently, Eq. (5) can be approximately expressed as:

$$P(Y|T, W, do(H)) \approx f\left(T, W, H, \sum_{z\in\hat{\mathcal{Z}}'} zP(z)\right). \tag{8}$$

where $\hat{\mathcal{Z}}' = \{z, z^1, \ldots, z^n, \ldots\}$ is the approximate set of $\mathcal{Z}'$.

To judge the approximation effect in Eq. (8), we exploit the Jensen gap [33] to measure the approximation error δ:

$$\delta = |\mathbb{E}_z[f(T, W, H, z)] - f(T, W, H, \mathbb{E}_z[z])|, \tag{9}$$

where $\mathbb{E}_z$ displaces the summation operation for clarity.

Theorem 1: If $f : I \to \mathbb{R}$, where I is a closed subset of $\mathbb{R}$ and $\mu \in I$, satisfies the conditions: 1) f is bounded on any compact subset of I; 2) $|f(x) - f(\mu)| = O(|x-\mu|^\alpha)$ at $x \to \mu$ for $\alpha > 0$; 3) $|f(x)| = O(|x|^n)$ at $x \to \infty$ for $n \geq \alpha$. Then

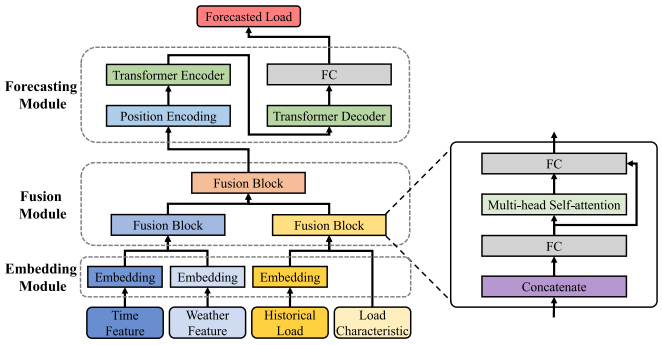

Fig. 3. The architecture of the proposed load forecasting model.

for a random variable X with probability distribution $\mathcal{P}$ and expectation μ, the inequality holds:

$$|\mathbb{E}[f(X) - f(\mu)]| \leq M\left(\sigma_\alpha^\alpha + \sigma_n^n\right),$$

where $M = \sup_{x \in I/\mu} \frac{|f(x)-f(\mu)|}{|x-\mu|^\alpha + |x-\mu|^n}$ does not depend on the probability distribution $\mathcal{P}$; and $\sigma_n = \sqrt[n]{\mathbb{E}[|X-\mu|^n]}$.

Proof: Refer to [33] for detailed proof. ■

It can be proven that most deep learning models including our proposed model $f(\cdot)$, satisfy conditions in Theorem 1, and the upper bound is small [34]. Therefore, according to Theorem 1, there is also a small upper bound of δ, especially when the distribution of $\mathcal{Z}'$ concentrates around its expectation, thus guaranteeing the approximation effect of Eq. (8).

C. Load Characteristic-Injected Load Forecasting Model

According to Eq. (8), the load forecasting model is supposed to receive four types of features as input, i.e., time, weather, historical loads, and corresponding load characteristics, which differs from most existing methods. To this end, we propose a characteristic-injected load forecasting model, which utilize the attention mechanism since input features are time-series. Specifically, the proposed model is composed of three main modules: embedding module, fusion module, and forecasting module, which is illustrated in Figure 3. These three modules are revealed in turn below. Besides, the generalization bound proof of the proposed model can be found in the Appendix.

1) Embedding Module: For better data representation, we apply the embedding transformation to input features. Since time features t are qualitative (e.g., date), we convert them into computable vectors $\boldsymbol{x}_t$ via one-hot encoding function [35]:

$$\boldsymbol{x}_t = FC\big(OneHot(t)\big), \tag{10}$$

where $FC(\cdot)$ denotes the fully-connected layer. Furthermore, although the weather features w are quantitative data, we also embed them into $\boldsymbol{x}_w$ to enhance their information capacity:

$$\boldsymbol{x}_w = FC(w) \ + PE\big(FC(w)\big), \tag{11}$$

where $PE(\cdot)$ represents the positional encoding function that provides relative sequence information [30]. Moreover, since there is only one value at each time point in historical loads h, we employ the fully-connected layer for data embedding:

$$\boldsymbol{x}_h = FC(h) \ + PE\big(FC(h)\big), \tag{12}$$

where $\boldsymbol{x}_h$ is the embed vector of h. Besides, load characteristics z are also converted into $\boldsymbol{x}_z$ by fully-connected layers.

2) Fusion Module: After performing the data embedding, we need to integrate these features that contain different types of information for load forecasting. Moreover, since the load characteristics Z are extracted from historical loads H, it is necessary to blend them in a non-redundant manner, otherwise it may degrade the model performance. For this purpose, we design the fusion block $Fusion(\cdot)$, which employs the attention mechanism and residual connection for better feature fusion:

$$\begin{aligned} Fusion(\boldsymbol{x}_1, \boldsymbol{x}_2) &= FC\big(MHSA(\boldsymbol{x}_{12}) \oplus \boldsymbol{x}_{12}\big) \\ \text{where } \boldsymbol{x}_{12} &= FC\big(Concat(\boldsymbol{x}_1, \boldsymbol{x}_2)\big), \end{aligned} \tag{13}$$

where $MHSA(\cdot)$ denotes the multi-head self-attention function [30]; $Concat(\cdot)$ and $\oplus$ represent the vector concatenation and element-wise addition operators [36]. Symbols $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are two example features to be fused. According to the feature types, the model inputs can be classified into load-relevant (H and Z) and load-irrelevant (T and W). To better dig out feature information and combine them, we adopt a hierarchical route based on $Fusion(\cdot)$. Specifically, we first internally merge the load-relevant and load-irrelevant features separately, and then integrate them to obtain the final fusion feature $\boldsymbol{x}_{\text{fusion}}$:

$$\boldsymbol{x}_{\text{fusion}} = Fusion_3\big(Fusion_1(\boldsymbol{x}_t, \boldsymbol{x}_w), Fusion_2(\boldsymbol{x}_h, \boldsymbol{x}_z)\big), \tag{14}$$

where $Fusion_1$, $Fusion_2$, and $Fusion_3$ are three fusion blocks with different parameters, respectively.

3) Forecasting Module: Considering that we focus on day-ahead load forecasting here, the model needs to output the load sequence of the whole day. To handle time-series relationships, we utilize Transformer once again, where the encoder taps into valuable information from $\boldsymbol{x}_{\text{fusion}}$ and the decoder reconstructs them into target loads. Thus, the model output y is written as:

$$y = FC\Big(TranDec\Big(TranEnc\big(PE(\boldsymbol{x}_{\text{fusion}})\big)\Big)\Big), \tag{15}$$

where $TranEnc(\cdot)$ and $TranDec(\cdot)$ are the encoder and decoder of Transformer, which are implemented with reference to [31].

V. Case Studies

A. Experiment Settings

1) Dataset: We choose smart meter data from Low Carbon London program for the experiments, which includes half-hour electricity loads of more than 5000 customers [37]. After data preprocessing, we retain 375 customers with complete load data from August 1, 2012, to February 27, 2014. To improve the dataset diversity, we randomly select and aggregate 10 customers each time, and finally obtain 200 aggregated customers.

In order to construct the generalization scenarios in Section II-B, we first randomly select 180 aggregated customers with full data from August 2012, to July 2013 (365 days in total) as training dataset. Then we pick all load data from August 2013 to February 2014 (211 days in total) of these 180 aggregated customers, as test dataset for the distribution drift scenario. Finally, we use the rest 20 aggregated customers with a total of 576 days as test dataset for the unseen customer scenario.

TABLE I
IMPLEMENTATION DETAILS OF CASE STUDIES

Parameter	Definition	Value
E	the training epoch number	200
B	the training data batch size	16
α_{adam}	the learning rate of Adam	0.001
(β_1, β_2)	the decay rates of Adam	(0.9, 0.999)
N_{train}	the encoder/decoder layer number	6
h_{head}	the head number of $MHSA(\cdot)$	4
d_{model}	the embedding vector dimension	16

2) Implementation: We implement the proposed method with the open-source machine learning framework PyTorch, and employ the Adam algorithm [38] with mini-batch scheme for model training. Moreover, all experiments are conducted on an Ubuntu 18.04 LTS platform, which is equipped with the Intel Core i9-10980XE CPU and NVIDIA GeForce RTX 3090 GPU. The implementation details are summarized in Table I.

3) Benchmarks and Metrics: To verify the effectiveness and superiority of our proposed method, we select the following six benchmarks from state-of-the-art studies for comparison:

- B1: A convolutional LSTM-based neural network with selected autoregressive features proposed in 2021 by Li et al. [39] to improve short-term forecasting accuracy.
- B2: A hybrid deep learning model combining LSTM and self-attention mechanism proposed in 2021 by Zang et al. [40] for day-ahead residential load forecasting.
- B3: An online adaptive RNN-based method with continuous learning proposed in 2021 by Fekri et al. [41] to handle newly arriving data and adapt to new patterns.
- B4: An adaptive sparse attention network proposed in 2023 by Deng et al. [42] to increase the anti-interference ability for electric load forecasting.
- B5: An online-offline deep kernel learning with deep soft Spiking Neural Networks proposed in 2023 by Li et al. [18] to address the high uncertainty of residential loads.
- B6: A temporal data pooling framework with recurrent deep embedding and meta-initialization proposed in 2022 by Yang and Youn [19] to achieve the robust accuracy under concept drift for short-term individual load forecasting.

In addition, to evaluate the load forecasting effect, we adopt three common performance metrics: RMSE, MAE and MAPE.

B. Performance Comparison With Load Forecasting Methods

In this part, we validate the load forecasting performance of our proposed method in two generalization scenarios. For brevity, we denote unseen customer scenario as Scenario A and distribution drift scenario as Scenario B. As stated in Section V-A1, since the time period of the test dataset for Scenario B lasts for 7 months, we believe that there is the distribution drift in these test data. Besides, the aggregated customers combined from individuals randomly, facilitate to simulate Scenario A, because their load patterns have commonalities but are not identical. Therefore, we believe the test datasets are representative of two generalization scenarios. To comprehensively verify the proposed method, we compare its performance with the above benchmarks. Moreover, we repeat all experiments 5 times to avoid interference errors and get average outcomes.

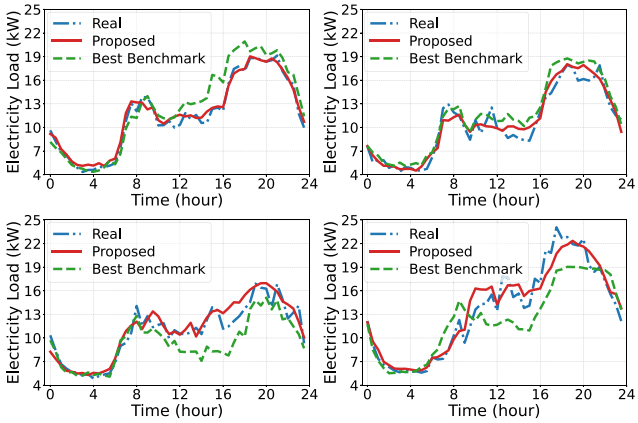

Fig. 4. Examples of load forecasting effect under Scenario A. Each figure represents a single day's load profiles for each unknown customer.

Table II summarizes the performance comparison results. It can be observed that our proposed method outperforms all benchmarks in Scenario A, with the MAPE of 9.03%. This consequence also holds true in Scenario B, where the MAPE is within 8.44%. Specifically, compared with all benchmarks, the decline ranges of our proposed method in Scenario A vary from 37.1% to 66.5% and from 34.7% to 69.9% in terms of RMSE and MAE, respectively. A similar situation occurs in Scenario B, even to a greater extent, e.g., the RMSE reduction rate of the proposed method is between 42.2% to 70.9%, and MAE between 37.2 to 71.9%. It should be noted that all evaluation metrics of our proposed method in Scenario B are lower than those in Scenario A, which also basically applies to all benchmarks. We believe it is reasonable because the data distribution gap incurred by the distribution drift should be relatively smaller than the unknown customer. Furthermore, for benchmarks that consider model generalization (B5 and B6), their forecasting performance is significantly improved compared to other benchmarks, where the MAPEs are reduced at least 0.98% and 1.03% in two scenarios. However, although B5 and B6 enhance the model generalization ability from the model perspective (i.e., increase model complexity), their RMSE and MAE are still on average 0.73 kW and 0.59 kW higher than our proposed model. This indicates the effectiveness and necessity of improving model generalization from the data perspective, which is exactly adopted in the proposed method.

To intuitively inspect the performance improvement of our proposed method, we provide a visual demonstration of load forecasting compared with benchmarks. As for Scenario A, we first randomly select four customers from the corresponding test dataset, and then pick up their one-day load profile as examples. We only display forecasting results by our proposed method and best benchmark for the sake of brevity, as shown in Fig. 4. It is clear our proposed method achieves

TABLE II
NUMERICAL RESULTS OF PERFORMANCE COMPARISON FOR LOAD FORECASTING METHODS UNDER TWO SCENARIOS

	Scenario A - Unseen Customer			Scenario B - Distribution Drift		
	RMSE (kW)	MAE (kW)	MAPE (%)	RMSE (kW)	MAE (kW)	MAPE (%)
B1	3.3078 (1.4024)	3.0548 (1.2985)	16.2373 (3.2276)	3.2014 (0.7197)	2.8838 (0.7414)	15.9228 (3.4006)
B2	2.9556 (1.5268)	2.8943 (1.1042)	15.9046 (3.0509)	2.8724 (0.6837)	2.8263 (0.7342)	15.5755 (3.3049)
B3	2.6719 (1.1585)	2.3377 (0.9934)	15.1672 (3.2453)	2.3430 (0.6271)	2.2283 (0.6346)	14.9936 (2.7045)
B4	2.4382 (0.9067)	2.1851 (0.8496)	14.7206 (2.9971)	2.0958 (0.3567)	1.6581(0.4339)	14.2118 (2.7204)
B5	1.9558 (0.6183)	1.6304 (0.7142)	13.7382 (2.8786)	1.6549 (0.2773)	1.4871 (0.3237)	13.1857 (2.6187)
B6	1.7612 (0.4709)	1.4074 (0.4515)	13.2068 (2.1562)	1.6127(0.2601)	1.2895(0.2369)	12.9843 (2.7188)
Proposed	**1.1076 (0.2683)**	**0.9188 (0.2005)**	**9.0286 (1.1393)**	**0.9314 (0.2040)**	**0.8095 (0.1892)**	**8.4381(2.0597)**

*Each entry gives a pair of mean and standard deviation.

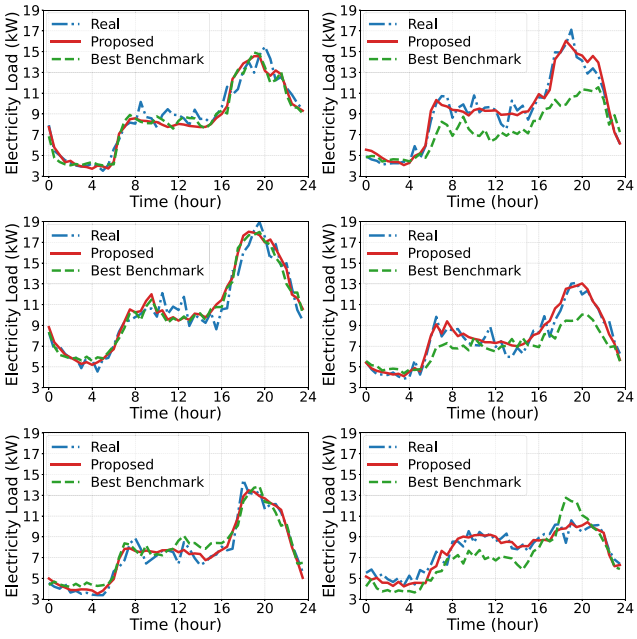

Fig. 5. Examples of load forecasting effect under Scenario B. Each row represents two daily load profiles for each customer (left column: before distribution drift; right column: after distribution drift).

good forecasting effects, regardless of the pattern of real load profiles. In contrast, the benchmark suffers performance fluctuation due to the generalization issue. According to these examples, the accuracy and stability of our proposed method are verified.

Similarly, we also choose three customers for Scenario B at random, and select their daily load profiles from the training and test dataset as examples, respectively. Fig. 5 illustrates the impact of distribution drift in load forecasting. Specifically, for load profiles prior to the distribution drift event, our proposed method and benchmark can both realize accurate forecasting, reflected in the left column of Fig. 5. However, when distribution drift occurs (the right column), the benchmark shows performance degradation while our proposed method maintains a small forecasting error as before. Considering the stochastic nature of distribution drift, this stable performance further validates the generalization of our proposed model.

TABLE III
NUMERICAL RESULTS OF PERFORMANCE COMPARISON WITH THE STATE-OF-THE-ART TIME SERIES FORECASTING METHODS

Method	RMSE (kW)	MAE (kW)	MAPE (%)
B7-1	1.2164 (0.2879)	1.0149 (0.2739)	9.1327 (2.2072)
B7-2	1.1984 (0.3105)	0.9908 (0.2415)	9.0266 (2.0872)
B7-3	1.1496 (0.2661)	0.9553 (0.2289)	8.8605 (2.1064)
B7-4	1.1072 (0.2715)	0.9080 (0.1965)	8.6121 (2.0242)
Proposed	**0.9652 (0.2218)**	**0.8158 (0.1936)**	**7.9193 (1.9325)**

*Each entry gives a pair of mean and standard deviation.

C. Performance Comparison With State-of-the-Art Domain Generalization Methods

In this part, we further verify the superiority of our proposed method by comparing it with advanced domain generalization methods. Specifically, we select four state-of-the-art studies for time series forecasting as benchmarks, as follows:

- B7-1: A gradient interpolation-based model with time-sensitive parameters proposed in 2021 by Nasery et al. [43] to allow the decision boundary to change along time.
- B7-2: An attention-based shared module using domain-invariant latent features proposed in 2022 by Jin et al. [44] to enable joint training on source and target domains.
- B7-3: A temporal domain generalization with drift-aware dynamic neural network framework proposed in 2023 by Bai et al. [45] to predict in the future without future data.
- B7-4: A domain discrepancy regularization-based time series model proposed in 2024 by Deng et al. [46] to enforce consistent performance across different domains.

To ensure a consistent comparison, we use the same dataset of generalization scenarios described in Section V-A1. Similarly, all numerical experiments are repeated 5 times to prevent human interference, and we calculate the average value as the results. Moreover, we combine the forecasting results of two scenarios for demonstration, which are presented in Table III.

It can be seen that our proposed method achieves the best prediction performance, with MAPE within 8%. Although these advanced methods have made progress compared to other benchmarks (i.e., B1–B6), they are still inferior to our proposed model. Specifically, on the basis of the proposed model, there is at least a rise of 14.7% and 11.3% for these four benchmarks in terms of RMSE and MAE, respectively.

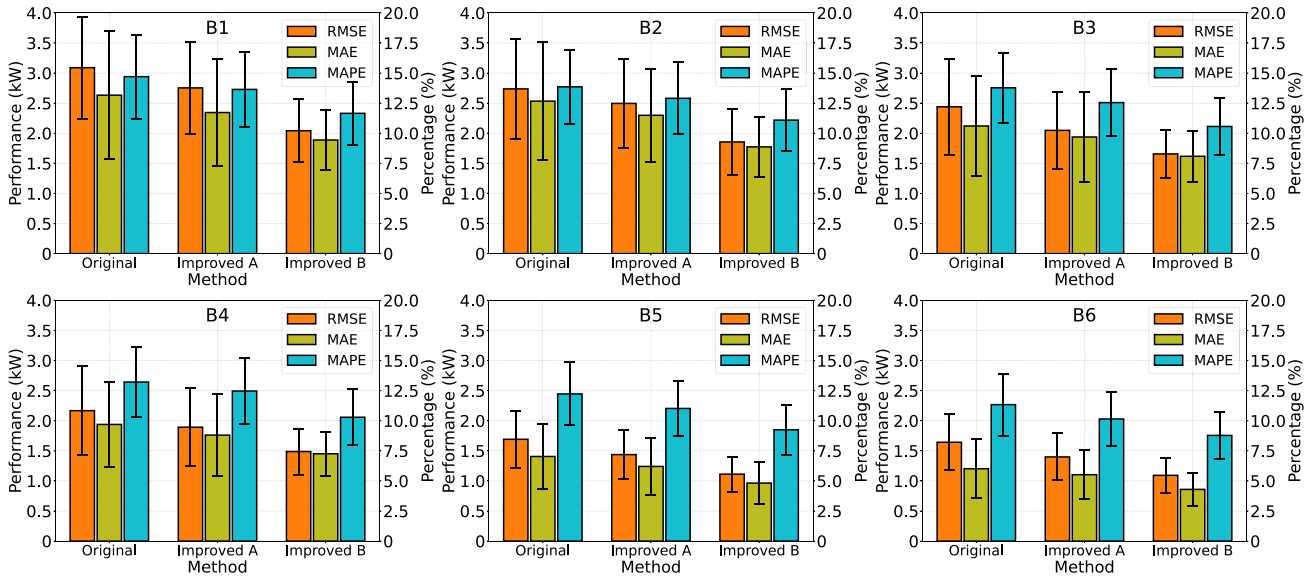

Fig. 6. The performance improvement of benchmarks from load forecasting studies (i.e., B1–B6) by adopting the components in our proposed framework. Improved A: only inject load characteristics as extra model input; Improved B: inject load characteristics and apply characteristic pool approximation.

Moreover, the forecast error percentage (MAPE) is also increased by up to 15.3%. Despite this, both benchmarks and our proposed method exhibit good performance in load forecasting, where each of their MAPEs falls within 9.2%. However, our proposed method is the only one that can reach a MAPE within 8%, and its RMSE and MAE are controlled within 1 kW and 0.9 kW. Therefore, this comprehensive comparison further validates the effectiveness and superiority of our proposed method.

D. Forecasting Performance Improvement of Benchmarks

In this part, we conduct the ablation study to explore the impact of each component in our proposed framework on the model performance, which further demonstrates the effectiveness of the proposed method. According to the introduction in Section IV, there are mainly three parts in our proposed framework: 1) load characteristic extraction; 2) characteristic pool approximation; 3) characteristic-injected load foresting model. Since the first two parts are model-agnostic that can be applied to different models, we apply them to benchmarks sequentially to explore their importance differences. Specifically, we first add load characteristics as an additional input of benchmarks (denoted as Improved A), and compare its performance with the original form of benchmarks. On this basis, we continue to add the characteristic approximation to benchmarks (denoted as Improved B), and analyze its impact on load forecasting performance. In addition, we discern the efficacy of the proposed model by replacing it in our framework with benchmarks. Note that we conduct the experiment under two scenarios, and blend the forecasting results to calculate average values.

Fig. 6 visually displays the performance improvement of all benchmarks by using components from our proposed method. It can be easily seen that after injecting load characteristics as extra input, benchmarks' evaluation metrics show a certain decline, with a maximum reduction in MAPE of 1.2%. This proves that introducing causal items (i.e., load characteristics) as model inputs is beneficial to improve model generalization. When we further apply the characteristic approximation to benchmarks, there is a significant improvement in forecasting accuracy. In particular, compared to solely adding load characteristics, the RMSE and MAE are reduced by an average of about 26.8% and 28.9%, respectively. This may be because benchmarks are no longer disturbed by spurious correlations, with the help of the characteristic approximation. Moreover, the performance stability of benchmarks has a greater enhancement after adding the characteristic approximation, where the standard deviation of all metrics shows a more substantial decline. Therefore, the characteristic approximation (i.e., causal intervention with do-calculus) makes more contribution to the model generalization improvement.

After employing the first two components of the proposed framework, all benchmarks make great strides in model performance. Taking B6 as an example, with the load characteristics and characteristic approximation, its RMSE and MAE are both within 1.1 kW as well as its MAPE is around 8.8%, which is a dramatic diminution from the original version. However, its evaluation metrics are still inferior to those of our proposed method. Specifically, the performance discrepancies between B6 and the proposed method are 0.15 kW, 0.13 kW, and 1.38% in terms of RMSE, MAE, and MAPE, respectively. This indicates that the characteristic-injected model also contributes to addressing generalization issues, especially the fusion module for integrating input features. Therefore, the effectiveness and superiority of our proposed method are further demonstrated.

VI. Conclusion

In this paper, we concentrate on short-term customer load forecasting with model generalization issues. Owing to the

heterogeneity and stochasticity of customers' electricity loads, existing machine learning-based load forecasting methods encounter serious challenges of performance degradation. To address this problem, we propose a causal inference-based method to build the generalization-improved load forecasting models. Specifically, we introduce load characteristics as extra input with causality, and exploit the causal intervention with do-calculus to remove spurious correlations for generalization improvement. Furthermore, we propose a novel load forecasting framework to efficiently implement the causal intervention using only observational data. Case studies comprehensively validate that the proposed method outperforms all benchmarks from the state-of-the-art studies in two generalization scenarios. The RMSE and MAE of our proposed method are controlled within 1 kW and 0.9 kW, and its MAPE is the only method within 8%. Moreover, the performance improvement of benchmarks can reach up to within 1.1 kW for both RMSE and MAE. This further demonstrates that our proposed method can be applied to different load forecasting models.

With the penetration of distributed energy resources, the actual load profiles recorded by smart meters will be affected by renewable generation, which will further exacerbate the distribution drift of load data. Therefore, it is necessary to make our proposed method adaptable to customers with renewable generation, which will be considered in our future work. In addition, since it is difficult to identify causality between variables, we intend to design the causal discovery method to efficiently find causal terms in the future.

APPENDIX
GENERALIZATION BOUND PROOF

This Appendix provides a theoretical analysis and proof of the generalization bound for our proposed method. In this paper, we focus on the short-term customer load forecasting task, which is formulated in Eq. (1). Hence, the model prediction error can be written as follows:

$$R(f) = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim P_{XY}}\big[\ell(f(\boldsymbol{x};\boldsymbol{\theta}),\boldsymbol{y})\big], \tag{16}$$

where $R(f)$ is called the generalization risk, which is the model prediction error on the overall dataset. However, the overall dataset is not available, so we usually use the model prediction error on the training dataset as an approximation:

$$\hat{R}(f) = \frac{1}{|D|}\sum_{i=1}^{|D|}\big[\ell\big(f(\boldsymbol{x}_i;\boldsymbol{\theta}),\boldsymbol{y}_i\big)\big], \tag{17}$$

where $\hat{R}(f)$ is called the empirical risk; D denotes the training dataset with the size of $|D|$. Our goal is to find the optimal parameters so that $R(f)$ will be the smallest, but we can only measure $\hat{R}(f)$. Therefore, we use the generalization bound to bound the difference between $R(f)$ and $\hat{R}(f)$ [47].

Let us first consider the case where the hypothesis space $\mathcal{H}$ is finite, with size $\dim(\mathcal{H}) = |\mathcal{H}|$. In other words, we select a hypothesis (i.e., load forecasting model) from a finite list.

Theorem 1: For any data distribution p^*, and any dataset $\mathcal{D}$ of size m drawn from p^*, the probability that the generalization error will be more than $\epsilon \in (0, 1)$, is upper bounded:

$$P\Big(|R(f) - \hat{R}(f)| > \epsilon\Big) \le 2|\mathcal{H}|e^{-2m\epsilon^2}, \forall f \in \mathcal{H}. \tag{18}$$

Proof: Refer to [47] for detailed proof. ■

Since we adopt neural networks for load forecasting, $\mathcal{H}$ is usually infinite [48] and we cannot use $\dim(\mathcal{H}) = |\mathcal{H}|$. Thus, we exploit the Vapnik–Chervonenkis (VC) dimension [48] to measure the degrees of freedom of $\mathcal{H}$. Because the calculation of VC dimension is hard, we use the Sauer's Lemma [48] to compute the upper bound of the VC dimension by the growth function. Now we consider the case where $\mathcal{H}$ is infinite.

Theorem 2: For a hypothesis space $\mathcal{H}$ and any dataset $\mathcal{D}$ of size m, the following generalization bound holds for ϵ:

$$P\Big(|R(f) - \hat{R}(f)| > \epsilon\Big) \le 4\Pi_{\mathcal{H}}(2m)e^{-\frac{m\epsilon^2}{8}}, \forall f \in \mathcal{H}. \tag{19}$$

Proof: Refer to [47] for detailed proof. ■

According to Eq. (19), we can obtain the upper bound of the generalization risk $R(f)$ based on the VC dimension:

$$R(f) \le \hat{R}(f) + \sqrt{\frac{8d\ln\frac{2em}{d} + 8\ln\frac{4}{\delta}}{m}}, \forall f \in \mathcal{H}, \tag{20}$$

where d is the VC dimension; $\delta \in (0, 1)$ is confidence level.

According to Eq. (20), the optimism of $R(f)$ increases with d but decreases with $|D| = m$, as is to be expected. Compared with existing methods, our proposed method does not increase the model complexity and we use the same dataset for model training (i.e., d is not risen and m is constant). Therefore, the second term on the right side of Eq. (20) does not increase, which proves the effectiveness of our proposed method.

REFERENCES

[1] T. M. Khanna et al., "A multi-country meta-analysis on the role of behavioural change in reducing energy consumption and CO2 emissions in residential buildings," *Nat. Energy*, vol. 6, no. 9, pp. 925–932, 2021.

[2] I. Antonopoulos et al., "Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review," *Renew. Sustain. Energy Rev.*, vol. 130, Sep. 2020, Art. no. 109899.

[3] S. Aslam, H. Herodotou, S. M. Mohsin, N. Javaid, N. Ashraf, and S. Aslam, "A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids," *Renew. Sustain. Energy Rev.*, vol. 144, Jul. 2021, Art. no. 110992.

[4] Z. Wang, P. Yu, and H. Zhang, "Privacy-preserving regulation capacity evaluation for HVAC systems in heterogeneous buildings based on federated learning and transfer learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 3535–3549, Sep. 2023.

[5] J. Luo, T. Hong, Z. Gao, and S.-C. Fang, "A robust support vector regression model for electric load forecasting," *Int. J. Forecast.*, vol. 39, no. 2, pp. 1005–1020, 2023.

[6] P. Li, B. Zhang, Y. Weng, and R. Rajagopal, "A sparse linear model and significance test for individual consumption prediction," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4489–4500, Nov. 2017.

[7] T. Teeraratkul, D. O'Neill, and S. Lall, "Shape-based approach to household electric load curve clustering and prediction," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5196–5206, Sep. 2017.

[8] J. Li et al., "A novel hybrid short-term load forecasting method of smart grid using MLR and LSTM neural network," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2443–2452, Apr. 2021.

[9] W. Lin, D. Wu, and B. Boulet, "Spatial-temporal residential short-term load forecasting via graph neural networks," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5373–5384, Nov. 2021.

[10] Y. Zhou, Z. Ding, Q. Wen, and Y. Wang, "Robust load forecasting towards adversarial attacks via Bayesian learning," *IEEE Trans. Power Syst.*, vol. 38, no. 2, pp. 1445–1459, Mar. 2023.

[11] C. Wan, Z. Cao, W.-J. Lee, Y. Song, and P. Ju, "An adaptive ensemble data driven approach for nonparametric probabilistic forecasting of electricity load," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 5396–5408, Nov. 2021.

[12] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019.
[13] M. Grabner, Y. Wang, Q. Wen, B. Blažič, and V. Štruc, "A global modeling framework for load forecasting in distribution networks," *IEEE Trans. Smart Grid*, vol. 14, no. 6, pp. 4927–4941, Nov. 2023.
[14] Y. Yang, W. Li, T. A. Gulliver, and S. Li, "Bayesian deep learning-based probabilistic load forecasting in smart grids," *IEEE Trans. Ind. Informat.*, vol. 16, no. 7, pp. 4703–4713, Jul. 2020.
[15] D. Qin, C. Wang, Q. Wen, W. Chen, L. Sun, and Y. Wang, "Personalized federated DARTS for electricity load forecasting of individual buildings," *IEEE Trans. Smart Grid*, vol. 14, no. 6, pp. 4888–4901, Nov. 2023.
[16] L. Von Krannichfeldt, Y. Wang, and G. Hug, "Online ensemble learning for load forecasting," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 545–548, Jan. 2021.
[17] P. Yu, H. Zhang, Y. Song, H. Hui, and G. Chen, "District cooling system control for providing operating reserve based on safe deep reinforcement learning," *IEEE Trans. Power Syst.*, vol. 39, no. 1, pp. 40–52, Jan. 2024.
[18] Y. Li, F. Zhang, Y. Liu, H. Liao, H.-T. Zhang, and C. Chung, "Residential load forecasting: An online-Offline deep kernel learning method," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4264–4278, Mar. 2024, doi: 10.1109/TPWRS.2023.3299637.
[19] E. Yang and C.-H. Youn, "Temporal data pooling with Meta-Initialization for individual short-term load forecasting," *IEEE Trans. Smart Grid*, vol. 14, no. 4, pp. 3246–3258, Jul. 2022.
[20] J. Wang et al., "Generalizing to unseen domains: A survey on domain generalization," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 8, pp. 8052–8072, Aug. 2023.
[21] J. G. Richens, C. M. Lee, and S. Johri, "Improving the accuracy of medical diagnosis with causal machine learning," *Nat. Commun.*, vol. 11, no. 1, p. 3923, 2020.
[22] P. Cui and S. Athey, "Stable learning establishes some common ground between causal inference and machine learning," *Nat. Mach. Intell.*, vol. 4, no. 2, pp. 110–115, 2022.
[23] Z. Zeng, W. Peng, and D. Zeng, "Improving the stability of intrusion detection with causal deep learning," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 4750–4763, Dec. 2022.
[24] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access J. Power Energy*, vol. 7, pp. 376–388, 2020.
[25] Z. Wang and H. Zhang, "Customized load profiles synthesis for electricity customers based on conditional diffusion models," *IEEE Trans. Smart Grid*, vol. 15, no. 4, pp. 4259–4270, Jul. 2024.
[26] B. Li, G. Xiao, R. Lu, R. Deng, and H. Bao, "On feasibility and limitations of detecting false data injection attacks on power grid state estimation using D-FACTS devices," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 854–864, Feb. 2020.
[27] J. Runge, A. Gerhardus, G. Varando, V. Eyring, and G. Camps-Valls, "Causal inference for time series," *Nat. Rev. Earth Environ.*, vol. 4, no. 7, pp. 487–505, 2023.
[28] J. Pearl, *Causality*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[29] J.-A. Chen, W. Niu, B. Ren, Y. Wang, and X. Shen, "Survey: Exploiting data redundancy for optimization of deep learning," *ACM Comput. Surv.*, vol. 55, no. 10, pp. 1–38, 2023.
[30] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
[31] Z. Wang and H. Zhang, "Customer baseline load estimation for virtual power plants in demand response: An attention mechanism-based generative adversarial networks approach," *Appl. Energy*, vol. 357, May 2024, Art. no. 122544.
[32] W. Wang, F. Feng, X. He, X. Wang, and T.-S. Chua, "Deconfounded recommendation for alleviating bias amplification," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2021, pp. 1717–1725.
[33] S. Abramovich and L.-E. Persson, "Some new estimates of the 'Jensen gap'," *J. Inequal. Appl.*, vol. 2016, pp. 1–9, Feb. 2016. [Online]. Available: https://link.springer.com/article/10.1186/s13660-016-0985-4
[34] X. Gao, M. Sitharam, and A. E. Roitberg, "Bounds on the Jensen gap, and implications for mean-concentrated distributions," 2017, *arXiv:1712.05267*.
[35] S. Okada, M. Ohzeki, and S. Taguchi, "Efficient partition of integer optimization problems with one-hot encoding," *Sci. Rep.*, vol. 9, no. 1, 2019, Art. no. 13036.
[36] H. Huang, T. Li, B. Li, W. Wang, and Y. Sun, "A bidirectional differential evolution based unknown cyberattack detection system," *IEEE Trans. Evol. Comput.*, early access, Feb. 13, 2024, doi: 10.1109/TEVC.2024.3365101.
[37] J. R. Schofield et al., 2015, "Low carbon London project: Data from the dynamic time-of-use electricity pricing trial, 2013," Dataset, UK Data Service. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://doc.ukdataservice.ac.uk/doc/7857/mrdoc/pdf/7857_userguide.pdf
[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[39] L. Li, C. J. Meinrenken, V. Modi, and P. J. Culligan, "Short-term apartment-level load forecasting using a modified neural network with selected auto-regressive features," *Appl. Energy*, vol. 287, Apr. 2021, Art. no. 116509.
[40] H. Zang et al., "Residential load forecasting based on LSTM fusing self-attention mechanism with pooling," *Energy*, vol. 229, Aug. 2021, Art. no. 120682.
[41] M. N. Fekri, H. Patel, K. Grolinger, and V. Sharma, "Deep learning for load forecasting with smart meter data: Online adaptive recurrent neural network," *Appl. Energy*, vol. 282, Jan. 2021, Art. no. 116177.
[42] Y. Deng, X. Wang, and Y. Liao, "ASA-net: Adaptive sparse attention network for robust electric load forecasting," *IEEE Internet Things J.*, vol. 11, no. 3, pp. 4668–4678, Feb. 2024, doi: 10.1109/JIOT.2023.3300695.
[43] A. Nasery, S. Thakur, V. Piratla, A. De, and S. Sarawagi, "Training for the future: A simple gradient interpolation loss to generalize along time," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 19198–19209.
[44] X. Jin, Y. Park, D. Maddix, H. Wang, and Y. Wang, "Domain adaptation for time series forecasting via attention sharing," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 10280–10297.
[45] G. Bai, C. Ling, and L. Zhao, "Temporal domain generalization with drift-aware dynamic neural networks," 2022, *arXiv:2205.10664*.
[46] S. Deng, O. Sprangers, M. Li, S. Schelter, and M. de Rijke, "Domain Generalization in time series forecasting," *ACM Trans. Knowl. Disc. Data*, vol. 18, no. 5, pp. 1–24, 2024.
[47] K. P. Murphy, *Probabilistic Machine Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2022.
[48] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.

Zhenyi Wang (Graduate Student Member, IEEE) received the B.E. degree in cybersecurity from Sichuan University, Chengdu, China, in 2021. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of Macau, Macau, China.

His research interests include data analytics in demand response and electricity market, trustworthy and data-centric AI for power systems, and the intersection of causal inference with AI.

Hongcai Zhang (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 2013 and 2018, respectively.

He is currently an Assistant Professor with the State Key Laboratory of Internet of Things for Smart City and the Department of Electrical and Computer Engineering, University of Macau, Macau, China. From 2018 to 2019, he was a Postdoctoral Scholar with the University of California at Berkeley, Berkeley. His current research interests include Internet of Things for smart energy, optimal operation and optimization of power and transportation systems, and grid integration of distributed energy resources. He is an Associate Editor of IEEE TRANSACTIONS ON POWER SYSTEMS and of *Journal of Modern Power Systems and Clean Energy*.

Ruixiong Yang received the B.E. degree in electrical engineering and automation from the South China University of Technology, Guangzhou, Guangdong, China, in 2009.

He is currently a researcher with the DC Distribution and Consumption Center, Guangdong Power Grid Company Ltd. His research interests include flexible distribution network planning and optimized operation, distribution network fault detection, and self-healing.

Yong Chen received the B.E. degree in electrical engineering from Sichuan University, Chengdu, China, in 2004, and the master's degree in power system engineering from Xi'an Jiaotong University in 2007.

He is currently the Deputy Director of the DC Distribution and Consumption Center, Guangdong Power Grid Company Ltd. His research interests include modeling of power electronic converters, integration of power electronic systems, and clean energy supply systems.