

# Customized Load Profiles Synthesis for Electricity Customers Based on Conditional Diffusion Models

Zhenyi Wang, *Graduate Student Member, IEEE*, and Hongcai Zhang, *Senior Member, IEEE*

**Abstract**—Customers’ load profiles are critical resources to support data analytics applications in modern power systems. However, there are usually insufficient historical load profiles for data analysis, due to the collection cost and data privacy issues. To address such data shortage problems, load profiles synthesis is an effective technique that provides synthetic training data for customers to build high-performance data-driven models. Nonetheless, it is still challenging to synthesize high-quality load profiles for each customer using generation models trained by the respective customer’s data owing to the high heterogeneity of customer load. In this paper, we propose a novel customized load profiles synthesis method based on conditional diffusion models for heterogeneous customers. Specifically, we first convert the customized synthesis into a conditional data generation issue. We then extend traditional diffusion models to conditional diffusion models to realize conditional data generation, which can synthesize exclusive load profiles for each customer according to the customer’s load characteristics and application demands. In addition, to implement conditional diffusion models, we design a noise estimation model with stacked residual layers, which improves the generation performance by using skip connections. The attention mechanism is also utilized to better extract the complex temporal dependency of load profiles. Finally, numerical case studies based on a public dataset are conducted to validate the effectiveness and superiority of the proposed method.

**Index Terms**—Conditional diffusion models, data analytics applications, deep generative model, load profiles synthesis

## I. INTRODUCTION

**I**N order to build a new generation of power systems to achieve carbon neutrality, it is necessary to improve the digitalization and intelligence of the grid [1]. As an advanced type of metering infrastructure, smart meters are widely deployed to facilitate the integration of energy flow and information flow, consequently improving the efficiency and reliability of power systems [2]. On the one hand, relying on a large amount of fine-grained load profiles collected by smart meters, utilities can provide personalized and high-quality services to customers, and reduce their operation costs. On the other hand, customers can be more aware of their power consumption characteristics, save energy costs, and even promote participation in demand response [3]–[5].

However, building data-driven models for data analytics applications, especially deep learning models, usually requires massive amounts of training data [6]. Unfortunately, because

of the cost, regulatory, and privacy concerns, it is often challenging for utilities, let alone customers, to acquire sufficient training data [7]. To tackle data shortages, some researchers have proposed federated learning to collaboratively train models among multiple customers, which has been extensively used in power systems [8]. Federated learning can improve the performance of data-driven models, but it requires cooperation among customers to complete tasks. Because customers may have different goals and needs, they are likely to be unwilling to collaborate on certain tasks, which in turn will lead to federated learning not meeting each customer’s demands.

In contrast to federated learning, load profiles synthesis does not directly train specialized models for customers. Instead, it generates massive high-fidelity load data for customers so that they can train their own data-driven models. Thus, load profiles synthesis is another effective approach to tackle data deficiency, and many studies have been devoted to this field in recent years. The published methods can generally be divided into two categories: *model-based* methods and *data-driven* methods. The model-based methods generate load profiles based on mathematical physical modeling of customer loads. López et al. [9] simulated residential energy consumption by modeling the physical properties of household appliances and buildings. Diao et al. [10] developed a physical model-based load synthesis method for residential buildings, which identifies and simulates the load pattern of each household component. However, these methods require accurate physical models and domain knowledge, which makes them difficult to popularize across different scenarios. Furthermore, model-based methods often adopt approximation techniques, which may result in large deviations in generated load profiles.

To solve the above problems, data-driven approaches that learn the load characteristics based on historical load profiles and then generate new load data for customers, have been widely adopted. Al-Wakeel et al. [11] proposed a load estimation method based on the k-means algorithm to generate missing and future values of smart meters. Labeeuw et al. [12] combined the mixture model clustering and Markov models to synthesize residential loads in a top-down manner. Furthermore, with the rapid development of deep learning, some researchers have proposed load profiles generation methods based on deep generative models, especially adversarial generative networks (GAN). For example, El Kababji et al. [13] proposed a flexible framework based on GAN to generate synthetic labeled load patterns and usage habits at the appliance level. Song et al. [14] leveraged a GAN-based model and weather data to recover high-resolution load profiles from low-resolution load profiles through a two-stage process.

This paper is funded in part by the Science and Technology Development Fund, Macau SAR (File no. 001/2024/SKL, and File no. 0011/2021/AGJ). (Corresponding author: *Hongcai Zhang*.)

Z. Wang, and H. Zhang are with the State Key Laboratory of Internet of Things for Smart City and Department of Electrical and Computer Engineering, University of Macau, Macao, 999078 China (email: hc Zhang@um.edu.mo).

Huang et al. [15] developed a Wasserstein GAN approach with differential privacy to transform a real-world dataset into an anonymous synthetic load dataset to preserve customers' data privacy. However, most existing data-driven methods are unconditional, which means that the characteristics and patterns of the load profiles to be synthesized cannot be specified. Thus, these unconditional generation methods cannot satisfy heterogeneous customers' demands to synthesize highly personalized load profiles [16].

To address the above issue, conditional information (e.g., season or typical load) needs to be added to the load profiles synthesis process as target guidance. Recently, some studies have focused on conditional data generation in power systems. Chen et al. [16] presented a novel load generation model using conditional GAN to produce personalized energy consumption data according to the customer label. Moreover, Chen et al. [17] proposed a conditional GAN-based approach for scenario generation, which generates realistic wind and photovoltaic power profiles conditioned on weather events or time information. However, these methods can only use discrete values as conditions, which is not suitable for power systems where most information, e.g., historical loads, are continuous [18]. To aid in data generation using continuous values as conditional information, Zhang et al. [18] proposed flow-based conditional generative models with reversible transformation architectures to forecast residential load profiles according to the previous day's load observations. However, the generalizability of flow-based models is limited because they need specialized architectures designed case-by-case to construct reversible transforms.

To address the above problems, we exploit diffusion models to perform the load profiles synthesis. Diffusion models add random noises to original data through a diffusion Markov process, and then learn to denoise the data via the reverse Markov process for data generation [19]. Hence, load profiles can be generated by sampling a random vector from the prior distribution, followed by denoising through the reverse Markov chain. Unlike flow-based models, diffusion models learn the characteristics of load profiles through a fixed Markov process, so they have higher reusability and generalization. Recently, diffusion models have been broadly applied in data generation tasks such as image, speech, and time series synthesis [20]. However, diffusion models are rarely used in power systems. Therefore, how to adopt diffusion models for customized load profiles synthesis is a valuable and necessary study.

To close the aforementioned research gaps, we propose a novel framework to provide customized load profiles synthesis for varied electricity customers according to their power consumption habits and individual demands. To the best of our knowledge, we are the first to exploit the conditional diffusion models for load profiles synthesis. Specifically, we first formulate the customized load profiles synthesis as the conditional generative issue that considers the load characteristics and requirements of electricity customers. Then, we extend the traditional diffusion models to the conditional diffusion models, and design a deep learning model based on the attention mechanism and residual connection to realize conditional data generation. Compared with the published

literature, the main contributions of this paper are summarized as follows:

- 1) We propose a customized load profiles synthesis method based on conditional diffusion models. Our proposed method achieves exclusive, high-quality load profiles synthesis for each customer according to their respective continuous conditional information. Unlike popular GAN-based methods, our proposed method can generate more diverse data samples to avoid missing minority modes of historical load profiles. Furthermore, compared with flow-based models, the proposed method has a fixed learning procedure and structure, which is more flexibly adapted to electricity customers with different load patterns, thus being more suitable for customized synthesis.
- 2) We design a noise estimation model based on the attention mechanism and residual layers with skip connections to implement conditional diffusion models. The designed model can accurately predict added noises by using the attention mechanism for temporal feature extraction of data, thus enabling excellent data denoising. Moreover, we adopt skip connections to aggregate the output of each residual layer, which enables obtaining high-quality estimation results by considering the information from different estimation stages of the designed model.

The rest of this paper is organized as follows. Section II describes the customized load profiles synthesis problem. The proposed method is elaborated in Section III. Section IV validates the effectiveness of the proposed method and Section V presents the conclusion and future work of this paper.

## II. PROBLEM STATEMENT

In this section, we introduce and formulate the problem of customized load profiles synthesis, and then provide two typical realistic application scenarios.

To implement data analytics applications (e.g., load forecasting and anomaly detection), electricity customers usually employ the data-driven model, which requires large amounts of historical load data for training. However, it is often difficult for a single customer to meet that data volume requirement, so customers need synthetic load data to solve the data insufficiency issue. Relying on data-driven models trained with synthetic data, customers can realize their individual applications and targets, e.g., improving energy efficiency or participating in demand response programs.

In this paper, we broadly divide the load profiles synthesis task into the following two scenarios according to whether the customer has load data to train a model:

1) *Data Generation*: Some customers may have little or even no historical data, rendering them unable to train their data-driven models. They can apply the load profiles synthesis to generate synthetic load data for model training.

2) *Data Augmentation*: Some customers may have insufficient historical load data that can only train under-fitting data-driven models. In this circumstance, they can apply the load profiles synthesis to augment existing load data and thereby enhance the model performance.

Because load profiles are highly personalized, the electricity customer's load pattern is different from each other. To achieve

high-quality data synthesis for each customer, we consider the customized load profiles synthesis that can provide specific generated load data based on the customer's respective load characteristics and target requirements, e.g., the start date and range of load profiles to be synthesized.

For historical load data that can reveal the load pattern of the corresponding customer, we mark them as  $\mathbf{x}_{\text{load}}$ . In addition, we suppose that this paper generates the daily load profiles, and denote the date of the load data to be generated as  $d$ . Considering that  $\mathbf{x}_{\text{load}}$  and  $d$  reflect the load characteristic and target requirement, respectively, we intend to fulfill the customized load profiles synthesis as follows:

$$\mathbf{y}_d = f_{\theta}(\mathbf{x}_{\text{load}}, d), \quad (1)$$

where  $\mathbf{y}_d$  denotes the generated load profiles for the target date; and  $f_{\theta}$  is the deep learning model with parameters  $\theta$ .

### III. PROPOSED METHODOLOGY

In this section, we elaborate on the proposed method for the customized load profiles synthesis. We first outline the framework and then explain the key techniques in detail.

#### A. Customized Load Profiles Synthesis Framework

The goal of our proposed method is to provide customized load profiles synthesis for electricity customers by taking their personalized needs into account. With the help of the generated data, the customers or utilities can build high-performance data-driven models for their specific tasks.

Because electricity customers have varied task demands and historical load data volumes, it seems more reasonable to build a model for each customer individually. However, this approach is costly and may encounter the data shortage problem. Therefore, we exploit the cross-learning approach [21] that establishes a generic load profiles synthesis model by treating each customer as a different sample, to carry out customized load synthesis for every customer. To accomplish this, we introduce a server that is acted by the utility company or non-profit organization. In addition, we adopt the client-server architecture for the proposed method, as illustrated in Fig. 1. Specifically, there are two types of entities in the framework, described as follows:

1) *Server*: The server gathers load profiles from customers, and trains the global data generation model with the collected data (see details in Section III-B). Moreover, the server provides the customized load profiles synthesis service for each customer based on their personalized requirements, so that they have enough data to build their own data-driven models.

2) *Customer*: Each customer is responsible for uploading their own historical data to the server to help train the global model. In return, they can obtain customized synthesized load profiles from the server when needed. It is worth noting that uploaded data not only assist in model training but also enable customers to obtain more realistic synthetic load profiles. For new customers with no historical data, the customized data generation service is also available from the server. Although the quality of synthetic data may be slightly poorer, it can

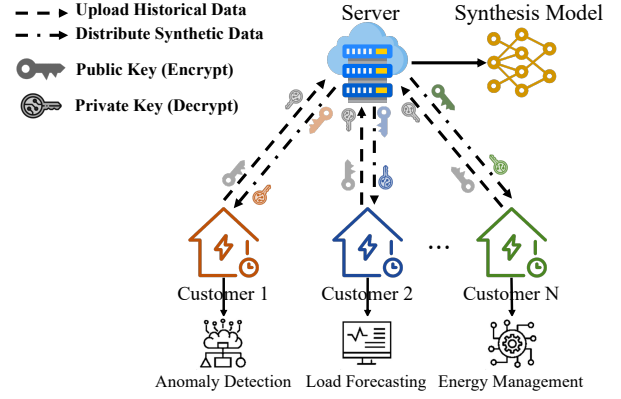


Fig. 1. Customized load profiles synthesis framework.

preliminarily support their data-driven applications. After attaining the customized synthetic load profiles, customers can build high-performance data-driven models according to their respective application tasks, e.g., load forecasting, anomaly detection, and energy management, etc.

#### B. Conditional Diffusion Models-Based Generation Method

As the customized synthesis is converted into a conditional data generation form in Eq. (1), we employ diffusion models, which is an advanced deep generative method, to accomplish high-quality load profiles synthesis. The diffusion models are latent variable models that learn the dataset's latent structure to approximate the real data distribution [22]. What makes diffusion models different from other deep generative methods is that they can generate high-quality samples with diversity, which avoids the instability training problem of GAN-based methods. Moreover, diffusion models have a fixed learning procedure and their latent variable has high dimensionality, so their data generation quality is better and more stable compared to autoencoder-based and flow-based methods [23].

1) *Diffusion Models*: In general, diffusion models are composed of two processes: the *diffusion process* and the *reverse process*. Diffusion models are distinguished from other latent variable models in that the posterior is derived from a Markov chain. Specifically, the Gaussian noise is gradually added to the original data through the diffusion process, which can be formulated as:

$$q(\mathbf{x}_{1:T}|\mathbf{x}_0) := \prod_{t=1}^T q(\mathbf{x}_t|\mathbf{x}_{t-1}), \quad (2)$$

where  $q(\mathbf{x}_{1:T}|\mathbf{x}_0)$  is the diffusion process;  $\mathbf{x}_0$  denotes original data sampled from the real data distribution  $q(\mathbf{x}_0)$ ;  $\{\mathbf{x}_t\}_{t=1}^T$  are latent variables (i.e., noisy data) with the same dimensionality as  $\mathbf{x}_0$ ;  $t$  and  $T$  are timestep and diffusion total timestep, respectively. Since the diffusion process takes the form of a Markov chain, where  $q(\mathbf{x}_t)$  only depends on  $q(\mathbf{x}_{t-1})$ , the joint distribution of noisy data  $q(\mathbf{x}_{1:T})$  can be written as the product of successive diffusion steps.

For each diffusion step  $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ , it takes in  $\mathbf{x}_{t-1}$  and returns  $\mathbf{x}_t$  by adding some noise, which is defined as:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) := \mathcal{N}(\mathbf{x}_t; \sqrt{1 - \beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \quad (3)$$

where  $\beta_t \in (0, 1)$  is the noise schedule. Finally,  $\mathbf{x}_T$  will follow an isotropic Gaussian distribution when  $T \rightarrow \infty$ .

Because the diffusion process is defined by a Markov chain, the Gaussian noise is iteratively added to  $\mathbf{x}_0$ , so it will cost much time to sample  $\mathbf{x}_t$ . To improve the sampling efficiency, we rewrite the diffusion process in Eq. (2) in a closed form for any arbitrary timestep  $t$ , using the reparameterization trick:

$$q(\mathbf{x}_t|\mathbf{x}_0) := \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t) \mathbf{I}), \quad (4)$$

$$\mathbf{x}_t := \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{(1 - \bar{\alpha}_t)} \boldsymbol{\epsilon}, \quad (5)$$

where  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  is Gaussian noise;  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ ; and  $\alpha_t = 1 - \beta_t$ .

The reverse process aims to denoise  $\mathbf{x}_t$  to recover  $\mathbf{x}_0$  so that it can eventually recreate data samples from the Gaussian noise. Because the reverse step is hard to directly and explicitly formulate like Eq.(3), we adopt the neural network model for approximation. Similar to the diffusion step, the reverse step can be written as follows:

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \sigma_\theta(\mathbf{x}_t, t) \mathbf{I}), \quad (6)$$

where  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  denotes the reverse step implemented by the neural network;  $\boldsymbol{\mu}_\theta(\mathbf{x}_t, t)$  and  $\sigma_\theta(\mathbf{x}_t, t)$  are two neural network models with parameters  $\theta$ , which take  $\mathbf{x}_t$  and  $t$  as inputs and then output the estimated mean and variance of Gaussian distribution, respectively.

Since the reverse process is also defined by a Markov chain, it can be formulated according to the reverse step, as follows:

$$p_\theta(\mathbf{x}_{0:T}) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \quad (7)$$

where the joint distribution  $p_\theta(\mathbf{x}_{0:T})$  is the reverse process; and  $p(\mathbf{x}_T) = \mathcal{N}(\mathbf{0}, \mathbf{I})$  represents the Gaussian noise.

To denoise the data, it is important to accurately estimate the noise added to the original data at each timestep. Given that the reverse process is the inverse of the diffusion process, the objective of the diffusion models can be expressed as:

$$\min \left( \text{KL}(p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)) \right), 1 \leq t \leq T, \quad (8)$$

where  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  is the posterior conditional probability of the diffusion process (i.e., the ground truth of added noises); and KL is the abbreviation for Kullback–Leibler divergence, which is to be minimized by using diffusion models.

2) *Conditional Diffusion Models*: To realize the customized synthesis, the proposed method needs to generate specified load data according to each customer's own information and requirements. Therefore, we expand the traditional diffusion models to the conditional one, which is illustrated in Fig. 2. The conditional diffusion models can synthesize customized load profiles by the guidance of the typical load and target date that implicitly represent the load pattern and individual demand of customers, respectively.

According to Eq. (7), the reverse process of diffusion models can only generate non-specific load profiles because it starts from a Gaussian noise and does not involve any additional information to control data generation [24]. Therefore, we propose conditional diffusion models to consider

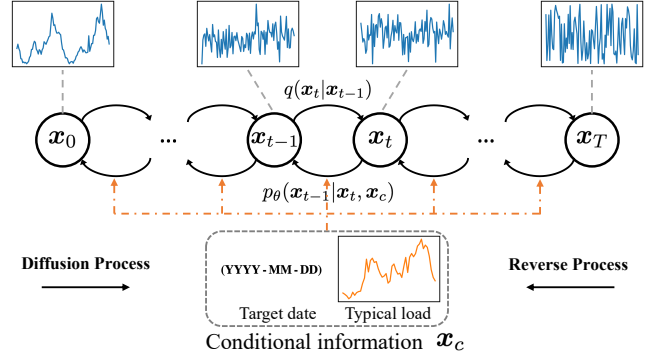


Fig. 2. Illustration of conditional diffusion models for customized load profiles synthesis.

the customer's personalized demands (i.e., continuous values), which can produce customized load profiles. Specifically, we extend the reverse process to the conditional form, as follows:

$$p_\theta(\mathbf{x}_{0:T}|\mathbf{x}_c) := p(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_c), \quad (9)$$

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_c) := \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t|\mathbf{x}_c), \sigma_\theta(\mathbf{x}_t, t|\mathbf{x}_c) \mathbf{I}), \quad (10)$$

where  $\mathbf{x}_c$  denotes the conditional information, which changes with customers, thus accounting for each electricity customer.

Considering that the posterior conditional probability in Eq. (8) represents the noise added to the data, it should follow a Gaussian distribution. To obtain an explicit expression for easy calculation, we rewrite  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  into a Gaussian form based on Bayes' rule:

$$q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) := \mathcal{N}(\mathbf{x}_{t-1}; \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0), \tilde{\beta}_t \mathbf{I}), \quad (11)$$

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0) &:= \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon} \right), \\ \tilde{\beta}_t &:= \frac{1 - \bar{\alpha}_{t-1}}{1 - \bar{\alpha}_t} \beta_t. \end{aligned} \quad (12)$$

Note that  $\tilde{\boldsymbol{\mu}}_t(\mathbf{x}_t, \mathbf{x}_0)$  is only related to  $\mathbf{x}_t$  and  $\boldsymbol{\epsilon}_t$ , while  $\tilde{\beta}_t$  is the constant associated with  $\beta_t$ . We will not discuss  $\tilde{\beta}_t$  subsequently since no unknown variables are involved.

Inspired by the Eq. (12), the mean expression in the conditional reverse process can be reformulated in a similar way:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t|\mathbf{x}_c) := \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t|\mathbf{x}_c) \right), \quad (13)$$

where  $\boldsymbol{\epsilon}_\theta$  is the noise estimation model with parameter  $\theta$ .

Because the KL divergence in Eq. (8) is difficult to calculate directly, we need to convert it into a convenient expression for gradient calculation, which enables the neural network model to update its parameters. Since both  $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)$  and  $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0)$  follow a Gaussian distribution, the KL divergence can be calculated in a Rao-Blackwellized fashion with closed form expressions. Therefore, we combine Eqs. (8)–(13) to formulate the loss function of diffusion models, and we obtain the explicit expressions under the conditional form:

$$L(\theta) := \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), t} \left\| (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t|\mathbf{x}_c)) \right\|_2^2. \quad (14)$$

It is worth noting that we have removed the coefficient of

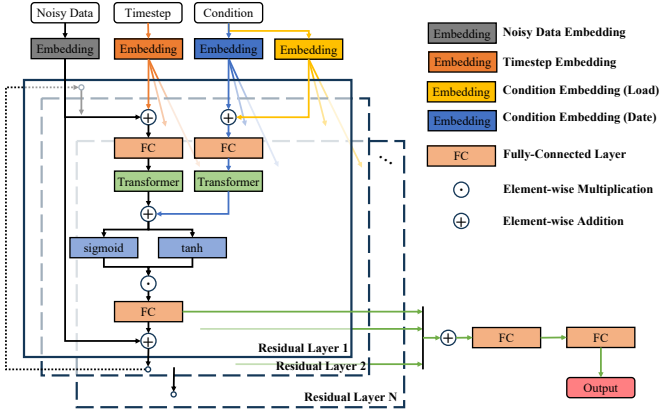


Fig. 3. The neural network architecture of noise estimation model.

loss function to improve the training quality and stability [19].

**Remark 1.** *The customized load profiles synthesis is formulated as a conditional data generation problem and intended to be solved by the proposed conditional diffusion models. Based on the customer's conditional information, the well-trained neural network model of the conditional reverse process can denoise from Gaussian noise to generate load profiles specific to that customer, which enables the high-quality customized synthesis. To achieve excellent denoising, the challenge now is to accurately predict the noise added during the diffusion process, which will be addressed in Section III-C.*

### C. Attention Mechanism-Based Noise Estimation Model

Because all inputs (e.g., noisy data and conditional information) of  $\epsilon_\theta$  in Eq. (14) are time-series, we design a deep learning model based on the attention mechanism [25] to predict the noises that are added to the original data during the diffusion process. The designed model is composed of  $N$  stacked residual layers, where the architecture is illustrated in Fig. 3. In particular, the inputs include the latent variable  $\mathbf{x}_t$ , timestep  $t$ , and condition information  $\mathbf{x}_c$ . Furthermore, we aggregate the output of each residual layer to form the final model output (i.e., estimated noise) by using skip connections [26]. The designed model primarily consists of an embedding module, a residual module, and an output module, which are described respectively below.

1) *Embedding Module:* In order to preserve the temporal relationship of data, we perform the embedding transformation for each input [25]. First, we convert timestep  $t$  into an encoding vector  $\mathbf{t}_{\text{emb}}$ , because the model needs to yield different outputs  $\epsilon_\theta(\mathbf{x}_t, t|\mathbf{x}_c)$  for different  $t$ :

$$\begin{aligned} \mathbf{t}_{\text{emb}}(2i) &= \sin\left(10 \frac{2ix8}{d_{\text{model}}} t\right), \\ \mathbf{t}_{\text{emb}}(2i+1) &= \cos\left(10 \frac{2ix8}{d_{\text{model}}} t\right), \end{aligned} \quad (15)$$

where  $d_{\text{model}}$  denotes the encoding vector's dimensionality; and  $i \in [0, d_{\text{model}}/2]$  is the index of data dimension.

Then, since  $\mathbf{x}_t$  is from the same space as  $\mathbf{x}_0$ , it is also the sequence data. To get the corresponding embedding  $\mathbf{x}_{t\text{emb}}$ , we apply the linear layer for data transformation:

$$\mathbf{x}_{t\text{emb}} = FC(\mathbf{x}_t) + PE(FC(\mathbf{x}_t)), \quad (16)$$

where  $FC(\cdot)$  and  $PE(\cdot)$  denote the fully-connected layer and positional encoding function, respectively.

Finally, we consider that continuous conditional information  $\mathbf{x}_c$  contains the customer's typical load  $\mathbf{x}_{c,\text{load}}$  and target date  $\mathbf{x}_{c,\text{date}}$ , so we need to process them separately. For  $\mathbf{x}_{c,\text{load}}$ , we utilize the sequence data transformation in Eq. (16), and then adopt the one-hot encoding method [5] for  $\mathbf{x}_{c,\text{date}}$ , as follows:

$$\begin{aligned} \mathbf{x}_{c,\text{load}\text{emb}} &= FC(\mathbf{x}_{c,\text{load}}) + PE(FC(\mathbf{x}_{c,\text{load}})), \\ \mathbf{x}_{c,\text{date}\text{emb}} &= FC(\text{OneHot}(\mathbf{x}_{c,\text{date}})), \end{aligned} \quad (17)$$

where  $\text{OneHot}(\cdot)$  denotes the one-hot encoding function. In this manner, we improve the representation capability of all inputs and make them more suitable for model operation.

2) *Residual Module:* Since each embedding vector represents a different type of content, the naive addition or merging will reduce the information's diversity. Therefore, we deploy the attention mechanism to extract time-series characteristics of these vectors, and then combine them organically.

We divide the above four embedding vectors into two parts according to whether they are related to the condition, and then we fuse each part. Next, we distill temporal features of each part separately by using the attention function, followed by combining two parts to obtain the integrated feature  $\mathbf{x}_{\text{total}}$ :

$$\begin{aligned} \mathbf{x}_{\text{total}} &= MHSa(FC(\mathbf{x}_{t\text{emb}} \oplus \mathbf{t}_{\text{emb}})) \oplus \\ &MHSa(FC(\mathbf{x}_{c,\text{load}\text{emb}} \oplus \mathbf{x}_{c,\text{date}\text{emb}})), \end{aligned} \quad (18)$$

where operation  $\oplus$  is the element-wise addition; and  $MHSa(\cdot)$  denotes the multi-head self-attention function. Henceforth, we utilize  $FC(\cdot)$  to enhance the non-linear fitting ability of the designed model, instead of representing data transformation.

To further facilitate the fusion of conditional information, we split  $\mathbf{x}_{\text{total}}$  into two chunks, which will be processed by different activation functions, and then merge them again. We adopt different activation functions to avoid the information loss caused by the same processing operation, thus ensuring the diversity of information [26]. In addition, we also deploy the residual connection to effectively alleviate the vanishing gradient problem, as follows:

$$\begin{aligned} \mathbf{x}_{\text{output}} &= FC(\sigma(\mathbf{x}_{\text{sub},1}) \odot \tanh(\mathbf{x}_{\text{sub},2})), \\ \mathbf{x}_{\text{res}} &= \mathbf{x}_{\text{output}} \oplus \mathbf{x}_{t\text{emb}}, \end{aligned} \quad (19)$$

where  $\sigma(\cdot)$  and  $\tanh(\cdot)$  denote the sigmoid and tanh activation function;  $\mathbf{x}_{\text{sub},1}$  and  $\mathbf{x}_{\text{sub},2}$  are two chunks of  $\mathbf{x}_{\text{total}}$ ; and operation  $\odot$  represents the element-wise multiplication.

For each residual layer, its  $\mathbf{x}_{\text{res}}$  will serve as  $\mathbf{x}_t$  of the next layer, and so on. Meanwhile, we collect  $\mathbf{x}_{\text{output}}$  of each residual layer, which will work in the next module.

3) *Output Module:* As the designed model is composed of multiple stacked residual layers, some valuable information may be missed if we directly adopt the output of the last layer as the model result. Hence, we take account of each layer's output to jointly produce the ultimate output (i.e., the estimated noise), through integrating skip connections:

$$\epsilon_\theta(\mathbf{x}_t, t|\mathbf{x}_c) = FC(\text{Concat}(\mathbf{x}_{\text{skip},1}, \dots, \mathbf{x}_{\text{skip},N})), \quad (20)$$

where  $\text{Concat}(\cdot)$  is the vector concatenation operator;  $\mathbf{x}_{\text{skip},i}$  denotes  $\mathbf{x}_{\text{output}}$  of the  $i$ -th residual layer; and  $N$  is the total



**Algorithm 1: Conditional Diffusion Models Training**


---

**Input** : The model parameter  $\theta$ , diffusion total timestep  $T$ , noise schedules  $\{\beta_t\}_{t=1}^T$ , load profiles dataset  $D$ , condition dataset  $D_c$ , batch size  $B$ , total epoch number  $E$ , Adam parameters  $\alpha^{Adam}, \beta_1^{Adam}, \beta_2^{Adam}$ .

**Output** : The comprehensive trained model.

**Procedure:**

- 1 **for**  $e = 1, \dots, E$  **do**
- 2     **for** each batch of training data **do**
- 3         **for**  $i = 1, \dots, B$  **do**
- 4             Sample original data  $\mathbf{x}_0 \sim D$ , diffusion timestep  $t \sim \text{Uniform}(1, T)$ , Gaussian noise  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;
- 5             Select the corresponding condition  $\mathbf{x}_c$  from  $D_c$ ;
- 6             Calculate  $\mathbf{x}_t \leftarrow \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{(1 - \alpha_t)}\epsilon$ ;
- 7             Obtain estimated noise  $\epsilon_\theta(\mathbf{x}_t, t|\mathbf{x}_c)$ ;
- 8             Compute model's loss  $L^{(i)} \leftarrow \|(\epsilon - \epsilon_\theta(\mathbf{x}_t, t|\mathbf{x}_c))\|_2^2$ ;
- 9         **end**
- 10         Update model's parameters based on Adam algorithm  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{B} \sum_{i=1}^B L^{(i)}, \alpha^{Adam}, \beta_1^{Adam}, \beta_2^{Adam})$ ;
- 11     **end**
- 12 **end**
- 13 **return**  $\theta$

---

number of residual layers.

**Remark 2.** *The proposed model exploits the attention mechanism to extract features of all inputs separately and then combine them together, followed by using the skip connection to produce the estimated noise. The model comprehensively considers the distilled information from each estimation stage of the model so that it can realize accurate noise estimation.*

#### D. Model Training and Synthesis Execution

After establishing the neural network model for noise estimation, we train the designed model  $\epsilon_\theta$  by minimizing the loss function in Eq. (14). The detailed training process is summarized in Algorithm 1. It should be mentioned that the noise schedule  $\{\beta_t\}_{t=1}^T$  is a linear schedule based on the original implementation described in [22], which is calculated by linear interpolation between 0.0001 and 0.5. In addition, to improve the quality and speed of training convergence, we use the Adam algorithm to optimize the model's parameters [27], which adjusts the learning rate adaptively and uses momentum to amend the direction of parameter update.

With comprehensive training, the proposed method can synthesize exclusive load profiles in an iterative manner according to customers' individual conditions, as presented in Algorithm 2. Specifically, we first determine the diffusion total timestep and accordingly obtain the noise schedules (see details in Section III-B). Then, we sample a random noise  $\mathbf{x}_T$  from the Gaussian distribution. Moreover, we pick the customer's typical load and corresponding target date of data synthesis as conditional information  $\mathbf{x}_c$ , which will be used as the model's inputs with the random noise and timestep (see details in Section III-C). Subsequently, we calculate noisy data  $\mathbf{x}_t$  based on the model's output, and repeat this process until  $t = 0$ . At this moment, we take eventual denoised data  $\mathbf{x}_0$  as the synthetic load profiles, which is customized for a customer.

## IV. CASE STUDIES

In this section, we conduct extensive experiments to verify our proposed method. First, we introduce the experiment

**Algorithm 2: Customized Load Profiles Synthesis**


---

**Input** : The well-trained model  $\epsilon_\theta$ , diffusion total timestep  $T$ , noise schedules  $\{\beta_t\}_{t=1}^T$ , condition dataset  $D_c$ .

**Output** : Customized synthetic load profiles.

**Procedure:**

- 1 Sample  $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;
- 2 Select typical load  $\mathbf{x}_{c,\text{load}}$  and target date  $\mathbf{x}_{c,\text{date}}$  according to the customer and its load profiles synthesis demand;
- 3 Obtain  $\mathbf{x}_c \leftarrow (\mathbf{x}_{c,\text{load}}, \mathbf{x}_{c,\text{date}})$ ;
- 4 **for**  $t = T, \dots, 1$  **do**
- 5     Sample  $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ ;
- 6     **if**  $t = 0$  **then**
- 7          $\epsilon \leftarrow \mathbf{0}$ ;
- 8     **end**
- 9     Calculate latent variable  $\mathbf{x}_{t-1}$  according to Eqs. (11)–(13)
- 10      $\mathbf{x}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}} \left( \mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}} \epsilon_\theta(\mathbf{x}_t, t|\mathbf{x}_c) \right) + \frac{1-\alpha_{t-1}}{1-\alpha_t} \beta_t \epsilon$ ;
- 11 **end**
- 12 **return**  $\mathbf{x}_0$

---

settings, including the dataset, evaluation metrics, benchmarks, and implementation details. Then, we demonstrate the performance of the proposed method by comparing it with four benchmarks in two scenarios, respectively. In addition, we conduct ablation experiments to validate the effectiveness of the proposed noise estimation model. Finally, we further evaluate the effectiveness of our proposed method for applying to the load forecasting task by comparing it with federated learning.

#### A. Experiment Settings

1) *Data Preparation:* We make use of the smart meter data collected in the Low Carbon London program for the numerical experiments [28]. This dataset contains electricity consumption data of 5198 customers from November 2011 to February 2014 with a time granularity of 30 minutes. After data preprocessing for missing values, we ultimately obtain full data across 1069 customers for the entire calendar year of 2013, i.e., 17520 pieces of recorded load data per customer. Since we consider daily load profiles in case studies, each sample contains 48 recorded load data. Moreover, all data are divided into the training, validation, and test sets, accounting for 60%, 20%, and 20% of all data, respectively. Furthermore, we manually crawl the London weather data in 2013 for the load forecasting task.

2) *Evaluation Metrics:* Since it is challenging to measure the quality of synthetic data, especially for time series data [15], an effective evaluation indicator is necessary for load profiles synthesis. Moreover, the goal of the load profiles synthesis is to make the generated load data conform to the customer's load pattern, i.e., follow the distribution of real data. However, when the synthetic sample obeys the real data's distribution, the discrepancy between it and the real sample may still be large [29]. Therefore, we design a comprehensive evaluation scheme that is suitable for both two scenarios.

For the data generation scenario, we select four evaluation metrics from the perspective of sample level and distribution level, including root mean square error (RMSE), mean absolute error (MAE), maximum mean discrepancy (MMD), and Wasserstein distance (WD). The first two are sample-level

classic metrics, and the latter two are used to quantify the difference between two distributions, expressed as follows:

$$\text{MMD} = \frac{1}{M^2} \left( \sum_{i=1}^M \sum_{j=1}^M (k(y_i, y_j) + k(\hat{y}_i, \hat{y}_j) - 2 \cdot k(y_i, \hat{y}_j)) \right), \quad (21)$$

$$\text{WD} = \int_{-\infty}^{+\infty} |F_1(z) - F_2(z)| dz, \quad (22)$$

where  $y_i$  and  $\hat{y}_i$  denote the  $i$ -th real and synthetic samples of data;  $M$  is the total amount of data;  $k(\cdot)$  denotes the Gaussian kernel function;  $F_1(z)$  and  $F_2(z)$  are the respective cumulative distribution functions of real and synthetic data. It is worth noting that a smaller value represents better data generation performance for these four metrics. In this way, the synthetic load profiles will be compared with real load profiles both numerically and distributionally to evaluate synthesis methods.

For the data augmentation scenario, there are generally two objectives: 1) minimizing the distribution shift from training data; and 2) increasing the amount of effective training data, thus improving the model generalization and performance. Considering that measuring the effect of augmented data is related to the downstream application task, we choose the load forecasting as the example in this paper. We adopt the *Affinity* to quantify the impact of augmented data on the model decision boundary (i.e., performance), and the *Diversity* to quantify the extent of augmented data on preventing the model overfitting, formulated as follows:

$$\text{Affinity} = \mathcal{M}(g_\omega, D'_{\text{val}}) - \mathcal{M}(g_\omega, D_{\text{val}}), \quad (23)$$

$$\text{Diversity} = \mathbb{E}_{D'_{\text{train}}} L_{\text{train}}, \quad (24)$$

where  $D_{\text{train}}$  and  $D_{\text{val}}$  are the original training and validation datasets for the load forecasting task;  $D'_{\text{train}}$  and  $D'_{\text{val}}$  denote the corresponding augmented datasets;  $g_\omega$  is the load forecasting model trained on  $D_{\text{train}}$ ;  $\mathcal{M}(g_\omega, D_{\text{val}})$  denotes the performance result of the trained data-driven model in the load prediction task evaluated on dataset  $D_{\text{val}}$ ; and  $L_{\text{train}}$  represents the model's training loss. Considering that augmented data will shift the decision boundary of data-driven models, which indicates that the value of affinity will not exceed 0, a larger value of affinity represents less impact on the model performance. In addition, a larger value of diversity implies a greater extent to which augmented data improve model training performance that can be improved by augmented data, and vice versa. To sum up, unlike before, the larger value of affinity and diversity stands for better augmentation performance.

3) *Benchmarks*: To verify the effectiveness and superiority of our proposed method, we select the following benchmarks based on the current state-of-the-art methods for comparison. Note that the first two are unconditional, while the latter two adopt the conditional data generation model.

- Benchmark A: A GAN-based framework proposed in 2020 by Kababji et al. [13] for load pattern synthesis.
- Benchmark B: A least square GAN proposed in 2021 by Li et al. [30] for renewable scenario generation.
- Benchmark C: A flow-based conditional generative model proposed in 2019 by Zhang et al. [18] for scenario

TABLE I  
IMPLEMENTATION DETAILS

Parameter	Definition	Value
$T$	diffusion total timestep	50
$(\beta_1, \beta_T)$	noise schedule start and end	(0.0001, 0.5)
$d_{\text{model}}$	the embedding vector dimension	16
$N$	the number of residual layers	6
$E$	the number of training epochs	200
$B$	the batch size of training data	16
$\alpha$	the learning rate of Adam	0.001

forecasting of residential load profiles.

- Benchmark D: A Wasserstein deep convolutional conditional GAN proposed in 2023 by Chen et al. [16] to generate energy consumption data.

4) *Implementation Details*: The proposed method is implemented by the open source machine learning framework PyTorch<sup>1</sup>. We conduct all experiments on an Ubuntu 18.04 LTS platform, which is equipped with the Intel Core i9-10980XE CPU and NVIDIA GeForce RTX 3090 GPU. Table I summarizes the details of the model and training parameters.

### B. Performance in Data Generation Scenario

In this part, we validate the performance of our proposed method in the data generation scenario by comparing it with the aforementioned four benchmarks. For the comprehensive validation, we randomly select 200 customers from the test dataset as cases, and hide their load data to imitate customers with little or no load data for the data generation scenario. After the load profiles synthesis, we release hidden load profiles and exploit the designed evaluation scheme to inspect the effect of synthetic load profiles from the sample and distribution levels. In addition, to avoid random errors, we repeat all experiments 10 times and then calculate the average values. The statistical results are summarized in Table II.

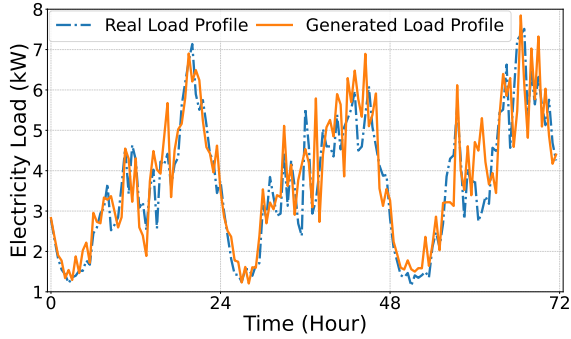
It can be seen that our proposed method achieves the best performance in all metrics compared with benchmarks. The evaluation metric results of the proposed method are both less than 1 kW, and its MAE is even close to 0.7 kW. In contrast, the results of the four benchmarks are relatively poor, where the two error metrics (RMSE and MAE) are at least 19% and 26% higher, respectively. For the other two metrics (MMD and WD), the values of our proposed method are 0.41 and 0.36, whereas those of all benchmarks exceed 0.45. Moreover, the proposed method also has the smallest standard deviation in both sample-level and distribution-level metrics, with a maximum reduction of more than 50% compared to the benchmarks. It is worth mentioning that both our proposed method and the latter two benchmarks achieve significant performance improvements compared to the first two unconditional benchmarks. This indicates the necessity of using conditional information for customized load profiles synthesis. Therefore, our proposed method attains the best load profiles synthesis performance in the data generation scenario in terms of accuracy and stability.

<sup>1</sup><https://pytorch.org>

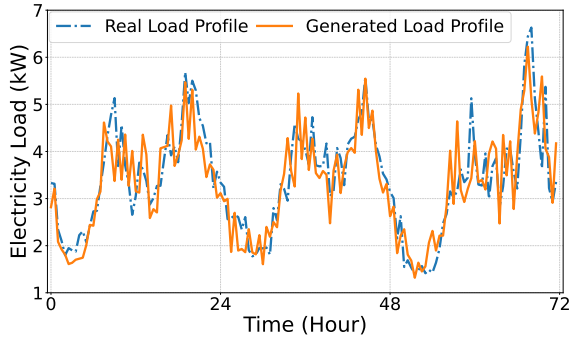
TABLE II  
NUMERICAL RESULTS OF PERFORMANCE COMPARISON UNDER DATA GENERATION SCENARIO

Metric	Benchmark A	Benchmark B	Benchmark C	Benchmark D	Proposed
<b>RMSE (kW)</b>	1.5584 (0.3099)	1.5104 (0.2923)	1.2271 (0.2153)	1.1694 (0.2046)	<b>0.9806 (0.1512)</b>
<b>MAE (kW)</b>	1.2264 (0.2350)	1.1571 (0.2138)	0.9824 (0.1402)	0.9205 (0.1304)	<b>0.7299 (0.1041)</b>
<b>MMD</b>	0.5727 (0.1023)	0.5646 (0.09676)	0.4986 (0.0904)	0.4759 (0.0862)	<b>0.4092 (0.0667)</b>
<b>WD</b>	0.6526 (0.1366)	0.6496 (0.1315)	0.4968 (0.1080)	0.4515 (0.0989)	<b>0.3582 (0.0749)</b>

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



(a)



(b)

Fig. 4. Load profiles synthesis results of two examples under data generation scenario in (a) summer and (b) winter.

In order to provide a visual demonstration of the load profiles synthesis effects, we randomly choose one customer in summer and winter respectively as examples, and the visualization results are shown in Fig. 4. The generated load curves of the proposed method are close to the real curves for both the summer and winter cases, and they also follow the real load variation trends. Of particular importance is that our proposed method can accurately synthesize the daily peaks and valleys of load profiles, which is essential for the operation and control of power systems.

Similarly, to intuitively compare the performance of different methods, we randomly select another customer and make the 2-D visualization of its full-year synthetic load profiles that are generated by the proposed method and benchmarks, as shown in Fig. 5. This figure consists of a scatterplot in the center and density curves on the marginal axes. Specifically, each data point in the scatterplot represents a daily load profiles after dimensionality reduction to 2 dimensions by using umap [31]. Each density curve on the axes describes the

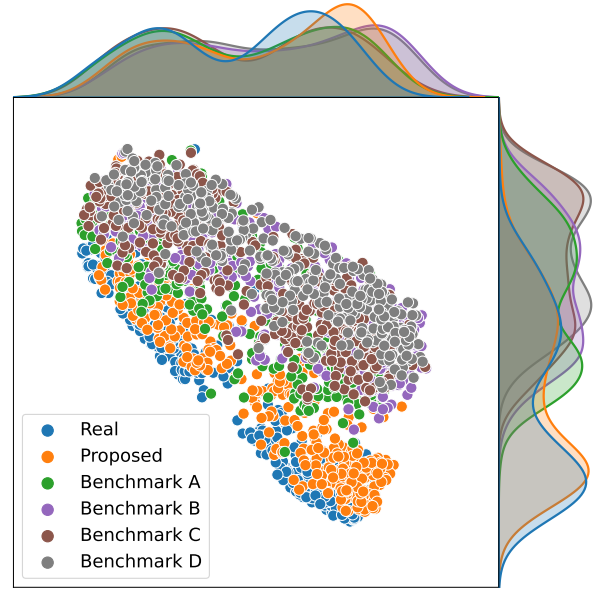


Fig. 5. Two-dimensional visualization of synthesis performance comparison under data generation scenario. A data point represents one daily load profiles after dimensionality reduction. The curves on the axes depict the density curves for each type of load profiles in the two reduction dimensions.

distribution of each type of scatter in two reduced dimensions. It is clear that the synthesis load profiles of our proposed method are closest to the real load profiles, and have the most similar shape to the real load profiles compared to other benchmarks. Moreover, from the perspective of density curves, the proposed method is also closest to the real load profiles in both dimensions, especially the vertical axis. This corresponds to the above four numerical metrics, thus demonstrating that our proposed method achieves the best synthesis performance.

### C. Performance in Data Augmentation Scenario

In this part, we further verify the performance of the proposed method in the data augmentation scenario. Following the same steps, we pick 200 customers without bias and augment each customer's full-year load profiles 50-fold. We present two augmentation examples using our proposed method, which are arbitrarily chosen and shown in Fig. 6. It should be noted that the gray curves represent augmented load profiles, and the gray area is formed by the superposition of 50 gray curves. Moreover, the augmented centroid is calculated by averaging 50 pieces of augmented load profiles. From the figure, the augmented load profiles not only cover the real load profile but also are mostly in the vicinity of the real curves, which proves



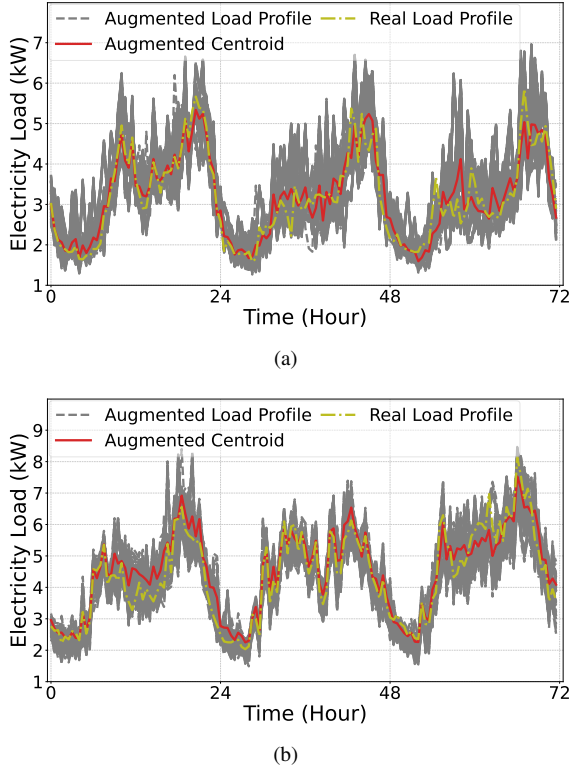


Fig. 6. Load profiles synthesis results of two examples under data augmentation scenario in (a) summer and (b) winter.

TABLE III  
PERFORMANCE COMPARISON UNDER DATA AUGMENTATION SCENARIO

Method	Affinity	Diversity	Improvement <sup>1</sup> (%)
Benchmark A	-0.2922	0.7997	3.9882
Benchmark B	-0.2340	0.8634	4.1311
Benchmark C	-0.1293	1.3521	7.3723
Benchmark D	-0.1108	1.4563	7.5538
<b>Proposed</b>	<b>-0.0775</b>	<b>1.6381</b>	<b>8.9184</b>

\*Each entry gives the mean value.

<sup>1</sup>This represents performance improvement in load forecasting.

the effectiveness of the data augmentation. Furthermore, the centroid of augmented data is also close to the real data, indicating that the statistical properties of the augmentation follow the ground truth.

To explore the differences between distributions of the augmented and real load profiles, we employ umap for dimension reduction once again, and the visualization results of two examples are presented in Fig. 7. For the sake of clarity, we only select the best benchmark in experiments for comparison. Compared with the best benchmark, the augmented data of our proposed method are more similar in shape to the original data and have more sample point intersections. However, these two types of data are not entirely overlapping, thus the proposed method does not purely simulate original data, but learns the intrinsic manifold of load profiles, making it capable of generating augmented data with diversity.

To reveal the role of load profiles synthesis on real-world data analytics applications, we train the respective models for the load forecasting task, using the augmented data from

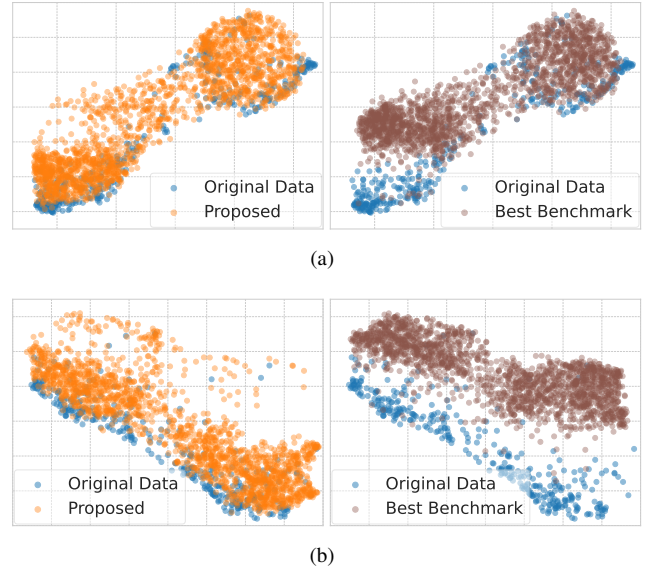


Fig. 7. Two-dimensional visualization of synthesis performance comparison with the best benchmark under data augmentation scenario. The original data are the measured real load profiles, and the other are augmented load profiles by using the proposed method or the best benchmark. A data point represents one daily load profiles after dimensionality reduction to two dimensions. The more similar shape and more overlap of the two types of data points signify the closer pattern of augmented load profiles and real load profiles.

each benchmark and our proposed method, respectively. Then, we exploit the designed evaluation metrics (i.e., affinity and diversity) to quantify the data augmentation performance. Furthermore, we train the load forecasting baseline model with original real load profiles, and accordingly calculate the performance improvement of load forecasting models after using the augmented data. All numerical results are listed in Table III. It can be observed that the proposed method has the largest affinity value exceeding  $-0.1$ , as well as the highest diversity value with at least 10% improvement over the benchmarks. Therefore, our proposed method can complement effective training data that have minimal impact on the model decision boundary and improve model generalization. Therefore, our proposed method can complement effective training data that have minimal impact on the model performance and improve model generalization best. In addition, the performance improvement results further prove that our proposed method can enhance the model performance on the load forecasting task, with an extent of nearly 9% on prediction accuracy.

#### D. Performance Comparison for Ablation Experiments

To further demonstrate the effectiveness of the proposed noise estimation model, we conduct ablation experiments to reveal the impact of key components (i.e., the attention mechanism and skip connections) on model performance. According to the structure of our proposed model, we construct three new benchmarks, as follows:

- Benchmark E1: the Transformer in the residual module is substituted by the long short-term memory (LSTM) network, and the skip connections operation in the output module is deleted.

TABLE IV  
PERFORMANCE COMPARISON FOR ABLATION EXPERIMENT

Metric	E1	E2	E3	Proposed
<b>RMSE</b>	1.7018	1.3509	1.6165	<b>0.9806</b>
<b>MAE</b>	1.3460	1.0262	1.3356	<b>0.7299</b>
<b>MMD</b>	0.6686	0.5569	0.6354	<b>0.4092</b>
<b>WD</b>	0.7809	0.5921	0.7212	<b>0.3582</b>
<b>Affinity</b>	-0.3965	-0.1875	-0.3273	<b>-0.0775</b>
<b>Diversity</b>	0.5016	0.9863	0.6185	<b>1.6381</b>

\*Each entry gives the mean value.

\*\*RMSE and MAE are all in kilowatts(kW).

- Benchmark E2: the Transformer in the residual module is substituted by the LSTM network, but the skip connections operation in the output module is reserved.
- Benchmark E3: the Transformer in the residual module is reserved, but the skip connections operation in the output module is deleted.

Note that when the skip connections operation is removed, the output of benchmarks is the output of the last residual layer in their models. In addition, our proposed method and benchmarks utilize the same data for performance comparison.

We repeat the experiments 10 times to calculate the average values, and the results are shown in Table IV. It can be seen that the performance degradation is significant, with the absence of the attention mechanism or skip connections. Specifically, compared to our proposed method, when we remove only the attention mechanism or skip connections, the RMSE of benchmarks increases by 37.8% and 64.8%, respectively. Similarly, their diversity metrics drop by 39.8% and 62.2%, respectively. This demonstrates the importance of the attention mechanism and skip connections on the model performance, and the attention mechanism is more crucial than skip connections. In addition, when we remove both the attention mechanism and skip connections, the synthesis performance further declines. For example, compared to Benchmark E2 and E3, the MMD of Benchmark E1 raises by 20.1% and 5.2%, respectively. Likewise, its affinity metric also decreases by 111.5% 21.1%, respectively. This further indicates that the attention mechanism has more contribution to the model performance. Therefore, even with the same structure, the model performance will suffer an obvious degradation without the attention mechanism and skip connections.

#### E. Performance Comparison with Federated Learning in Load Forecasting Task

In this part, we evaluate the performance of our proposed method on downstream application tasks, where the load forecasting task is selected as an example. We randomly select 50 customers with a full year of load profiles for the experiment. Considering that the historical load data of a single customer is insufficient for training a data-driven model, we adopt the load profiles synthesis and federated learning [32] to deal with the data shortage problem, respectively. Specifically, on the one hand, we utilize our proposed method and four

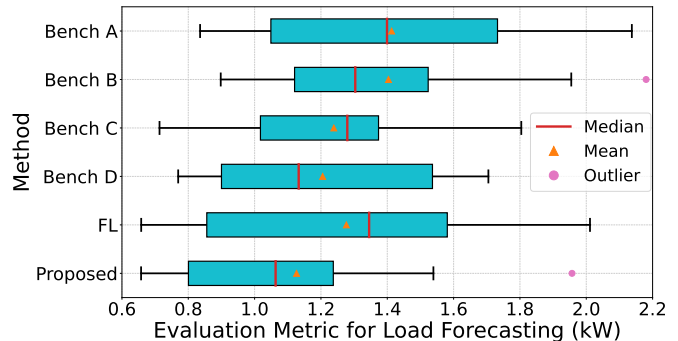


Fig. 8. Performance comparison of the proposed method, benchmarks, and federated learning in the load forecasting task. RMSE is selected as the performance evaluation metric for load forecasting. FL stands for federated learning, and Bench is short for “Benchmark”.

benchmarks to augment each customer’s load profiles 50-fold respectively, and then train each customer’s individual load forecasting model by using augmented data. On the other hand, we adopt federated learning to collaboratively train a global load forecasting model among the 50 customers. In this way, both two approaches increase the amount of training data in essence. It is worth noting that for a single customer, the amount of training data for the load forecasting model is the same in the above two approaches. Furthermore, we use RMSE as the metric to evaluate the performance of trained load forecasting models in the load forecasting task.

We summarize the load forecasting results of the 50 customers and calculate the statistical values, which are shown in Fig. 8. It is clear that our proposed method outperforms the benchmarks in load forecasting. This makes sense since the forecasting performance is determined by the quality of synthesized load profiles (see details in Sections IV-B and IV-C). The model trained using augmented data from our proposed method achieves the lowest mean and median of RMSE, implying the highest forecasting accuracy. Moreover, the RMSE range of the federated learning is wider than the proposed method, indicating that the global model’s accuracy for each customer is relatively poor. This is because the global model trained through federated learning loses some accuracy to guarantee generalization. Therefore, at the customer level, our proposed method is a better choice to improve the model performance on real-world data analytics applications.

## V. CONCLUSION

In this paper, we concentrate on customized load profiles synthesis for electricity customers. High-quality customized synthetic load profiles enable customers to build personalized data-driven models for various data analytics applications, e.g., load forecasting and energy management. We propose a novel customized load profiles synthesis method based on conditional diffusion models, which generates tailored load profiles conditioned on the customer’s load characteristics and individual demands. Furthermore, we design a noise estimation model with the attention mechanism and skip connection to implement the conditional diffusion models. Case studies comprehensively verify that our proposed method can realize

customized load profiles synthesis and outperform existing methods. The average enhancement of evaluation metrics is 17.5% and 21% in the data generation and data augmentation scenarios, respectively. Moreover, the proposed method aids customers in improving the performance of their data-driven models on application tasks, such as load forecasting, by providing synthetic training data.

Due to the widespread deployment of smart meters and the increasing attention on data security, how to design efficient and secure data protection approaches is an important research direction. We will consider this topic in our future work.

## REFERENCES

- [1] L. Xie, C. Singh, S. K. Mitter, M. A. Dahleh, and S. S. Oren, "Toward carbon-neutral electricity and mobility: Is the grid infrastructure ready?," *Joule*, vol. 5, no. 8, pp. 1908–1913, 2021.
- [2] D. Alahakoon and X. Yu, "Smart electricity meter data intelligence for future energy systems: A survey," *IEEE Trans. Industr. Inform.*, vol. 12, no. 1, pp. 425–436, 2015.
- [3] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018.
- [4] Y. Chen, L. Xu, A. Egea-Alvarez, and B. Marshall, "Accurate and general small-signal impedance model of lcc-hvdc in sequence frame," *IEEE Trans. Power Deliv.*, vol. 38, no. 6, pp. 4226–4241, 2023.
- [5] 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, p. 122544, 2024.
- [6] 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, 2024.
- [7] S. Lee and D.-H. Choi, "Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources," *IEEE Trans. Industr. Inform.*, vol. 18, no. 1, pp. 488–497, 2020.
- [8] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated learning for internet of things: Recent advances, taxonomy, and open challenges," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 3, pp. 1759–1799, 2021.
- [9] J. M. G. López, E. Poursmaeil, C. A. Cañizares, K. Bhattacharya, A. Mosaddegh, and B. V. Solanki, "Smart residential load simulator for energy management in smart grids," *IEEE Trans. Ind. Electron.*, vol. 66, no. 2, pp. 1443–1452, 2018.
- [10] L. Diao, Y. Sun, Z. Chen, and J. Chen, "Modeling energy consumption in residential buildings: A bottom-up analysis based on occupant behavior pattern clustering and stochastic simulation," *Energy Build.*, vol. 147, pp. 47–66, 2017.
- [11] A. Al-Wakeel, J. Wu, and N. Jenkins, "K-means based load estimation of domestic smart meter measurements," *Appl. Energy*, vol. 194, pp. 333–342, 2017.
- [12] W. Labeeuw and G. Deconinck, "Residential electrical load model based on mixture model clustering and markov models," *IEEE Trans. Industr. Inform.*, vol. 9, no. 3, pp. 1561–1569, 2013.
- [13] S. El Kababji and P. Srikantha, "A data-driven approach for generating synthetic load patterns and usage habits," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 4984–4995, 2020.
- [14] L. Song, Y. Li, and N. Lu, "Profiles-gan: A gan based super-resolution method for generating high-resolution load profiles," *IEEE Trans. Smart Grid*, vol. 13, no. 4, pp. 3278–3289, 2022.
- [15] J. Huang, Q. Huang, G. Mou, and C. Wu, "Dpwgan: High-quality load profiles synthesis with differential privacy guarantees," *IEEE Trans. Smart Grid*, vol. 14, no. 4, pp. 3283–3295, 2023.
- [16] Z. Chen, J. Li, L. Cheng, and X. Liu, "Federated-wdpcgan: A federated smart meter data sharing framework for privacy preservation," *Appl. Energy*, vol. 334, p. 120711, 2023.
- [17] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, "Model-free renewable scenario generation using generative adversarial networks," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3265–3275, 2018.
- [18] L. Zhang and B. Zhang, "Scenario forecasting of residential load profiles," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 1, pp. 84–95, 2019.
- [19] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020.
- [20] L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, Y. Shao, W. Zhang, B. Cui, and M.-H. Yang, "Diffusion models: A comprehensive survey of methods and applications," *arXiv preprint arXiv:2209.00796*, 2022.
- [21] J.-F. Toubeau, F. Teng, T. Morstyn, L. Von Krannichfeldt, and Y. Wang, "Privacy-preserving probabilistic voltage forecasting in local energy communities," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 798–809, 2022.
- [22] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *International Conference on Machine Learning*, pp. 2256–2265, PMLR, 2015.
- [23] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah, "Diffusion models in vision: A survey," *arXiv preprint arXiv:2209.04747*, 2022.
- [24] Y. Tashiro, J. Song, Y. Song, and S. Ermon, "Csd: Conditional score-based diffusion models for probabilistic time series imputation," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 24804–24816, 2021.
- [25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.
- [26] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *International Conference on Learning Representations*, pp. 1–17, 2021.
- [27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [28] J. R. Schofield, R. Carmichael, S. Tindemans, M. Bilton, M. Woolf, G. Strbac, et al., "Low carbon london project: Data from the dynamic time-of-use electricity pricing trial, 2013," *UK Data Service, SN*, vol. 7857, no. 2015, pp. 7857–7851, 2015.
- [29] Y. Luo, X. Cai, Y. ZHANG, J. Xu, and Y. xiaojie, "Multivariate time series imputation with generative adversarial networks," in *Advances in Neural Information Processing Systems*, vol. 31, 2018.
- [30] Y. Li, J. Li, and Y. Wang, "Privacy-preserving spatiotemporal scenario generation of renewable energies: A federated deep generative learning approach," *IEEE Trans. Industr. Inform.*, vol. 18, no. 4, pp. 2310–2320, 2021.
- [31] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.
- [32] 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, 2023.



**Zhenyi Wang** (S'22) 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 at University of Macau, Macao, 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** (S'14–M'18–SM'23) received the B.S. and Ph.D. degree 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 Department of Electrical and Computer Engineering, University of Macau, Macao, China. In 2018–2019, he was a postdoctoral scholar with the University of California, Berkeley. He is an Associate Editor of IEEE Transactions on Power Systems and Associate Editor of Journal of Modern Power Systems and Clean Energy.

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.