

# A Two-Stage Generative Architecture for Renewable Scenario Generation: Combining Temporal Scenario Representation and Diffusion Models

Chenglong Xu, Peidong Xu, Yuxin Dai, Shi Su, Jun Zhang, *Senior Member, IEEE*, Yuyang Bai, Tianlu Gao, Qingyang Xie, Lei Shang, and David Wenzhong Gao, *Fellow, IEEE*

**Abstract**—Scenario generation is an effective method for addressing uncertainties in stochastic programming for power systems with integrated renewable resources. In recent years, numerous studies have explored the application of deep generative modeling to scenario generation. However, most applications do not take into account the temporal correlation inherent in renewable energy output, resulting in an insufficient description of uncertainty. In this paper, we propose a novel two-stage generative architecture for renewable scenario generation using diffusion models. Specifically, in the first stage the temporal features of the renewable energy output are learned and encoded into the hidden space by means of a representational model with an Encoder-Decoder structure, which provides the inductive bias of the scenario for generation. In the second stage, the real distribution of vectors in the hidden space is learned based on the conditional diffusion model, and the generated scenario are obtained through decoder mapping. The case study demonstrates the effectiveness of this architecture in generating high-quality renewable scenarios. In comparison to advanced deep generative models, the proposed method achieves state-of-the-art performance in a comprehensive evaluation.

**Index Terms**—Renewable resource, generative model, diffusion model, scenario generation, scenario representation.

## I. INTRODUCTION

THE increasing penetration of renewable resources bring about heightened intermittency and randomness, significantly impacting the stable operation of power systems. Within the analysis of power systems incorporating renewable resources, modeling the uncertainty inherent in it has emerged as a pivotal concern. Scenario analysis, by reflecting the probabilistic characteristics of uncertain variables, can describe the uncertainty of renewable resource output. Based on this approach, stochastic planning models can be established for objectives such as economic dispatch [1] and unit commitment [2]. The fundamental concept of scenario analysis involves discretizing a random vector with a continuous probability

This research was funded by Science and Technology Program of Yunnan PowerGrid (YNKJXM20222105).

C. Xu, P. Xu, Y. Dai, J. Zhang (corresponding author), Y. Bai, T. Gao, and L. Shang are with the School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China. (email: jackie.xu@whu.edu.cn; xupd@whu.edu.cn; YuxinDai@whu.edu.cn; jun.zhang.ee@whu.edu.cn; baiyuyang@whu.edu.cn; tianlu.gao@whu.edu.cn; shanglei@whu.edu.cn)

S. Su and Q. Xie are with Electric Power Research Institute of Yunnan Power Grid Co., Ltd, Kunming 650000, Yunnan Province, China. (email: 472732908@qq.com; qingyxyie@qq.com)

D. W. Gao is with Department of Electrical and Computer Engineering, Denver, Colorado, 80208, USA (email: wenzhong.2001.gao@ieee.org).

distribution into a set of scenarios, effectively transforming a stochastic optimization problem into a deterministic one. The precision of the scenarios constructed in scenario analysis directly influences the proximity of the solution to the corresponding optimal value in the actual stochastic optimization problem. Therefore, accurately generating renewable scenarios stands out as a crucial research direction.

In numerous prior studies, the generation of renewable scenarios has commonly relied on a sampling method grounded in mathematical modeling. That is, a probabilistic model that conforms to the distribution pattern of renewable energy output is built from historical data, and then new scenarios for analysis are generated through sampling. Specific methods include the Monte Carlo (MC) [3], Latin Hypercube Sampling (LHS) [4], and Copula function [5]. The MC method has the advantages of simplicity and speed. However, when the MC method is combined with simple random sampling, it faces the challenges of prolonged computation and heavy storage requirements. The LHS method has been proposed to address this issue [6]. The Copula method involves scenarios that are obtained by sampling the copula function. Camal et al. [7] proposes a multivariate copula modeling approach to predict related scenarios for renewable resources and applied it to unit commitment, effectively reducing system operating costs. However, methods based on sampling usually necessitate assumptions about prior probabilities. Therefore, the challenge of the methods lie in enhancing the similarity between the scenario set and the problem to be addressed by establishing an appropriate probability model. In practice, the variability of climate, the nonlinearity of energy conversion in generators, and the intricate temporal-spatial relationships between sites make it challenging to construct accurate probability models.

Another commonly adopted method is the optimization-based scenario generation approach. This method involves scenario reduction to obtain suitable scenarios from the set of all possible scenarios, including clustering, forward selection, moment matching, and others.

With the advancements in artificial intelligence technology, data-driven methods are emerging in power system scenario generation research. A novel approach integrates an extreme learning machine with the pairs bootstrap method for probabilistic forecasting of wind power generation [8]. In recent years, generative models, represented by Variational Auto-Encoders (VAE) [9] and Generative Adversarial Networks (GANs) [10], have found numerous applications in the gener-

ation of renewable scenarios. In [11], a GANs based method is firstly applied to generate renewable scenarios, which is able to effectively learn the features of renewable energy output and provides a robust generative idea. In [12], a privacy-preserving method is proposed for renewable scenario generation by integrating federated learning and Least Square Generative Adversarial Networks (LSGANs). In [13], GANs is used to learn the distribution of historical data and employ stochastic constrained optimization to generate predictive scenarios. In [14], the utilization of VAE is implemented for scenario generation in the context of coordinated optimization within a multi-energy system. In [15], a conditional VAE based approach is proposed for wind power forecasting. However, the performance of the VAE model in practical applications is limited by its posterior distribution. Ensuring the convergence and stability of GANs during the training process is challenging. This may result in insufficient diversity of samples generated by GANs and difficulty in capturing the entire distribution of real data. Diffusion models constitute a new category of data-driven generative models, originating from [16]. Subsequently, it has demonstrated superior capabilities in the field of images compared to GANs. [17] indicates that diffusion models outscore GANs in diverse tests. The introduction of Stable Diffusion further proves its superior performance in terms of image generation quality [18]. Nevertheless, diffusion models encounter challenges, including slow sampling speed and a constrained ability to generalize to various data types. The above generative models exclusively focused on learning the global data distribution. When applied to renewable scenarios, most of them do not consider modeling temporal characteristics, thereby overlooking the crucial feature of time-related relationships.

In addition, renewable scenarios, as a type of time series, possess notable temporal features. Time series representation learning methods aim to adeptly extract meaningful features from such data, transforming the raw data into a more informative representation. Franceschi et al. proposes a method combining causal dilated convolutions with time-based negative sampling to obtain the general-purpose representation of multivariate time series [19]. In [20], a method based on the Transformer encoder structure is first proposed for multivariate time series characterization and achieves satisfactory results in classification tasks. However, these methods provide a coarse-grained representation of the entire time series, which may not achieve adequate performance. If the Transformer structure is applied for encoding, there is a problem of quadratic growth in computational complexity, leading to diminish the efficacy in renewable scenarios with strong uncertainty and long time series.

To address the above issues, we propose a two-stage generative architecture that combines scenario representation with diffusion models for renewable scenario generation. In the first stage, we introduce a time series representation model with an Encoder-Decoder structure to enhance the temporal features of renewable scenarios within the latent space. In the second stage, we employ an conditional diffusion model to learn the joint distribution of latent space variables. The diffusion model maps random noise to real data and achieves the generation of

renewable scenarios through the decoder of the first stage. The main contributions of this paper are summarized as followings:

- 1) This paper proposes a novel two-stage generative architecture for renewable scenario generation using diffusion models. The architecture innovatively introduces a priori temporal knowledge learning, which solves the problem that diffusion models are difficult to model temporal correlations. The diffusion model is extended to the conditional implicit one, enabling deterministic generation of specific scenarios.
- 2) This paper introduces a time series representation module suitable for renewable resources, which is a crucial part of scenario generation. This module partitions the scenario into multiple short sequences through patching, and with the incorporation of positional encoding, it achieves the modeling of time series, addressing the challenge of capturing temporal relationships in long renewable sequences.
- 3) Through comprehensive experiments, this paper substantiates the effectiveness and reliability of the proposed method. Compared with the current mainstream deep generative models, including GANs, VAE, and diffusion models, the results of qualitative and quantitative evaluations from various perspectives indicate that the proposed method achieves state-of-the-art performance in scenario generation task.

The rest of this paper is organized as follows. Section II describes the basic principles of time series representation and diffusion models. The proposed two-stage generative architecture is elaborated in Section III. Section IV validates the effectiveness of the proposed method, and Section V presents the conclusions of this paper.

## II. TIME SERIES REPRESENTATION AND DIFFUSION MODELS

### A. Problem Formulation

Assume that the number of observed renewable resources sites is  $M$ . Let  $x_s \in \mathbb{R}^L$  denotes the output value of renewable site  $s$  in the historical observation data, where  $s = 1, \dots, M$ ,  $L$  is the sequence length. The time series representation aims to study a non-linear projection function which maps each  $x_s$  to its optimal self-described representation  $r_s$ . The  $r_s$  can be expressed by the joint probability distribution  $p_r(r_s|x_s)$ . The purpose of scenario generation is to train a generative model  $G_\theta(\cdot)$  with parameter  $\theta$ . Through the learning of historical data, it achieves the mapping from distribution  $p_r(r_s|x_s)$  to synthetic data  $\hat{x}_s \sim p_x(\hat{x}_s|r_s)$  whose distribution is as close as possible to the unknown distribution  $p_x(x_s)$  of historical data. These generated time series data constitute the scenarios and adhere to the following objective function:

$$\min \|P_x[\hat{x}_s] - P_x[x_s]\| \leq \varepsilon \quad (1)$$

where  $P_x(\cdot)$  is the probability operator and  $\varepsilon$  denotes an acceptable distribution error. The set of scenarios that satisfy objective (1) can depict a stochastic process with similar characteristics to the real renewable output data.

### B. Time Series Representation

Consider scenario sample  $x_s \in \mathbb{R}^L$ , the time masked encoder module was applied to perform temporal feature extraction for renewable scenarios. Inspired by [21], we introduce patches to chunk the original time series.

1) *Patching The Series*: Each input sample undergoes an initial segmentation into multiple non-overlapping patches. Denote the length of a patch as  $P$  and the interval between two consecutive patches as  $S$ . Consequently, this patching procedure yields a sequence of patches  $x_p \in \mathbb{R}^{P \times N}$ , where  $N = \frac{L+S}{P+S}$  is the number of patches.

2) *Masked Mechanism*: After patching, a subset of patches is selected and masked to value 0 while recording its indices. The selection strategy employed here is a simple uniform distribution random selection to prevent potential center bias.

3) *The Transformer Based Encoder*: To achieve the representation of renewable scenarios, we use a vanilla Transformer encoder to map the patched samples to the latent space. Neural networks with a Transformer structure inherently possess the capability for temporal modeling due to the fact that they carry encoding of sequence position information. The input patches are projected to a potential representation with  $D$  dimensions via a learnable embedding layer  $\mathbf{W}_p \in \mathbb{R}^{D \times P}$ . Furthermore, an additional positional information encoding  $\mathbf{W}_{pos} \in \mathbb{R}^{D \times N}$  is incorporated to delineate the sequential order of patches. The embedded sample  $x_d$  is articulated as follows:

$$x_d = \mathbf{W}_p x_p + \mathbf{W}_{pos} \quad (2)$$

Subsequently,  $x_d$  will be fed into the Transformer encoder to generate the representation information of the input sample denoted as  $r \in \mathbb{R}^{D \times N}$  in the latent space.

4) *Loss Function*: A linear head  $\mathbf{W}_o \in \mathbb{R}^{P \times D}$ ,  $b_o \in \mathbb{R}^{P \times N}$  is applied over the ultimate representation vector  $r$ , forecasting the  $\hat{x}_p$  for the input patches  $x_p$ :

$$\hat{x}_p = \mathbf{W}_o r + b_o \quad (3)$$

Typically, the mean square error (MSE) loss is used to measure the discrepancy between the estimated and true masked patches:

$$\mathcal{L}_{MSE} = \frac{1}{M} \sum_{i \in M} (\hat{x}_i - x_i)^2 \quad (4)$$

### C. Denoising Diffusion Probabilistic Models

The denoising diffusion probabilistic model (DDPM) is a generative model, employing variational inference to train a parameterized Markov chain capable of generating samples that align with the original data. To distinguish the time steps between the concept of diffusion models and the time series, we denote  $T$  as the diffusion steps and  $L$  as the number of periods per time series. We consider a dataset  $O = \{r^i\}_{i=1}^N$  comprising  $N$  continuous variables  $r^i \in \mathbb{R}^{L \times F}$  sampled from the real represented distribution  $p_r(r)$ , where  $F$  is the feature dimension. Let  $r_0$  denote the 0th diffusion step of  $r^i$ ,  $r_1, \dots, r_t, \dots, r_T$  are diffusion variables of the same dimensionality as the data  $r_0$ . The DDPM aims to learn the underlying distribution  $p_\theta(r_0)$  that is similar to  $p_r(r_0)$ . On a high level, the DDPM samples noise from a Gaussian

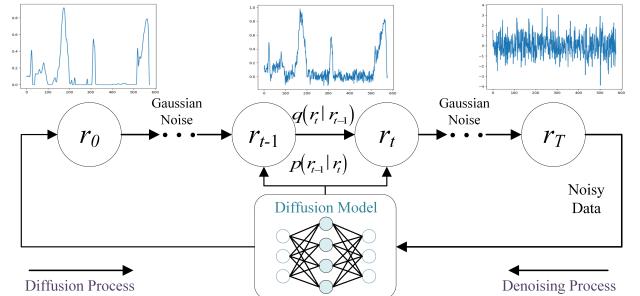


Fig. 1. Illustration of diffusion models.

distribution and transforms it to conform to the distribution  $p_r(r)$  through the reversal of a gradual noising process. In particular, sampling starts with noise  $r_T$  and produce less-noisy samples  $r_{T-1}, r_{T-2}, \dots$ , gradually, until reaching a final sample  $r_0$ . Formally, DDPM is a class latent variable model of the form:

$$p_\theta(r_0) := \int p_\theta(r_{0:T}) dr_{1:T} \quad (5)$$

As illustrated in Fig. 1, the training process of DDPM primarily consists of two parts: the diffusion process and the denoising process. The diffusion process involves incrementally adding noise to the data through a Markov chain until the signal is degraded. The denoising process involves learning to reverse a diffusion process, gradually eliminating noise in the opposite direction of diffusion.

1) *The Diffusion Process*: This process is not learned but fixed, distinguishing diffusion models from other latent variable models. Indeed, the approximate posterior  $q(r_{1:T} | r_0)$  is derived from a Markov chain that systematically introduces Gaussian noise to the data  $r_0$  which can be formulated as:

$$q(r_{1:T} | r_0) := \prod_{t=1}^T q(r_t | r_{t-1}) \quad (6)$$

For each diffusion step  $q(r_t)$ , it depends only on the preceding step  $q(r_{t-1})$ , obtained by adding noise to the input  $r_{t-1}$ , and can be defined as follows:

$$q(r_t | r_{t-1}) := \mathcal{N}\left(r_t; \sqrt{1 - \beta_t^2} r_{t-1}, \beta_t^2 \mathbf{I}\right) \quad (7)$$

where  $\beta_t \in (0, 1)$  is a predetermined schedule of variance with no learnable parameters. This procedure is capable of distorting the original signal, inducing it to adhere to a standard Gaussian distribution  $\mathcal{N}(0, \mathbf{I})$ .

2) *The Denoising Process*: This part anticipates finding  $p(r_0 | r_T)$  to eliminate the noise introduced by the diffusion process for facilitating the recovery of  $r_0$ . As  $p(r_{t-1} | r_t)$  is unknown, the reverse process cannot obtain an explicit expression similar to (6) directly. Under the given condition for  $r_0$ , the application of Bayes' theorem yields:

$$p(r_{t-1} | r_t, r_0) = \frac{p(r_t | r_{t-1}) p(r_{t-1} | r_0)}{p(r_t | r_0)} \quad (8)$$

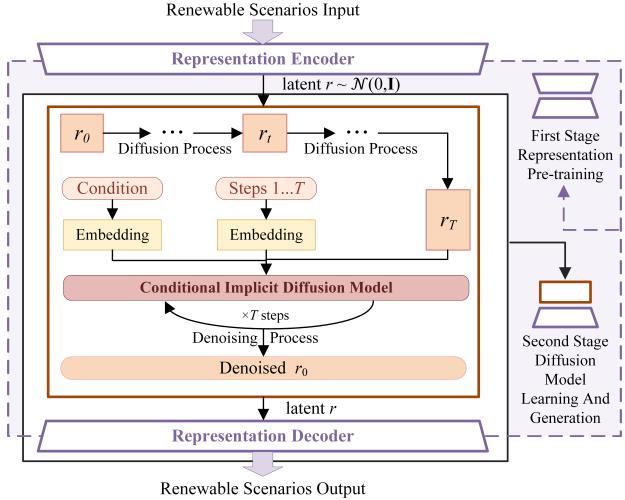


Fig. 2. Training and generation process of the two-stage architecture using diffusion model.

Let  $\alpha_t = \sqrt{1 - \beta_t^2}$ ,  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$ ,  $\bar{\beta}_t = \sqrt{1 - \bar{\alpha}_t^2}$ , (8) can be rewritten as:

$$p(r_{t-1}|r_t, r_0) = \mathcal{N}\left(r_{t-1}; \frac{\alpha_t \bar{\beta}_{t-1}^2}{\beta_t^2} r_t + \frac{\bar{\alpha}_{t-1} \beta_t^2}{\beta_t^2} r_0, \frac{\bar{\beta}_{t-1}^2 \beta_t^2}{\beta_t^2} \mathbf{I}\right) \quad (9)$$

where  $r_0$  is the result we ultimately want to generate, which makes the expression conflict with our purpose. Therefore a neural network  $\bar{u}(r_t)$  is introduced to estimate  $r_0$ , the loss function is  $\|r_0 - \bar{u}(r_t)\|^2$ . From the diffusion process it can be deduced that  $r_t = \bar{\alpha}_t r_0 + \bar{\beta}_t \varepsilon$ , where  $\varepsilon$  is a known Gaussian noise.  $\bar{u}_t$  can be parameterized as:

$$\bar{u}(r_t) = \frac{1}{\bar{\alpha}_t} (r_t - \bar{\beta}_t \varepsilon_\theta(r_t, t)) \quad (10)$$

The loss function is derived as follows:

$$\mathcal{L}_{DM} = \mathbb{E}_{\varepsilon \sim \mathcal{N}(0, I)} \left[ \frac{\bar{\beta}_t^2}{\sqrt{\bar{\alpha}_t^2}} \|\varepsilon - \varepsilon_\theta(\bar{\alpha}_t r_0 + \bar{\beta}_t \varepsilon, t)\|^2 \right] \quad (11)$$

The conclusive denoising process is also constrained to a Markov chain, parameterized by the neural network and defined as follows:

$$p_\theta(r_0 | r_T) = p(r_T) \prod_{t=1}^T p_\theta(r_{t-1} | r_t) \quad (12)$$

### III. METHODOLOGY

When applied to renewable power output scenarios generation, the DDPM fails to accurately model their temporal correlation, resulting in an insufficient description of uncertainty. In this section, we elaborate on the proposed method for renewable scenario generation. It is a novelty two-stage generative architecture that combines time series representation and diffusion models, as depicted in Fig. 2.

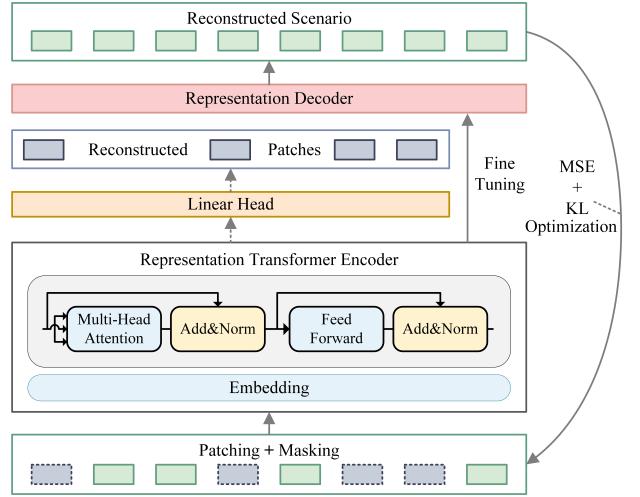


Fig. 3. Structure of the first stage scenario representation learning.

#### A. Scenario Representation of The First Stage

In the first stage, according to section II-B, we have previously trained the time-masked based representation model through unsupervised learning, which comprises the vanilla Transformer structure [22]. As shown in Fig. 3, the model can be viewed as an Encoder-Decoder architecture. Each set of training samples is divided into  $N$  patches before entering the encoder. In this work, we choose  $P = 12$  and  $S = 0$ , with a sample sequence length  $L = 288$ . Therefore,  $N = \frac{L+S}{P+S} = 24$ . Each sample is divided into tightly non-overlapping patches to ensure that the observed patches do not contain masked information. Following this, we randomly and uniformly select a subset of patches, which constitutes 50 % of the total patches. This subset is masked with 0 values, and the indices are recorded. The entirety of patches, including the masked subset, is subsequently fed into the encoder to derive their representation within the latent space. The model's output layer uses a simple linear layer instead of the vanilla Transformer structure's autoregressive decoding output. This approach enables multi-step direct prediction to avoid error accumulation effects and slow decoding efficiency. Finally, the model is trained using MSE loss to reconstruct the masked patches based on their indices. In fact, our representation model includes the step to reconstruct the original data from the representation vectors in the latent space. Therefore, we removed the original output layer, retained the encoder module, and appended a linear layer as its decoder module. Network parameter fine-tuning was then performed, where the output dimension of the linear layer was aligned with the input dimension of the model. Note that, this step is essential not only to ensure the accuracy of reconstructing samples but also to constrain the scaling of the latent space within a certain range. It is to maintain stability in the subsequent learning of the diffusion model in the latent space and avoid distribution disturbances caused by outlier samples. Hence, the loss function for the fine-tuning process should include Kullback–Leibler (KL) regularization on the basis of (4), defined as follows:

$$\mathcal{L} = \mathcal{L}_{MSE} + \omega KL(\mathcal{N}(\mu, \sigma) \| \mathcal{N}(0, I)) \quad (13)$$

where  $\mu$  and  $\sigma$  are the mean vector and variance vector of a normal distribution, and  $\omega$  denotes the regularization weight. It can be explicitly expressed as follows:

$$\mathcal{L} = \frac{1}{N} \sum_{i \in N} (\hat{x}_i - x_i)^2 + \frac{1}{2} \sum_{j=1}^D (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1) \quad (14)$$

where  $D$  represents the dimension of the latent variable  $r$ . Through KL regularization, the model aligns the distribution of the encoded latent space with a standard normal distribution during the fine-tuning process, thus avoiding arbitrary dimension scaling.

### B. The Diffusion Model of The Second Stage

In the second stage, the diffusion model comprehensively learns the underlying data distribution in the latent space with temporal inductive bias, where all renewable scenario data undergo encoded by the pre-trained representation model encoder from the first stage, as shown in Fig. 2.

1) *Conditional Implicit Diffusion Model*: To synthesize customized renewable scenarios for specific scenario problem analysis, we extend the traditional DDPM to the conditional one. Given that the denoising process initiates with Gaussian noise and lacks information about the target output  $r_0$ , the generation procedure becomes inherently uncontrollable. We introduce the label  $c$  as conditional information for the sample and embed it into the prediction neural network  $\bar{u}(r_t)$  to guide conditional generation. Therefore,  $\bar{u}(r_t)$  can be parameterized as:

$$\bar{u}(r_t) = \frac{1}{\bar{\alpha}_t} (r_t - \bar{\beta}_t \varepsilon_\theta(r_t, t, c)) \quad (15)$$

From an advanced standpoint, although the denoising process mentioned in Section II-C forms a Markov chain,  $p(r_t | r_{t-1})$  does not directly contribute to the calculation of the final outcome. The term  $p(r_t | r_{t-1})$  can be omitted from the derivation procedure. At this point, the solution space for (8) becomes more expansive and needs to satisfy the marginal distribution condition:

$$\int p(r_{t-1} | r_t, r_0) p(r_t | r_0) dr_t = p(r_{t-1} | r_0) \quad (16)$$

Similar to (9), we solve the above (16) using the method of undetermined coefficients, assuming the following:

$$p(r_{t-1} | r_t, r_0) = \mathcal{N}(r_{t-1}; \kappa_t r_t + \lambda_t r_0, \sigma_t^2 \mathbf{I}) \quad (17)$$

where  $\kappa_t, \lambda_t, \sigma_t$  is the undetermined coefficient. Taking  $\sigma_t$  as the variable parameter, the final result can be derived according to (5):

$$p(r_{t-1} | r_t, r_0) = \mathcal{N}\left(r_{t-1}; \frac{\gamma}{\bar{\beta}_t} r_t + \left(\bar{\alpha}_{t-1} - \frac{\bar{\alpha}_t \gamma}{\bar{\beta}_t}\right) r_0, \sigma_t^2 \mathbf{I}\right) \quad (18)$$

where  $\gamma = \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}$ . As in (15),  $r_0$  is approximated by a neural network  $\bar{u}(r_t)$ . Let  $\sigma_t = \eta \frac{\bar{\beta}_{t-1} \bar{\beta}_t}{\bar{\beta}_t}$ , where  $\eta \in [0, 1]$ . The retained term  $\sigma_t$  in (18) can influence the noise intensity to control the randomness of the denoising process. When  $\eta = 1$ , (18) transforms into (9). When  $\eta$  is set to 0, the denoising process becomes a deterministic transformation:

$$r_{t-1} = \frac{1}{\bar{\alpha}_t} (r_t - (\bar{\beta}_t - \alpha_t \bar{\beta}_{t-1}) \varepsilon_\theta(r_t, t, c)) \quad (19)$$

2) *Training of The Diffusion Model*: For each sample  $r_0$ , the diffusion process adds Gaussian noise to it according to (6), resulting in  $r_T \sim \mathcal{N}(0, \mathbf{I})$ , where  $\beta$  schedule is a linear increasing schedule from  $\beta_1 = 10^{-4}$  to  $\beta_T = 0.1$ , and

$T = 100$ . In the denoising process, conditional information  $c$  such as weather, and time step  $t$  are introduced. According to (15) and (19), one can obtain  $r_{t-1}$  from the Bayesian transformation of  $r_t$ . Following  $T$  iterations, the noise added to  $r_T$  is systematically eliminated to restore the original  $r_0$ . Subsequently, the diffusion model undergoes optimization based on the loss function:  $\|\varepsilon - \varepsilon_\theta(\bar{\alpha}_t r_0 + \bar{\beta}_t \varepsilon, t, c)\|^2$ .

It can be seen that we explicitly incorporate the parameter  $t$  in the input. This is because, in principle, different time steps  $t$  correspond to different noise-level objects, necessitating diverse diffusion models. Ideally, there should be  $T$  distinct ones. Nevertheless, we opt to share parameters across all diffusion models. The time step  $t$ , converted into the corresponding position encoding, serves as a condition and is directly integrated into the embedding block. In this work, the neural network  $\varepsilon_\theta$  of the diffusion model adopts the U-Net structure as the backbone, and specific details can be referred to [23]. In this structure, the ResNet module is replaced with the ConvNeXt module [24].

Upon completion of the diffusion model training, the generation of new scenarios involves the combination of two-stage models. Setting  $\sigma_t = 0$ , a random  $r_T$  is sampled from a standard Gaussian distribution. The trained diffusion model from the second stage is then utilized to generate a latent space vector  $r_g$ , following the procedure outlined in (18). Subsequently, this vector  $r_g$  is mapped to the entire renewable scenario through the pre-trained representation decoder from the first stage.

## IV. CASE STUDIES

### A. Experiment Settings

1) *Data Preparation*: We employed the integrated wind and solar energy dataset sourced from NREL [25]. The experiment

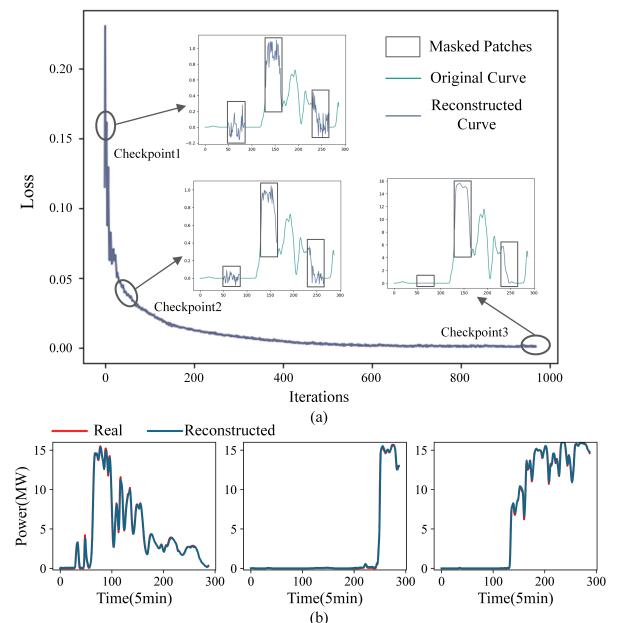


Fig. 4. Illustration of representation results.

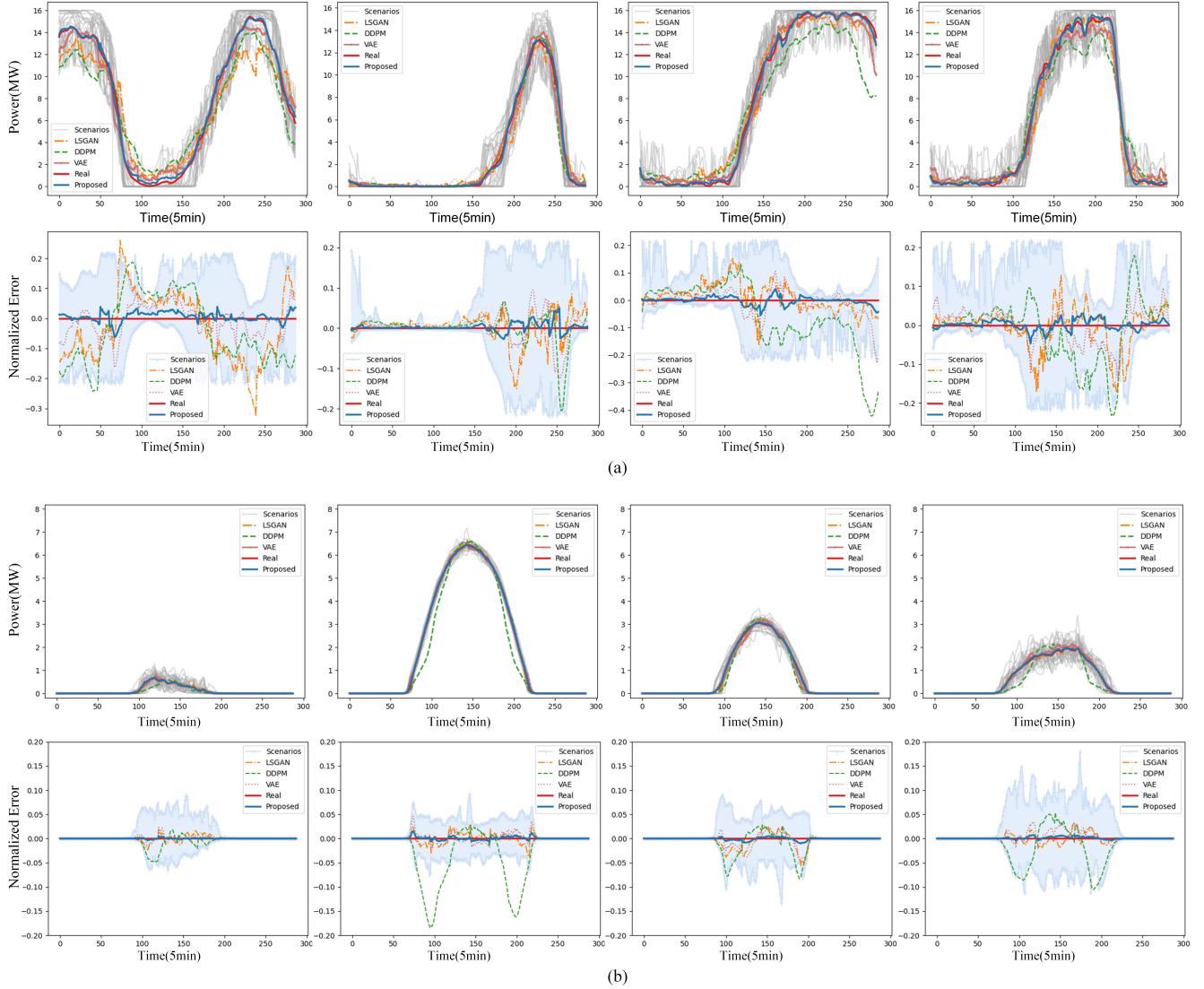


Fig. 5. The comparison of typical scenarios generated by different models. (a) is wind power and (b) is solar power.

utilized data from 24 wind and 32 solar sites situated in Washington state as input for the model. The selected wind and solar stations possess consistent installed capacities respectively, are geographically proximate, and display a certain level of correlation between the outputs of each site. The historical data exhibit a time resolution of 5 minutes. We use 80% of them as the training set, while the remaining 20% served as the test set for the model to assess against, thereby providing a genuine comparison baseline.

2) *Benchmarks:* In many previous studies, it has been demonstrated that artificial intelligence methods are superior to traditional probabilistic models. Therefore, this paper primarily compares several advanced deep generative models, including LSGANs [12], VAE [15], and DDPM [16]. In this work, we introduce the original DDPM as a benchmark for scenario generation. It can be considered as an ablation experiment to highlight the superiority of our proposed method. All experiments were conducted on the CentOS 7 platform,

equipped with 4 NVIDIA GeForce RTX 2080 Ti GPUs.

### B. Performance In Scenario Representation

As this work focuses on scenario generation, this section briefly illustrates the efficacy of the first-stage representation. As shown in Fig. 4(a), the loss value continuously decreases during the training process until convergence. We present the reconstruction results of different checkpoints during the training process in it. Furthermore, three randomly selected samples from the test set are visually displayed in Fig. 4(b), and the MSE calculated from the test set is less than 0.001. It can be observed that the reconstructed curves almost overlap with the original curves, demonstrating the effectiveness of the first-stage autoencoder representation learning.

### C. Performance In Scenario Generation

In this section, we validate the performance of our proposed method in scenario generation by comparing it with the

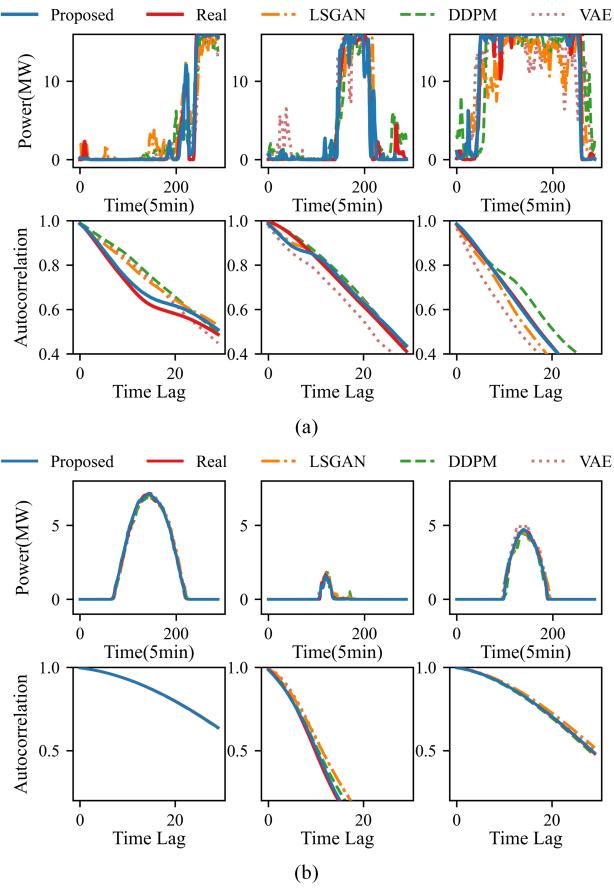


Fig. 6. The comparison of individual samples generated by different methods. (a) is wind power and (b) is solar power.

three benchmarks mentioned above. The NREL power data comprises 288 points per day. All three benchmark models convert the 2-day data (576 points) into a  $24 \times 24$  matrix as a sample for training. This is because most current deep generative models are initially designed for the image domain with uniform dimensions. When analyzing the data, they treat the generated 2-day data as two separate samples, inevitably leading to some deviation in describing the daily output of renewable energy. In our approach, during the training phase, 288 points are input as a time series into the representation encoder to derive its mapped latent vector. Subsequently, the diffusion model comprehensively learns the potential distribution of vectors within the latent space. Upon completing the training of all models, a substantial number of scenarios are randomly generated for consequent analysis of the results.

### 1) Numerous Scenarios Analysis:

Due to the strong randomness of renewable energy output and the data generated by different models, it is challenging to compare each sample individually. To address this, for both the generated scenario set and the test one, we employ the k-means clustering algorithm to partition them into 12 clusters. However, it's important to note that these clusters may not entirely summarize the output characteristics of renewable energy, particularly for wind power scenarios with pronounced fluctuations. To enhance comparability, we select several

categories of representative sets from these clusters. These scenarios are further matched with curves sharing similar random characteristics through Euclidean distance searching, facilitating the construction of a set of typical scenarios. Fig. 5 depicts the comparison between the real scenario set and the generated sets from four methods. For clarity, the gray curves represent the scenario set generated by our proposed method, while the remaining curves represent centroids obtained by computing the typical scenario sets from real data and various benchmarks. From the top rows of Fig. 5(a) and Fig. 5(b), it can be observed that the scenario set generated by the proposed method follows the trend of the actual curves, perfectly covering the real fluctuation range. Furthermore, the centroid of this scenario set is closest to the real one, which can be interpreted as the proposed method having the most similar maximum likelihood estimation to real data. Fig. 5 indicates that the proposed method has effectively learned the distinctive features of renewable energy generation, such as peak-valley variations, rapid power fluctuations, and ramp events in wind power, as well as day-night variations in solar power. It can accurately capture the dynamic characteristics of the practical renewable energy generation process.

For a more detailed display, the bottom rows of Fig. 5(a) and Fig. 5(b) normalize all errors of the curves for comparison. The baseline is the centroid of real samples, and the upper and lower bounds of the error represent the maximum and minimum normalized error values of the generated scenario set at different time points. It is evident that, for wind power scenarios, the centroid generated by the proposed method exhibits an absolute error within 0.05 relative to the real clustering center. Furthermore, the error bounds of its scenario set do not exceed 0.3. In contrast, the normalized error limits of other benchmarks generally surpass twice that of the proposed method. Notably, the original DDPM displays a substantial deviation, highlighting the significant improvement of our method over the basic DDPM. In the case of solar power, its distribution is more regular compared to wind power. All benchmarks perform well, with the proposed method's relative error astonishingly below 0.02.

As shown in Fig. 6, we selected individual samples with representative characteristics for comparison with real samples. It can be observed that they exhibit similar fluctuating and intermittent features. Then, we calculated the autocorrelation coefficients  $R(\tau)$  to verify their temporal correlation. The formula is shown below:

$$R(\tau) = \frac{E[(S_t - \mu)(S_{t+\tau} - \mu)]}{E(S_t - \mu)^2} \quad (20)$$

where  $S_t$  is the sample output at time  $t$ ,  $\mu$  represents the output series average,  $\tau$  is the time interval. It can be observed that the autocorrelation coefficient curves of the samples generated by the proposed method maintain a high consistency with those of the real samples. This demonstrates that the method can reflect the operational characteristics of real power generation scenarios in terms of temporal features.

### 2) Statistical analysis:

Apart from visual assessments of sample similarity, we conduct statistical validations to ascertain the reliability of the generated samples. This is achieved by calculating the

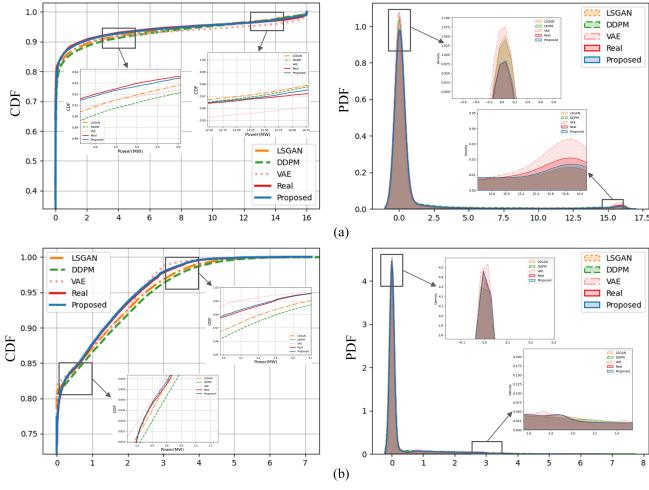


Fig. 7. The comparison of CDF and PDF obtained from different methods. (a) is wind power and (b) is solar power.

cumulative distribution function (CDF) and probability density function (PDF) of a substantial number of samples, where the PDF is obtained by Gaussian kernel density estimation. As illustrated in Fig. 7, we have zoomed in on local regions of the CDF and PDF curves for a clearer comparison of each method. It can be seen that the CDF and PDF of the proposed method are more fitting to the real curves compared to LSGANs, VAE, and DDPM. This indicates that our method not only learns the shape details of the scenario curves but also fully captures the distribution patterns of real data, validating that the generated scenarios and real scenarios share similar statistical properties.

#### D. Quality Results

Evaluating and comparing generative models poses a challenging task. Currently, there exists no consensus or standard guidelines on which metrics are ideal for assessing the capabilities and limitations of models. Remarkable results may be achieved for a specific metric, yet the same model might not perform equally well on other criteria. Hence, the incorporation of multiple complementary metrics to evaluate a model is considered a good practice. This study uses five complementary quality metrics: the root mean square error

TABLE I  
EVALUATION METRICS RESULTS

	Proposed	DDPM	VAE	LSGAN
Wind	RMSE <b>0.5184</b>	1.1210	0.8240	0.9100
	MAE <b>0.3857</b>	0.8236	0.6318	0.6774
	MMD <b>0.2374</b>	0.5342	0.3089	0.3937
	CPRS <b>2.2016</b>	2.2960	2.2269	2.2928
	ES <b>2.1907</b>	4.3107	3.7771	4.2895
	VS <b>23.6096</b>	92.2627	80.9852	85.7794
Solar	RMSE <b>0.1395</b>	0.9531	0.1824	0.2465
	MAE <b>0.0838</b>	0.5667	0.1101	0.1536
	MMD <b>0.2819</b>	3.2079	0.3660	0.3345
	CPRS <b>0.6437</b>	0.6995	0.6731	0.6729
	ES <b>0.2352</b>	1.9264	0.3550	0.3509
	VS <b>1.0492</b>	20.5207	2.4495	3.6483

(RMSE), mean absolute error (MAE), maximum mean discrepancy (MMD), continuous ranked probability score(CRPS), energy score (ES) and the variogram score (VS). All of them are negatively oriented.

1) *CRPS*: CRPS extends the mean absolute error of deterministic predictions to the case of probabilistic predictions and is one of the most widely used accuracy metrics. For a given day  $d$  of the observed set, the CRPS per marginal  $k = 1 \dots T$  is

$$\text{CRPS}_{d,k} = \frac{1}{M_g} \sum_{i=1}^{M_g} \left| \hat{x}_{d,k}^j - x_{d,k} \right| - \frac{1}{2M_g^2} \sum_{i,j=1}^{M_g} \left| \hat{x}_{d,k}^i - \hat{x}_{d,k}^j \right| \quad (21)$$

2) *MMD*: MMD quantify the difference between two distributions, expressed as follows:

$$\text{MMD} = \frac{1}{M_g^2} \left( \sum_{i=1}^{M_g} \sum_{j=1}^{M_g} (k(x_i, x_j) + k(\hat{x}_i, \hat{x}_j) - 2 \cdot k(x_i, \hat{x}_j)) \right) \quad (22)$$

where  $k(\cdot)$  denotes the Gaussian kernel function.

3) *ES*: The ES is a commonly employed metric for assessing a finite number of scenarios that model a distribution. For a given day  $d$  of the observed set, the ES is computed as:

$$\text{ES} = \frac{1}{D_o} \sum_{d \in D_o} \left( \frac{1}{M} \sum_{i=1}^M \left\| \hat{x}_d^i - x_d^i \right\| - \frac{1}{2M^2} \sum_{i,j=1}^M \left\| \hat{x}_d^i - \hat{x}_d^j \right\| \right) \quad (23)$$

4) *VS*: In contrast to the ES, the VS is sensitive to the mean, variance, and correlation of incorrect scenarios, making it capable of distinguishing the correlation structure significantly. It is defined as follows:

$$\text{VS}_d = \sum_{k,k'}^L \omega_{kk'} \left( |x_{d,k} - x_{d,k'}|^\gamma - \frac{1}{M} \sum_{i=1}^M \left| \hat{x}_{d,k}^i - \hat{x}_{d,k'}^i \right|^\gamma \right)^2 \quad (24)$$

where  $\omega$  is a non-negative weight. We employ  $\omega_{kk'} = 1$  and  $\gamma = 0.5$  in this study. The scenario time resolution used for VS calculation is 1 hour.

Table I indicates that, compared to the benchmarks, our proposed method achieves the best performance across all metrics. Conversely, the outcomes of the three baseline tests exhibit relative inadequacy, with a minimum 30% and 32% increase in the two error metrics (RMSE and MAE), respectively. Concerning other distribution-level metrics (MMD, ES, VS, CRPS), our proposed method registers the minimum values, underscoring its proximity to the distribution of real scenarios. Moreover, compared to the original DDPM, the improvements made by our method are undoubtedly significant. All qualitative comparative results substantiate the conclusions drawn in the preceding sections. Consequently, our proposed method achieves state-of-the-art accuracy and reliability in the scenario generation.

#### E. Conditional Scenario Generation

Given the consistency in the generation process for both wind power and solar scenarios, the former exhibits stronger fluctuations, rendering it a more representative case. Consequently, this section predominantly employs the generation of wind power scenarios as an exemplar for analysis.

##### 1) Specific Scenario Generation:

In the course of scenario analysis, it is necessary to generate scenario data under certain special conditions. For instance,

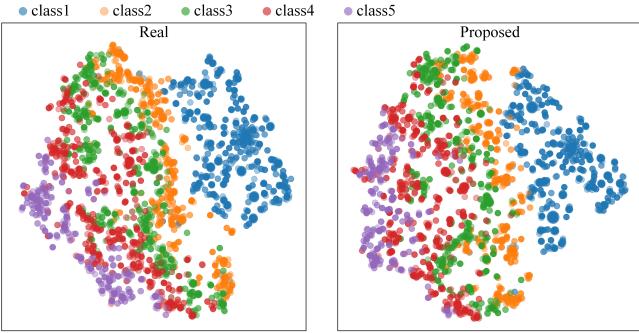


Fig. 8. Two-dimensional visualization of synthesis performance comparison.

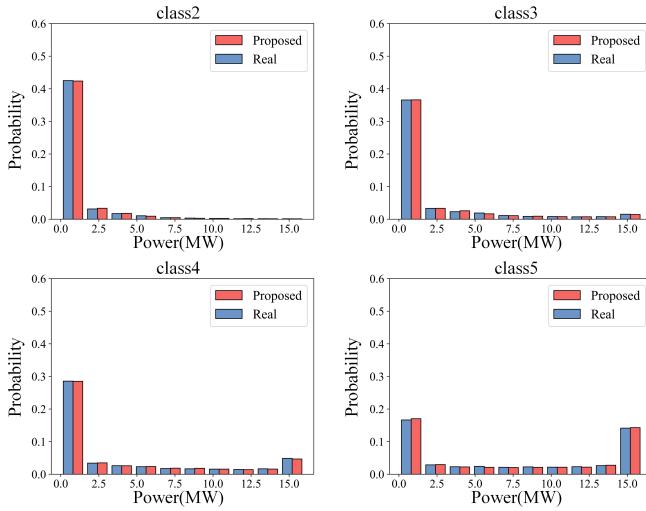


Fig. 9. Comparison of real and synthesized scenarios marginal distribution under different classes.

wind power data under particular weather conditions. To accomplish this, these conditions can be assigned as labels to the training set. Then, the conditional diffusion model, as elucidated earlier, can be employed to derive the generative model. The resultant scenario data will exhibit statistical characteristics akin to those encapsulated in the labeled training set. In this work, we categorize wind power scenarios into 5 classes based on mean value  $v(x)$ :  $v(x) < 0.5$ ,  $v(x) < 2$ ,  $v(x) < 3.5$ ,  $v(x) < 6.5$  and  $v(x) > 6.5$ , denoting different daily wind strength. The class information is encoded into one-hot vectors and embedded into the conditional implicit diffusion model for generating scenarios corresponding to the specified label, as outlined in (15).

To intuitively illustrate the performance of the proposed method, we present a visualization of the generated multi-class scenarios alongside real scenarios in a two-dimensional space, as shown in Fig. 8. Specifically, we treat the 288 outputs for each day as features, then use t-SNE to reduce the feature dimensions for better visualization, and color-code the scenarios based on different categories. It can be seen that the arrangement of generated scenarios resembles that of real scenarios. In particular, the class 1 exhibits a large number of samples clustered around the center. Through numerical analysis, this pattern is attributed to the prevalence

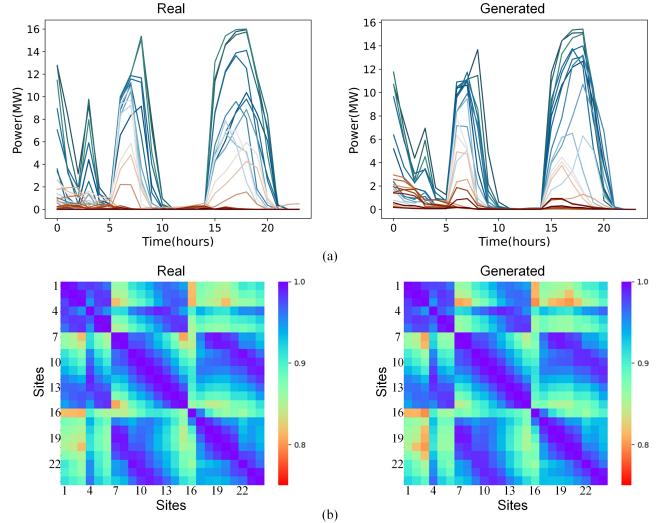


Fig. 10. The spatial correlation coefficients matrix colormap for a set of real and synthesized multisites power output.

of windless days characterized by sustained output close to 0. This observation demonstrates the capability of the proposed method to generate scenarios with a distribution pattern in the sample space closely resembling that of real scenarios. We evaluate the samples generated under these conditions by examining the marginal distributions for each category within the range of 0 to 16 MW, divided into 10 intervals, as shown in Fig. 9. The data produced by the proposed method exhibits a commendable fit to the probability distribution of each category in the test set. For instance, on days with light wind, the probability of the site's output being less than 2 MW surpasses 50%, and as the wind speed increases, the center of the probability distribution gradually shifts to the right. This alignment with real data patterns attests that the generated samples by the proposed method adhere to the same marginal distribution as the corresponding validation samples.

## 2) Multisites Scenario Generation:

In large wind farms, there are multiple power generation sites that are geographically close. When generating scenarios, in addition to considering the probability distribution of data similar to the previous context, it is also necessary to take into account the correlation between these generation sites. In this study, 24 power generation sites were selected, and the sampling time interval for each site was increased from 5 minutes to 1 hour. Thus, each sample is a matrix of size  $24 \times 24$ . To analyze the spatial relationships between multiple sites, we calculate the Pearson correlation coefficient between each pair of sites for both real and generated data. The Pearson correlation coefficient  $\rho$  is:

$$\rho = \frac{\sum_{i=1}^M (x_i - \bar{x}_i)(x_j - \bar{x}_j)}{\sqrt{\sum_{i=1}^M (x_i - \bar{x}_i)^2 \sum_{i=1}^M (x_j - \bar{x}_j)^2}} \quad (26)$$

Fig. 10(a) illustrates a collection of real and generated scenarios for a group of 24 power generation stations. The visual representation of correlation coefficients among the power stations is presented in Fig. 10(b). The results indicate

that the generated multi-site scenarios are consistent with real scenarios in terms of correlation patterns. The proposed method is capable of simultaneously capturing the complex temporal and spatial distribution characteristics of renewable energy generation, contributing to the construction of a reliable and effective scenario set.

## V. CONCLUSION

This paper proposes a two-stage generative architecture based on diffusion models for renewable scenario generation. In the first stage, the representation learning of time series through the Encoder-Decoder structure achieves the encoding of renewable scenarios in the latent space. This design imparts the specific inductive bias of renewable scenario for subsequent diffusion model learning, thereby facilitating the assimilation of temporal information. In the second stage, we introduce a conditional implicit diffusion model to learn the features of renewable scenarios in the latent space, and the ultimate generated scenarios were then obtained through the decoder of the first stage.

Case studies indicate that, compared to other advanced deep generative models, the proposed method stands out as the most competitive. Unlike other models that learn from a graph feature perspective essentially neglect the temporal characteristic of renewable scenarios, resulting in inaccurate descriptions of uncertainty. The proposed method, by learning the intricate time-dependent relationships of renewable resource output, can generate high-quality scenarios with full diversity. These scenarios effectively capture the underlying distribution of real data rather than simply simulating. We validated the effectiveness of each method in generating scenarios through a series of visual and statistical approaches. The diverse quality evaluation metrics employed demonstrate that the proposed method has achieved state-of-the-art performance.

In future research, we propose to incorporate this study for the planning and optimal operation of an electrical system with renewables.

## REFERENCES

- [1] M. Xie, J. Xiong, S. Ke, and M. Liu, “Two-Stage Compensation Algorithm for Dynamic Economic Dispatching Considering Copula Correlation of Multiwind Farms Generation,” *IEEE Transactions on Sustainable Energy*, vol. 8, no. 2, pp. 763–771, Apr. 2017.
- [2] J. Wang, M. Shahidehpour, and Z. Li, “Security-Constrained Unit Commitment With Volatile Wind Power Generation,” *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1319–1327, Aug. 2008.
- [3] E. K. Hart and M. Z. Jacobson, “A Monte Carlo approach to generator portfolio planning and carbon emissions assessments of systems with large penetrations of variable renewables,” *Renewable Energy*, vol. 36, no. 8, pp. 2278–2286, Aug. 2011.
- [4] H. Yu, C. Y. Chung, K. P. Wong, H. W. Lee, and J. H. Zhang, “Probabilistic Load Flow Evaluation With Hybrid Latin Hypercube Sampling and Cholesky Decomposition,” *IEEE Transactions on Power Systems*, vol. 24, no. 2, pp. 661–667, May 2009.
- [5] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, “From probabilistic forecasts to statistical scenarios of short-term wind power production,” *Wind Energy*, vol. 12, no. 1, pp. 51–62, 2009.
- [6] Y. Chen, J. Wen, and S. Cheng, “Probabilistic Load Flow Method Based on Nataf Transformation and Latin Hypercube Sampling,” *IEEE Transactions on Sustainable Energy*, vol. 4, no. 2, pp. 294–301, Apr. 2013.
- [7] S. Camal, F. Teng, A. Michiorri, G. Kariniotakis, and L. Badesa, “Scenario generation of aggregated Wind, Photovoltaics and small Hydro production for power systems applications,” *Applied Energy*, vol. 242, pp. 1396–1406, May 2019.
- [8] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, “Probabilistic Forecasting of Wind Power Generation Using Extreme Learning Machine,” *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1033–1044, May 2014.
- [9] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, and L. Carin, “Variational autoencoder for deep learning of images, labels and captions,” in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS’16. Red Hook, NY, USA: Curran Associates Inc., Dec. 2016, pp. 2360–2368.
- [10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial networks,” *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020.
- [11] Y. Chen, Y. Wang, D. Kirschen, and B. Zhang, “Model-Free Renewable Scenario Generation Using Generative Adversarial Networks,” in *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Aug. 2019, pp. 1–1.
- [12] Y. Li, J. Li, and Y. Wang, “Privacy-Preserving Spatiotemporal Scenario Generation of Renewable Energies: A Federated Deep Generative Learning Approach,” *IEEE Transactions on Industrial Informatics*, vol. 18, no. 4, pp. 2310–2320, Apr. 2022.
- [13] C. Jiang, Y. Mao, Y. Chai, and M. Yu, “Day-ahead renewable scenario forecasts based on generative adversarial networks,” *International Journal of Energy Research*, vol. 45, no. 5, pp. 7572–7587, 2021.
- [14] H. Zhang, W. Hua, R. Yub, M. Tangb, and L. Dingc, “Optimized Operation of cascade reservoirs Considering Complementary Characteristics between wind and photovoltaic based on variational auto-encoder,” in *MATEC Web of Conferences*, X. Lei, S. Cai, Y. Yu, and H. Li, Eds., vol. 246, 2018, p. 01077.
- [15] Z. Zheng, L. Yang, and Z. Zhang, “Conditional Variational Autoencoder Informed Probabilistic Wind Power Curve Modeling,” *IEEE Transactions on Sustainable Energy*, vol. 14, no. 4, pp. 2445–2460, Oct. 2023.
- [16] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS ’20. Red Hook, NY, USA: Curran Associates Inc., Dec. 2020, pp. 6840–6851.
- [17] P. Dhariwal and A. Nichol, “Diffusion models beat gans on image synthesis,” *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.
- [18] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-Resolution Image Synthesis with Latent Diffusion Models,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 674–10 685, Jun. 2022.
- [19] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi, “Unsupervised scalable representation learning for multivariate time series,” in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., Dec. 2019, no. 418, pp. 4650–4661.
- [20] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff, “A Transformer-based Framework for Multivariate Time Series Representation Learning,” in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, ser. KDD ’21. New York, NY, USA: Association for Computing Machinery, Aug. 2021, pp. 2114–2124.
- [21] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, “Masked Autoencoders Are Scalable Vision Learners,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15 979–15 988, Jun. 2022.
- [22] R. Al-Rfou, D. Choe, N. Constant, M. Guo, and L. Jones, “Character-Level Language Modeling with Deeper Self-Attention,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, pp. 3159–3166, Jul. 2019.
- [23] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.
- [24] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, pp. 11 966–11 976.
- [25] C. Draxl, A. Clifton, B.-M. Hodge, and J. McCaa, “The Wind Integration National Dataset (WIND) Toolkit,” *Applied Energy*, vol. 151, pp. 355–366, Aug. 2015.