A Learning to Optimize Approach to Accelerating Distributed Optimal Power Flow Solving

Huihuang Cai, Huan Long, Zhi Wu, Wei Gu, and Jingtao Zhao

***Abstract*—As the scale of power system continues to grow, a fast and accurate distributed optimal power flow solver becomes crucial for the effective dispatch of power system. This paper presents a learning to optimize (L2O) approach to accelerating the distributed optimal power flow solving. The final convergence values of global variables and Lagrange multipliers of the alternating direction method of multipliers (ADMM) are estimated as its warm-start solution. A long short-term memory-variational auto-encoder (LSTM-VAE) model is developed as the core for estimating the convergence value, and the LSTM-VAE assisted ADMM is proposed. The LSTM generates low-dimensional representations of global variables and Lagrange multipliers, while the decoder part of VAE reconstructs the high-dimensional asymptotic convergence values. A novel loss function is designed in the form of a quadratic sum penalty term to incorporate the constraint violations of the Lagrange multipliers. Additionally, a two-stage training data generation strategy is proposed to efficiently generate substantial data within a limited amount of time. The effectiveness of the LSTM-VAE assisted ADMM is evaluated using the modified IEEE 123-bus system, a synthetic 500-bus system, and a 793-bus system.**

***Index Terms*—Optimal power flow, learning to optimize, alternating direction method of multipliers (ADMM), long short-term memory, variational auto-encoder.**

I. Introduction

OPTIMAL power flow (OPF) plays a fundamental role in modern power systems. OPF aims to optimize a specific objective of the power system by regulating the controllable variables while ensuring that the operational constraints are satisfied. Traditionally, centralized OPF has been widely utilized, as it achieves global optimization and ensures consistency and coordination across the power system [1]. However, with the increasing integration of renewable energy sources and the growing scale of power system, centralized OPF faces challenges in meeting the requirements for computational efficiency. As a result, distributed OPF has gained prominence. Unlike centralized OPF, distributed OPF divides the power system into multiple sub-areas and solves regional OPFs in parallel, thereby reducing the computational time. Furthermore, distributed OPF preserves the privacy by only exchanging partial information between sub-areas [2].

Constrained by limitations in information sharing, the consistency of boundary variables presents a significant challenge in distributed OPF. To address this issue, several alternating distributed algorithms have been proposed in recent years, including the auxiliary problem principle [3], the alternating direction method of multipliers (ADMM) [4], analytical target cascading [5], and the augmented Lagrangian alternating direction inexact Newton method [6]. Among these, ADMM has demonstrated superior performance. However, it still requires a large number of iterations to achieve consistency of boundary variables. To improve the efficiency of ADMM, various enhancements have been proposed, including adaptive parameter selection and modifications to consistency conditions [4], [7]. Despite these improvements, a gap remains between the current efficiency of ADMM and the stringent time requirements of scheduling the modern and complex power system.

Learning to optimize (L2O) [8] approach integrates machine learning (ML) with traditional optimization algorithms, and uses ML to accelerate the optimization process or estimate optimal solutions. The primary frameworks of L2O approaches are categorized into end-to-end and hybrid frameworks [9]. In the end-to-end framework, ML directly generates solutions, improving the computational efficiency. However, this framework cannot guarantee the solution feasibility [10]. In contrast, the hybrid framework leverages ML to enhance the performance of traditional optimization solvers. For example, [11] employs decision trees to provide a warm-start solution as the initial solution for the traditional optimization solver. Although the hybrid framework typically requires more time to reach the final optimal solution compared to the end-to-end framework, it can ensure both optimality and feasibility.

Recently, numerous studies have employed the hybrid framework of L2O to enhance the efficiency of ADMM, aiming to reduce computational time while ensuring feasibility [12]-[16]. One typical way is that the warm-start solution is directly estimated using global load information. For instance, [12] introduces a deep neural network (DNN) to estimate global variables and Lagrange multipliers as initial val-

Manuscript received: September 19, 2024; revised: December 24, 2024; accepted: February 21, 2025. Date of CrossCheck: February 21, 2025. Date of online publication: March 13, 2025.

This work was supported in part by Jiangsu Industry Outlook and Key Technology Research Project (No. BE2023093-2) and the Young Elite Scientists Sponsorship Program by CAST (No. 2023QNRC001).

H. Cai, H. Long (corresponding author), Z. Wu, and W. Gu are with School of Electrical Engineering, Southeast University, Nanjing 210018, China (e-mail: huihuangcai@seu.edu.cn; hlong@seu.edu.cn; zwu@seu.edu.cn; wgu@seu.edu.cn).

J. Zhao is with State Grid Electric Power Research Institute, Nanjing 211102, China (e-mail: zhaojingtao@sgepri.sgcc.com.cn).

DOI: 10.35833/MPCE.2024.001036

ues for ADMM using load data as input. However, since each sub-area provides all of its load information, this approach not only significantly increases data exchange between sub-areas and the central controller, but also poses a serious risk to the privacy of sub-areas.

Another way generates the warm-start solution by capturing the periodic iteration characteristics of ADMM. Unlike the previously mentioned way, it only requires the boundary information of sub-areas, thereby protecting their privacy. Reference [15] explores the dynamics of ADMM convergence, demonstrating that the acceleration of ADMM could be formulated as a time-series regression problem. Consequently, the recurrent neural network (RNN), an effective tool for capturing periodic features, has emerged as a promising approach for estimating global variables and Lagrange multipliers. Reference [13] uses the gated recurrent unit (GRU) to estimate global variables and Lagrange multipliers as the warm-start solution for ADMM. Long short-term memory (LSTM) is introduced to estimate the warm-start solution for ADMM in [15]. Studies in [13], [15], and [16] show that RNNs are an effective approach for accelerating ADMM. However, the high dimensionality of global and dual variables in complex distributed networks presents a significant challenge to RNNs.

Furthermore, to ensure the convergence of ADMM, the values of the Lagrange multipliers are constrained during the iterative process. However, since the warm-start solution is estimated by a neural network, it often fails to satisfy these constraints, which can significantly hinder the convergence of ADMM. Despite the importance of this issue, it has received limited attention in prior research. To address convergence problems arising from constraint violations, [14] explores the constraints on dual variable values and proposes a strategy to estimate a subset of variables ensuring that these constraints are satisfied. However, this strategy relies on manual selection, and the acceleration effect is highly dependent on the choice of dual variables. To overcome these limitations, this paper designs a loss function for the model training, which ensures that the values of the Lagrange multipliers satisfy the constraints without requiring manual selection. Additionally, regardless of the approaches in [13], [14], and [16] to accelerating ADMM, a large amount of data are usually required. The dataset is typically generated by executing the traditional ADMM multiple times, which is time-consuming. Therefore, a two-stage training data generation strategy is proposed to reduce the time required for generating training data.

This paper aims to utilize an L2O approach to accelerating the distributed OPF solving. The final convergence values of global variables and Lagrange multipliers are estimated, which serve as the warm-start solution for ADMM. An LSTM-variational auto-encoder (LSTM-VAE) model is proposed as the core model to estimate the convergence value. A novel loss function is designed in the form of a quadratic sum penalty term to incorporate the constraint violations of the Lagrange multipliers. Additionally, a two-stage training data generation strategy is proposed to efficiently generate substantial data within a constrained timeframe. The main contributions of this paper are summarized as follows.

1) An L2O approach is employed to estimate the convergence values of Lagrange multipliers and global variables. The LSTM part processes high-dimensional temporal characteristics of global variables and Lagrange multipliers, extracting their latent temporal patterns to generate low-dimensional representations. Subsequently, the decoder part of VAE nonlinearly maps these compressed latent vectors back to the high-dimensional space, reconstructing the asymptotic convergence values of ADMM variables. These estimated values are subsequently used as the warm-start solution for ADMM, accelerating its iteration process.

2) A loss function is designed to guide the training of the LSTM-VAE model to satisfy the constraints of the Lagrange multipliers. The loss function consists of two components: accuracy error and penalty error. The mean squared error (MSE) serves as the accuracy error to ensure the precision of the estimated values. The penalty error, in the form of a quadratic sum, is developed to measure the violations of the constraints of the Lagrange multipliers across adjacent sub-areas.

3) The performance of learning-based approaches relies heavily on the training data, and generating these data can be time-consuming. To address potential computational resource limitations during offline training, a two-stage training data generation strategy is proposed to reduce the time required for generating training data for the LSTM-VAE model. In the first stage, traditional ADMM is executed to generate a low-precision dataset, which is then used to train a low-precision LSTM-VAE$^{(l)}$ model. In the second stage, the LSTM-VAE$^{(l)}$ model is employed to facilitate the generation of a high-precision training dataset, which is used to train the high-precision LSTM-VAE$^{(h)}$ model. The LSTM-VAE assisted ADMM is proposed, which demonstrates superior suitability for real-world application in large-scale power system.

The remainder of this paper is organized as follows. Section II introduces the details of distributed OPF with ADMM. Section III presents the LSTM-VAE assisted ADMM. Section IV reports the case study and discusses the numerical results. Finally, Section V gives the conclusion of this paper.

II. Distributed OPF with ADMM

In distributed OPF, the distribution network is first divided into sub-areas based on the principle of high intra-area connectivity and low inter-area connectivity. And then, the distributed optimization algorithm, ADMM, is applied to solve the distributed OPF.

A. Formulation of Distributed OPF

Considering the physical characteristics of the distribution network topology, the electrical coupling strength e_{ij} between node i and node j is utilized to assess the connectivity between the nodes [17]. $\boldsymbol{E}$ is a symmetry matrix that contains a set of elements e_{ij}.

To partition a distribution network, it is treated as a graph. Community detection, an approach used in graph theory to

identify clusters of nodes that are densely connected, is then applied to the sub-division with high connectivity, making it effective for partitioning the distribution network [18]. Based on community detection, the modularity Q is formulated as (1). The objective for partitioning the distribution network is to maximize the modularity Q.

$$\max Q = \frac{1}{2m}\sum_{e_{ij}\in \boldsymbol{E}}\left(e_{ij}-\frac{d_i d_j}{2m}\right)\delta(C_i, C_j) \tag{1}$$

$$m = \frac{1}{2}\sum_{e_{ij}\in \boldsymbol{E}} e_{ij} \tag{2a}$$

$$\delta(C_i, C_j) = \begin{cases} 0 & C_i = C_j \\ 1 & \text{otherwise} \end{cases} \tag{2b}$$

where d_i is the sum of the elements in the i^{th} row of matrix $\boldsymbol{E}$; and C_i is the community to which node i is assigned.

The power system is partitioned into several sub-areas according to (1) and (2), as shown in Fig. 1, where P_{ij} and Q_{ij} are the active and reactive power in line (ij), respectively. The auxiliary nodes i' and j', along with the auxiliary lines (ij') and ($i'j$), are introduced to decompose the adjacent sub-areas.

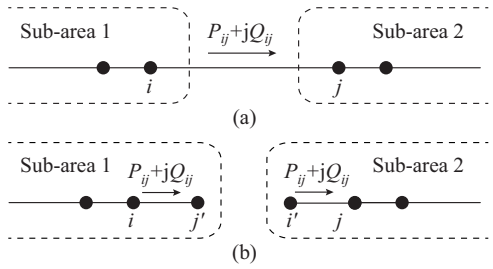

Fig. 1. Decomposition of adjacent sub-areas. (a) Boundary of adjacent sub-areas. (b) Boundary of decomposed sub-areas.

The squares of the boundary node voltages V_i, $V_{i'}$, V_j, $V_{j'}$, active power $P_{ij'}$, $P_{i'j}$, reactive power $Q_{ij'}$, $Q_{i'j}$, and the squares of boundary line currents $l_{ij'}$, $l_{i'j}$ are defined as boundary variables. To confirm the feasibility of the solution, the consistency constraints of boundary variables in (3)-(6) are introduced.

$$\begin{cases} V_i = V_{i'} \\ V_j = V_{j'} \end{cases} \tag{3}$$

$$P_{ij} = P_{ij'} = P_{i'j} \tag{4}$$

$$Q_{ij} = Q_{ij'} = Q_{i'j} \tag{5}$$

$$l_{ij} = l_{ij'} = l_{i'j} \tag{6}$$

Suppose $\mathcal{N}$ is the set of sub-areas in the distributed network. Let $\boldsymbol{x}$, $\boldsymbol{x}_n$, $f(\boldsymbol{x})$, $f_n(\boldsymbol{x}_n)$ denote the vector of independent variables in the distributed network, the set of independent variables in sub-area n, $n\in\mathcal{N}$, the active power loss of the whole distributed network, and the active power loss of sub-area n, respectively. The distributed OPF can be formulated as (7) - (16). Equations (8) - (13) represent the power flow constraints in sub-area n, while (14)-(16) represent the security constraints in sub-area n. For any $n\in\mathcal{N}$, constraints (8)-(16) should be satisfied.

$$\min f(\boldsymbol{x}) = \sum_{n\in\mathcal{N}} f_n(\boldsymbol{x}_n) = \sum_{n\in\mathcal{N}}\sum_{ij\in\mathcal{L}^n} l_{ij} r_{ij} \tag{7}$$

s.t.

$$P_i = P_i^{gen} - P_i^{load} \quad \forall i \in \mathcal{B}^n \tag{8}$$

$$Q_i = Q_i^{gen} + Q_i^{SVC} - Q_i^{load} \quad \forall i \in \mathcal{B}^n \tag{9}$$

$$P_j = \sum_{jk\in\mathcal{L}^n} P_{jk} - \sum_{ij\in\mathcal{L}^n}(P_{ij} - r_{ij} l_{ij}) \quad \forall j \in \mathcal{B}^n \tag{10}$$

$$Q_j = \sum_{jk\in\mathcal{L}^n} Q_{jk} - \sum_{ij\in\mathcal{L}^n}(Q_{ij} - x_{ij} l_{ij}) \quad \forall j \in \mathcal{B}^n \tag{11}$$

$$V_i - V_j = 2(P_{ij} r_{ij} + Q_{ij} x_{ij}) + l_{ij}(r_{ij}^2 + x_{ij}^2) \quad \forall i \in \mathcal{B}^n, ij \in \mathcal{L}^n \tag{12}$$

$$\left\| \begin{matrix} 2P_{ij} \\ 2Q_{ij} \\ V_i - l_{ij} \end{matrix} \right\|_2 \le V_i + l_{ij} \quad \forall i \in \mathcal{B}^n, ij \in \mathcal{L}^n \tag{13}$$

$$V_{i,\min} \le V_i \le V_{i,\max} \quad \forall i \in \mathcal{B}^n \tag{14}$$

$$l_{ij,\min} \le l_{ij} \le l_{ij,\max} \quad \forall ij \in \mathcal{L}^n \tag{15}$$

$$Q_{i,\min}^{SVC} \le Q_i^{SVC} \le Q_{i,\max}^{SVC} \quad \forall i \in \mathcal{B}^{SVC,n} \tag{16}$$

where $\mathcal{B}^n$ and $\mathcal{L}^n$ are the sets of nodes and lines in sub-area n, respectively; $\mathcal{B}^{SVC,n}$ is the set of nodes with static var compensator (SVC) in sub-area n; P_i and Q_i are the injected active and reactive power at bus i, respectively; P_i^{gen} and P_i^{load} are the generated active power and load active power at bus i, respectively; Q_i^{gen} and Q_i^{load} are the generated reactive power and load reactive power at bus i, respectively; Q_i^{SVC} is the reactive power generated by SVC at bus i; r_{ij} and x_{ij} are the resistance and reactance in line (ij), respectively; and $V_{i,\min}$, $V_{i,\max}$, $l_{ij,\min}$, $l_{ij,\max}$, and $Q_{i,\min}^{SVC}$, $Q_{i,\max}^{SVC}$ are the minimum and maximum values of V_i, l_{ij}, and Q_i^{SVC}, respectively.

B. ADMM

ADMM is a distributed optimization algorithm that solves complex problems by decomposing them into several sub-problems [19]. To apply ADMM to solve distributed OPF, the global variable $\boldsymbol{z}$ is introduced to coordinate boundary variables and ensure their consistency. With the introduction of global variables, the distributed OPF in (7) is decomposed into several optimization sub-problems. For sub-area n, $n\in\mathcal{N}$, the sub-problem is formulated as (17) and (18). The objective is the minimization of line loss in sub-area n in (17). And (18) is the consistency constraint to ensure that the boundary variables are equal to the global variables.

$$\min_{\boldsymbol{x}_n\in\boldsymbol{X}_n} f_n(\boldsymbol{x}_n) = \sum_{ij\in\mathcal{L}^n} l_{ij} r_{ij} \tag{17}$$

s.t.

$$\boldsymbol{x}_n^b = \boldsymbol{M}_n \boldsymbol{z} \tag{18}$$

where $\boldsymbol{x}_n$ is divided into the inner variables $\boldsymbol{x}_n^{in}$ and the boundary variables $\boldsymbol{x}_n^b$; $\boldsymbol{M}_n$ is a selection matrix (0-1 matrix) that serves as a mapping operator; and $\boldsymbol{X}_n$ is the feasible region of $\boldsymbol{x}_n$ constrained by (8)-(16). The 2-sub-area system illustrated in Fig. 1 is taken as an example. The boundary variables of sub-area 1 are defined as $\boldsymbol{x}_1^b = [V_i, V_{j'}, P_{ij'}, Q_{ij'}, l_{ij'}]^{\mathrm{T}}$,

while they are given by $\boldsymbol{x}_2^b=[V_{i'}, V_j, P_{i'j}, Q_{i'j}, l_{i'j}]^{\mathrm{T}}$ in sub-area 2. A global variable $\boldsymbol{z}=[z_{V_i}, z_{V_j}, z_{P_{ij}}, z_{Q_{ij}}, z_{l_{ij}}]^{\mathrm{T}}$ is introduced to ensure the consistency.

Let $\boldsymbol{\lambda}_n$ denote the Lagrange multipliers of sub-area n, and ρ denote the penalty parameter. Equations (17) and (18) can be transformed to (19) by the augmented Lagrangian $\mathcal{L}_n$.

$$\mathcal{L}_n(\boldsymbol{x}_n, \boldsymbol{z}, \boldsymbol{\lambda}_n)=\boldsymbol{f}_n(\boldsymbol{x}_n)+\boldsymbol{\lambda}_n^{\mathrm{T}}(\boldsymbol{x}_n^b-\boldsymbol{M}_n\boldsymbol{z})+\frac{\rho}{2}\left\|\boldsymbol{x}_n^b-\boldsymbol{M}_n\boldsymbol{z}\right\|_2^2 \quad (19)$$

The variable $\boldsymbol{x}_n^{(k+1)}$ is first updated based on $\boldsymbol{z}^{(k)}$ and $\boldsymbol{\lambda}_n^{(k)}$ at the $(k+1)^{\text{th}}$ iteration by (20).

$$\boldsymbol{x}_n^{(k+1)}=\underset{\boldsymbol{x}_n\in\boldsymbol{X}_n}{\arg\min}\ \mathcal{L}_n(\boldsymbol{x}_n, \boldsymbol{z}^{(k)}, \boldsymbol{\lambda}_n^{(k)}) \quad \forall n\in\mathcal{N} \quad (20)$$

For any global variable $z_g\in\boldsymbol{z}$, define $\mathcal{H}(z_g)$ as the set of all boundary variables x_s^g that are coordinated by z_g. To ensure the consistency condition (18), it is required that $x_s^g=z_g$ for all $x_s^g\in\mathcal{H}(z_g)$. The global variable $z_g^{(k+1)}$ and Lagrange multiplier $\boldsymbol{\lambda}_n^{(k+1)}$ are then updated by (21) and (22) at the $(k+1)^{\text{th}}$ iteration, respectively.

$$z_g^{(k+1)}=\frac{\sum_{x_s^g\in\mathcal{H}(z_g)} x_s^{g(k+1)}}{\mathrm{card}(\mathcal{H}(z_g))} \quad (21)$$

$$\boldsymbol{\lambda}_n^{(k+1)}=\boldsymbol{\lambda}_n^{(k)}+\rho(\boldsymbol{x}_n^{b(k+1)}-\boldsymbol{M}_n\boldsymbol{z}^{(k+1)}) \quad (22)$$

where $\mathrm{card}(\mathcal{H}(z_g))$ is the number of elements in $\mathcal{H}(z_g)$.

To ensure the convergence of ADMM, the Lagrange multipliers must satisfy certain constraint conditions, as described in Lemma 1.

Lemma 1: let $\boldsymbol{\lambda}_{z_g}$ denote the vector composed of all Lagrange multipliers λ_s^g corresponding to x_s^g, $x_s^g\in\mathcal{H}(z_g)$, where λ_s^g is the Lagrange multiplier associated with the constraint $x_s^g=z_g$. At each iteration, the sum of all elements in vector $\boldsymbol{\lambda}_{z_g}^{(k)}$ should be equal to the sum in $\boldsymbol{\lambda}_{z_g}^{(0)}$ at the initial state.

Assuming $\boldsymbol{\lambda}_{z_g}^{(0)}=\boldsymbol{0}$, the sum of all elements in vector $\boldsymbol{\lambda}_{z_g}$ should remain 0 at the k^{th} iteration, as shown in (23).

$$\sum_{\lambda_s^g\in\boldsymbol{\lambda}_{z_g}} \lambda_s^{g(k)}=0 \quad \forall z_g\in\boldsymbol{z} \quad (23)$$

The convergence criterion of ADMM includes two parts, i.e., the primal residual $\varepsilon_p^{(k)}$ and dual residual $\varepsilon_d^{(k)}$, calculated as (24) and (25), respectively.

$$\varepsilon_p^{(k+1)}=\left\|\boldsymbol{z}^{(k+1)}-\boldsymbol{z}^{(k)}\right\|_\infty \quad (24)$$

$$\varepsilon_d^{(k+1)}=\frac{1}{\rho}\left\|\boldsymbol{\lambda}^{(k+1)}-\boldsymbol{\lambda}^{(k)}\right\|_\infty \quad (25)$$

When a specified precision tolerance ε^* is given, the convergence criterion for ADMM is defined as:

$$\varepsilon^{(k+1)}=\left\|[\varepsilon_p^{(k+1)} \quad \varepsilon_d^{(k+1)}]\right\|_\infty<\varepsilon^* \quad (26)$$

III. LSTM-VAE Assisted ADMM

A. Framework of LSTM-VAE Assisted ADMM

The framework of the LSTM-VAE assisted ADMM is shown in Fig. 2. It can be divided into two stages, i.e., offline training and online solving. The offline training employs a two-stage data generation strategy to reduce the computational time required for generating training data. The online solving stage involves solving the distributed OPF using the LSTM-VAE assisted ADMM.

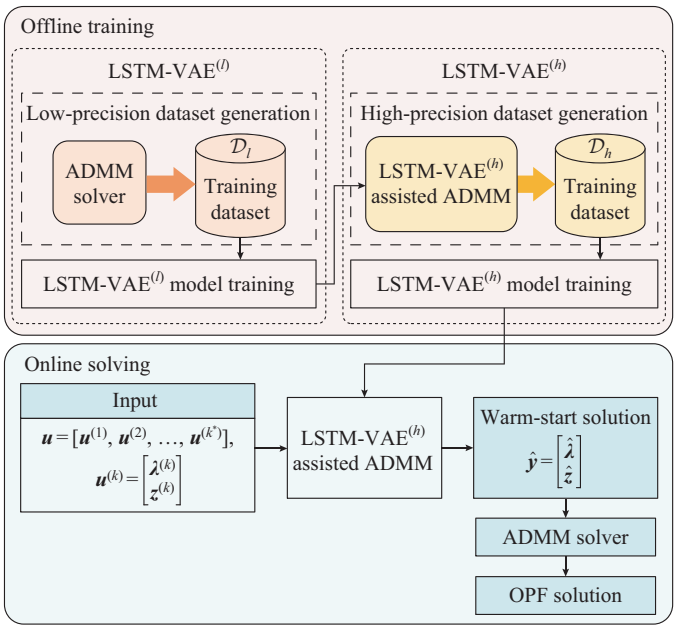

Fig. 2. Framework of LSTM-VAE assisted ADMM.

The offline training stage includes two steps, i.e., training the low-precision LSTM-VAE$^{(l)}$ model and training the high-precision LSTM-VAE$^{(h)}$ model.

In the step of training LSTM-VAE$^{(l)}$ model, traditional ADMM with large convergence tolerance is first executed M_l times to generate the low-precision dataset $\mathcal{D}_l$, which is utilized to train LSTM-VAE$^{(l)}$ model. In the step of training the LSTM-VAE$^{(h)}$ model, the LSTM-VAE$^{(l)}$ model is employed to accelerate ADMM with small convergence tolerance to generate the high-precision dataset $\mathcal{D}_h$. $\mathcal{D}_h$ is then used to train the LSTM-VAE$^{(h)}$ model, which is applied to expedite the solution of the distributed OPF.

In the online solving stage, the proposed LSTM-VAE assisted ADMM first performs k^* iterations of ADMM to generate the vector $\boldsymbol{u}=[\boldsymbol{u}^{(1)}, \boldsymbol{u}^{(2)}, \ldots, \boldsymbol{u}^{(k^*)}]$. Each $\boldsymbol{u}^{(k)}$ includes the global variable $\boldsymbol{z}^{(k)}$ and Lagrange multiplier $\boldsymbol{\lambda}^{(k)}$, $k=1, 2, \ldots, k^*$. The vector $\boldsymbol{u}$ is then used as the input of LSTM-VAE$^{(h)}$ model to estimate the warm-start solution $\hat{\boldsymbol{y}}=\begin{bmatrix}\hat{\boldsymbol{\lambda}}\\ \hat{\boldsymbol{z}}\end{bmatrix}$ for ADMM. Based on the warm-start solution, ADMM is carried out until the convergence condition is satisfied to obtain the optimal solution of distributed OPF.

In the offline training stage, the computational time consists of the time required for data generation and model training. Due to the offline training of LSTM-VAE model, its computational time does not impact the online solving stage. Thus, in practical applications, the time complexity of the proposed LSTM-VAE assisted ADMM is usually determined by the online solving stage. Since the computational time of LSTM-VAE model is significantly smaller than that of one ADMM iteration, the time complexity of the proposed LSTM-VAE assisted ADMM is not greater than the traditional ADMM.

B. LSTM-VAE

Assuming that the ADMM iteration process resembles the time series, both global variables $\boldsymbol{z}$ and Lagrange multipliers $\boldsymbol{\lambda}$ exhibit distinct temporal characteristics. As the scale of the power system and the number of partitions grow, the output dimensions of $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ expand, which complicates the accurate estimation. Therefore, an LSTM-VAE model is proposed to estimate the convergence value of global variables and Lagrange multipliers. The LSTM [20] part is introduced to process high-dimensional temporal characteristics of $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, extracting their temporal patterns to generate low-dimensional representations. Subsequently, the decoder part of VAE [21] reconstructs these compressed latent vectors back to the high-dimensional asymptotic convergence values of $\boldsymbol{z}$ and $\boldsymbol{\lambda}$.

The network structure of LSTM-VAE model is shown as Fig. 3. Let $\boldsymbol{y}=\begin{bmatrix}\boldsymbol{z}\\ \boldsymbol{\lambda}\end{bmatrix}$ denote the target vector. The true value of target vector $\boldsymbol{y}$ is the true convergence value of $\boldsymbol{z}$ and $\boldsymbol{\lambda}$, denoted as $\boldsymbol{y}^*=\begin{bmatrix}\boldsymbol{z}^*\\ \boldsymbol{\lambda}^*\end{bmatrix}$. The estimated value of $\boldsymbol{y}$ by LSTM-VAE is represented as $\hat{\boldsymbol{y}}=\begin{bmatrix}\hat{\boldsymbol{z}}\\ \hat{\boldsymbol{\lambda}}\end{bmatrix}$. C_t and h_t are the current state and hidden state, respectively; *sig* is the sigmoid function; and tanh is the activation function.

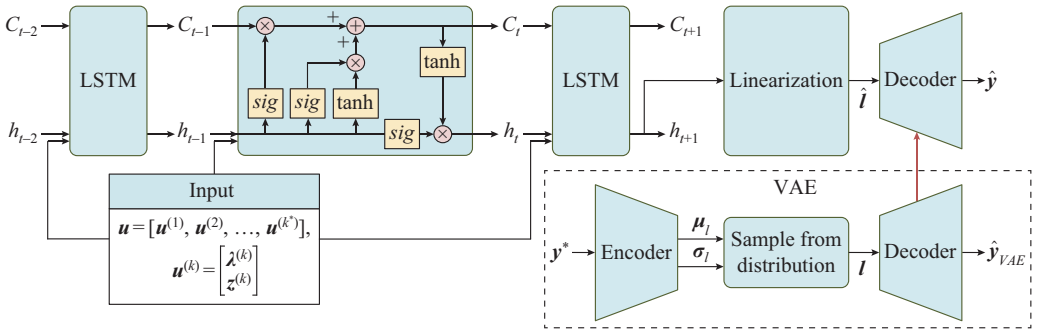

Fig. 3. Network structure of LSTM-VAE model.

The VAE is composed of an encoder and a decoder. It is first individually trained and frozen for the subsequent LSTM training. The target vector $\boldsymbol{y}$ is used as both of the input and output of VAE. $\boldsymbol{y}$ is encoded into a latent vector $\boldsymbol{l}$ with dimension d, which follows the normal distribution $\boldsymbol{l}\sim(\boldsymbol{\mu}_l,\boldsymbol{\sigma}_l^2)$, where $\boldsymbol{\mu}_l=[\mu_{l(1)},\mu_{l(2)},\ldots,\mu_{l(d)}]^{\mathrm{T}}$ and $\boldsymbol{\sigma}_l^2=[\sigma_{l(1)}^2,\sigma_{l(2)}^2,\ldots,\sigma_{l(d)}^2]^{\mathrm{T}}$ are the mean and variance of $\boldsymbol{l}$, respectively. $\hat{\boldsymbol{y}}_{VAE}$ is the true value reconstructed by VAE. The loss function of VAE comprises reconstruction loss and Kullback-Leibler divergence loss as (27). Through the encoding and decoding process of VAE, the latent vector $\boldsymbol{l}$ can represent the compressed target vector $\boldsymbol{y}$ to reduce the dimensionality of $\boldsymbol{y}$. The LSTM is trained to estimate the latent vector $\boldsymbol{l}$ instead of target vector $\boldsymbol{y}$ accordingly.

$$\mathcal{L}_{VAE}=\left\|\hat{\boldsymbol{y}}_{VAE}-\hat{\boldsymbol{y}}^*\right\|_2^2-\sum_{i=1}^{d}\frac{1}{2}\left(1+2\ln\sigma_{l(i)}-\mu_{l(i)}^2-\sigma_{l(i)}^2\right) \quad (27)$$

The LSTM comprises k^* LSTM cells, designed to extract the time-series characteristics of $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ from the first k^* iterations of the ADMM process. ADMM is first executed k^* times to generate time sequences $\boldsymbol{u}=[\boldsymbol{u}^{(1)},\boldsymbol{u}^{(2)},\ldots,\boldsymbol{u}^{(k^*)}]$, where $\boldsymbol{u}^{(k)}=\begin{bmatrix}\boldsymbol{\lambda}^{(k)}\\ \boldsymbol{z}^{(k)}\end{bmatrix}$. $\boldsymbol{u}$ is utilized as the input and the latent vector $\boldsymbol{l}$ is the output of LSTM. The estimated latent vector $\hat{\boldsymbol{l}}$ is finally transformed into the frozen decoder of VAE, and $\hat{\boldsymbol{l}}$ is reconstructed into the estimated target value $\hat{\boldsymbol{y}}=\begin{bmatrix}\hat{\boldsymbol{\lambda}}\\ \hat{\boldsymbol{z}}\end{bmatrix}$. The loss of the LSTM is computed based on the estimated value $\hat{\boldsymbol{y}}$ and true value $\boldsymbol{y}^*$.

Based on the trained LSTM-VAE model, the acceleration process of LSTM-VAE assisted ADMM is outlined in Algorithm 1. The input is the given tolerance ε^*, the size of input vector $\boldsymbol{u}$ of LSTM k^*, and the trained LSTM-VAE model. The output is the optimal solution of distributed OPF. The ADMM is first executed k^* times to generate $\boldsymbol{u}=[\boldsymbol{u}^{(1)},\boldsymbol{u}^{(2)},\ldots,\boldsymbol{u}^{(k^*)}]$. And then, $\boldsymbol{u}$ is input into the LSTM-VAE model to obtain the estimated convergence values $\hat{\boldsymbol{z}}$ and $\hat{\boldsymbol{\lambda}}$, which are then utilized as warm-start solution for ADMM. The ADMM then continues to execute until (26) is satisfied and outputs the optimal solution of distributed OPF.

Algorithm 1: acceleration process of LSTM-VAE assisted ADMM

Input: ε^*, k^*, and LSTM-VAE model
Output: optimal solution of distributed OPF
$k\leftarrow 0$
while $\varepsilon>\varepsilon^*$ **do**
  if $k<k^*$ **then**
    Execute ADMM with (20)-(22) to obtain $\boldsymbol{u}^{(k+1)}=\begin{bmatrix}\boldsymbol{\lambda}^{(k+1)}\\ \boldsymbol{z}^{(k+1)}\end{bmatrix}$
  else if $k=k^*$ **then**
    Input $\boldsymbol{u}$ into LSTM-VAE to estimate $\hat{\boldsymbol{y}}=\begin{bmatrix}\hat{\boldsymbol{\lambda}}\\ \hat{\boldsymbol{z}}\end{bmatrix}$
    Correct Lagrange multipliers $\hat{\boldsymbol{\lambda}}$
    Set $\hat{\boldsymbol{y}}$ as the warm-start solution
  else if $k>k^*$ **then**
    Execute ADMM with (20)-(22) to update $\boldsymbol{x}$, $\boldsymbol{z}$, and $\boldsymbol{\lambda}$
  end
  $k\leftarrow k+1$
end

C. Loss Function Considering Lagrange Multipliers Constraint

The LSTM-VAE model is designed to estimate the convergence values of global variables and Lagrange multipliers. However, the estimated values of Lagrange multipliers, $\hat{\boldsymbol{\lambda}}$, usually violate the constraint in (23), adversely affecting the convergence of ADMM. To improve the accuracy of the

LSTM-VAE model, a specific loss function is designed to guide the training process effectively.

The loss function is composed of two parts, i.e., the accuracy error term $\mathcal{L}_{acc}$ and the constraint penalty term $\mathcal{L}_{pen}$. The accuracy error term $\mathcal{L}_{acc}$ is calculated using the mean squared error (MSE) between the estimated value $\hat{\boldsymbol{y}}$ and the true value $\boldsymbol{y}^*$ as (28).

$$\mathcal{L}_{acc} = \frac{1}{\text{card}(\boldsymbol{y}^*)} \left\| \hat{\boldsymbol{y}} - \boldsymbol{y}^* \right\|_2^2 \tag{28}$$

where $\text{card}(\boldsymbol{y}^*)$ denotes the number of elements in $\boldsymbol{y}^*$.

The penalty error $\mathcal{L}_{pen}$ reflects the degree of constraint violation and guides $\hat{\boldsymbol{\lambda}}$ towards satisfying the constraint in (23). Define $\hat{\boldsymbol{\lambda}}_{z_g}$ as the vector composed of estimated Lagrange multipliers $\hat{\lambda}_s^g$ corresponding to x_s^g. For any global variable $z_g \in \boldsymbol{z}$, the penalty error $\mathcal{L}_{pen}^{z_g}$ is computed as:

$$\mathcal{L}_{pen}^{z_g} = \sum_{\hat{\lambda}_s^g \in \hat{\boldsymbol{\lambda}}_{z_g}} (\hat{\lambda}_s^g)^2 \tag{29}$$

For different $z_g \in \boldsymbol{z}$, the magnitude of $\mathcal{L}_{pen}^{z_g}$ varies significantly. Furthermore, the order of magnitude between $\mathcal{L}_{acc}$ and $\mathcal{L}_{pen}^{z_g}$ is also greatly different, which can diminish the impact of penalty term on training the LSTM-VAE model. Therefore, for each $z_g \in \boldsymbol{z}$, the normalization parameter τ_{z_g} is defined to normalize $\mathcal{L}_{pen}^{z_g}$ as:

$$\tau_{z_g} = \frac{1}{\text{card}(\boldsymbol{\lambda}_{z_g}^*)} \sum_{\lambda_s^{g*} \in \boldsymbol{\lambda}_{z_g}^*} \left(\max_{\mathcal{D}} (\lambda_s^{g*}) - \min_{\mathcal{D}} (\lambda_s^{g*}) \right) \tag{30}$$

where $\max_{\mathcal{D}} (\lambda_s^{g*})$ and $\min_{\mathcal{D}} (\lambda_s^{g*})$ are the maximum and minimum values of λ_s^{g*} in the training dataset $\mathcal{D}$, respectively; and $\text{card}(\boldsymbol{\lambda}_{z_g}^*)$ denotes the number of elements in $\boldsymbol{\lambda}_{z_g}^*$.

Based on (29) and (30), the normalized penalty error $\overline{\mathcal{L}}_{pen}^{z_g}$ is computed as:

$$\overline{\mathcal{L}}_{pen}^{z_g} = \frac{\mathcal{L}_{pen}^{z_g}}{\tau_{z_g}} \quad \forall z_g \in \boldsymbol{z} \tag{31}$$

According to (28)-(31), the loss function designed for the LSTM-VAE model is formulated as:

$$\mathcal{L} = \omega_{acc} \mathcal{L}_{acc} + \omega_{pen} \sum_{z_g \in \boldsymbol{z}} \overline{\mathcal{L}}_{pen}^{z_g} \tag{32}$$

where ω_{acc} and ω_{pen} are the weights of the accuracy error term and penalty term in the loss function, respectively.

Although the designed loss function in (32) can effectively alleviate the constraint violation, it cannot be ensured that the constraint in (23) is satisfied. Thus, the correction value $\Delta\hat{\lambda}^g$ is first computed by averaging the constraint violation. Then, it is uniformly reduced by subtracting $\Delta\hat{\lambda}^g$ from each Lagrange multipliers $\hat{\lambda}_s^g \in \hat{\boldsymbol{\lambda}}_{z_g}$ to ensure that the constraints are satisfied. This correction approach is presented as:

$$\lambda_s^{g(k^*)} = \hat{\lambda}_s^g - \Delta\hat{\lambda}^g \tag{33}$$

$$\Delta\hat{\lambda}^g = \frac{1}{\text{card}(\hat{\boldsymbol{\lambda}}_{z_g})} \sum_{\hat{\lambda}_s^g \in \hat{\boldsymbol{\lambda}}_{z_g}} \hat{\lambda}_s^g \quad \forall z_g \in \boldsymbol{z} \tag{34}$$

where $\Delta\hat{\lambda}^g$ is the correction value corresponding to $\boldsymbol{\lambda}_{z_g}$; and $\text{card}(\hat{\boldsymbol{\lambda}}_{z_g})$ denotes the number of elements in $\hat{\boldsymbol{\lambda}}_{z_g}$.

D. Two-stage Training Data Generation Strategy

Although the complexity of the proposed LSTM-VAE assisted ADMM in practical applications is primarily associated with the online solving stage, there are also significant challenges in the offline training stage. In practice, to achieve good estimation performance of the LSTM-VAE model, it is necessary to have enough and various training data, and ADMM needs to be executed many times, which is time-consuming. To enhance the efficiency of offline training of LSTM-VAE model, a two-stage training data generation strategy is proposed. Its pseudo code is shown in Algorithm 2.

Algorithm 2: two-stage training data generation strategy

Input: ε_l^*, ε_h^*, M_l, and M_h
Output: high-precision dataset $\mathcal{D}_h$
for $m = 1, 2, \ldots, M_l$ **do**
  Execute ADMM with (20)-(22) to obtain the sample $\mathcal{D}_l[m]$
  Add $\mathcal{D}_l[m]$ into $\mathcal{D}_l$
end
Train LSTM-VAE$^{(l)}$ with $\mathcal{D}_l$
for $m = 1, 2, \ldots, M_h$ **do**
  Execute LSTM-VAE$^{(l)}$ assisted ADMM to obtain the sample $\mathcal{D}_h[m]$
  Add $\mathcal{D}_h[m]$ into $\mathcal{D}_h$
end

1) Stage 1: the distributed OPF with ADMM is executed M_l times to generate a low-precision dataset $\mathcal{D}_l$ with a specified low-precision tolerance ε_l^*.

Subsequently, a low-precision LSTM-VAE$^{(l)}$ is trained based on $\mathcal{D}_l$. The trained LSTM-VAE$^{(l)}$ is then utilized to assist in the high-precision data generation. Since the accuracy of the dataset $\mathcal{D}_l$ is not directly used for acceleration, the required samples in $\mathcal{D}_l$ is small and the value of M_l is relatively small.

2) Stage 2: to generate the dataset $\mathcal{D}_h$ with the desired precision tolerance ε_h^*, the LSTM-VAE$^{(l)}$ assisted ADMM is executed M_h times. The trained LSTM-VAE$^{(l)}$ is used to estimate the global variables and Lagrange multipliers. Subsequently, ADMM is executed until it reaches the given precision ε_h^*. Due to $\mathcal{D}_h$ directly impacting the acceleration performance, the value of M_h should be large.

Remark 1: since the computational time of each iteration of ADMM is much longer than that of LSTM-VAE, the whole computational time depends on the iteration numbers of ADMM, and the computational time of LSTM-VAE can be ignored. Suppose $\bar{t}_l$ and $\bar{t}_h$ are the average iteration numbers of ADMM to solve distributed OPF at precision ε_l^* and ε_h^*, respectively. Let $\bar{t}'$ denote the average iteration number of ADMM in LSTM-VAE$^{(l)}$ assisted ADMM to solve distributed OPF at precision ε_h^*. The total iteration number of ADMM for generating training data directly using traditional ADMM is $t_1 = M_h \bar{t}_h$, whereas the iteration number using the proposed two-stage training data generation strategy is $t_2 = M_l \bar{t}_l + M_h \bar{t}'$. Since $\varepsilon_l^* > \varepsilon_h^*$, it is evident that $M_l \le M_h$, $\bar{t}_l \le \bar{t}_h$, and $\bar{t}' \le \bar{t}_h$. Suppose $M_l = \omega_1 M_h$, $\bar{t}_l = \omega_2 \bar{t}_h$, $\bar{t}' = \omega_3 \bar{t}_h$, $\omega_1, \omega_2, \omega_3 \in (0, 1]$. The difference of iteration number is $\Delta t = t_1 - t_2 = M_h \bar{t}_h - M_l \bar{t}_l - M_h \bar{t}' = M_h \bar{t}_h - \omega_1 \omega_2 M_h \bar{t}_h - \omega_3 M_h \bar{t}_h = (1 - \omega_1 \omega_2 - \omega_3) M_h \bar{t}_h$. Thus, when

$\omega_1\omega_2+\omega_3<1$, $\Delta t>0$, and the proposed strategy can reduce the data generation time.

IV. Case Study

This section provides a detailed description of the experimental settings and performance of the proposed LSTM-VAE assisted ADMM. A modified IEEE 123-bus system, a synthetic 500-bus system, and a 793-bus system are utilized to test the acceleration and convergence performance. The average normalized correction of the Lagrange multipliers is then compared to assess the effectiveness of the designed loss function. Finally, the impact of dataset precision on acceleration performance and the efficiency of the two-stage training data generation strategy is evaluated.

All experiments are implemented using Python 3.9.13 and COPT 6.5.1 on a personal computer equipped with an AMD Ryzen 7 3700X 8-Core processor, an NVIDIA GeForce GTX 1660 SUPER graphics card, and 32 GB of RAM.

A. Experimental Settings

1) System Parameters

The proposed LSTM-VAE assisted ADMM is evaluated based on the modified IEEE 123-bus system [22], a synthetic 500-bus system [23], and a 793-bus system [24]. The total load for the modified IEEE 123-bus system is 7.060 MW/3.890 Mvar. A total of 12 SVCs are installed, and 12 photovoltaic (PV) units with a combined capacity of 270 kW are installed on the modified IEEE 123-bus system. The system is partitioned into five sub-areas using community detection approaches, as depicted in Fig. 4. The total load of the synthetic 500-bus system is 16.136 MW/4.134 Mvar, and the detailed topology information can be accessed via [25]. In this system, 44 SVCs are installed with a total capacity of 1.05 Mvar, and 42 PV units are configured, providing a combined capacity of 905 kW. The synthetic 500-bus system is partitioned into 21 sub-areas. The total load of the 793 bus system is 23.86 MW/7.48 Mvar, and the detailed topology information can be accessed via [26]. The system is equipped with 59 SVCs providing a total reactive power of 3.43 Mvar and 61 PV units with a total output of 1.57 MW. The 793-bus system is also partitioned into 21 sub-areas.

2) Algorithm Parameters

In ADMM, the feasibility tolerance ε^* is set to be 5×10^{-4} and the penalty parameter ρ is 0.05. Given that day-ahead load prediction accuracy typically exceeds 90% in power systems, historical load data are simulated with active and reactive power fluctuating within the range of [90%, 110%] based on the reference load. To expedite the data generation process, $\varepsilon_l^*=5\times10^{-3}$, $\varepsilon_h^*=5\times10^{-4}$, $M_l=100$, and $M_h=300$ for modified IEEE 123-bus system, while $M_h=500$ for synthetic 500-bus system in Algorithm 2. Parameter settings of LSTM-VAE model are detailed in Table I. Both ω_{acc} and ω_{pen} in (32) are set to be 0.5. Additionally, 80% of the data are utilized to train the LSTM-VAE model, while the remaining 20% are used to evaluate the performance.

To examine the time-series behavior of global variables and Lagrange multipliers, their values are analyzed throughout the iteration process.

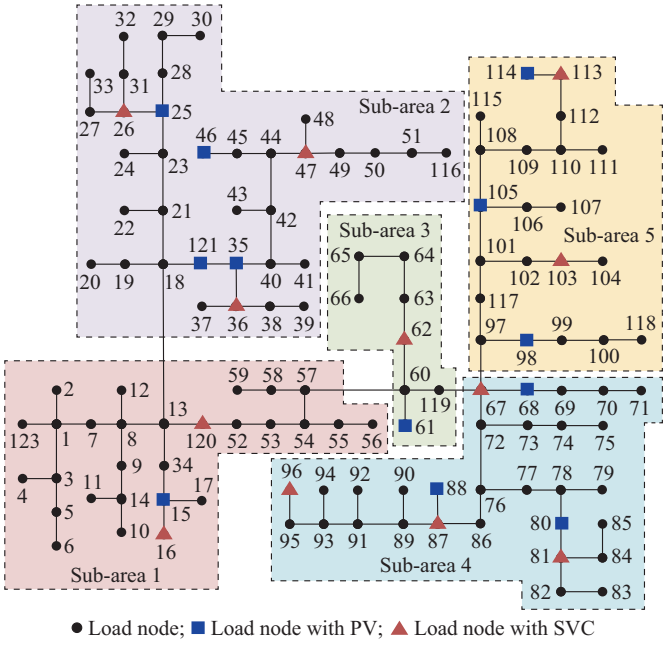

Fig. 4. Sub-area partition of modified IEEE 123-bus system.

TABLE I
Parameter Settings of LSTM-VAE Model

Part	Parameter	Modified IEEE 123-bus system	Synthetic 500-bus system	793-bus system
VAE	Hidden layer size	32	128	128
	Latent layer size	16	32	64
	Learning rate	0.001	0.001	0.001
	Batch size	32	32	32
LSTM	Sequence length k^*	10	30	35
	Number of layers	2	2	2
	Layer size	(128, 128)	(128, 128)	(128, 128)
	Learning rate	0.001	0.001	0.001
	Batch size	32	32	32

Figure 5 presents the variations of partial global variables and Lagrange multipliers in the synthetic 500-bus system, where different colors in Fig. 5(a)-(c) represent partial global variables of different node voltages, active power, and reactive power, respectively; and different colors in Fig. 5(d) represent different Lagrange multipliers of global variables. The partial global variables and Lagrange multipliers exhibit periodic fluctuations around their convergent values, with the amplitude of these fluctuations gradually decreasing over iterations, ultimately converging to a stable value.

B. Performance of Proposed LSTM-VAE Assisted ADMM

To verify the acceleration performance, the number of iterations is compared between the proposed LSTM-VAE assisted ADMM, other learning assisted ADMM, and the traditional ADMM. Subsequently, both the primal and dual residual curves are then utilized to analyze the convergence performance of proposed LSTM-VAE assisted ADMM. To evaluate the effectiveness of the designed loss function, the average normalized correction of the Lagrange multipliers is compared under different loss function parameter settings.

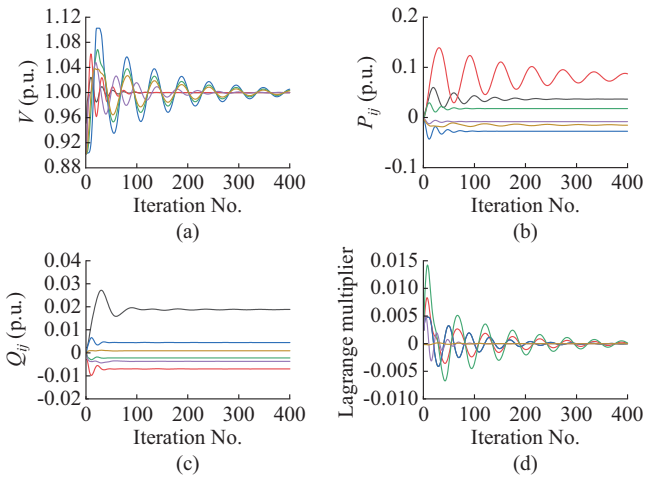

Fig. 5. Variations of partial global variables and Lagrange multipliers in synthetic 500-bus system. (a) Global variable V. (b) Global variable P_{ij}. (c) Global variable Q_{ij}. (d) Lagrange multipliers.

Furthermore, the impact of dataset precision on the acceleration performance and the efficiency of proposed data generation strategy are also analyzed. At last, the impact of estimation performance of LSTM-VAE model on acceleration performance is analyzed.

1) Acceleration Performance

Since the computational time of the LSTM-VAE model is substantially shorter than that of each iteration of ADMM, the overall solution time for the distributed OPF is primarily determined by the iteration number of ADMM. To evaluate the acceleration performance, the speedup ratio S_p is denoted by the ratio of the total iteration numbers of the LSTM-VAE assisted ADMM and traditional ADMM, as shown in (35).

$$S_p = \frac{\bar{t}_h}{\bar{t}'} \tag{35}$$

The iteration numbers required to solve the distributed OPF using traditional ADMM and LSTM-VAE assisted ADMM are presented in Table II. In the modified IEEE 123-bus system, the proposed LSTM-VAE assisted ADMM reduces the iteration number from 87 to 60, resulting in a speedup ratio of 1.45. Similarly, in the synthetic 500-bus system, the required iteration number decreases from 447 to 277, achieving a speedup ratio of 1.614. Compared with the modified IEEE 123-bus system, the synthetic 500-bus system is larger in scale and has more sub-areas. Consequently, the iteration number of the synthetic 500-bus system is larger than modified IEEE 123-bus system. The corresponding acceleration performance in synthetic 500-bus system is more obvious than that in the modified IEEE 123-bus system.

TABLE II
ITERATION NUMBERS REQUIRED TO SOLVE DISTRIBUTED OPF USING TRADITIONAL ADMM AND LSTM-VAE ASSISTED ADMM

System	Iteration number		
	Traditional ADMM	LSTM-VAE assisted ADMM	S_p
Modified IEEE 123-bus	87	60 (10+50)	1.450
Synthetic 500-bus	447	277 (30+247)	1.614
793-bus	852	627 (35+592)	1.359

The proposed LSTM-VAE assisted ADMM executes a total of 60 iterations to solve the distributed OPF in modified IEEE 123-bus system. The first 10 iterations are specifically dedicated to generate the input vector $\boldsymbol{u}$ for the LSTM-VAE model. Then the LSTM-VAE model leverages $\boldsymbol{u}$ to produce a warm-start solution for ADMM. Based on the warm-start solution, the ADMM executes additional 50 iterations to reach the convergence.

In the synthetic 500-bus system, the effectiveness of the proposed LSTM-VAE assisted ADMM becomes more evident. The require iterations using traditional ADMM to solve the distributed OPF is 447, while the proposed LSTM-VAE assisted ADMM only needs 277 iterations. After 30 iterations of traditional ADMM, $\boldsymbol{u}$ is input into the LSTM-VAE model. The warm-start solution is then generated and ADMM is converged with just additional 247 iterations, proving the efficiency gains brought by LSTM-VAE model.

In the 793-bus system, the proposed LSTM-VAE assisted ADMM also effectively accelerates the convergence process of ADMM. Specifically, while the traditional ADMM requires 852 iterations to achieve the target residual, the proposed LSTM-VAE assisted ADMM reduces the number of iterations to 627, resulting in a speedup ratio of 1.359.

The LSTM-VAE model captures periodic variations within the iteration process, enabling it to generate a good warm-start solution from only a few early ADMM iterations. The proposed LSTM-VAE assisted ADMM reduces the total number of iterations required for convergence, enhancing computational efficiency, especially when it is applied in complex large-scale system.

The sequence length k^* influences the number of iterations in solving distributed OPF. To evaluate the sensitivity of k^*, the total number of iterations of the proposed LSTM-VAE assisted ADMM with different k^* in modified IEEE-123 bus system is presented in Fig. 6. When k^*=11, the optimal acceleration performance is achieved, and the total iteration is reduced to 52 with a speedup ratio of 1.673. When k^* is too small, the LSTM-VAE model fails to adequately capture the periodic features of ADMM, resulting in limited acceleration of the distributed OPF solution. Conversely, as k^* increases, the input sequence length grows, which can diminish acceleration due to an increased number of iterations required for input sequence generation. Moreover, longer input sequences may introduce redundant information, adversely affecting the acceleration performance.

2) Convergence Performance

To evaluate the convergence performance of proposed LSTM-VAE assisted ADMM, both the primal and dual residuals are analyzed in the modified IEEE 123-bus system, synthetic 500-bus system, and 793-bus system.

The residuals in modified IEEE 123-bus system are presented in Fig. 7, where the red dashed line is the tolerance ε^* in (26). At the 11th iteration, the LSTM-VAE model estimates the convergence values of global variables and Lagrange multipliers. The estimated values for global variables differ from those at the 10th iteration, causing a sudden increase in the primal residual and leading to a temporal overall increase in residuals at subsequent iterations.

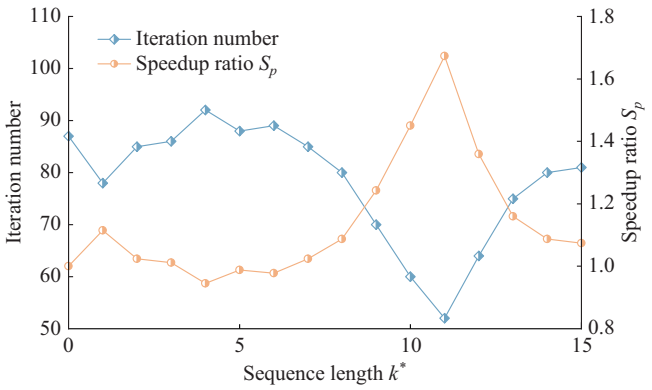

Fig. 6. Total number of iterations of proposed LSTM-VAE assisted ADMM with different k^* in modified IEEE 123-bus system.

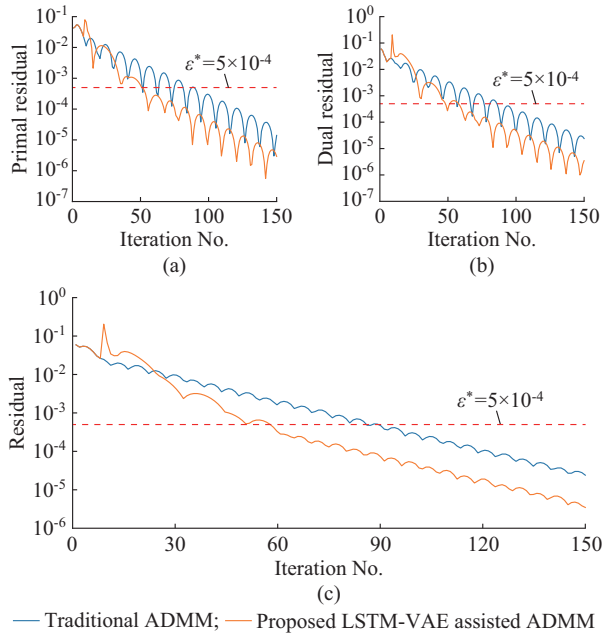

Fig. 7. Comparison of residuals in modified IEEE 123-bus system. (a) Primal residual. (b) Dual residual. (c) Residual.

Despite this, the proposed LSTM-VAE assisted ADMM still converges faster than traditional ADMM. It is because the residuals decrease more rapidly compared with traditional ADMM based on the estimated values.

The residual curves of the synthetic 500-bus system with various approaches are presented in Fig. 8. Similar to the modified IEEE 123-bus system, the residual curve of the modified synthetic 500-bus system also experiences a sudden increase at the 31st iteration. This increase is attributed to the estimated global variables differing from the values at the 30th iteration. Unlike in the IEEE 123-bus system, the dual residual in synthetic 500-bus system decreases suddenly at the 31st iteration, due to relatively minor change in the Lagrange multipliers. Despite the reduction in dual residual, the overall residual still increases because of the rise in primal residual.

Figure 9 compares the residuals in the 793-bus system. Consistent with the observations in the modified 123-bus and synthetic 500-bus systems, the residual in the 793-bus system exhibits a sudden increase at the 35th iteration, followed by a rapid decrease at the 36th iteration. This distinctive behavior reflects the impact of the warm-start solution, which adjusts the global variables and Lagrange multipliers to approach their convergence values more quickly. Such a sudden rise and subsequent rapid decrease indicate the effectiveness of the LSTM-VAE assisted ADMM, allowing the solver to bypass intermediate states that would otherwise require more iterations in traditional ADMM. Although this characteristic behavior is pronounced in small-scale systems, it still contributes to accelerating convergence in the 793-bus system, reducing the required number of iterations and improving overall efficiency.

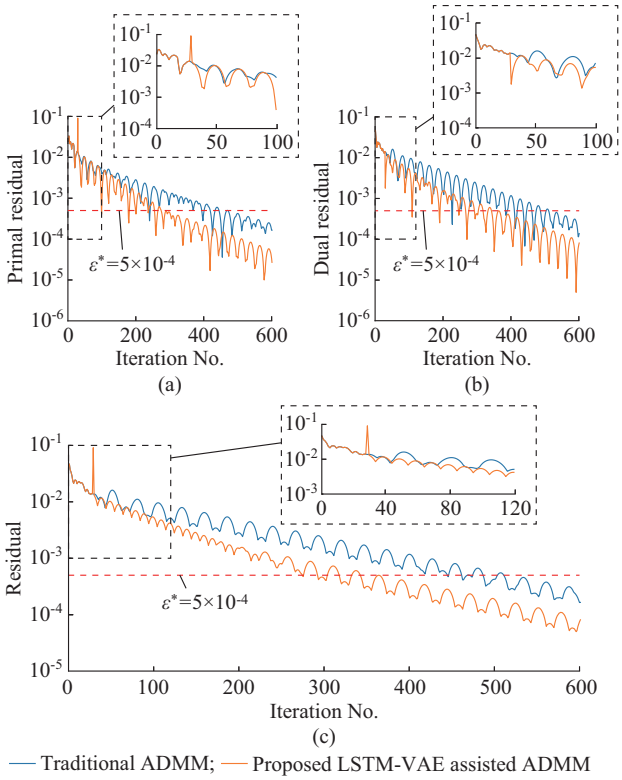

Fig. 8. Comparison of residuals in synthetic 500-bus system. (a) Primal residual. (b) Dual residual. (c) Residual.

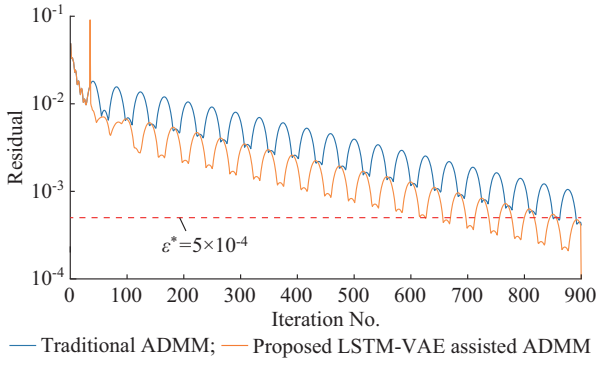

Fig. 9. Comparison of residuals in 793-bus system.

3) Comparison with Other Learning Assisted ADMMs

The proposed LSTM-VAE assisted ADMM is compared with other learning assisted ADMMs to evaluate its effective-

ness. Typical machine learning algorithms including LSTM, GRU, and support vector regression (SVR) are employed to estimate the convergence values of ADMM. Experimental results indicate that all four learning assisted ADMMs are capable of accelerating the convergence compared to the traditional ADMM. Figure 10 presents the residuals of various learning assisted ADMMs.

Among the evaluated approaches, SVR demonstrates the least effective performance. Nevertheless, the SVR assisted ADMM achieves faster convergence than the traditional ADMM, highlighting the general advantage of incorporating learning-based estimations. In contrast, deep learning assisted approaches such as LSTM and GRU exhibit superior acceleration performance.

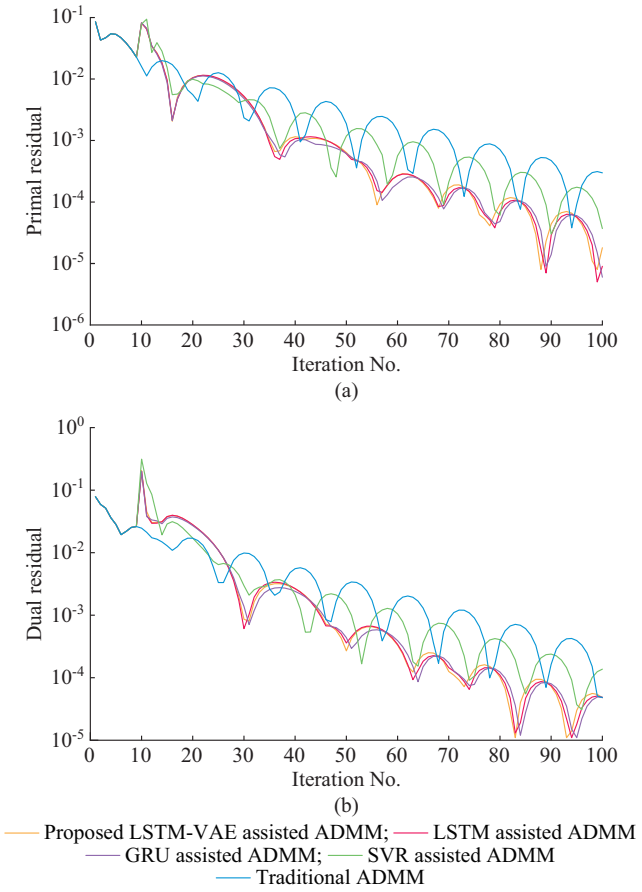

Fig. 10. Residuals of various learning assisted ADMMs in modified IEEE 123-bus system. (a) Primal residuals. (b) Dual residuals.

Notably, compared with LSTM and GRU assisted ADMMs, the proposed LSTM-VAE assisted ADMM introduces the decoder part of VAE to reduce the dimensionality of global variables and Lagrange multipliers.

Although the overall acceleration performances of LSTM, GRU, and LSTM-VAE assisted ADMMs are comparable, the dimensionality reduction provided by the VAE enables the LSTM-VAE assisted ADMM to converge to a local minimum faster than other learning assisted ADMMs. Specifically, after the 50th iteration, the LSTM-VAE assisted ADMM begins to exhibit a more rapid reduction in local residuals and reaches the local minimum point earlier than the LSTM and GRU assisted ADMMs.

4) Performance of Designed Loss Function

To measure the impact of the designed loss function on the constraint in (23), the average normalized correction $\Delta\bar{C}_\lambda$ is defined as (36).

$$\Delta\bar{C}_\lambda = \frac{1}{\mathrm{card}(\boldsymbol{\lambda}_c)} \sum_{\Delta\hat{\lambda}^g \in \boldsymbol{\lambda}_c} \frac{\left|\Delta\hat{\lambda}^g\right|}{\tau_{z_g}} \tag{36}$$

where $\boldsymbol{\lambda}_c$ is the vector of $\Delta\hat{\lambda}^g$; and $\mathrm{card}(\boldsymbol{\lambda}_c)$ is the number of elements in $\boldsymbol{\lambda}_c$.

The results of $\Delta\bar{C}_\lambda$ under different penalty weights are presented in Table III. When $\omega_{acc}=1$ and $\omega_{pen}=0$, the designed loss function is simplified to traditional MSE loss function. In this case, as no constraint violation penalty is imposed, $\Delta\bar{C}_\lambda = 4.11\times10^{-3}$, which is greatly higher than other weight settings. As ω_{pen} increases, $\Delta\bar{C}_\lambda$ decreases correspondingly, indicating that the inclusion of the penalty term effectively mitigates the violation rate of $\boldsymbol{\lambda}$.

TABLE III
RESULTS OF $\Delta\bar{C}_\lambda$ UNDER DIFFERENT PENALTY WEIGHTS

$(\omega_{acc}, \omega_{pen})$	$\Delta\bar{C}_\lambda$ (10^{-3})	$\mathcal{L}_{acc}$ (10^{-2})
(1, 0)	4.11	2.23
(0.9, 0.1)	1.94	3.35
(0.75, 0.25)	1.55	3.68
(0.5, 0.5)	1.42	3.84
(0.25, 0.75)	1.35	3.95

When ω_{pen} is relatively small, $\Delta\bar{C}_\lambda$ experiences rapid and significant changes as ω_{pen} increases, indicating a strong sensitivity of the penalty term. Once ω_{pen} surpasses a threshold, the changes in $\Delta\bar{C}_\lambda$ become moderate. This implies that further increase of ω_{pen} yields diminishing effects on $\Delta\bar{C}_\lambda$.

Besides, although $\Delta\bar{C}_\lambda$ decreases as ω_{pen} increases, $\mathcal{L}_{acc}$ in the loss function increases, which adversely affects the estimation accuracy of the LSTM-VAE. This compromises the accelerate performance of LSTM-VAE assisted ADMM in turn.

5) Acceleration Performance of Two-stage Training Data Generation Strategy

The estimation performance of LSTM-VAE is impacted by the quality of training dataset. The values of the precision are set different and the corresponding quality of training dataset is also different. Figure 11 presents the convergence curves of the LSTM-VAE assisted ADMM trained under different precisions of 5×10^{-3} and 5×10^{-4}, separately.

When the convergence tolerance ε^* is set to be 5×10^{-4} (red dashed line), the required numbers of iterations of these two LSTM-VAE assisted ADMMs are almost the same. When the convergence tolerance ε^* is set to be 5×10^{-5} (blue dashed line), the LSTM-VAE model trained with the precision 5×10^{-4} can assist the ADMM in converging faster than that with the precision 5×10^{-3}. Therefore, the LSTM-VAE model trained under the dataset with high precision can assist the ADMM in accelerating the convergence process. To reduce the generation time of the dataset with high precision, the two-stage training data generation strategy is proposed.

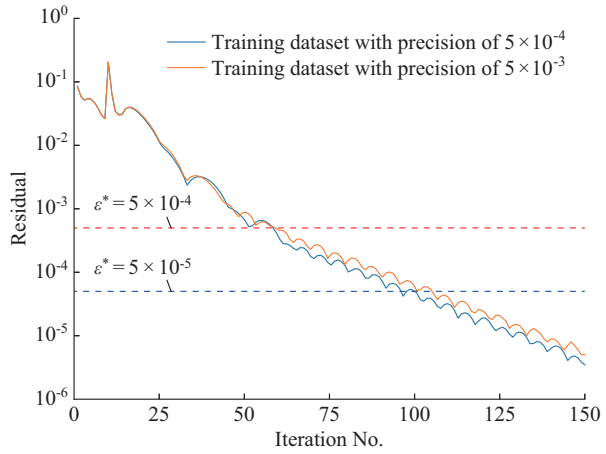

Fig. 11. Convergence curves of LSTM-VAE assisted ADMM trained under different precisions.

To further validate the acceleration performance of proposed training data generation strategy, the numbers of iterations required to generate the training datasets using traditional ADMM and two-stage training data generation strategy are presented in Table IV.

In the modified IEEE 123-bus system, the total number of iterations to generate 300 samples with ε_h^* using traditional ADMM is $300\times87=26100$. The proposed two-stage training data generation strategy requires only $100\times40+300\times60=22000$ iterations, achieving a speedup ratio of 1.19. In the synthetic 500-bus system, using traditional ADMM to generate 500 samples requires $500\times447=223500$ iterations, while using the proposed two-stage training data generation strategy, only $100\times117+500\times315=169200$ iterations are needed and a speedup ratio of 1.32 is achieved.

TABLE IV
NUMBERS OF ITERATIONS REQUIRED TO GENERATE TRAINING DATASETS USING TRADITIONAL ADMM AND TWO-STAGE TRAINING DATA GENERATION STRATEGY

System	Training dataset $\mathcal{D}$ of traditional ADMM		Two-stage data generation strategy				S_p
			Low-precision dataset $\mathcal{D}_l$		High-precision dataset $\mathcal{D}_h$		
	Size of dataset	Average iteration	Size of dataset	Average iteration	Size of datasets	Average iteration	
Modified IEEE 123-bus	300	87	100	40	300	60	1.19
Synthetic 500-bus	500	447	100	117	500	315	1.32

Compared with simple distribution networks, the acceleration performance of the proposed training data generation strategy is more pronounced in larger and more complex large-scale distribution networks, demonstrating significant potential for real-world applications.

6) Discussion

The impact of the LSTM-VAE estimation performance on the acceleration and optimal distributed OPF solution is discussed.

Regarding the acceleration performance, both the best- and worst-case scenarios are analyzed. In the best-case scenario, the LSTM-VAE estimates of Lagrange multipliers and global variables perfectly match the optimal values, allowing ADMM to converge without requiring additional iterations. However, as the system complexity increases, the estimation error may also increase. In the worst-case scenario, the estimation error exceeds a certain threshold, causing the LSTM-VAE to fail in providing a warm-start solution close to the optimal values. Consequently, ADMM proceeds with a random initial solution, and the proposed LSTM-VAE assisted ADMM degrades to the traditional ADMM. Since each ADMM iteration takes significantly longer than the LSTM-VAE, the computational time of the proposed LSTM-VAE assisted ADMM becomes nearly equivalent to that of the traditional ADMM in the worst-case scenario. In terms of the optimal distributed OPF solution, the LSTM-VAE model is only used to provide a warm-start solution, with subsequent ADMM iterations refining this initial solution. The final optimal solution is guaranteed by ADMM. Therefore, even in the worst-case scenario, the estimation error from the LSTM-VAE model does not affect the final OPF result.

V. CONCLUSION

To accelerate the convergence of ADMM, this paper proposes an LSTM-VAE assisted ADMM for solving distributed OPF. The LSTM-VAE model is designed to estimate the convergence values of global variables and Lagrange multipliers, which are then utilized as the warm-start solution for ADMM. To specifically address the constraints associated with Lagrange multipliers in ADMM, a novel loss function is designed to guide the training of the LSTM-VAE model. Additionally, a two-stage training data generation strategy is proposed to efficiently generate substantial data within a limited timeframe. The proposed LSTM-VAE assisted ADMM is validated on a modified IEEE 123-bus system, a synthetic 500-bus system, and a 793-bus system. The results demonstrat that the proposed LSTM-VAE assisted ADMM can significantly accelerate the convergence of ADMM, achieving speedup ratios of 1.45 for the modified IEEE 123-bus system, 1.614 for the synthetic 500-bus system, and 1.359 for the 793-bus system, respectively. Moreover, the proposed two-stage training data generation strategy proves to be effective, enhancing the data generation process with speedup ratios of 1.19 for the modified IEEE 123-bus system and 1.32 for the synthetic 500-bus system. Overall, the proposed LSTM-VAE assisted ADMM is highly suitable for applications in complex large-scale power systems.

REFERENCES

[1] Y. Lin, X. Zhang, J. Wang *et al.*, "Voltage stability constrained optimal power flow for unbalanced distribution system based on semidefinite programming," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 6, pp. 1614-1624, Nov. 2022.

[2] R. Dobbe, O. Sondermeijer, D. Fridovich-Keil *et al.*, "Toward distributed energy services: decentralizing optimal power flow with machine learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1296-1306, Mar. 2020.
[3] B. Kim and R. Baldick, "Coarse-grained distributed optimal power flow," *IEEE Transactions on Power Systems*, vol. 12, no. 2, pp. 932-939, May 1997.
[4] X. Liang, Z. Li, W. Huang *et al.*, "Relaxed alternating direction method of multipliers for hedging communication packet loss in integrated electrical and heating system," *Journal of Modern Power Systems and Clean Energy*, vol. 8, no. 5, pp. 874-883, Sept. 2020.
[5] A. Kargarian, Y. Fu, and Z. Li, "Distributed security-constrained unit commitment for large-scale power systems," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1925-1936, Jul. 2015.
[6] A. Engelmann, Y. Jiang, T. Muhlpfordt *et al.*, "Toward distributed OPF using ALADIN," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 584-594, Jan. 2019.
[7] L. Yang, J. Luo, Y. Xu *et al.*, "A distributed dual consensus admm based on partition for DC-DOPF with carbon emission trading," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1858-1872, Mar. 2019.
[8] T. Chen, X. Chen, W. Chen *et al.*, "Learning to optimize: a primer and a benchmark," *Journal of Machine Learning Research*, vol. 23, no. 189, pp. 1-59, Jun. 2022.
[9] X. Zhang, Z. Wu, Q. Sun *et al.*, "Application and progress of artificial intelligence technology in the field of distribution network voltage control: a review," *Renewable and Sustainable Energy Reviews*, vol. 192, p. 114282, Mar. 2024.
[10] W. Liao, J. Chen, Q. Liu *et al.*, "Data-driven reactive power optimization for distribution networks using capsule networks," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 5, pp. 1274-1287, Sept. 2022.
[11] Y. Cao, H. Zhao, G. Liang *et al.*, "Fast and explainable warm-start point learning for AC optimal power flow using decision tree," *International Journal of Electrical Power & Energy Systems*, vol. 153, p. 109369, Nov. 2023.
[12] T. W. K. Mak, M. Chatzos, M. Tanneau *et al.*, "Learning regionally decentralized ac optimal power flows with ADMM," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4863-4876, Nov. 2023.
[13] D. Biagioni, P. Graf, X. Zhang *et al.*, "Learning-accelerated ADMM for distributed DC optimal power flow," *IEEE Control Systems Letters*, vol. 6, pp. 1-6, 2022.
[14] M. Li, S. Kolouri, and J. Mohammadi, "Learning to optimize distributed optimization: ADMM-based DC-OPF case study," in *Proceedings of 2023 IEEE PES General Meeting (PESGM)*, Orlando, USA, Jul. 2023, pp. 1-5.
[15] B. Li and Q. Xu, "A machine learning-assisted distributed optimization method for inverter-based volt-var control in active distribution networks," *IEEE Transactions on Power Systems*, vol. 39, no. 2, pp. 2668-2681, Mar. 2024.
[16] A. Mohammadi and A. Kargarian, "Learning-aided asynchronous admm for optimal power flow," *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 1671-1681, May 2022.
[17] M. Z. Oskouei, B. Mohammadi-Ivatloo, O. Erdinc¸ *et al.*, "Optimal allocation of renewable sources and energy storage systems in partitioned power networks to create supply-sufficient areas," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 2, pp. 999-1008, Apr. 2021.
[18] H. Ruan, H. Gao, Y. Liu *et al.*,"Distributed voltage control in active distribution network considering renewable energy: a novel network partitioning method," *IEEE Transactions on Power Systems*, vol. 35, no. 6, pp. 4220-4231, Nov. 2020.
[19] T. Erseghe, "Distributed optimal power flow using ADMM," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2370-2380, Sept. 2014.
[20] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
[21] D. P. Kingma and M. Welling. (2013, Dec.). Auto-encoding variational bayes. [Online]. Available: https://api.semanticscholar.org/CorpusID:216078090
[22] K. P. Schneider, B. A. Mather, B. C. Pal *et al.*, "Analytic considerations and design basis for the IEEE distribution test feeders," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 3181-3188, May 2018.
[23] A. B. Birchfield, T. Xu, K. M. Gegner *et al.*, "Grid structural characteristics as validation criteria for synthetic networks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3258-3265, Jul. 2017.
[24] R. Huang and A. Tbaileh. (2020, Dec.). A 793 bus model created by the sustainable data evolution technology (SDET) tool for use in ARPAE's grid optimization competition, challenge 1. [Online]. Available: https://gocompetition.energy.gov/challenges/22/datasets
[25] IEEE PES Power Grid Library. (2023, Jul.). pglib-opf/pglib_opf_case500_goc.m. [Online]. Available: https://github.com/power-grid-lib/pglib-opf/blob/master/pglib_opf_case500_goc.m
[26] IEEE PES Power Grid Library. (2023, Jul.). pglib-opf/pglib_opf_case793_goc.m. [Online]. Available: https://github.com/power-grid-lib/pglib-opf/blob/master/pglib_opf_case793_goc.m

Huihuang Cai received the B. S. degree in electrical engineering from Southeast University, Nanjing, China, in 2023. He is currently working toward the M. S. degree in the School of Electrical Engineering, Southeast University. His research interest includes distributed optimization assisted by machine learning.

Huan Long received the B.Eng. degree from Huazhong University of Science and Technology, Wuhan, China, in 2013, and the Ph. D. degree from City University of Hong Kong, Hong Kong, China, in 2017. She is currently an Associate Professor with the School of Electrical Engineering, Southeast University, Nanjing, China. Her research interests include application of artificial intelligence in modeling, optimizing, and monitoring renewable energy system and power system.

Zhi Wu received the B.Eng. degree in mathematics from Southeast University, Nanjing, China, in 2009, the M. Sc. degree in electrical engineering from the School of Electrical Engineering, Southeast University, in 2012, and the Ph.D. degree from The University of Birmingham, Birmingham, U.K., in 2016. He is currently an Associate Professor with Southeast University. His research interests include renewable energy, and planning and optimization technique.

Wei Gu received the B.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2001 and 2006, respectively. From 2009 to 2010, he was a Visiting Scholar in the Department of Electrical Engineering, Arizona State University, Pheonix City, USA. He is now a Professor at the School of Electrical Engineering, Southeast University. He is the Director of the Institute of Distributed Generations and Active Distribution Networks. His research interests include distributed generation and microgrid, and integrated energy system.

Jingtao Zhao is with State Grid Electric Power Research Institute, Nanjing, China. His research interests include distribution network automation, AC/DC hybrid distribution network, distributed generation, and microgrid.