Multi-kernel Collaborative Graph Convolution Neural Network for Operational Reliability Assessment Considering Varying Topologies

Xinyu Liu, Maosheng Gao, Juan Yu, Zhifang Yang, and Wenyuan Li

***Abstract*—Operational reliability assessment (ORA), which evaluates the risk level of power systems, is hindered by accumulated computational burdens and thus cannot meet the demands of real-time assessment. Recently, data-driven methods with fast calculation speeds have emerged as a research focus for online ORA. However, the diverse contingencies of transformers, power lines, and other components introduce numerous topologies, posing significant challenges to the learning capabilities of neural networks. To this end, this paper proposes a multi-kernel collaborative graph convolution neural network (GCNN) for ORA considering varying topologies. Specifically, a physics law-informed graph convolution kernel derived from the Gaussian-Seidel iteration is introduced. It effectively aggregates node features across different topologies. By integrating additional advanced graph convolution kernels with a novel self-attention mechanism, the multi-kernel collaborative GCNN is constructed, which enables the extraction of diverse features and the construction of representative node feature vectors, thereby facilitating high-precision reliability assessments. Furthermore, to enhance the robustness of multi-kernel collaborative GCNN, the inherent pattern of the load-shedding model is analyzed and utilized to design a specialized supervised loss function, which allows the neural network to explore a broader feature space. Compared with the existing data-driven methods, the multi-kernel collaborative GCNN, combined with supervised exploration, can accommodate a wider range of contingencies and achieve superior assessment accuracy.**

***Index Terms*—Reliability assessment, multi-kernel collaborative design, self-attention graph convolution neural network (GCNN), topology.**

I. Introduction

OPERATIONAL reliability assessment (ORA) is a vital tool to evaluate operational risk and carry out early warning in the power system [1]. In the context of developing renewable energy worldwide, the power system faces significant uncertainties due to the intermittent and stochastic characteristics of wind and solar [2]. It is necessary to implement the ORA in real time [3]. Thus, in China, the National Energy Administration is actively promoting the establishment of a power system reliability management framework, which would facilitate the real-time collection of vital power equipment reliability data and foster the development of rapid and intelligent ORA methodologies for future advancements.

However, with the increasing collection of abundant reliability data for vital power equipment, conventional numerical and data-driven ORA methods face significant challenges. On the one hand, the traditional numerical ORA methods, such as the Monte Carlo simulation method, require iteratively solving load-shedding optimization problems under a vast number of system states. This results in high computational complexity, which is particularly time-consuming for large-scale power systems and hinders real-time reliability assessment due to the accumulated computational burden. On the other hand, with the availability of extensive data on critical power equipment, it is necessary to simulate more complex and diverse system states, which introduces various system topologies. This, in turn, poses a significant challenge to the effective learning capability of neural networks employed in data-driven ORA methods.

The conventional ORA methods can be categorized into two types: analytical method and simulation method. According to the probabilistic model of uncertainty and contingency probability of critical equipment, the analytical method directly derives an analytical formulation for computing the operational reliability [4], [5]. This method performs effectively for small-scale power systems with a limited number of system states and manageable computational complexity. However, when dealing with complex operational conditions and a large number of severe events in large-scale power systems, the analytical method becomes overly intricate, which may fail to account for certain operational scenarios. In contrast, the simulation method, which treats the problem as a

Manuscript received: November 21, 2024; revised: February 21, 2025; accepted: April 23, 2025. Date of CrossCheck: April 23, 2025. Date of online publication: August 22, 2025.

This work was supported by the National Natural Science Foundation of China (No. 52377076).

X. Liu is with Tsinghua University, Beijing 100084, China, and he is also with Chongqing Electric Power Company of State Grid Corporation of China, Chongqing 400015, China (e-mail: 1162573948@qq.com).

M. Gao (corresponding author), J. Yu, Z. Yang, and W. Li are with the State Key Laboratory of Power Transmission Equipment Technology, College of Electrical Engineering, Chongqing University, Chongqing 400044, China (e-mail: maoshenggao@outlook.com; 148454745@qq.com; zfyang@cqu.edu.cn; wenyuan.li@ieee.org).

DOI: 10.35833/MPCE.2024.001249

series of experiments and subsequently calculates operational reliability indices, does not suffer from these limitations [6]. It can simulate not only the operational conditions, contingency events, and uncertainties but also the non-electrical system factors, such as reservoir operational conditions and weather effects, which are typically beyond the scope of the analytical method. However, one drawback that impedes the real-time application of simulation methods is the cumulative computational burden [7]. Since the simulation method needs to simulate all the operational conditions and solve the load-shedding problem to compute the operational reliability indices, it often fails to complete within the operational time window, such as 5 min, making it unsuitable for online assessment of power systems. To address this computational burden issue associated with traditional numerical simulation methods, data-driven ORA methods have been proposed in recent years.

The data-driven method is a further improvement on the traditional simulation method. Its essence lies in leveraging the rapid computational capabilities of neural networks to expedite ORA. For example, neural networks can be employed to swiftly classify success states [8], [9], substitute for solving load-shedding problems [10], [11], or directly predict the reliability indices [12]. Those neural networks, characterized by their robust classification and approximation abilities, can precisely determine the success/failure states, the load curtailment amounts, and even the reliability indices while accounting for the stochastic nature of renewable energy and load fluctuations. Moreover, their applications extend beyond ORA problems. The exceptional computational speed and intelligent decision-making capabilities of neural networks have been extensively explored in other power system optimization challenges, such as optimal power flow calculations, economic dispatch, and unit commitment solutions [13]. Regarding the ORA, neural networks aim to approximate the non-linear and non-convex mapping from those continuous features to the success/failure states or the load curtailment amounts. However, ORA encompasses not only continuous features but also discrete features, which typically arise from topology changes in the power grid, such as contingencies in power equipment. This dynamic topology characteristic complicates the load-shedding mapping and poses significant challenges to neural network learning.

To cope with this issue in the data-driven ORA method, two strategies can be employed to handle the discrete topology features: feature engineering and application of graph convolutional neural networks (GCNNs).

Feature engineering involves constructing a representative feature vector that indicates topology changes, enabling neural networks to effectively utilize topology information for approximating target outputs (including load curtailments). For instance, the diagonal element of the susceptance matrix [14], the magnitude and angle of the diagonal element of the admittance matrix [15], or the upper triangular part of the susceptance matrix [12] have been used as the topology feature vector. Besides, based on the power flow model, differences in voltage magnitude and branch power flow between the original and new topology are also regarded as representative topology features [16], [17]. However, due to the consideration of the input feature dimension, feature engineering often incurs feature loss to maintain an acceptable input dimensionality. Furthermore, while encoding topology into a feature vector, feature engineering may overlook the correlation between topology and node features.

On the contrary, the application of GCNNs can effectively address this issue. By constructing a GCNN with a structure analogous to that of the power grid, both node features (such as loads and renewable energy generation) and topology features (including interconnections between nodes, branch susceptance and conductance, and contingency scenarios) can be seamlessly integrated into the neural network without concerns about scale explosion. This method ensures compatibility with the evolving scale and structure of the power grid at both input and output stages [18], [19]. As a result, GCNN has been widely applied in various aspects of power systems, including optimization [20], state estimation [21], [22], and forecasting [23], [24]. Moreover, it is an ideal tool for ORA, particularly when handling varying topologies. There are three types of GCNNs with different graph convolution kernels that can be applied to the ORA problem. Firstly, the standard graph convolution kernel derived from the Laplacian matrix based on the spectral graph theory is utilized for load-shedding calculation problems [25], which embeds the topology of the power grid into the forward propagation of neural networks, thereby enhancing the adaptability for varying topologies. The second type of graph convolution kernel refers to the impedance matrix in power systems. It is an impedance-based Gaussian kernel for graph convolution [26], [27], which makes the weight in the convolution kernel correlate with the branch impedance and further improves the ability of GCNN to extract topology features. The third type of graph convolution kernel is designed in accordance with the power flow model [28], regarding power flow equations as the neighborhood aggregation function and forcing the neural network to extract the complex physics features. Currently, the GCNNs with the power flow model embedded are more suitable for addressing varying topologies in the ORA problem.

Although the advanced GCNNs with the power flow model embedded make it possible to predict the load curtailments under varying topologies, there is still room for potential improvements. Specifically, during the derivation of the forward propagation function in those GCNNs, the variables, e. g., voltage magnitude and phase angle, are partially assigned to the preceding graph convolution layer and partially assigned to the subsequent one. This separation violates the physics laws and may result in insufficient or inaccurate physics feature extraction. Besides, the three types of graph convolution kernels help the neural network pay more attention to different features in the power system individually. Integrating different graph convolution kernels may compel neural networks to extract more represenive features. Consequently, this would enhance the accuracy and robust topology adaptability of the data-driven ORA method. To this end, a multi-kernel collaborative GCNN is proposed for the ORA, specifically addressing the varying topologies. By integrating multiple kernels, including spectral graph convolution kernel, impedance-based Gaussian kernel, power flow model embedded graph convolution kernel, and the proposed physics law-informed graph convolution kernel, this GCNN can effectively handle complex scenarios involving changing topologies and achieve more accurate ORAs. Additionally, a new learning strategy is developed based on the inherent pattern of the ORA model to improve the robustness of neural networks. The main contributions are summarized as follows:

1) A novel physics law-informed graph convolution kernel is constructed. Different from the variable assumption, this kernel is rooted in the Gaussian-Seidel iteration of the complex power flow equations, which can efficiently aggregate graph node features in a manner consistent with physical laws such as Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL).

2) A multi-kernel collaborative GCNN is designed. By leveraging the feature extraction capabilities of various advanced graph convolution kernels, this GCNN aggregates node features effectively while incorporating a novel self-attention mechanism to construct a highly representative node feature vector. This updated vector encompasses both global topology and physical law-associated features, which lays the foundation for high-precision solving of load-shedding problems by capturing essential data characteristics.

3) An inherent pattern-guided learning strategy is proposed to strengthen the robustness of the multi-kernel collaborative GCNN. Specifically, an inherent pattern between the load and its curtailment is derived, enabling direct calculation of the load curtailment amount without iterative solving. Thus, a specialized supervised learning loss function based on such an inherent pattern is constructed, which makes it possible to add random disturbances to the load during the training process without label concerns. As a result, the neural network can explore a broader feature space, thereby enhancing the robustness of neural networks.

The rest of the paper is organized as follows. In Section II, the conventional ORA method and proposed ORA method are introduced. The architecture of multi-kernel collaborative GCNN is presented in Section III. Section IV presents the inherent pattern-guided learning. Section V presents the case study. Section VI presents the discussion. And the conclusion is given in Section VII.

II. Conventional ORA Method and Proposed ORA Method

A. Conventional ORA Method

The conventional ORA method of power systems is the Monte Carlo simulation method. Its main steps can be concluded as follows.

1) System State Sampling

The system states are usually sampled based on the probabilistic distribution of loads and renewable energy and the failure probability of vital power equipment.

2) System State Analysis

After obtaining the system states, the next step is determining whether the power system can operate safely. So, the power flow calculation is implemented to check whether any violation exists. If it is true, optimize the load-shedding problem. The load-shedding problem is to minimize the load curtailment under the practical constraints of power systems, such as power flow constraints and generator ramping limitations. The mathematical formulation of the load-shedding problem is shown in (1)-(10).

$$\min \sum_{i \in S_{\mathrm{B}}} \rho_{\mathrm{P}i}\left(C_{\mathrm{P}i} + \rho_{\mathrm{Q}i} C_{\mathrm{Q}i}\right) \tag{1}$$

$$\begin{cases} P_{\mathrm{G}i} u_i - P_{\mathrm{D}i} + C_{\mathrm{P}i} = V_i \sum\limits_j V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}\right) \\ Q_{\mathrm{G}i} u_i - Q_{\mathrm{D}i} + C_{\mathrm{Q}i} = V_i \sum\limits_j V_j \left(G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}\right) \end{cases} \quad i,j \in S_{\mathrm{B}} \tag{2}$$

$$\underline{P_{\mathrm{G}i}} \le P_{\mathrm{G}i} \le \overline{P_{\mathrm{G}i}} \quad i \in S_{\mathrm{G}} \tag{3}$$

$$P_{\mathrm{G}i} - P_{\mathrm{G}i,\mathrm{last}} \le P_{\mathrm{G}i,\mathrm{rampup}} \quad i \in S_{\mathrm{G}} \tag{4}$$

$$P_{\mathrm{G}i,\mathrm{last}} - P_{\mathrm{G}i} \le P_{\mathrm{G}i,\mathrm{rampdown}} \quad i \in S_{\mathrm{G}} \tag{5}$$

$$\underline{Q_{\mathrm{G}i}} \le Q_{\mathrm{G}i} \le \overline{Q_{\mathrm{G}i}} \quad i \in S_{\mathrm{G}} \tag{6}$$

$$0 \le C_{\mathrm{P}i} \le P_{\mathrm{D}i} \quad i \in S_{\mathrm{L}} \tag{7}$$

$$0 \le C_{\mathrm{Q}i} \le Q_{\mathrm{D}i} \quad i \in S_{\mathrm{L}} \tag{8}$$

$$\underline{V_i} \le V_i \le \overline{V_i} \quad i \in S_{\mathrm{B}} \tag{9}$$

$$\begin{cases} P_{\mathrm{L},ij} = V_i V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}\right) - V_i^2 G_{ij} \\ \underline{P_{\mathrm{L},ij}} \le P_{\mathrm{L},ij} \le \overline{P_{\mathrm{L},ij}} \end{cases} \quad i,j \in S_{\mathrm{B}} \tag{10}$$

where $C_{\mathrm{P}i}$ and $C_{\mathrm{Q}i}$ are the active and reactive load curtailments at bus i, respectively; $\rho_{\mathrm{P}i}$ and $\rho_{\mathrm{Q}i}$ are the cost coefficients at bus i; $P_{\mathrm{D}i}$ and $Q_{\mathrm{D}i}$ are the active and reactive power demands at bus i, respectively; $P_{\mathrm{G}i}$ and $Q_{\mathrm{G}i}$ are the active and reactive power generations of generator i, respectively; V_i and θ_i are the voltage magnitude and phase angle at bus i, respectively; θ_{ij} is the phase angle difference between buses i and j; G_{ij} and B_{ij} are the elements of the conductance and susceptance matrices in the ith row and jth column, respectively; $P_{\mathrm{L},ij}$ is the active branch power between buses i and j; $P_{\mathrm{G}i,\mathrm{last}}$ is the power generation of generator i in the last timestep; $P_{\mathrm{G}i,\mathrm{rampup}}$ and $P_{\mathrm{G}i,\mathrm{rampdown}}$ are the maximum and minimumn ramping amounts of generator i, respectively; ¯ and _ denote the upper and lower limits, respectively; and S_{G}, S_{B}, and S_{L} are the index sets of generators, buses, and branches, respectively.

3) Reliability Index Calculation

The basis indices of operational reliability are the adequacy indices, e.g., probability of load curtailment (PLC), and expected demand not supplied (EDNS). The PLC and EDNS are calculated in this paper after obtaining the load curtailment under different system states, as shown in (11) and (12).

$$PLC = \sum_{i \in S} p_i \tag{11}$$

$$EDNS = \sum_{i \in S} \sum_{j \in S_{\mathrm{B}}} p_i C_{\mathrm{P}j,i} \tag{12}$$

where p_i is the probability of system state i; S is the set of system states where load curtailment occurs; and $C_{\mathrm{P}j,i}$ is the load curtailment amount at bus j under system state i.

4) Convergence Criterion

When implementing the Monte Carlo simulation method, it is necessary to determine when to stop. Two mainstream convergence criteria are the maximum sample number and the variance of reliability indices. In this paper, both the maximum sample number and the variance of PLC and EDNS are used.

B. Framework of ORA Method

According to the Monte Carlo simulation method, it is required to solve the load-shedding problem (1)-(10) while giving different system states. Usually, thousands of system states would be sampled and analyzed. However, the Monte Carlo simulation method cannot be applied in real-time scenarios due to its significant computational burden. To address this issue, the data-driven ORA method leverages the neural network with fast computational speed. Nevertheless, its computational accuracy has been a concern, particularly when numerous contingencies that alter the topology of power system are considered.

This paper improves the neural network structure and learning strategy to address the varying topologies in the data-driven ORA method, referred to as the proposed ORA method. For better clarity, the framework of the proposed ORA method is illustrated in Fig. 1. The proposed ORA method still follows the data-driven simulation structure but exploits the neural networks to solve the load-shedding problem when given different system states or features. At the offline phase, a neural network agent is constructed to accurately predict the minimum load curtailment under various contingencies. At the online phase, this agent predicts load curtailments under different system states and evaluates the operational reliability of the power system. This paper makes two distinct contributions to effectively training a neural network for adapting to varying topologies. First, we design the GCNN structure by proposing a physics law-informed graph convolution kernel and integrating it with other advanced kernels to construct a multi-kernel collaborative GCNN. Second, we develop an inherent pattern-guided learning strategy by deriving an inherent pattern from the load-shedding model and designing a supervised learning loss function. This enables the neural network to explore a broader feature space during training. Consequently, by applying the neural network architecture and learning strategy, the topology adaptability of the proposed ORA method can be enhanced, leading to more accurate reliability assessments.

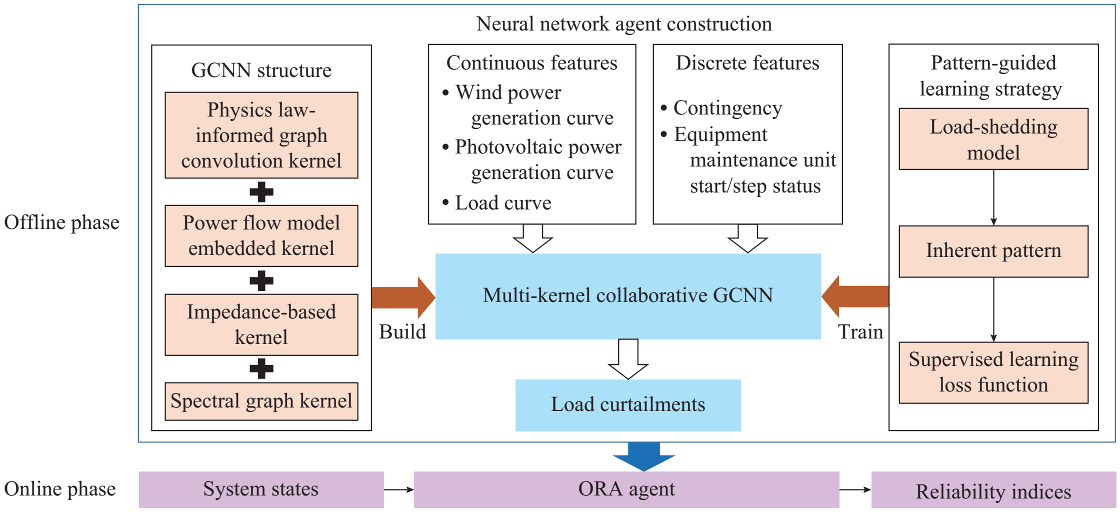
Neural network agent construction
Offline phase
GCNN structure
Physics law-informed graph convolution kernel
Power flow model embedded kernel
Impedance-based kernel
Spectral graph kernel
Build
Continuous features
• Wind power generation curve
• Photovoltaic power generation curve
• Load curve
Discrete features
• Contingency
• Equipment maintenance unit start/step status
Multi-kernel collaborative GCNN
Load curtailments
Train
Pattern-guided learning strategy
Load-shedding model
Inherent pattern
Supervised learning loss function
Online phase
System states
ORA agent
Reliability indices

Fig. 1. Framework of proposed ORA method.

III. Architecture of Multi-kernel Collaborative GCNN

In the conventional ORA method, the repeated calculation of the highly complex and time-consuming load-shedding model (1)-(10) hinders its online practical application. If a neural network can solve the load-shedding model quickly, the ORA will be fundamentally accelerated. In the load-shedding model, equipment failures and other factors will change the system topology, making it more difficult for the neural network to learn. Effectively extracting the impact of topology changes on power flow distribution is one of the strategies to make the data-driven ORA method more practical. Graph neural network is an effective tool for dealing with topology changes in the proposed ORA method by embedding the topology into the forward propagation of the neural network and performing feature aggregation calculations based on the input topology. However, in the load-shedding model, the nonlinear and complex power flow model makes it difficult for graph neural networks to extract important features that affect the assessment results accurately. To this end, this section focuses on how to accurately consider the impact of the power flow model in the load-shedding model in graph convolution and designs a multi-kernel collaborative GCNN architecture for ORA. Next, we introduce some existing graph convolution methods and propose a physics law-informed graph convolution kernel. Finally, the design scheme of a multi-kernel collaborative GCNN is introduced.

A. Brief Introduction to GCNN

GCNN is a kind of neural network designed to handle graph-structured data. Generally, there are two operations in the forward propagation of GCNNs: aggregation and convolution [29]. The aggregation operation collects features from the neighborhoods and combines them with their features to form new node feature vectors. Then, the convolution operation would update the node features with convolution parameters.

$$x_i^{l+1}=f_{CONV}\Big(f_{AGGRE}\Big(x_j^l\Big|\, j\in N(i)\Big),H\Big) \tag{13}$$

where x_i^{l+1} is the feature of node i in the $(l+1)^{\text{th}}$ layer; $N(i)$ is the set of adjacent nodes and itself for node i; f_{CONV} and f_{AGGRE} are the convolution and aggregation operation functions, respectively; and H is the convolution parameter.

To reduce the scale of trainable parameters in GCNNs, most f_{AGGRE} functions merely have known parameters, and how to determine the f_{AGGRE} function becomes a significant task in constructing GCNNs. The f_{AGGRE} function of the standard GCNN is constructed based on the adjacent matrix $\boldsymbol{A}$ to consider the internal effects of the neighborhoods based on the spectral graph theory. Its mathematical formulation in matrix form is:

$$f_{AGGRE1}\big(\boldsymbol{X}^l\big)=\hat{\boldsymbol{D}}^{-\frac{1}{2}}\hat{\boldsymbol{A}}\hat{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{X}^l \tag{14}$$

where $\boldsymbol{X}^l\in\mathbf{R}^{n\times k}$ is the matrix of input features with n nodes and k-dimensional features; and $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{D}}$ are the normalized adjacent matrix and degree matrix based on the re-normalization ticks, respectively [30].

In the applications of GCNN in power systems, the physics parameters and physics model that can describe the internal effects of neighborhood nodes are exploited, e.g., branch impedance and power flow model. Hence, some physics-informed GCNNs in power systems are proposed. First, the branch impedance is utilized to construct the aggregation weights of neighborhood features. This kind of GCNN adopts a Gaussian kernel $W_{ij}=\exp\Big(-\kappa\big|z_{ij}\big|^2\Big)$, where κ is a scaling factor [26]; and z_{ij} is the impedance of the power line connecting nodes i and j. In different power systems, κ is adjusted to ensure that the weights of power lines are in a reasonable range, and no power line is ignored due to too small weight. The aggregation function of this GCNN can be indicated as:

$$f_{AGGRE2}\big(\boldsymbol{X}^l\big)=\boldsymbol{W}\boldsymbol{X}^l \tag{15}$$

where $\boldsymbol{W}$ is the weight adjacent matrix calculated using the branch impedance.

To further improve the feature aggregation ability, the processing strategy based on the re-normalization tricks is also implemented in the impedance-based aggregation $\boldsymbol{W}$. The aggregation function is:

$$\begin{cases} f_{AGGRE3}\big(\boldsymbol{X}^l\big)=\hat{\boldsymbol{D}}_w^{-\frac{1}{2}}\hat{\boldsymbol{W}}\hat{\boldsymbol{D}}_w^{-\frac{1}{2}}\boldsymbol{X}^l \\ \hat{\boldsymbol{W}}=\boldsymbol{W}+\boldsymbol{I}_n \\ \hat{\boldsymbol{D}}_w=\operatorname{diag}\left(\sum\limits_{j=1}^{n}\hat{W}_{ij}\right) \end{cases} \tag{16}$$

where $\boldsymbol{I}_n$ is the identity matrix, whose diagonal elements are 1 and other elements are 0; and $\hat{W}_{ij}$ is the i^{th} row and j^{th} column element of $\hat{\boldsymbol{W}}$.

In addition, the power flow model in the Cartesian coordinate system, indicated in (17), is also embedded into the aggregation function to enhance the feature extraction ability of GCNNs [31], which sets $e_i=V_i\cos\theta_i$ and $f_i=V_i\sin\theta_i$ as the coupled node feature in the graph and takes advantage of the power flow model (17) as the aggregation function to update them. Assume that the features of the central node i are unknown and updated by its neighborhoods. Then, the elements in (17) including e_i or f_i are kept on the right-hand side and the other elements are moved to the left-hand side. By solving e_i and f_i in the reformulated power flow equations, the node feature aggregation function could be derived, as shown in (18).

$$\begin{cases} P_{\mathrm{G}i}-P_{\mathrm{D}i}=\sum\limits_{j\in S_B}\Big[e_i\big(G_{ij}e_j-B_{ij}f_j\big)+f_i\big(G_{ij}f_j+B_{ij}e_j\big)\Big] \\ Q_{\mathrm{G}i}-Q_{\mathrm{D}i}=\sum\limits_{j\in S_B}\Big[f_i\big(G_{ij}e_j-B_{ij}f_j\big)-e_i\big(G_{ij}f_j+B_{ij}e_j\big)\Big] \end{cases} \tag{17}$$

$$\begin{cases} f_{AGGRE4}^{e}\big(\boldsymbol{e}^l,\boldsymbol{f}^l\big)=\dfrac{\boldsymbol{\delta}\odot\boldsymbol{\alpha}-\boldsymbol{\lambda}\odot\boldsymbol{\beta}}{\boldsymbol{\alpha}\odot\boldsymbol{\alpha}+\boldsymbol{\beta}\odot\boldsymbol{\beta}} \\ f_{AGGRE4}^{f}\big(\boldsymbol{e}^l,\boldsymbol{f}^l\big)=\dfrac{\boldsymbol{\delta}\odot\boldsymbol{\beta}+\boldsymbol{\lambda}\odot\boldsymbol{\alpha}}{\boldsymbol{\alpha}\odot\boldsymbol{\alpha}+\boldsymbol{\beta}\odot\boldsymbol{\beta}} \end{cases} \tag{18}$$

$$\boldsymbol{\alpha}=\boldsymbol{G}_{ndiag}\boldsymbol{e}^l-\boldsymbol{B}_{ndiag}\boldsymbol{f}^l \tag{19}$$

$$\boldsymbol{\beta}=\boldsymbol{G}_{ndiag}\boldsymbol{f}^l+\boldsymbol{B}_{ndiag}\boldsymbol{e}^l \tag{20}$$

$$\boldsymbol{\delta}=-\boldsymbol{P}_D-\big(\boldsymbol{e}^l\odot\boldsymbol{e}^l+\boldsymbol{f}^l\odot\boldsymbol{f}^l\big)\boldsymbol{G}_{diag} \tag{21}$$

$$\boldsymbol{\lambda}=-\boldsymbol{Q}_D-\big(\boldsymbol{e}^l\odot\boldsymbol{e}^l+\boldsymbol{f}^l\odot\boldsymbol{f}^l\big)\boldsymbol{B}_{diag} \tag{22}$$

where $\boldsymbol{e}^l$ and $\boldsymbol{f}^l$ are the coupled node feature vectors of the l^{th} layer; f_{AGGRE4}^{e} and f_{AGGRE4}^{f} are the node feature aggregation functions by embedding the power flow equations, respectively; $\boldsymbol{P}_D$ and $\boldsymbol{Q}_D$ are the active and reactive power vectors, respectively; $\odot$ is the Hadamard (entry-wise) product; $\boldsymbol{G}_{ndiag}$ and $\boldsymbol{B}_{ndiag}$ are the admittance and susceptance matrices without diagonal elements, respectively; and $\boldsymbol{G}_{diag}$ and $\boldsymbol{B}_{diag}$ are the diagonal elements of the admittance and susceptance matrices, respectively.

Although aggregation methods in the GCNN help extract the topology and physical features, the assumption in the derivation violates the law of power flow. While iterating several times, node features could diverge [31]. Therefore, there are still potential improvements to design a well-performed aggregation function and physics law-informed GCNN in power systems.

B. Physics Law-informed GCNN

To follow the physics law in the feature aggregation operation, there must be no assumptions in the derivation based on the power flow model. The Gaussian-Seidel iteration is an ideal tool, which does not break the initial equations, and the physical property can be well preserved. Besides, the Gaussian-Seidel iteration is applied to solve the power flow equations. Therefore, referring to the process of solving the power flow equation by Gaussian-Seidel iteration, this subsection mainly introduces how to embed the Gaussian-Seidel iteration of power flow equations into the graph convolution.

The power flow model in complex form is:

$$\begin{cases} \boldsymbol{Y}\boldsymbol{V}=\boldsymbol{I} \\ \boldsymbol{I}=\left[\boldsymbol{S}/\boldsymbol{V}\right] \end{cases} \tag{23}$$

where $\boldsymbol{Y}$ is the admittance matrix; $\boldsymbol{V}$ is the voltage vector in complex form; $\boldsymbol{I}$ is the injection current vector in complex form; and $\boldsymbol{S}$ is the power injection vector in complex form.

When decomposing the matrix $\boldsymbol{Y}$ into its diagonal form, denoted as $\boldsymbol{Y}_{diag}$, and the off-diagonal form $\boldsymbol{Y}_{ndiag}$, (23) can be reformulated as (24). Hence, the Gaussian-Seidel iteration function of V_i in the $(l+1)^{\text{th}}$ iteration can be formularized as shown in (25).

$$\boldsymbol{V}=\boldsymbol{Y}_{diag}^{-1}\left(\boldsymbol{I}-\boldsymbol{Y}_{ndiag}\boldsymbol{V}\right) \tag{24}$$

$$V_i^{l+1}=\frac{1}{Y_{ii}}\left(\frac{S_i}{V_i^l}-\sum_{j=1}^{n}Y_{ij}V_j^l\right) \tag{25}$$

where Y_{ij} is the i^{th} row and j^{th} column of $\boldsymbol{Y}$; and S_i is the i^{th} element of $\boldsymbol{S}$.

It can be observed that (25) is an aggregation function to update $\boldsymbol{V}$ using the system state (power injection vector $\boldsymbol{S}$) and the grid parameters (conductance and susceptance matrices $\boldsymbol{G}$ and $\boldsymbol{B}$). There are no assumptions and the physics laws (like KCL and KVL) are still satisfied. Thus, we take (25) as the aggregation function and design a new graph convolution method. However, all the elements in (25) are complex values, and it is difficult to exploit directly in the forward propagation of neural networks. Rewriting it in the Cartesian coordinate system to decouple node features where $\boldsymbol{V}=\boldsymbol{e}+\text{j}\boldsymbol{f}$, the aggregation function could be:

$$e_i^{l+1}+\text{j}f_i^{l+1}=\frac{P_i+\text{j}Q_i}{\left(G_{ii}+\text{j}B_{ii}\right)\left(e_i^l+\text{j}f_i^l\right)}-\frac{1}{G_{ii}+\text{j}B_{ii}}\sum_{j=1}^{n}\left(G_{ij}+\text{j}B_{ij}\right)\left(e_i^l+\text{j}f_i^l\right) \tag{26}$$

where P_i and Q_i are the real and imaginary parts of S_i, respectively.

Then, decoupling the real and imaginary parts in (26), the aggregation function can be written as:

$$\begin{cases} e_i^{l+1}=\dfrac{\alpha_i' G_{ii}}{G_{ii}^2+B_{ii}^2}+\dfrac{\beta_i' B_{ii}}{G_{ii}^2+B_{ii}^2} \\ f_i^{l+1}=\dfrac{\beta_i' G_{ii}}{G_{ii}^2+B_{ii}^2}-\dfrac{\alpha_i' B_{ii}}{G_{ii}^2+B_{ii}^2} \end{cases} \tag{27}$$

where e_i^{l+1} and f_i^{l+1} are the node feature elements in the $(l+1)^{\text{th}}$ iteration. α_i' and β_i' can be written as:

$$\begin{cases} \alpha_i'=\dfrac{P_i e_i^l+Q_i f_i^l}{\left(e_i^l\right)^2+\left(f_i^l\right)^2}-\displaystyle\sum_{j=1}^{n}G_{ij}e_j^l+\sum_{j=1}^{n}B_{ij}f_j^l \\ \beta_i'=\dfrac{Q_i e_i^l-P_i f_i^l}{\left(e_i^l\right)^2+\left(f_i^l\right)^2}-\displaystyle\sum_{j=1}^{n}G_{ij}f_j^l-\sum_{j=1}^{n}B_{ij}e_j^l \end{cases} \tag{28}$$

In (27) and (28), $G_{ii}^2+B_{ii}^2$ and $\left(e_i^l\right)^2+\left(f_i^l\right)^2$ are in the denominator. $G_{ii}^2+B_{ii}^2$ is a nonzero constant for different nodes in a given topology. e_i^l and f_i^l may have zero elements in the hidden layers and cause a calculation error in the neural network. To avoid this situation, a small constant δ is added to $\left(e_i^l\right)^2+\left(f_i^l\right)^2$. The aggregation function of this GCNN can be written as shown in (29)-(31) in matrix form.

$$\begin{cases} f_{AGGRE5}^{e}\left(\boldsymbol{e}^l,\boldsymbol{f}^l\right)=\dfrac{\boldsymbol{\alpha}'\odot\boldsymbol{G}_{diag}+\boldsymbol{\beta}'\odot\boldsymbol{B}_{diag}}{\boldsymbol{G}_{diag}\odot\boldsymbol{G}_{diag}+\boldsymbol{B}_{diag}\odot\boldsymbol{B}_{diag}} \\ f_{AGGRE5}^{f}\left(\boldsymbol{e}^l,\boldsymbol{f}^l\right)=\dfrac{-\boldsymbol{\alpha}'\odot\boldsymbol{B}_{diag}+\boldsymbol{\beta}'\odot\boldsymbol{G}_{diag}}{\boldsymbol{G}_{diag}\odot\boldsymbol{G}_{diag}+\boldsymbol{B}_{diag}\odot\boldsymbol{B}_{diag}} \end{cases} \tag{29}$$

$$\boldsymbol{\alpha}'=\frac{\boldsymbol{P}\odot\boldsymbol{e}^l+\boldsymbol{Q}\odot\boldsymbol{f}^l}{\boldsymbol{e}^l\odot\boldsymbol{e}^l+\boldsymbol{f}^l\odot\boldsymbol{f}^l+\delta}-\boldsymbol{G}_{ndiag}\boldsymbol{e}^l+\boldsymbol{B}_{ndiag}\boldsymbol{f}^l \tag{30}$$

$$\boldsymbol{\beta}'=\frac{\boldsymbol{Q}\odot\boldsymbol{e}^l-\boldsymbol{P}\odot\boldsymbol{f}^l}{\boldsymbol{e}^l\odot\boldsymbol{e}^l+\boldsymbol{f}^l\odot\boldsymbol{f}^l+\delta}-\boldsymbol{G}_{ndiag}\boldsymbol{f}^l-\boldsymbol{B}_{ndiag}\boldsymbol{e}^l \tag{31}$$

where $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the active and reactive power injection vectors, respectively; and f_{AGGRE5}^{e} and f_{AGGRE5}^{f} are the physics law-informed node feature aggregation functions.

Eventually, using this aggregation function (29), a new GCNN preserved the physics law can be obtained as:

$$\begin{cases} \boldsymbol{e}^{l+1}=f_{CONV}\left(f_{AGGRE5}^{e}\left(\boldsymbol{e}^l,\boldsymbol{f}^l\right),\boldsymbol{H}_e\right) \\ \boldsymbol{f}^{l+1}=f_{CONV}\left(f_{AGGRE5}^{f}\left(\boldsymbol{e}^l,\boldsymbol{f}^l\right),\boldsymbol{H}_f\right) \end{cases} \tag{32}$$

where $\boldsymbol{H}_e$ and $\boldsymbol{H}_f$ are the trainable parameters in the convolution layer.

The physics law-informed GCNN directly incorporates the Gaussian-Seidel iteration process into the node feature aggregation mechanism. Unlike conventional methods that approximate or simplify power flow equations, it retains the structure of the original equations during the aggregation of features, thereby avoiding distortions of the underlying physics laws. Although fixed-step unrolling does not theoretically guarantee convergence, this design ensures that the aggregation process adheres to the power flow equations at each iteration step. Therefore, there are two benefits as follows. First, by embedding the Gaussian-Seidel-based aggregation, the model explicitly captures the interdependencies between neighboring nodes governed by power flow equations, enabling more accurate feature propagation and aggregation, which improves the ability to learn the load-shedding model that respects grid physics. Second, the direct integration of power flow constraints reduces the complexity of learning the mapping between grid states and the minimum load-shedding values.

C. Multi-kernel Collaborative GCNN

Different from convolution in matrices or images, graph convolution decomposes convolutions into aggregation and node convolution operations to handle the node degree disunity in a graph. However, the design of fixed parameters in aggregation operations leads to the limitation of the feature extraction capability of graph neural networks. Recent research works mainly discover an aggregation operation method that can effectively extract graph features and further add attention mechanisms to weigh the node features [32]. The aforementioned aggregation operations and GCNNs are gradually improved to extract the topology and physical features in the power systems. But those GCNNs still exploit a single aggregation operation, which results in extracting features from neighborhoods primarily focusing on a fixed form. It is difficult for neural networks to accurately approximate the complex nonlinear and non-convex mapping of load-shedding problems using such extracted features in a fixed form. Hence, the feature extraction ability of the aggregation operation in GCNNs needs to be further improved. To this end, this paper proposes an architecture of multi-kernel collaborative GCNN, as shown in Fig. 2, where σ is the activation function.

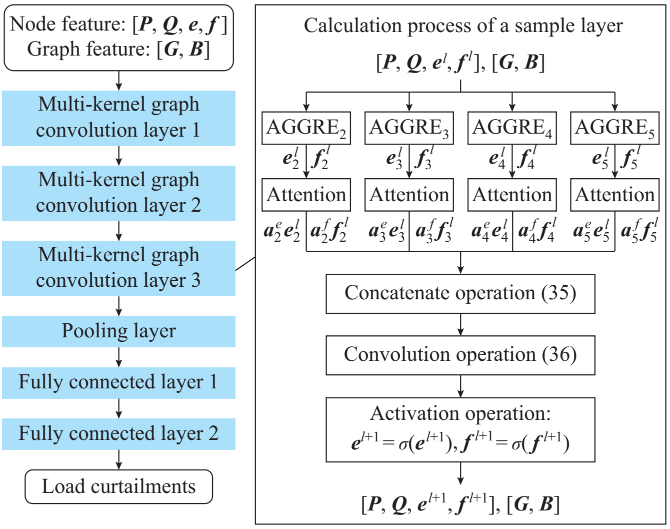
Node feature: [$\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{e}$, $\boldsymbol{f}$]
Graph feature: [$\boldsymbol{G}$, $\boldsymbol{B}$]
Multi-kernel graph convolution layer 1
Multi-kernel graph convolution layer 2
Multi-kernel graph convolution layer 3
Pooling layer
Fully connected layer 1
Fully connected layer 2
Load curtailments
Calculation process of a sample layer
[$\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{e}^l$, $\boldsymbol{f}^l$], [$\boldsymbol{G}$, $\boldsymbol{B}$]
AGGRE$_2$
AGGRE$_3$
AGGRE$_4$
AGGRE$_5$
$\boldsymbol{e}_2^l$ $\boldsymbol{f}_2^l$
$\boldsymbol{e}_3^l$ $\boldsymbol{f}_3^l$
$\boldsymbol{e}_4^l$ $\boldsymbol{f}_4^l$
$\boldsymbol{e}_5^l$ $\boldsymbol{f}_5^l$
Attention
Attention
Attention
Attention
$\boldsymbol{a}_2^e\boldsymbol{e}_2^l$ $\boldsymbol{a}_2^f\boldsymbol{f}_2^l$
$\boldsymbol{a}_3^e\boldsymbol{e}_3^l$ $\boldsymbol{a}_3^f\boldsymbol{f}_3^l$
$\boldsymbol{a}_4^e\boldsymbol{e}_4^l$ $\boldsymbol{a}_4^f\boldsymbol{f}_4^l$
$\boldsymbol{a}_5^e\boldsymbol{e}_5^l$ $\boldsymbol{a}_5^f\boldsymbol{f}_5^l$
Concatenate operation (35)
Convolution operation (36)
Activation operation:
$\boldsymbol{e}^{l+1}=\sigma(\boldsymbol{e}^{l+1})$, $\boldsymbol{f}^{l+1}=\sigma(\boldsymbol{f}^{l+1})$
[$\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{e}^{l+1}$, $\boldsymbol{f}^{l+1}$], [$\boldsymbol{G}$, $\boldsymbol{B}$]

Fig. 2. Architecture of multi-kernel collaborative GCNN.

Specifically, there are two improvements in the multi-kernel collaborative GCNN. One is that the multiple aforementioned aggregation operations, e.g., (15), (16), (18), and (29), are utilized in the graph convolution layer to break feature extraction limitations with a single aggregation method. The other is that a self-attention mechanism is integrated into the multi-kernel graph convolution layer to determine the attention for extracted features by different aggregation operation methods. Regarding $\boldsymbol{e}^l,\boldsymbol{f}^l \in \mathbf{R}^{n\times k}$ as the coupled features of all node inputs to the l^{th} graph convolution layer, the aggregation features $\boldsymbol{e}_s^l,\boldsymbol{f}_s^l \in \mathbf{R}^{n\times k}$ of the s type of aggregation methods can be indicated by:

$$\begin{cases} \boldsymbol{e}_s^l = f_{AGGREs}^e\left(\boldsymbol{e}^l,\boldsymbol{f}^l\right) \\ \boldsymbol{f}_s^l = f_{AGGREs}^f\left(\boldsymbol{e}^l,\boldsymbol{f}^l\right) \end{cases} \tag{33}$$

In this paper, the aggregation methods in (15), (16), (18), and (29) are integrated in the multi-kernel collaborative GCNN, and therefore $s=\{2,3,4,5\}$. In (15) and (16), the coupled relationship of node representations $\boldsymbol{e}$ and $\boldsymbol{f}$ is not considered and therefore $\boldsymbol{e}^l$ and $\boldsymbol{f}^l$ are used individually to calculate the aggregated node representations $\boldsymbol{e}_s^l$ and $\boldsymbol{f}_s^l$. Since different aggregation methods extract different types of features, concatenating them together can enrich representations of node features and further improve prediction accuracy. However, uniformly concatenating the node representations $\boldsymbol{e}_s^l$ and $\boldsymbol{f}_s^l$ calculated by different aggregation methods may add redundant or unimportant features, resulting in important features that affect prediction accuracy not being focused on. To this end, a weight (observed as a kind of attention) is multiplied by $\boldsymbol{e}_s^l$ and $\boldsymbol{f}_s^l$ to scale the feature values. If the aggregated node representations have a significant impact on the output of neural networks, the weight could be large. Otherwise, it could be a small value. Different from the attention mechanism in [32] and [33], we employ a global pooling layer and a fully connected layer to determine it, denoting it as (34). It is noted that there are two fully connected layers used for $\boldsymbol{e}$ and $\boldsymbol{f}$, respectively.

$$\begin{cases} \boldsymbol{a}_s^e = f_{fc}^e\left(f_{pooling}\left(\boldsymbol{e}_s^l\right)\right) \\ \boldsymbol{a}_s^f = f_{fc}^f\left(f_{pooling}\left(\boldsymbol{f}_s^l\right)\right) \end{cases} \tag{34}$$

where $\boldsymbol{a}_s^e$ and $\boldsymbol{a}_s^f$ are the attentions of the node features in the s type of aggregations; $f_{pooling}(\cdot)$ is the global pooling calculation and its output dimension is $1\times k$; and $f_{fc}^e(\cdot)$ and $f_{fc}^f(\cdot)$ represent two fully connected layers and the output dimension is 1×1.

Eventually, the aggregation operation of the coupled feature and the forward propagation function of neural networks in the l^{th} graph convolution layer can be represented by (35) and (36). Combining the pooling layer and fully connected layers, the complete architecture of multi-kernel collaborative GCNN can be constructed.

$$\begin{cases} \boldsymbol{e}^{l'} = \boldsymbol{a}_2^e\boldsymbol{e}_2^l \| \boldsymbol{a}_3^e\boldsymbol{e}_3^l \| \boldsymbol{a}_4^e\boldsymbol{e}_4^l \| \boldsymbol{a}_5^e\boldsymbol{e}_5^l \\ \boldsymbol{f}^{l'} = \boldsymbol{a}_2^f\boldsymbol{f}_2^l \| \boldsymbol{a}_3^f\boldsymbol{f}_3^l \| \boldsymbol{a}_4^f\boldsymbol{f}_4^l \| \boldsymbol{a}_5^f\boldsymbol{f}_5^l \end{cases} \tag{35}$$

$$\begin{cases} \boldsymbol{e}^{l+1} = f_{CONV}\left(\boldsymbol{e}^{l'},\boldsymbol{H}_e\right) \\ \boldsymbol{f}^{l+1} = f_{CONV}\left(\boldsymbol{f}^{l'},\boldsymbol{H}_f\right) \end{cases} \tag{36}$$

where $\|$ denotes the concatenation of different features; and $\boldsymbol{e}^{l'}$ and $\boldsymbol{f}^{l'}$ are the aggregated coupled node features.

In Fig. 2, regarding node feature representation, $\boldsymbol{P}$ and $\boldsymbol{Q}$ are the system states, which change with the sampling of system states during the ORA process. For node voltages $\boldsymbol{e}$ and $\boldsymbol{f}$, we give 1 and 0 in the initial state. In the forward propagation of the neural network layer by layer, they change with the distribution of power injections. The core of the multi-kernel collaborative GCNN is to accurately extract the feature representation of node voltages $\boldsymbol{e}$ and $\boldsymbol{f}$ at the injected power, which can be used as a basis for load-shedding judgment.

IV. Inherent Pattern-Guided Learning Strategy

A. Inherent Pattern Derivation

The load-shedding problem is to reduce the minimum loads so that all the operational constraints are not violated. The outputs are highly relative to the loads. Is there a load perturbation strategy that can directly calculate the load curtailment without changing other optimization variables under the KKT condition? Intuitively, if the load needs to be reduced on a certain node, it means that the node is overloaded. If we continue to increase the load on this node, the increased load will also need to be reduced. If the load curtailment on the node is not sufficient, it does not need to be reduced. This rule can be written as:

$$C_{\mathrm{P}i}+\Delta_{\mathrm{P}i}=f_i\left(P_{\mathrm{D}i}+\Delta_{\mathrm{P}i},\dots\right)\quad C_{\mathrm{P}i}>0, C_{\mathrm{P}i}+\Delta_{Pi}>0 \tag{37}$$

where $\Delta_{\mathrm{P}i}$ is the load changing amount on node i; and $f_i(\cdot)$ is the function of the load-shedding model from input features to load curtailments on node i. According to the optimality and feasibility analysis, the above pattern can be directly proved. The details are shown in the Supplementary Material A. Under this circumstance, the load curtailment can be directly calculated, and if this pattern is applied in the training, neural network learning would not be limited to the given training data.

B. Analysis of Inherent Pattern-guided Learning Strategy

The learning of neural network is to optimize the trainable parameter. The typical learning is the supervised learning with sample labels, which minimizes the loss functions L, as shown in (38). However, it demands a lot of samples so that the neural network can be trained over the entire feature space. In data-driven load-shedding computation, due to complicated non-linearity and the combinatorial explosion of topology, training a load-shedding calculation neural network would require hundreds of thousands or even more samples. We construct a new learning strategy that can automatically add perturbed samples and reduce the training sample demanded.

$$L=\frac{1}{M}\sum_{k=1}^{M}\sum_{i=1}^{N}\left(C_{\mathrm{P,Out},i}^{k}-C_{\mathrm{P},i}^{k}\right)^2 \tag{38}$$

where $C_{\mathrm{P},i}^{k}$ and $C_{\mathrm{P,Out},i}^{k}$ are the i^{th} outputs of neural networks and the corresponding labels for the k^{th} sample, respectively; M is the training data number; and N is the output dimension.

According to the inherent pattern of the load-shedding model, a perturbation can be added to the loads and their curtailments, and the optimality and feasibility do not change if the initial load curtailment is not zero. Thus, we add perturbations to all the samples with nonzero load curtailment to generate more training data. The loss function can be rewritten as (39), and $\Delta_{\mathrm{P},i}$ is in $(-1,1]$.

$$L=\frac{1}{M}\sum_{k=1}^{M}\sum_{i}^{N}\left(C_{\mathrm{P,Out},i}^{k}-C_{\mathrm{P},i}^{k}\right)^2+\frac{\mu}{M}\sum_{k=1}^{M}\sum_{i}^{N}\left[\hat{C}_{\mathrm{P,Out},i}^{k}-C_{\mathrm{P},i}^{k}\odot\left(1+\Delta_{\mathrm{P}i}\right)\right]^2 \tag{39}$$

where $\hat{C}_{\mathrm{P,Out},i}^{k}$ is the load curtailment of the i^{th} output of neural networks when the load is input; and μ is the weight and set to be 0.1.

According to the gradient analysis shown in the Supplementary Material A, this new loss function can explore a wider feature space and thereby alleviate the imbalance with the load-shedding sample. Eventually, the inherent pattern-guided learning strategy is formulated as shown in Supplementary Material A Algorithm SA1, which generates the perturbation samples first and then trains the neural networks with the original samples from the training dataset.

V. Case Study

In this section, the proposed ORA method is implemented in several power systems to demonstrate its effectiveness, considering varying topologies.

A. Case Setting

The ORA in power systems usually involves load fluctuations, stochastic renewable energy generations, and outages of vital power equipment. As for load fluctuations, we assume they follow normal distribution with the default value as the means and 0.3 as the standard deviation. Wind power and photovoltaic power are considered in the simulations. We assume the wind speed and the solar irradiance follow the Wei-bull distribution and Beta distribution, respectively. The details of the distribution are identical to [28]. We integrate several wind farms and photovoltaic stations in different buses so that the penetration of renewable energy is more than 20%. For instance, in the IEEE 39-bus system, four wind farms with 200 MW capacity and four photovoltaic stations with 200 MW capacity are randomly integrated into the power system. The penetration of renewable energy is 20.26%. To reduce the imbalance of training data, we sample the $N-1$ and $N-2$ contingencies to cover all situations for training. At the assessment stage, we utilize a 1% failure probability for the reliability assessment. 20000 and 10000 samples are generated for training and testing.

To demonstrate the effectiveness of the proposed ORA method, the conventional ORA method (M0) and nine data-driven ORA methods (M1-M9) are compared, as shown in Table I.

Those methods and relative descriptions are introduced in the following. All neural networks are constructed and trained in the PyTorch framework on a PC with Intel$^{(\mathrm{R})}$ Core$^{(\mathrm{TM})}$ i7-10700K CPU @ 3.80 GHz, 16 GB RAM, and NIVIDIA GeForce RTX 2080Ti. The Adam optimizer is used with a 0.001 learning rate.

1) M0: conventional ORA method uses the interior point method to solve the load-shedding problem, which is the benchmark of data-driven ORA method.

2) M1: the data-driven ORA method exploits the typical GCNN to solve the load-shedding problems, whose graph convolution kernel is derived from the Laplacian matrix following the spectral graph theory [25].

3) M2: it is identical to M1 except for the graph convolution kernel. The impedance-based Gaussian kernel is utilized as the graph convolution kernel [26].

4) M3: it is identical to M1, but the power flow model-embedded graph convolution kernel is exploited [31].

5) M4: it is identical to M1, but the proposed physics law-

informed graph convolution kernel is utilized.

6) M5: it is identical to M1, but the design of multi-kernel collaboration GCNN is applied to the graph convolution layer.

7) M6: it has the identical neural network structure and graph convolution method, but the proposed inherent pattern-guided learning strategy is applied.

8) M7: the existing data-driven ORA method uses the tacked denoising auto-encoder to solve the load-shedding problem [11].

9) M8: the existing data-driven ORA method uses deep neural networks to filter the normal system states [16].

10) M9: the data-driven ORA method uses convolution neural networks to calculate the load-shedding problem [15].

TABLE I
PURPOSE OF COMPARISON METHODS

Comparison method	Purpose
M1, M2, M3 M4, M5	Demonstrate the effectiveness of multi-kernel collaborative GCNN
M5, M6	Demonstrate the effectiveness of inherent pattern-guided learning
M0, M6, M7 M8, M9	Compare the assessment accuracy and speed of different ORA methods

The mean absolute errors *MAE* and the related errors e_{related} are utilized to measure the performance of different methods. The mathematical formulations are:

$$MAE = \frac{1}{N_D}\sum_{i=1}^{N_D}\left|y-y'\right| \tag{40}$$

$$e_{\text{related}} = \frac{\left|y'-y\right|}{y_{\text{base}}} \tag{41}$$

where y is the load curtailment calculated by solving the load-shedding model using the numerical method; y' is the prediction load curtailment using neural networks; N_D is the number of testing data; and y_{base} is the reference value. As for operational reliability indicators, the accurate value evaluated by M0 is the reference value.

B. Validation of Proposed Multi-kernel Collaborative GCNN

M1-M5 are simulated and compared in this subsection to demonstrate the effectiveness of the design of multi-kernel collaborative GCNN. First, we implement M1-M5 in the IEEE 39-bus system. The δ in M4 and M5 is set to be 0.1. The five methods are trained for 2000 epochs, and the mean absolute errors of training data and testing data in different epochs are shown in Figs. 3 and 4, respectively. It can be observed that M1 and M2 converge in several epochs but with the largest mean validation errors. The convergences are limited. Fortunately, in M3, the power flow model-embedded design effectively reduces the training error of neural networks. When changing M3 to M4, the training and testing errors could further decrease. The *K*-fold cross-validation and related results are introduced in the Supplementary Material A, which shows that M5 can obtain the smallest errors and standard deviations. Besides, to understand why M5 can achieve the best results, the attention analysis and the computational complexity analysis are also included in the Supplementary Material A.

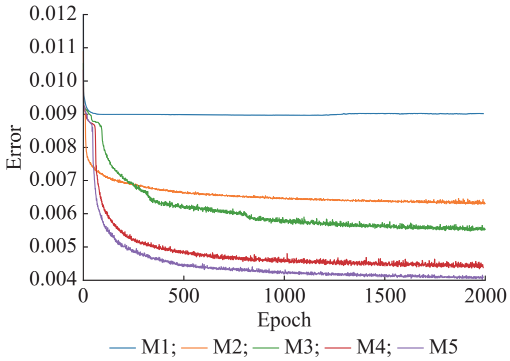

Fig. 3. Mean absolute errors of training data when different methods are tested in training process.

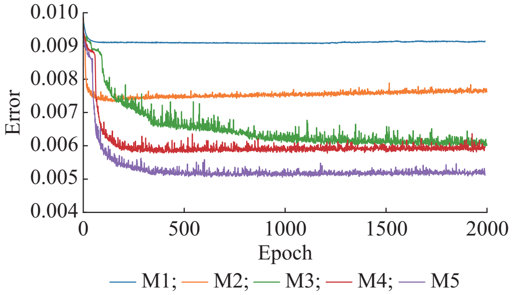

Fig. 4. Mean absolute errors of testing data when different methods are tested in training process.

To demonstrate the effectiveness of the inherent pattern-guided learning strategy, M5 and M6 are implemented with different numbers of training samples. After training for 1000 epochs with a learning rate of 0.001 using the Adam optimizer, all the neural networks are tested in 10000 testing samples, and the corresponding mean absolute errors of load curtailments in M5 and M6 are plotted in Fig. 5. It can be observed that the testing errors of M6 are always smaller than those of M5 under different numbers of training data. Besides, with the decrease in training samples, the error of M5 increases dramatically, especially when the amount of training data is less than 6000. On the contrary, the errors in M6 increase slowly and only begin to rise when the amount of data is less than 1000. The reason is that the inherent pattern-guided learning strategy is not limited to the given training data, and it can exploit those known samples to generate more load-shedding samples. It can not only overcome the unbalanced problem of load-shedding samples but also reduce the dependence of neural networks on large numbers of training samples. The effectiveness of the inherent pattern-guided learning strategy is verified.

Further, the trained neural networks in M1-M6 are used to evaluate the operational reliability of the IEEE 39-bus system, where the contingency probability of vital equipment is set to be 1%. The maximum iteration number is 30000. The corresponding reliability indices and relative errors of M0-M6 are listed in Table II. It can be observed that M1, with the typical aggregation method derived based on the adjacent matrix, has the largest assessment errors E_{EDNS}.

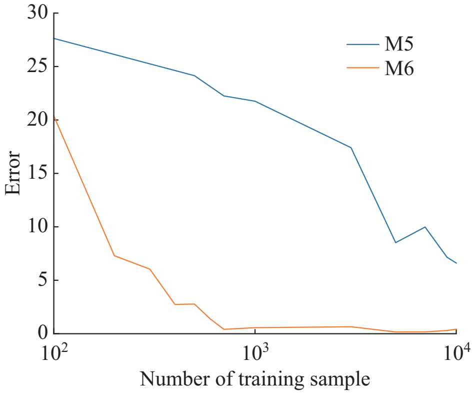

Fig. 5. Mean absolute errors of load curtailments for neural networks trained with different numbers of training sample.

When gradually constructing the aggregation method informed by the physics variables, e. g., line impedance, and physics model, the assessment errors decrease, but errors of PLC E_{PLC} are still larger than 10%, which is mainly caused by the unbalanced load-shedding samples. Fortunately, the inherent pattern-guided learning strategy can faster expand the load-shedding samples during the training process. It is helpful to overcome the effects of unbalanced training data. As a result, E_{PLC} is further reduced to 4.82%, and the effectiveness of the proposed multi-kernel collaborative GCNN is demonstrated. Considering the small difference in M5 and M6, additional paired t-tests and Wilcoxon signed-rank tests are implemented to establish the statistical difference in predicting MAEs of load curtailments. The statistical tests yield vanishingly small p-values (effectively zero), strongly rejecting the null hypothesis that the differences between M5 and M6 are due to random chance. Furthermore, the 95% confidence intervals of the MAEs for M5 and M6 are (0.0022, 0.0025) and (0.0015, 0.0018), respectively. The non-overlapping provides additional evidence that the observed differences are statistically significant.

TABLE II
RELIABILITY INDICES AND RELATIVE ERRORS OF M0-M6 IN IEEE 39-BUS SYSTEM

Method	PLC	E_{PLC} (%)	EDNS (MW)	E_{EDNS} (%)
M0	0.04479		9.900	
M1	0.01912	57.31	3.098	68.71
M2	0.03211	28.30	7.397	25.28
M3	0.03356	25.06	7.142	27.85
M4	0.03551	20.71	7.923	19.96
M5	0.04932	10.11	10.158	2.61
M6	0.04263	4.82	10.035	1.36

In conclusion, the proposed multi-kernel collaborative GCNN can enhance the feature extraction ability of neural networks so that their convergences and robustness can outperform other GCNNs. Besides, the inherent pattern-guided learning strategy can expand the load-shedding sample in the training process and explore a broader feature space. Therefore, the scale of training data can be reduced, and operational reliability can be evaluated to achieve higher accuracy.

C. Comparison with Existing Data-driven ORA Methods

To further demonstrate the effectiveness of the proposed ORA method, it is compared with three data-driven ORA methods in four power systems. There are 20000 samples generated for training, and the number of assessment system states is 30000. The assessment results of different methods in different power systems are listed in Table III.

TABLE III
RELIABILITY RESULTS OF DIFFERENT METHODS IN DIFFERENT POWER SYSTEMS

System	Method	PLC	E_{PLC} (%)	EDNS (MW)	E_{EDNS} (%)	Assessment time (s)
IEEE 39-bus system	M0	0.04479		9.900		353.00
	M6	0.04260	4.82	10.035	1.36	1.63
	M7	0.03090	30.98	4.967	49.85	0.55
	M8	0.03670	16.62	9.039	8.74	241.00
	M9	0.03590	18.39	2.073	79.06	1.23
IEEE 57-bus system	M0	0.25495		4.162		819.00
	M6	0.25106	1.42	4.136	0.81	3.17
	M7	0.19560	23.25	2.303	44.65	0.55
	M8	0.19580	23.20	4.008	3.70	612.33
	M9	0.17410	31.69	2.643	36.48	1.75
IEEE 118-bus system	M0	0.09970		2.712		571.00
	M6	0.10056	0.64	2.714	0.07	2.02
	M7	0.12623	25.51	2.345	13.53	0.36
	M8	0.09410	5.72	2.331	14.05	388.83
	M9	0.08210	17.68	2.079	23.34	0.54
Provincial 661-bus system	M0	0.36710		363.340		35472.50
	M6	0.38530	4.96	366.430	0.85	46.47
	M7	0.44960	22.45	167.410	53.92	0.86
	M8	0.33294	9.32	302.830	16.65	24659.20
	M9	0.17400	52.50	176.770	51.35	2.33

As for the benchmark method M0, there are five threads utilized to solve the load-shedding problem with different system states. All the neural networks in M6-M9 are implemented on a GPU. It can be observed that M6 outperforms M7-M9 in the simulations where 1% failure probability of power lines and generators is used. In the IEEE 39-bus system, the benchmarks of PLC and EDNS are 0.04479 and 9.90 MW, respectively, and the relative errors of M6 are less than 5%, which is 4.82% for E_{PLC} and 1.36% for E_{EDNS}, respectively. However, in other data-driven ORA methods, the relative errors are all larger than 15%. Under this circumstance, the topology changes in different system states, and the mapping of the load-shedding problem becomes discrete and complicated. Besides, abnormal system states are much fewer than the normal ones, even in the training data where all the $N-1$ contingencies of power lines and generators are included. But M6 can still achieve a relative error of less than 5%. In the IEEE 57-bus and 118-bus systems, PLC increases, which means there are more load-shedding samples in the training data, and the training data are less unbalanced. E_{PLC} of M6 is decreased to 1.42%. But in the provincial 661-bus system, E_{PLC} increases due to the complexity of power systems, and M6 is still the best one compared with other methods. Therefore, M6 with the multi-kernel collaborative GCNN and inherent pattern-guided learning strategy can calculate more accurate results and achieve a relative error of less than 5% under the varying topology circumstances, which outperforms many other existing methods.

Regarding the assessment time, M6, M7, and M9 can complete the reliability assessment within a few seconds in the IEEE testing system. However, M0 and M8 require several hundred seconds. As the size of the power system increases, it would take even more time for M0 and M8. Specifically, in the provincial 661-bus system, it takes 35472 s to complete the reliability assessment. The longer duration for solving load-shedding issues is attributed to the larger scale of the power system. Nevertheless, M6 still manages to evaluate within a mere 46 s, demonstrating the efficiency of the proposed multi-kernel collaborative GCNN.

In the case study, we primarily utilize power line and $N-1$ and $N-2$ contingency data of the unit for training the neural network. However, actual scenarios involve more intricate failures. The graph neural network is compatible with changes in topology, and the graph convolution method with physics model embedded design can better extract physical features and promote the learning of neural networks. Moreover, employing a learning method guided by inherent patterns can further enhance the performance of neural networks. Therefore, training the neural network with various complex scene data in reality can make it more adaptable to the operation scenarios of the power grid and realize accurate online ORA.

VI. Discussion

The proposed multi-kernel collaborative GCNN aims to enhance the adaptability of existing data-driven ORA methods to complex topology changes. It integrates conventional aggregation methods with physics law-informed aggregation methods to improve feature extraction capabilities. Additionally, an inherent pattern-guided learning strategy is constructed, leveraging the consistent properties of the load-shedding model. The proposed multi-kernel collaborative GCNN can consider a broader range of contingencies and achieve improved assessment accuracy.

However, when applying the proposed multi-kernel collaborative GCNN in practice, there are still several challenges and limitations. Even though the proposed inherent pattern-guided learning strategy can reduce the training data demanded, it still requires a basic training dataset. How to efficiently generate training data or generalize it to a sufficient and representative base dataset also challenges the data-driven ORA application. Additionally, the utilization of multiple kernels and the associated self-attention mechanism adds to the computational cost. Even though the sparse technique is applied, advanced computers or servers are still required to complete the training for large-scale power systems. It is crucial to quantify the contribution of each kernel to the overall performance improvement relative to the computational cost and finally simplify the graph convolution calculation. Thus, further research is needed to simplify the structure of neural networks and realize the actual application of practical engineering.

VII. Conclusion

This paper proposes a multi-kernel collaborative GCNN for ORA considering varying topologies. There are three main contributions in the paper. First, a physics law-informed graph convolution kernel rooted in the Gaussian-Seidel iteration is derived, which can effectively aggregate node features under varying topologies. Second, a multi-kernel GCNN is constructed by integrating other advanced graph convolution kernels with a novel self-attention mechanism. It can extract different features and build a representative node feature vector for high-precision assessment of reliability. Last, to further enhance the robustness of GCNN, the inherent pattern of the load-shedding model is derived and used to construct the specialized supervised loss function so that a wider feature space can be explored by neural networks. According to the simulation results in different power systems, the GCNN with a multi-kernel design and specialized supervised exploration is capable of considering a broader range of contingencies, achieving topology change adaptation and improved assessment accuracy, which outperforms the existing data-driven ORA methods.

References

[1] R. Billinton and W. Li, *Reliability Assessment of Electric Power Systems Using Monte Carlo Methods.* New York: Springer Science & Business Media, 2013.

[2] A. F. Zobaa and S. A. Aleem, *Uncertainties in Modern Power Systems.* Cambridge: Academic Press, 2020.

[3] Z. Dong, B. Li, J. Li *et al*., "Online reliability assessment of energy systems based on a high-order extended-state-observer with application to nuclear reactors," *Renewable and Sustainable Energy Reviews*, vol. 158, p. 112159, Apr. 2022.

[4] A. Tabares, G. Munoz-Delgado, J. F. Franco *et al*., "An enhanced algebraic approach for the analytical reliability assessment of distribution systems," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 2870-2879, Jul. 2019.

[5] Z. Li, W. Wu, B. Zhang *et al*., "Analytical reliability assessment method for complex distribution networks considering post-fault network reconfiguration," *IEEE Transactions on Power Systems*, vol. 35, no. 2, pp. 1457-1467, Mar. 2020.

[6] R. A. Gonzalez-Fernandez, A. M. L. da Silva, L. C. Resende *et al*., "Composite systems reliability evaluation based on Monte Carlo simulation and cross-entropy methods," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 598-606, Nov. 2013.

[7] Z. Zhu and X. Du, "Reliability analysis with Monte Carlo simulation and dependent kriging predictions," *Journal of Mechanical Design*, vol. 138, no. 12, p. 121403, Dec. 2016.

[8] D. Urgun and C. Singh, "A hybrid Monte Carlo simulation and multi-label classification method for composite system reliability evaluation," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 908-917, Mar. 2019.

[9] A. M. L. da Silva, L. C. de Resende, L. A. da F. Manso *et al*., "Composite reliability assessment based on monte carlo simulation and artificial neural networks," *IEEE Transactions on Power Systems*, vol. 22, no. 3, pp. 1202-1209, Aug. 2007.

[10] M. Kamruzzaman, N. Bhusal, and M. Benidris, "A convolutional neural network-based approach to composite power system reliability evaluation," *International Journal of Electrical Power & Energy Systems*, vol. 135, p. 107468, Feb. 2022.

[11] Z. Dong, K. Hou, H. Meng *et al*., "Data-driven power system reliability evaluation based on stacked denoising auto-encoders," *Energy Reports*, vol. 8, pp. 920-927, Apr. 2022.

[12] Y. Zhu and C. Singh, "Assessing bulk power system reliability by end-

to- end line maintenance-aware learning," *IEEE Access*, vol. 11, pp. 49639-49649, May 2023.

[13] Q. Zhang, J. Ling, W. Luo *et al*., "Policy-assisted graph reinforcement learning for real-time economic dispatch," *Journal of Modern Power Systems and Clean Energy*, vol. 13, no. 6, pp. 1896-1908, Nov. 2025.

[14] Y. Du, F. Li, J. Li *et al*., "Achieving 100x acceleration for $N-1$ contingency screening with uncertain scenarios using deep convolutional neural network," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3303-3305, Jul. 2019.

[15] Y. Zhou, W.-J. Lee, R. Diao *et al*., "Deep reinforcement learning based real-time ac optimal power flow considering uncertainties," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 5, pp. 1098-1109, Sept. 2022.

[16] Y. Yang, J. Yu, Z. Yang *et al*., "A trustable data-driven framework for composite system reliability evaluation," *IEEE Systems Journal*, vol. 16, no. 4, pp. 6697-6707, Dec. 2022.

[17] M. Xiang, J. Yu, Z. Yang *et al*., "Probabilistic power flow with topology changes based on deep neural network," *International Journal of Electrical Power & Energy Systems*, vol. 117, p. 105650, May 2020.

[18] A. Donon, F. Cubelier, E. Karangelos *et al*., "Topology-aware reinforcement learning for tertiary voltage control," *Electric Power Systems Research*, vol. 234, p. 110658, Sept. 2024.

[19] Y. Zhu, Y. Zhou, L. Yan *et al*., "Scaling graph neural networks for large-scale power systems analysis: empirical laws for emergent abilities," *IEEE Transactions on Power Systems*, vol. 39, no. 6, pp. 7445-7448, Nov. 2024.

[20] A. Owerko, F. Gama, and A. Ribeiro, "Unsupervised optimal power flow using graph neural networks," in *Proceedings of ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Soul, Korea, Apr. 2024, pp. 6885-6889.

[21] Q.-H. Ngo, B. L. Nguyen, T. V. Vu *et al*., "Physics-informed graphical neural network for power system state estimation," *Applied Energy*, vol. 358, p. 122602, Mar. 2024.

[22] R. Madbhavi, B. Natarajan, and B. Srinivasan, "Graph neural network-based distribution system state estimators," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 12, pp. 11630-11639, Dec. 2023.

[23] Z. Wang, X. Liu, Y. Huang *et al*., "A multivariate time series graph neural network for district heat load forecasting," *Energy*, vol. 278, p. 127911, Sept. 2023.

[24] Y. Huo, Z. Chen, Q. Li *et al*., "Graph neural network based column generation for energy management in networked microgrid," *Journal of Modern Power Systems and Clean Energy*, vol. 12, no. 5, pp. 1506-1519, Sept. 2024.

[25] C. Kim, K. Kim, P. Balaprakash *et al*., "Graph convolutional neural networks for optimal load shedding under line contingency," in *Proceedings of 2019 IEEE PES General Meeting*, Atlanta, USA, Aug. 2019, pp. 1-5.

[26] D. Owerko, F. Gama, and A. Ribeiro, "Optimal power flow using graph neural networks," in *Proceedings of ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, online conference, May 2020, pp. 5930-5934.

[27] D. Wang, K. Zheng, Q. Chen *et al*., "Probabilistic power flow solution with graph convolutional network," in *Proceedings of 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, Hague, Netherlands, Oct. 2020, pp. 650-654.

[28] M. Gao, J. Yu, Z. Yang, *et al*., "Physics embedded graph convolution neural network for power flow calculation considering uncertain injections and topology," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 11, pp. 15467-15478, Jul. 2024.

[29] K. Xu, W. Hu, J. Leskovec *et al.* (2018, Oct.). How powerful are graph neural networks? [Online]. Available: https://arxiv. org/abs/1810.00826v3

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proceedings of 5th International Conference on Learning Representations*, Toulon, France, Apr. 2017, pp. 1-6.

[31] M. Gao, J. Yu, Z. Yang *et al*., "A physics-guided graph convolution neural network for optimal power flow," *IEEE Transactions on Power Systems*, vol. 39, no. 1, pp. 380-390, Jan. 2024.

[32] P. Velickovic, G. Cucurull, A. Casanova *et al*., "Graph attention networks," in *Proceedings of 5th International Conference on Learning Representations*, Toulon, France, Apr. 2017, pp. 1-6.

[33] Vaswani, N. Shazeer, N. Parmar *et al*., "Attention is all you need," in *Proceedings of 31st Annual Conference on Neural Information Processing Systems (NIPS)*, Long Beach, USA, Dec. 2017, pp. 1-5.

Xinyu Liu received the M.S. degree in electrical engineering from Sichuan University, Chengdu, China, in 2002. Currently, he is pursuing the Ph.D. degree in electrical engineering at Tsinghua University, Beijing, China, and is employed by Chongqing Electric Power Company of State Grid Corporation of China, Chongqing, China. His research interests include optimal dispatch, stability analysis, and artificial intelligence application in power systems.

Maosheng Gao received the B.S. degree from the Chongqing University-University of Cincinnati Joint Co-op Institute, Chongqing, China, in 2020. Currently, he is pursuing the Ph.D. degree in electrical engineering at the School of Electrical Engineering, Chongqing University, Chongqing, China. His research interests include deep learning and its applications in power and energy systems.

Juan Yu received the Ph.D. degree in electrical engineering from Chongqing University, Chongqing, China, in 2007. Currently, she is a Full Professor at Chongqing University. Her research interests include big data analytics and power system analysis.

Zhifang Yang received the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2018. He is now a Full Professor at Chongqing University, Chongqing, China. His research interests include power system optimization and electricity market.

Wenyuan Li received the B.S. degree from Tsinghua University, Beijing, China, in 1968, and the M.S. and Ph.D. degrees from Chongqing University, Chongqing, China, in 1982 and 1987, respectively, all in electrical engineering. He is currently a Professor with Chongqing University. He is a Foreign Member of the Chinese Academy of Engineering and a Fellow of the Canadian Academy of Engineering. His research interests include power system planning, operation, optimization, and reliability assessment.