Active Distribution System Synthesis via Unbalanced Graph Generative Adversarial Network

Rong Yan, Graduate Student Member, IEEE, Yuxuan Yuan, Graduate Student Member, IEEE, Zhaoyu Wang, Senior Member, IEEE, Guangchao Geng, Senior Member, IEEE, and Quanyuan Jiang, Senior Member, IEEE

Abstract—Real active distribution networks with associated smart meter (SM) data are critical for power researchers. However, it is practically difficult for researchers to obtain such comprehensive datasets from utilities due to privacy concerns. To bridge this gap, an implicit generative model with Wasserstein GAN objectives, namely unbalanced graph generative adversarial network (UG-GAN), is designed to generate synthetic three-phase unbalanced active distribution system connectivity. The basic idea is to learn the distribution of random walks both over a real-world system and across each phase of line segments, capturing the underlying local properties of an individual real-world distribution network and generating specific synthetic networks accordingly. Then, to create a comprehensive synthetic test case, a network correction and extension process is proposed to obtain time-series nodal demands and standard distribution grid components with realistic parameters, including distributed energy resources (DERs) and capacitor banks. A Midwest distribution system with 1-year SM data has been utilized to validate the performance of our method. Case studies with several power applications demonstrate that synthetic active networks generated by the proposed framework can mimic almost all features of real-world networks while avoiding the disclosure of confidential information.

Index Terms—Graph generative adversarial network, network synthesis, random walk, unbalanced active distribution system.

NOMENCLATURE

A. General Abbreviations
DER Distributed energy resource.
D Discriminator neural network.

B. Parameters and Functions of Wasserstein GAN and UG-GAN

\[ P \] Adjacency matrix.
\[ c \] Clipping parameter.
\[ \mathbb{E}(\cdot) \] Expectation function.
\[ f_{\theta} \] Kernel Sequential neutral network.
\[ g_{\theta}(\cdot) \] Initialization parametric function.
\[ m \] Batch size.
\[ n_{iter} \] Number of discriminator iterations per generator iteration.
\[ p_{\text{real}} \] Possibility of real of input data \( x \).
\[ P_{x} \] Distribution of the real samples \( x \).
\[ P_{z} \] Distribution of the noise signal \( z \).
\[ Q \] Scoring matrix.
\[ T \] Number of random walk step.
\[ V(\cdot) \] Value function.
\[ v_{i} \] Random walk vector of the \( i \)-th step.
\[ x \] Real data.
\[ x_{\text{fake}} \] Generated artificial data.
\[ z \] Noise signal data.
\[ \alpha \] Learning rate.
\[ \theta_{g} \] Learning parameter of \( G \).
\[ \theta_{d} \] Learning parameter of \( D \).
\[ \theta_{g0} \] Initial learning parameter of \( G \).
\[ \theta_{d0} \] Initial learning parameter of \( D \).
\[ \mathcal{N}(\cdot) \] Multivariate Gaussian distribution.
\[ \text{Cat}(\cdot) \] Category function.
\[ \sigma(\cdot) \] Sigmoid function.

C. Parameters and Variables of Network Correction and Extension

\[ E \] A \( N_{e} \times 2 \) matrix indicating from and to node indexes of the \( i \)-th edge.
\[ h_{j} \] Kernel bandwidth for the \( j \)-th variable.

0885-8950 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
I. INTRODUCTION

POWER researchers seek to understand how real-world systems work and how real-world systems can work better. Therefore, knowledge of real-world systems, including topologies, locations and parameters of electrical components, and customer consumption behaviors, is essential to their works. In practice, most utilities are hesitant to share their systems with the public due to data privacy concerns. One common solution is to use IEEE test feeders modified on real distribution systems for model validation and demonstration. However, the main challenge is that the number of standard test feeders is very limited. Hence, synthetic test systems have been developed as alternatives to represent various real networks flexibly. Basically, synthetic networks should exhibit the critical topological and electrical characteristics of real-world networks with user’s behaviors, but they are entirely fictitious, and users cannot extract any real-world network information from synthetic networks by reverse engineering.

Previous works mainly focus on generating synthetic transmission networks, which can be classified into two categories: statistics-based [1], [2], [3], [4], [5], [6], [7] and machine learning-based [8], [9] methods. The statistics-based methods performed extensive data analytics on a large amount of real-world power grid data to manually quantify the key properties, both topological and electrical, of network, such as node degree, load distribution, and parameters of grid components. Based on these properties, synthetic networks can be generated using graph theory and grid planning simulations. Specifically, reference [1] and [2] present the methods to get a set of statistical metrics by analyzing empirical probability density function of transmission network electrical parameters. These metrics are significantly important both in the network creation and validation stage. With these properties, reference [3] presents a systematic synthetic power grid creation method, which can be seen as a general solution for realizing this task. Latter researches based on statistics-based methods mostly focus on customizing a more realistic power grid for a specific study field, including testing the influence of geomagnetic disturbance [4], economic criteria [5], and communication and control network [6] on real-world grid. Instead of statistics-based methods, machine learning-based methods are also introduced in this field by predicting the connectivity of the grid directly according to the distribution of the training networks properties. In [8], [9], network imitating methods were proposed to generate grids with similar properties to the given networks. Both methods are based on the small-world assumption [7], which has been proven by most scholars in field of transmission systems, i.e., a type of system in which most buses are not directly connected, but the neighbor buses of any given bus are likely to be directly connected and most buses can be reached from other buses by a small number of buses. Recently, some works start rethinking whether small world is an accurate model for transmission grids [10] by a small group of researches, and attempt to design new techniques, e.g., methods based on system planning sensitivities [10], to produce a more realistic synthetic grid. It is worth noting that the network created by these methods is able to basically meet requirements of actual applications.

Compared to transmission network synthesis, research on active distribution network synthesis is still at a preliminary stage. Some studies [11], [12] have extended the transmission-level statistics-based methods to distribution grids by introducing several indices representing topological properties of distribution networks. However, distribution networks definitely no longer satisfy the small-world assumption, which impacts the performance of these methods. Moreover, the regional nature of the distribution systems is greatly ignored in these works. For example, urban and rural distribution networks have
different properties in both topology properties and power flow distribution. Based on our observations of real-world data, the characteristics of distribution networks depend heavily on street layout, space availability, customer density, and even utilities’ own preferences. Such observations indicate that each distribution network has a great deal of specificity. Consequently, some researchers have used local geographical and social statistic data, such as google maps and Census data, to simulate the system planning process for distribution network synthesis [13], [14], [15]. In fact, it is an alternative way since existing works cannot extract all key information for a specific distribution network. Although the best one among them [15] is able to create a realistic large-scale network, it still largely depends on the expert experience of planning and huge amount detailed local geographical, social statistic, and electrical data. Others try to develop representative synthetic test feeders directly from real systems using hierarchical clustering analysis manually. For example, in [16], 24 networks were presented from 575 real distribution feeders, which characterize distribution systems in different regions of the U.S. Apart from this, authors of [15] extend their works to synthetic combined transmission and distribution networks synthesis task [17] and do validate in an electrical manner, trying to build a more realistic synthetic grid with larger scale.

While previous works provide valuable insights, some challenges remain unanswered or only partly covered in this area and can be summarized as follows: (1) Existing statistics-based works [11], [12], [16] normally rely on a large amount of real-world data to extract statistical grid properties. Besides, other planning simulation-based works [13], [14], [15] also require a mass of detailed local geographical and social statistic data. Such a strategy not only poses a challenge for data acquisition and privacy but also raises concerns about the generalizability of the methods. When researchers generate synthetic grids for model development and validation, they need to first extend their datasets by collecting massive real-world data, which is very expensive. (2) Previous methods [11], [12], [16] ignore the significant diversity of distribution systems due to different geographic environments and grid infrastructures. For example, urban distribution systems show very different topological and electrical factors than rural systems. (3) For all existing works [11], [12], [13], [14], [15], [16], it is not well studied how to create realistic unbalanced active distribution systems, which is exactly one of the key features in practical distribution grids. (4) The previous works [11], [12], [13], [14], [15], [16] pay more attention to the grid connectivity generation, rather than the interaction between topology, loads, and electrical components. Besides, they do not provide time-series nodal load data reflecting the users’ behavior, and it limits the scope of application scenarios.

To address these challenges, we propose a data-driven framework that uses limited real-world data to generate a comprehensive active distribution test feeder. Here, “comprehensiveness” means that it contains time-series nodal demands and standard distribution grid components with realistic parameters. To achieve this, first, an unbalanced graph generative adversarial network (UG-GAN) method is designed to produce synthetic node connectivity. Specifically, we formulate the network synthesis problem as learning the distribution of biased random walks\(^1\) both over a single real-world network and across each phase of line segments. Also, we modify the standard GAN architecture to handle the discrete nature of the network data. When the UG-GAN is trained, synthetic node connectivity can be obtained by repeatedly generating random walks. Then, based on this synthetic topology, we utilize a non-parametric uncertainty quantification method known as kernel density estimation (KDE) to generate time-series load consumption data for each node. Finally, an optimization-based component placement model is proposed to determine the locations and parameters of various grid components. The goal of this optimization model is to consider the interactions between topology, loads, and electrical components in distribution systems. Unlike previous works that validate synthetic networks only in a statistical manner, our method is tested in a power system manner. More precisely, the generated test case is applied in three different power applications. Case studies demonstrate that our synthetic active distribution system has similar electrical properties and significantly different external characteristics to the input network, which respects the data autonomy of the data owner.

By using the proposed method, researchers and engineers can mimic one particular real-world network and generate a set of comprehensive testing cases with similar proprieties. As a result, data providers will no longer have any concerns about making desensitized data publicly available in response to requests from industry and academia. In other words, data providers will be more willing to share synthetic systems generated using our methods rather than sharing their real-world systems directly. Also, although this work is fine-tuned on our dataset to optimize the values of the model hyperparameters, the methodology is general and can be applied to any other radial distribution systems for system synthesis after retraining/fine-tuning to capture the unseen distribution of random walks. This is true for any data-driven solution. Furthermore, our model has good scalability. Specifically, the proposed method operates on random walks and only considers the non-zero entries of the adjacency matrix instead of generating the entire adjacency matrix, which requires computation and memory as a quadratic function of the number of nodes. Such a strategy efficiently exploits the sparsity of real-world active distribution systems to enhance scalability. Meanwhile, given that system synthesis is a purely offline analysis, the computation burden of the proposed UG-GAN does not directly impact the performance of our method.

In summary, the innovative contributions of this paper can be summarized as follows:

- The proposed model follows an adversarial generative framework that allows the use of limited real-world data (at least all key information of one real distribution network) to capture the specificity of individual three-phase unbalanced active distribution systems while maintaining confidential information.

\(^1\)Biased random walk is a randomly sampled path that consists of a succession of random steps on a given graph. Unlike in a pure random walk, the probabilities of the potential new states are unequal due to the topology of the given graph.
The proposed method can generate a comprehensive distribution test case that contains three-phase unbalanced topology, more detailed time-series nodal load data, and more types of standard grid components in order for broader application scenarios.

- Topological and electrical indices, together with three power applications, are introduced to verify that the generated active distribution systems are realistic.

II. UG-GAN Based Unbalanced Distribution Network Synthesis

In this section, a UG-GAN is proposed to generate unbalanced distribution networks by using a single network. To help the reader understand our model, we first review Wasserstein GAN, including basic idea, formulation, and training process, then describe the details of our UG-GAN.

A. Wasserstein Generative Adversarial Network

Wasserstein Generative Adversarial Network (Wasserstein GAN) is a novel GAN architecture [18] that improves the training stability and provides a loss function to describe the quality of the generated samples [19]. It is with the ability to learn the underlying distribution $P_x$ of the real samples $x$, by finding out a mapping relationship from a known sampled distribution $P_z$ (such as Gaussian distribution) to an artificial sample that follows $P_x$. This function can be realized by two deep neural networks: a generator ($G$) and a discriminator ($D$). The interaction between these two networks is formulated as a game-theoretic two-player nested min-max optimization $V(G,D)$. For concreteness, they are described as follows:

1) Generator Neural Network ($G$): $G$ defines an end-to-end neural network trained to transform a noise signal $z$ to the generated artificial data $x_{fake}$:

$$x_{fake} = G(z; \theta_g) \tag{1}$$

where $\theta_g$ denotes the learning parameter of $G$. $z$ is the noise signal with a known probability density distribution. In this work, we choose the noise with multivariate Gaussian distribution, shown as:

$$z = \mathcal{N}(0, \sigma) \sim P_z \tag{2}$$

General speaking, any machine learning model (like artificial neural network, convolutional neural network, long-short term memory or ensemble model) can be embedded into $G$, according to the specific requirements of different tasks, so that the generated artificial data satisfies the distribution of real data $P_x$.

2) Discriminator Neural Network ($D$): $D$ is trained to maximize the probability of assigning the correct labels to both real examples and artificially generated samples from $G$. It outputs a single scalar $p_{real}$ ranging from 0 to 1, representing the possibility that the input data $x$ is from the real dataset rather than generated artificially by $G$. The network with learning parameter $\theta_d$ is listed as:

$$p_{real} = D(x; \theta_d) \tag{3}$$

3) Value Function $V(G,D)$ and its Training Process: As mentioned above, $G$ can be regarded as a model to learn a mapping relationship $G(z; \theta_g)$ from noise with known distribution to real data space. Thus, the training object is obviously to make the generated artificial data as realistic as the real ones from the perspective of $D$, by maximizing the expectation of generated artificial data $E_z[D(G(Z))]$. Meanwhile, $D(x; \theta_d)$ is defined as another neural network to distinguish real data from artificial ones, with an objection maximizing the expectation difference between real data $E_x[D(x)]$ and generated data $E_z[D(G(Z))]$. Therefore, a suitable value function $V(G,D)$ for these two interconnected networks is the key idea of GAN, by modeling as a game-theoretic two-player minimax optimization problem. Noted that this value function is specially designed in Wasserstein GAN to improve the stability of the training process on the basis of traditional GAN, shown as:

$$\min_G \max_D V(G,D) = E_x[D(x)] - E_z[D(G(z))] \tag{4}$$

Two networks are trained simultaneously via an adversarial process using the above value function, until reaching a unique global optimum. More details can be found in [18].

B. UG-GAN for Unbalanced Distribution Network Synthesis

In power systems, despite novel generative models have great success in dealing with real-valued data, such as wind and outage scenario generation [20], [21], adapting generative models to handle discrete network data is still an open problem. Therefore, in this paper, we propose a new algorithm, UG-GAN, to address the needs of our task. The main idea is illustrated in Fig. 1. Basically, the proposed model captures graphical features of a network by learning the distribution of biased random walks over the network. As demonstrated concretely in [22], random walk is a stochastic sampled path that consists of a succession of random steps on a given network. A distribution grid can be decomposed into a set of random walks that contain both local and global graphical features. Generally speaking, similar networks share similar distribution of sampled random walks, as long as the sampled random walks are sufficient. Following this theory, random walk sampling is employed to convert network data to sequential data.

1) Random Walk Sampling and its Encoding Scheme: To indicate the process of random walk sampling and encoding
is determined for all possible nodes to be sampled at the next time step (5b)
\[ C \in \{ \text{can be A rows, where} \}
\]
\[ \begin{array}{c}
\sigma \\
\end{array} \]
\[ \begin{array}{c}
\text{and} \\
\end{array} \]
\[ \begin{array}{c}
\text{f} \\
\end{array} \]
\[ \begin{array}{c}
=0 \\
\end{array} \]
\[ \begin{array}{c}
\text{a} \\
\end{array} \]
\[ \begin{array}{c}
\text{two} \\
\end{array} \]
\[ \begin{array}{c}
\text{of} \\
\end{array} \]
\[ \begin{array}{c}
\text{length} \\
\end{array} \]
\[ \begin{array}{c}
\text{columns} \\
\end{array} \]
\[ \begin{array}{c}
\text{and} \\
\end{array} \]
\[ \begin{array}{c}
\text{of} \\
\end{array} \]
\[ \begin{array}{c}
\text{length} \\
\end{array} \]
\[ \begin{array}{c}
\text{columns} \\
\end{array} \]
\[ \begin{array}{c}
\text{and} \\
\end{array} \]
\[ \begin{array}{c}
\text{four} \\
\end{array} \]
\[ \begin{array}{c}
\text{rows}, \\
\end{array} \]
\[ \begin{array}{c}
\text{denoting} \\
\end{array} \]
\[ \begin{array}{c}
\text{the} \\
\end{array} \]
\[ \begin{array}{c}
\text{phase} \\
\end{array} \]
\[ \begin{array}{c}
\text{information} \\
\end{array} \]
\[ \begin{array}{c}
\text{of} \\
\end{array} \]
\[ \begin{array}{c}
\text{input} \\
\end{array} \]
\[ \begin{array}{c}
\text{network.} \\
\end{array} \]
\[ \text{As} \\
\]
\[ \text{shown} \\
\]
\[ \text{in} \\
\]
\[ \text{Fig.} \\
\]
\[ \text{3,} \\
\]
\[ \text{the} \\
\]
\[ \text{input} \\
\]
\[ \text{of} \\
\]
\[ \text{distribution} \\
\]
\[ \text{network,} \\
\]
\[ \text{defined} \\
\]
\[ \text{by} \\
\]
\[ \text{a} \\
\]
\[ \text{binary} \\
\]
\[ \text{adjacency} \\
\]
\[ \text{matrix}, \ A \in \{0, 1\}^{N_n \times N_n} \text{, we first sample a large number of random walks} \]
\[ \text{RW} := \{v_1, v_2, \ldots, v_T\} \text{ of length } T \text{ from } A \text{. Then, these random walks} \]
\[ \text{are used as the training set of } G, \text{ which can be formulated as follows:} \]
\[ (h_t, C_t, p_t) = f_\theta(h_{t-1}, C_{t-1}, v_{t-1}) \quad (5a) \]
\[ v_t \sim \text{Cat}(\sigma(p_t)) \quad (5b) \]
\[ (h_0, C_0) = g_\theta(z), \quad v_0 = 0 \quad (5c) \]
\[ \text{where } \sigma(\cdot) \text{ is the sigmoid function, } \text{Cat}(\cdot) \text{ is a category function, and } g_\theta(z) \text{ denotes a parametric function from the noise signal generated by the multivariate Gaussian distribution to initialize a sequential neutral network } f_\theta. \text{ In this work, a modified long short-term memory (LSTM) is utilized to represent } f_\theta. \text{ As shown in Fig. 4, for each time step } t, \text{ LSTM cell outputs two values: current state vector } h_t \text{ and } C_t, \text{ and discrete possibility vector } p_t \text{ for all possible nodes to be sampled at the next time step } t+1. \text{ Since sampling from a categorical distribution is the non-differentiable operation that impedes backpropagation, we have} \]
applied the Gumbel-Max trick to solve this problem [25]. After
relaxation, the exact node \( v_t \) of random walk can be sampled
according to \( p_t \) using (5b).

3) **Structure of Discriminator Neural Network in UG-GAN:**

\( D \) is based on the standard LSTM architecture to distinguish
sequential random walks generated by \( G \) from the ones sam-
pled from the real distribution network. Further, an input data
preprocessing and an output activation layer are added to \( D \).

More precisely, at each time step, the random walk vector \( v_t \)
encoded in a two-dimensional one-hot format is reshaped before
fed into LSTM as input. The output of the discriminator is a
scalar indicating the probability that the input random walk is
real.

4) **Training Algorithm of UG-GAN:** In this subsection, we
present the training algorithm of UG-GAN giving the reader
a clear picture of the training process. Frankly speaking, the
training process of UG-GAN follows the line of Wasserstein
GAN [18] with minor modifications, as it prevents mode collapse
and leads to more stable training. As shown in Algorithm 1, it re-
quires the original distribution network with several parameters
as input, and outputs the final parameters of \( G \) and \( D \).

After the training process, \( G \) can implicitly represent the un-
derlying distribution of biased random walks over the real-world
network and \( D \) cannot distinguish the true random walks from
the artificial random walks. The biased second-order random
walk sampling strategy described in [26] is utilized in \( G \). Based
on the random walks generated by \( G \), a scoring matrix \( Q \) is con-
structed, not only measuring the possibility of connectivity for
each node, but also providing phase information and conductor
configurations if connected.

### III. ACTIVE DISTRIBUTION NETWORK CORRECTION,
EXTENSION AND EVALUATION

When the graphical features of the real-world network are cap-
tured by the UG-GAN, an active distribution network correction
and extension framework is developed to provide a comprehen-
sive distribution test case, including realistic nodal load data and
standard grid components with detailed parameters.

#### A. Time-Series Load Data Synthesis

The basic idea of load data synthesis is to estimate the proba-
bility density of multiple load behaviors and then sample them
accordingly. However, considering the highly complex load
uncertainty, it is difficult to do utilizing traditional parametric
density estimation methods with Gaussian, beta, and GMM dis-
tribution model assumptions. This is because these methods rely
on model assumptions that may introduce significant modeling
bias in uncertainty quantification.

To address this challenge, a non-parametric method, known
as kernel density estimation (KDE), is employed to estimate the
probability density function (PDF) of different load behaviors,
and generate the time-series load data for each primary nodes by
sampling the estimated PDFs. For concreteness, the proposed
algorithm is summarized as three steps. The first step is to col-
clect the time-series load data of all types of users. Then, these
load data are classified using an unsupervised clustering algo-

\begin{algorithm}
\textbf{Algorithm 1: UG-GAN Training Algorithm.}
\begin{algorithmic}[1]
\Require
\begin{itemize}
  \item \( \alpha \), the learning rate.
  \item \( c \), the clipping parameter.
  \item \( m \), the batch size.
  \item \( n_d \), the number of iterations of the discriminator per
generator iteration.
  \item \( \theta_d \), initial discriminator’s parameters.
  \item \( \theta_g \), initial generator’s parameters.
\end{itemize}
\Ensure
\begin{itemize}
  \item \( \theta_d \), parameters of discriminator.
  \item \( \theta_g \), parameters of generator.
\end{itemize}
\end{algorithmic}
1: Sample a huge amount of random walks \( RW \) from the
input distribution network \( NET \), and encoding them
as the real input dataset \( x \).
2: while not converged do
3: for \( n_{iter} = 1, \ldots, n_d \) do
4: Sample \( \{x(i)\}_{i=1}^{m} \sim \mathbb{P}_x \) a batch from the real
data.
5: Sample \( \{z(i)\}_{i=1}^{m} \sim \mathbb{P}_z \) a batch of prior samples.
6: \( G_{\theta_d} \leftarrow \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} D(x(i); \theta_d) - \frac{1}{m} \sum_{i=1}^{m} D(G(z(i); \theta_g); \theta_d) \)
7: \( \theta_d \leftarrow \theta_d + \alpha \cdot \text{RMSProp}(\theta_d, G_{\theta_d}) \)
8: \( \theta_g \leftarrow \text{clip}(\theta_g, c, c) \)
9: end for
10: Sample \( \{z(i)\}_{i=1}^{m} \sim \mathbb{P}_z \) a batch of prior samples.
11: \( G_{\theta_g} \leftarrow -\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} D(G(z(i); \theta_g); \theta_d) \)
12: \( \theta_g \leftarrow \theta_g - \alpha \cdot \text{RMSProp}(\theta_g, G_{\theta_g}) \)
13: end while
\end{algorithm}

\end{algorithm}

2Electricity customers can be roughly divided into three main types with com-
pletely different consumption behaviors: residential, commercial and industrial
loads.
Joint framework of load assignment and topology correction is proposed, in order to assign all loads to the generated system while performing topology corrections. Specifically, this joint framework is cast as a Mixed Integer Quadratic Programming (MIQP) problem. Among them, 12 binary variables are defined to represent the connectivity between loads (including \(N_L\) single-phase and \(N_T\) three-phase loads) and the generated network with \(N_n\) nodes and \(N_e\) edges:

\[
\begin{align*}
    & u_{Lk}^A, u_{Lk}^B, u_{Lk}^C, u_{nj}^A, u_{nj}^B, u_{nj}^C, u_{nj}^A, \\
    & u_{nk}^A, u_{nk}^B, u_{nk}^C, u_{ei}^A, u_{ei}^B, u_{ei}^C \in \{0, 1\}
\end{align*}
\]  

The first three variables correspond to individual phase for \(k\)-th single-phase load connected to node \(M_{Lk}\), where \(k = 1, 2, 3, \ldots, N_L\). The fourth to sixth variables indicate individual phase for \(j\)-th three-phase load connected to node \(M_{Ij}\), where \(j = 1, 2, 3, \ldots, N_I\). The last six denote individual phase for node \(\zeta\) and edge \(i\), respectively.

First, optimization objective is formulated as follows to determine a final network according to matrix \(Q\):

\[
\text{Obj} = \sum_{i=1}^{N_n} ((Q_{i,1} - u_{ei}^A)^2 + (Q_{i,2} - u_{ei}^B)^2 + (Q_{i,3} - u_{ei}^C)^2)
\]

Second, several constraints are added to ensure consistency with grid physics. We will describe them one-by-one. For the \(k\)-th single-phase load, it can merely be assigned to a specific phase of the network. Thus, the constraints corresponding to the binary variables of each phase can be written as:

\[
u_{Lk}^A + u_{Lk}^B + u_{Lk}^C = 1
\]

In a similar manner, for the \(j\)-th three-phase load which is connected to the \(\zeta\)-th node, the constraints of phase-A binary variables can be written as:

\[
u_{nj}^A = 1, \quad u_{nj}^A = 1 \quad j \in \zeta
\]

For all customers connected to node \(M_{Lk}\) (\(k \in \zeta\)), the binary variables associated with the load and node satisfy boolean logical relationship “or”. We use phase-A as an example to explain this: \(u_{nk}^A\) will be 1 when a single-phase load connects to phase-A of this node or a three-phase load connects to this node, otherwise it will be 0. In this work, we convert this boolean operation to a set of constraints as follows:

\[
u_{nk}^A \geq u_{Lk}^A
\]

\[
u_{nk}^A \leq \sum_k u_{Lk}^A, \quad k \in \zeta
\]

Considering that the vast majority of distribution networks in normal operation are tree-like structures [12], the upstream and downstream edges and nodes in the generated topology should meet several rules. Obviously, when the upstream edge is a three-phase branch, the downstream one can be either a single or three-phase branch. In contrast, the downstream one can merely be a single-phase branch when the upstream edge is a single-phase branch. Meanwhile, the phase information of the downstream node should be aligned with that of upstream edges. These two rules are formulated as a set of constraints described in (13):

\[
u_{nk}^A \geq u_{nk}^A, \quad u_{ek}^A = u_{nk}^A
\]
where \( E \) denotes a \( N_e \times 2 \) matrix. In the \( i \)-th row, first and second column elements, \( E_{1i} \) and \( E_{2i} \) (\( E_{1i} < E_{2i} \)), are the from and to indexes of the \( i \)-th edge, \( i = 1, 2, 3, \ldots, N_e \). Further, a constraint is added to the model for avoiding overloads in the generated synthetic network:

\[
P_{E_{1i}E_{2i}} \leq P_{E_{1i}E_{2i}} \leq P_{E_{1i}E_{2i}}
\]

where, \( P_{E_{1i}E_{2i}} \) indicates the active power of \( i \)-th transmission line connecting node \( E_{1i} \) and \( E_{2i} \). \( P_{E_{1i}E_{2i}} \) and \( P_{E_{1i}E_{2i}} \) are the upper and lower bound of the active power of the certain line.

Finally, the following equations are added as constraints on the model in order to prevent unreasonable three-phase imbalance ratios in the synthetic network:

\[
P^A = \sum_j \frac{1}{3} u_{ij}^A P_{ij} + \sum_k u_{ik}^A P_{ik}
\]

\[
\Delta_{\min} \leq \Delta \leq \Delta_{\max}
\]

where \( P_{ij} \) and \( P_{ik} \) are the \( j \)-th three-phase and \( k \)-th single-phase active power load. \( \Delta_{\min} \) and \( \Delta_{\max} \) are the user-defined thresholds, near the imbalance ratios of original real-world unbalanced distribution systems.

C. Extension of Network With Grid Components

The proposed UG-GAN with the network correction process can generate a synthetic active distribution network with the related nodal consumption data. However, without standard grid components, the synthetic distribution system cannot be treated as a comprehensive test case. Thus, in this work, by imitating the real planning process, a Mixed Integer Second-order Cone Programming (MISCP) problem is formulated to place several grid components, including capacitor banks and distributed energy resources (DER), on the basis of the synthetic network. The objective function is written to minimize the power losses as follows:

\[
\min \sum_{(i,j)\in E} r_{ij} l_{ij}
\]

where \( r_{ij} \) denotes the resistance of line \( i - j \), \( l_{ij} = |I_{ij}|^2 \), i.e. the square of current, and \( v(i,j) \in E \). Obviously, reducing network losses is not the only factor to be considered in grid component planning. Some components are directly invested by customers with the goal of local economic optimization. Therefore, the objective function described above can be modified according to the actual needs of the generated synthetic networks. One point to note is that the modified function must still be a linear function of \( l_{ij} \) and \( v_{ij} \) to ensure the solvability of the formulated MISCP optimization problem.

Further, this optimization problem should be subject to multiple constraints to force the installed components to be realistic. In general, the constraints of this optimization problem can be divided into two parts. The first part shown in (18) restricts the location and capacity of each grid component. Among them, the first two inequality constraints restrict the active and reactive power injections of each grid component to be equipped. The third one describes the overall limits of active power, determining the possibility of power flow reversal. The last constraint refers to the limitation of the component number.

\[
\begin{align*}
& u_{ij} g_{ij} \leq p_{Gij} \leq u_{ij} g_{ij}, \quad j \neq 0 \\
& u_{ij} g_{ij} \leq q_{Gij} \leq u_{ij} g_{ij}, \quad j \neq 0 \\
& \sum_{j \in N_j \neq 0} p_{Gij} \leq \epsilon_p \sum_{j \in N_j} p_{Dij} \\
& \sum_{j \in N_j \neq 0} q_{Gij} \leq \epsilon_q \sum_{j \in N_j} q_{Dij}
\end{align*}
\]

where \( u_{ij} \) is a binary variable indicating whether the grid component with active capacity \( p_{Gij} \) and reactive capacity \( q_{Gij} \) is installed at node \( j \). \( p_{Dij} \) is the active load at node \( j \). \( g \) and \( b \) are the upper and lower bound of the variable.

The second part is the power flow constraints of the synthetic network. Considering that classic power flow constraints are non-linear equations, the overall optimization problem can only be formulated as a mixed integer non-linear programming problem, which is hard to solve. To alleviate such difficulty, a relaxed branch flow model [30] is employed in this subsection, which is thus modeled as a set of second-order cone constraints as follows:

\[
\begin{align*}
p_I &= \sum_{k = 1}^{K_j} P_{kj} - \sum_{k = 1}^{K_j} (P_{ij} - r_{ij} P_{ij}) + g_{ij} v_j \\
q_I &= \sum_{k = 1}^{K_j} (Q_{kj} - P_{ij} x_{ij} + x_{ij} P_{ij}) + b_{ij} v_j \\
v_j &= v_i - 2 r_{ij} P_{ij} + x_{ij} Q_{ij}) + (r_{ij} + x_{ij}^2) l_{ij} \\
2P_{ij} &\leq l_{ij} + v_i \\
2Q_{ij} &\leq l_{ij} + v_i \\
V_j^2 &\leq v_j \leq V_j^2 \\
T_{ij}^2 &\leq l_{ij} \leq T_{ij}^2
\end{align*}
\]

where \( v_j = |V_j|^2 \), \( P_{ij} \) and \( Q_{ij} \) are the active and reactive power flow of line \( i - j \), \( x_{ij} \) is the reactance of line \( i - j \).

Overall, various standard grid components, e.g., capacitor banks and DERs, are placed in this generated synthetic network using the proposed network extension method, changing or even reversing the distribution of synthetic network power flow. It enables the generated synthetic network is similar to a realistic active distribution network. It should be noted that the proposed network extension method cannot be integrated with our UG-GAN algorithm because the goal is to mimic a specific network rather than replicate the original network.

D. Performance Evaluation

In order to evaluate the performance of the proposed method, topological and electrical indices are defined as follows. Moreover, several power applications are introduced in this subsection to further demonstrate that our synthetic networks are useful for power researchers and utility engineers, replacing the unavailable real-world data.

1) Topological and Electrical Indices: Based on previous work [12], several statistical and electrical based metrics are utilized in both graph and power aspects to prove that our model reproduces the most known properties inherent to real-world networks, which are listed below:
The number of nodes and edges of synthetic active distribution network, which reflect the scale of the network.

$D_{avg}$, $D_{max}$, $D_{br}$, $p_{PC}$: These four node degree-based indices are average node degree, maximum node degree, branching rate and assortativity coefficient, respectively. Among them, node degree represents the number of edges that are incident to a certain node, branching rate denotes the percentage of the number of nodes with degree greater than three, and assortativity coefficient is examined in terms of node degrees using the Pearson Correlation coefficient. These indices reflect the local graph properties of the active distribution systems. For example, urban or higher voltage level networks normally tend to branch out more compared to rural or lower voltage level ones.

$D_{cmax}$: Maximum depth. It can be used to roughly describe the strength of the voltage drop in radial distribution systems.

$P_{L_{avg}}, P_{L_{max}}$: Average and maximum nodal active power of loads, which reflect the baseline load level of the generated network.

$\Delta$: Three-phase unbalanced ratio defined in (16). This index reveals the unbalanced degree of the network.

$P_b, Q_b$: Active and reactive power at the interface of transmission and active distribution network.

$PF$: Power factor of the generated system.

Meanwhile, to prove that our model is not to simply replicate the original network, the ratio of overlapping edges ($R_{ov}$) between the real system and our synthetic system.

2) Application Verification: To further demonstrate that our generated active distribution network is realistic and useful, we review a question, that is, how to truly define whether the generated network is successful or not. It is indeed a more challenging problem, even compared to the network synthesis task. Most of the previous works only rely on statistical indices, obtained from a large amount of real-world data [1], [2], [3], [4], [5], [6], [7], [11], [12], [16]. However, as we mentioned before, topology properties are quite different for various distribution networks. This can also be confirmed using real data, as shown in Fig. 5. This figure shows four different indices (i.e., $D_{avg}$, $D_{max}$, $D_{br}$, $p_{PC}$) for the three distribution systems in the same region. It is clear that the statistical indices of the three systems are quite different, especially for $D_{br}$ and $p_{PC}$. Thus, synthetic distribution system should be generated by a single network. Moreover, even if the statistical indices of synthetic networks are similar to those of real networks, it is difficult to guarantee that these networks can be used as alternatives for representing real networks. In our view, synthetic networks should be validated in a power system manner. In other words, the synthetic networks generated by our method should achieve similar results as the real network in various power applications. Hence, we have tested three common applications: power flow analysis, DERs placement, and transmission and distribution power flow co-analysis. Among them, power flow analysis is performed to verify that the synthetic system satisfies static stability limits, including voltage and line power flow limits. Besides, DERs placement and transmission and distribution power flow co-analysis are carried out to demonstrate that the co-operation of transmission system and active distribution network with partial reverse power flow is of no abnormality.

IV. ACTIVE DISTRIBUTION SYSTEM SYNTHESIS FRAMEWORK

In this section, we summarize the the proposed framework as a flowchart shown in Fig. 6, so as to present a clear view of the methodology. It can be observed that the whole process is divided into three parts: data processing stage, UG-GAN based network synthesis stage, and network correction, extension and evaluation stage.

In the first stage, data processing stage, the data needs to be collected and pre-processed in order to prepare for the latter two phases. Priority to listing the detailed data requirements, we should emphasize the purpose and the high-value use case of this paper again. When system operators need to share their networks and data with researchers or the third agents but have user privacy concerns, they can perform the proposed method to obtain the corresponding synthetic networks for different networks separately. Considering that different distribution networks may share different properties, all we need to generate a synthetic network is all key information of a single real-world network. The detailed information to be collected is as follows:

1) Detailed three-phase unbalanced distribution network topology information with its parameter, including
can reject the generated random walks and performs poorly (the generated network is quite different from the real one), so that a more realistic topology can be generated, as shown in Fig. 8(b)–(g). When the training process is iterated 3,000 times, the discriminator loss drops dramatically to a small value, as shown in Fig. 9. After that, the two deep neural networks of the discriminator and generator are updated simultaneously via the adversarial process. Besides, in the UG-GAN training process, all random walks are sampled from this specific distribution network, with the same number as the ones generated from $G$ network. In this case study, we sampled 128 random walks per iteration for discrimination and training in UG-GAN.

### B. Distribution Network Synthesis Results

In this subsection, the detailed synthesis process is illustrated, and selected statistical and electrical-based indices are compared with the real-world input network, in order to verify the proposed method.

1) Visualization of Topology Synthesis Process: In the first few iterations of UG-GAN training, $G$ and $D$ of UG-GAN are still in a preliminary state with the initial parameters, as shown in Fig. 8(a). As a result, the generated network has many drawbacks, like isolated nodes, circle topology and etc. Then, in the early stage of the UG-GAN training process, when $G$ performs poorly (the generated network is quite different from the real one), $D$ can reject the generated random walks with a high degree of confidence. Therefore, in this stage, the discriminator loss drops dramatically to a small value, as shown in Fig. 9. After that, the two deep neural networks of UG-GAN are updated simultaneously via the adversarial process so that a more realistic topology can be generated, as shown in Fig. 8(b)–(g). When the training process is iterated 3,000 times, see Fig. 8(h), all topological properties of the generated distribution network are similar to those of the original network.
Note that all edge-related information is determined at this stage by using UG-GAN, including distribution grid components (like circuit breaks) connected in series, cable type of each line, and etc.

2) Result of Load Data Synthesis Process: By using the proposed KDE-based method, the time-series data of 504 single-phase loads and 5 three-phase loads are generated and assigned to a certain phase on one of the 60 nodes in the generated network aforementioned. Fig. 11(a) and (b) illustrate the probability density diagram of a residential load and sampled time-series load data, respectively. To eliminate the possible customer’s private information, the available customer power measurements are aggregated at the secondary transformer level by summing them at different times. Then, nodal loads are assigned to a certain phase of the generated network with minor topology correction using the formulated MIQP optimization problem to ensure the unbalanced degree within certain limits.

3) Synthetic Distribution System Description: The generated synthetic unbalanced distribution network consists of a 13.8 kV 60-node primary feeder that is supplied by a 69-kV substation. In this network, there are 57 branches in total, 48 of which are three-phase branches using 4 types of overhead lines and underground cables, and 9 of which are single-phase branches with 3 types of single-phase cables. The total length of the synthetic system is 3.34 miles. The three different types of unbalanced loads are assigned to 46 different nodes via secondary distribution transformers. Among them, an industrial three-phase load is connected to node #41, and residential or commercial loads are mixed together and connected to other nodes. Based on the results of our optimization-based component placement model, a capacitor bank is equipped near node #41 to provide reactive power support. Besides, 3 normally-closed circuit breaks are equipped in this network on lines 0-1, 9-10, and 33-36. The detailed structure of the generated network is illustrated in Fig. 10.

4) Indices Comparison of Generated Network: When the synthetic network is obtained, the aforementioned indices, indicating both topological and electrical properties, are used to compare the original and generated distribution networks, as shown in Table I. It can be clearly observed that all the representative statistical and electrical indices are similar. Meanwhile, the ratio of overlapping edges between two networks is about 0.5, preventing extracting real network confidential information by reverse engineering. Considering that the use of visualization can improve the interpretation of the results, we present the two networks directly, as shown in Fig. 7 (original network)
To further demonstrate the effectiveness of our approach, we have conducted qualitative and numerical comparisons with the existing work. It is worth noting that the proposed method focuses on mimicking one particular network without any context data assumptions, e.g., local geographical and social statistic data, which poses a challenge for comparison with existing statistical-based methods. Also, the generated network on these methods are normally the three-phase balanced grid. Hence, to ensure a fair comparison among the existing grid synthesis method, we have compared the proposed method with a random tree algorithm that is the only method without involving any context data [32]. Specifically, by using this method, 50 different synthetic networks have been generated for investigation, as shown in Table II. Obviously, the topological indices of the synthetic networks generated by the previous method are far from the original network, especially for $D_{br}$, $\rho_{PC}$ and $D_{e_{max}}$. Moreover, based on our observations, almost all randomly generated networks fail to satisfy the physical laws of the actual distribution system. For example, the upstream edge is a one-phase branch while the downstream one is a three-phase branch. Thus, such synthetic networks cannot be used to represent real-world systems in power system studies. As for other methods, including statistics-based method and simulated planning-based method, our proposed method is completely different in terms of purpose, algorithm, and input and output data. Thus, it is difficult to perform a fair quantitative comparison, since other

and Fig. 10 (synthetic network), so as to directly visualize the differences between the two networks.

Fig. 12. Voltage and power flow of the synthetic distribution network.

Fig. 13. Test system of transmission and distribution network co-simulation.

Fig. 14. Test result of transmission and distribution system co-analysis.
methods cannot obtain a synthetic network using the same input data as our method, and vice versa. For example, statistic-based methods require thousands of distribution network for key statistical properties qualification, and simulated planning-based methods need a large amount of local geographical and social data. Therefore, we merely perform a qualitative comparison table shown in Table III, to illustrate the advantages of the proposed approach.

In addition, training time of UG-GAN and memory consumption are tested, to further evaluate the computational performance of the proposed method. The training time ranges from 1.4 to 1.8 seconds per iteration, with total time 4651 seconds for all 3000 iterations in this case. In terms of memory consumption, 1334 MB is used while training. In all, our method can be easily implemented on any standard computer with no additional configuration.

C. Application Examples

To further prove that the synthetic network generated by our model is realistic, a set of application examples are presented in this subsection.

1) Baseline Power Flow Analysis: Convergent AC power flow is the primary consideration to justify the network. Normally, when the load and system parameters are within reasonable limits, a converged AC power flow result can be obtained using the three-phase backward-forward algorithm [33]. The key point is to verify whether the voltage magnitude of each node is within the given limit (e.g., 0.95-1 p.u.). Fig. 12 illustrates the voltage magnitude of each node under the baseline power flow.

In addition, the size of the solid circle represents the load of the node, and the thickness of the line represents the size of the line power. Noted that the voltage of bus #1 is assumed to be 1 p.u. in this case. It can be seen that the voltage of the generated system is within 0.966 p.u. to 1 p.u., which satisfies the voltage limits requirement. Besides, we also calculate the power flow on all overhead lines and underground cables based on the generated time series loads data, and they are all within the limit of chosen line conductors. Among them, the power flow on line 2-3 is the closest to the conductor limit in high load periods. Noted that, it still remains power flow margin.

2) Distributed Energy Resources Placement for Loss Reduction: In actual active distribution systems, DERs are also possibly installed by utilities for network loss reduction or renewable energy consumption. In this case, industrial load located at node #41 accounts for nearly two-thirds of the total load, and thus has a great influence on the total loss of this system. Thus, we try to vary the power flow of each line by installing DERs to reduce the loss. Based on the predetermined specific type of the DER [34], DERs placement is similar to the capacitor banks installation using the proposed MISCP formulation with minor modifications regarding the constraints on power injection. Noted that the total capacity of the DERs cannot be greater than the maximum load of the system. The results, including optimal sizes, locations, and the total amount of loss reduction, are shown in Table IV. Installing DER is not only an application of the generated active distribution network, but also expanding the scope of the application.

3) Transmission and Distribution Power Flow Co-Analysis: As we discussed before, considering the unbalanced architecture of the distribution system, a zero-sequence current might be injected into the transmission system. Moreover, the characteristics of power flow are generally be changed with the installation of various components, such as capacitor banks and DERs, in current distribution networks. As a result, distribution network can no longer be directly regarded as an equivalent load of the transmission network. Transmission and distribution network time-series power flow co-analysis is important for ISO, using the detailed distribution network with similar properties. An application example is presented in this subsection.

The test system is obtained by replacing the aggregated load at bus 6 of the standard IEEE 9-bus transmission system with the generated distribution network, see Fig. 13. The test is carried out using Matlab and OpenDSS and the results are shown in Fig. 14. It can be observed that the voltage and power flow are within an acceptable range.

VI. Conclusion

This paper has proposed a deep learning-based framework to generate synthetic three-phase unbalanced active distribution networks using limited real data. Our method can implicitly capture the topological and electrical properties of real-world networks without revealing critical information. Moreover, the
proposed method not only outputs grid connectivity but also effectively generates relevant time-series load data and locations and capacity of various grid components to obtain a comprehensive test case. With the proposed method, utilities will no longer have any concerns about making desensitized data publicly available at the request of industry and academia. Moreover, it is also possible for ISO of the transmission system to carry out transmission and distribution co-simulation based on generated networks for joint evaluation of the mutual effect of different systems. The results of case studies illustrate that these expectations can be met using the proposed method. Overall, our proposed method is able to consider the sparse network connectivity of the synthetic network merely by learning the distribution of biased random walks, and as a result, it greatly reduces the computation burden and improves the scalability at the stage of network synthesis. However, global convergence of the optimization problems may affect the scalability of our method, posed by large-scale distribution networks, in the network correction and extension process. Thus, the direction of our future research will focus on this issue, so as to extend our method to larger-scale distribution network synthesis task.

REFERENCES


Rong Yuan (Graduate Student Member, IEEE) received the B.S. degree in electrical engineering from the School of Electrical Engineering, Wuhu University, Wuhu, China, in 2016, and the Ph.D. degree in electrical engineering from the College of Electrical Engineering, Zhejiang University, Hangzhou, China, in 2021. Since July 2021, he has been an Engineer with Power Dispatching and Control Center, China Southern Power Grid Co., Ltd, Guangzhou, China. From 2019 to 2020, he was a Visiting Student with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. His research interest includes the application of data-driven methods in power system stability analysis.
Yuxuan Yuan (Graduate Student Member, IEEE) received the B.S. degree in electrical and computer engineering from Iowa State University, Ames, IA, USA, in 2017, where he is currently working toward the Ph.D. degree. His research interests include distribution system state estimation, synthetic networks, data analytics, and machine learning.

Zhaoyu Wang (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Shanghai Jiaotong University, and the M.S. and Ph.D. degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, GA, USA. He is currently Northrop Grumman Endowed Associate Professor with Iowa State University, Ames, IA, USA. He is also the Principal Investigator for a multitude of projects funded by the National Science Foundation, the Department of Energy, National Laboratories, PSERC, and Iowa Economic Development Authority. His research interests include optimization and data analytics in power distribution systems and microgrids. He was the recipient of the National Science Foundation CAREER Award, Society-Level Outstanding Young Engineer Award from IEEE Power and Energy Society (PES), Northrop Grumman Endowment, College of Engineering’s Early Achievement in Research Award, and the Harpole-Pentair Young Faculty Award Endowment. He is the Chair of IEEE PES PSOPE Award Subcommittee, Vice Chair of PES Distribution System Operation and Planning Subcommittee, and Vice Chair of PES Task Force on Advances in Natural Disaster Mitigation Methods. He is the Associate Editor for IEEE TRANSACTIONS ON POWER SYSTEMS, IEEE TRANSACTIONS ON SMART GRID, IEEE OPEN ACCESS JOURNAL OF POWER AND ENERGY, IEEE POWER ENGINEERING LETTERS, and IET Smart Grid.

Guangchao Geng (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from the College of Electrical Engineering, Zhejiang University, Hangzhou, China, in 2009 and 2014, respectively. From 2012 to 2013, he was a Visiting Student with the Department of Electrical and Computer Engineering, Iowa State University, Ames, IA, USA. From 2014 to 2017, he was a Postdoctoral fellow with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China, and Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. He is currently an Associate Professor with the College of Electrical Engineering, Zhejiang University. His research interest includes power system stability and control, the applications of Internet of Things (IoT) technique and high performance computing (HPC) technique in power systems.

Quanyuan Jiang (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1997, 2000, and 2003, respectively. From 2006 to 2008, he was a Visiting Associate Professor with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA. He is currently a Professor with the College of Electrical Engineering and the Associate Dean of Undergraduate School, Zhejiang University, Hangzhou, China. His research interest includes power system stability and control, applications of energy storage systems, and high performance computing technique in power systems.