A Two-Stage Approach for Topology Change-Aware Data-Driven OPF

Yixiong Jia, Xian Wu, Zhifang Yang, Yi Wang

***Abstract*—Data-driven OPF has been widely studied recently to satisfy the real-time requirements of applications like economic dispatch, security analysis, etc. However, traditional data-driven models are typically trained for a specific system topology. When the system topology changes, the models must either be retrained (which demands a substantial amount of training data) or fine-tuned (which necessitates the selection of an appropriate pre-trained model). To this end, we propose a two-stage approach for topology change-aware data-driven OPF. It consists of: 1) generating data-driven models using a topology transfer framework; and 2) ensembling well-trained models. In Stage 1, GPR is employed to capture the nonlinear correlation between the new and predicted OPF data. The new data is obtained by solving the OPF problem using traditional optimization solvers under the new topology; the predicted data is obtained by inputting the same power demand into the data-driven OPF model trained on one of the historical datasets. This framework allows us to obtain sample-efficient topology transfer models. In Stage 2, a dynamic ensemble learning strategy is developed, where the weights and the topology transfer models that need to be ensembled are dynamically determined. This strategy allows us to avoid obtaining biased OPF solutions from sub-models. Numerical experiments on the modified IEEE 14- and TAS 97-bus test systems demonstrate that the proposed approach can obtain optimality-enhanced and equality function-satisfied OPF solutions as compared to other data-driven approaches.**

***Index Terms*—topology transfer framework, Gaussian process regression, Optimal power flow, dynamic ensemble learning**

ACRONYM LIST

OPF	Optimal Power Flow
DC-OPF	Direct-Current Optimal Power Flow
DNN	Deep Neural Networks
SELM	Stacked Extreme Learning Machines
GNN	Graph Neural Networks
CNN	Convolutional Neural Networks
PSCOPF	Preventive Security-Constrained Optimal Power Flow
CGAN	Conditional Generative Adversarial Network
MDP	Markov Decision Process
GCNN	Graph Convolutional Neural Network
GPR	Gaussian Process Regression
RBF	Radial Basis Function
TTF-OPF	Topology Transfer Framework based-OPF
OFE	Objective Function Error
PFEE	Power Flow Equation Error
MVR	Maximum Violation Rate
MVE	Mean Violation Error
TSU	Time Speed Up
BST	Block Spent Time
AQS	Average Quantile Score
AWS	Average Winkler Score
DSS	Deep Statistical Solvers
TCA-GNN	Topology Change Aware-GNN
SOCP-OPF	Second Order Cone Relaxation for OPF
MPCC	Mean Pearson Correlation Coefficient
MMIC	Maximal Information Coefficient
RF	Regression-Enhanced Random Forests
LMPs	Locational Marginal Prices

The work was supported in part by the Research Grants Council of the Hong Kong SAR (HKU 27203723), and in part by the Guangdong Basic and Applied Basic Research Foundation (2024A1515011266). (*Corresponding author: Yi Wang*)

Yixiong Jia, and Yi Wang are with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong SAR, China (e-mail: jiayx@connect.hku.hk, yiwang@eee.hku.hk).

Xian Wu is with the Department of Weiyang College, Tsinghua University, Beijing 100084, China (e-mail: wuxian21@mails.tsinghua.edu.cn).

Zhifang Yang is with the State Key Laboratory of Power Transmission Equipment Technology, College of Electrical Engineering, Chongqing University, Chongqing 400044, China. (e-mail: zfyang@cqu.edu.cn).

I. INTRODUCTION

SOLVING OPF is crucial for dispatch, security analysis, and other applications of the power grid. In the past, the OPF problem was solved by utilizing the interior-point method. However, due to the nonconvexity and nonlinearity of the OPF problem, it is difficult to satisfy the real-time requirements of such applications. To accelerate the solving process, the power grid solves other convex relaxation or linearization forms of the OPF problem, such as DC-OPF. However, solving such simplified OPF problems may obtain sub-optimal or infeasible solutions. Fortunately, with the acquired historical OPF dataset and advanced data-driven methods, it becomes possible to obtain an optimality and feasibility-enhanced OPF solution in real time by shifting the computational burden from online optimization to offline training.

Nowadays, many data-driven methods have been proposed to solve the OPF problem. Most of these methods are based on neural networks such as DNN [1] [2], SELM [3], GNN [4], and CNN [5]. The existing data-driven OPF methods can be classified into ***supervised learning approach***, ***unsupervised learning approach***, and ***reinforcement learning approach***.

For the ***supervised learning approach***, it can be further classified into *regression* and *classification*. Before introducing the core idea, it should be noted that ***supervised learning approach*** highly rely on the dataset which is constructed by using traditional solver. Even though the traditional solver may provide suboptimal OPF solutions when the initialization is far from the global optimal solution, it is the consensus to

treat this dataset as the optimal one in the model training. In addition, most existing methods are training and testing based on homothetic OPF dataset, which may provide bad OPF solutions for a practical power grid. In [6], some essential metrics (load distribution, generators outages, and etc.) are considered in the dataset construction step to provide non-homothetic OPF samples, which attract attentions recently.

The core idea behind the *regression* is to learn the mapping between the input (particularly the power demand) and the output (part or whole of the OPF solution). Then, the rest of the OPF solution can be obtained from the post-processing step (optional). In [7], two deep neural networks named voltage magnitude predictor and voltage angle predictor are utilized to predict the voltage magnitude and voltage angle, respectively. Then, the remaining OPF solution can be obtained based on the power flow equation. For the regression approach, because of the prediction error of the data-driven model, the inequality constraint of OPF may be violated. To improve the feasibility of the solution, different methods such as penalty approach [8] [9], post-processing approach [10] [11], mapping approach [12], and implicit layer embedded approach [13] [14] [15] are proposed. See [16] for the recent feasibility-enhanced data-driven OPF methods. Meanwhile, to alleviate the substantial computational burden of the training process for large power systems, feature reduction [17], compact learning [18], and decentralized (distributed) machine-learning [19] [20] methods are proposed. See [18] for the recent scalability-adapted data-driven OPF methods.

The core idea behind the *classification* is to identify the active or inactive set (inequality constraint or contingencies). Then, the OPF solution can be obtained by solving a machine learning assisted-OPF problem. In [21], the inactive voltage magnitude and branch flow constraints are identified using two classification neural networks. In [22], a relevant set is constructed. Then, this set is used in the intermediate step to obtain the final solution. In [23], a four-layer neural network is utilized to predict the binding contingencies vector. Then, this vector is utilized in the PSCOPF solver to obtain the final solution.

The core idea behind the ***unsupervised learning approach*** is to train a CGAN based on the dataset composed of feasible samples. Then feasible OPF solutions with different system costs can be generated by inputting the power demand and conditional vector. Additional refinement is also needed to find the optimal solutions. In [24], a machine learning model with three neural network components is proposed based on information theoretic generative adversarial networks. Then, an iterative update step is utilized to find the solutions that can minimize the approximated Lagrangian values of the OPF problem. In [25], the feasibility filter layer, comparison layer, and gradient-guided layer are embedded into the generative adversarial networks to improve the quality of generated solutions.

The core idea behind the ***reinforcement learning approach*** is to let the agent (generally the neural network) interact with the environment and learn how to achieve the best action (optimal solutions). In [26], a relaxed Lagrangian function is utilized as the action-value function. In [27], the power flow solver and convex safety layer are combined into the actor to provide safe action (feasible solutions). In [28], the multi-period OPF problem of the distribution network is formulated as a MDP first. Then, proximal policy optimization is utilized to solve the MDP.

Compared to using the interior-point method or solving the simplified OPF problem, the aforementioned data-driven OPF methods achieve high computational efficiency. However, these methods are designed for a specific system configuration, e.g., a fixed topology. In fact, the transmission lines may be switched to ensure the economical and reliable dispatch of the system [29] [30]. That means the aforementioned data-driven methods may fail to provide a correct and reliable OPF solution when the system's topology changes. To address the topology-change issue, the topology change-aware data-driven OPF methods are proposed, which can be classified into ***prior information embedded learning approach*** and ***sample-efficient learning approach***.

Note that the following methods only focus on the phenomenon that the topology changes are caused by the human operation, which means the topology information is known and the traditional solver can be directly utilized to obtain the training samples. Our method also focuses on this phenomenon, and the phenomenon that the contingencies cause the topology changes is out of the scope of the paper.

The core idea behind the ***prior information embedded learning approach*** is to incorporate the system's physical information into the model to guide the model's training. Once the topology changes, the model transfer can be achieved by modifying the parameter matrix and fine-tuning the model. In [31], the topology label and load data are input to the CNN to get the OPF solution. In [32], the line admittance and load data are used to train DNN, which account for any power network with the same bus, generation, and line capacity configurations but different topologies. In [33] and [34], the power flow equations and weighted adjacency matrix are embedded in the GCNN to extract the topology and physical features, respectively. It should be noted that some promising approaches to address the topology change issue of power flow calculation are proposed in recent year [35] [36]. Such methods try to learn the power flow mapping for a specific power grid with various topology, which can be seen as a potential solution for topology change-aware data-driven OPF.

The core idea behind the ***sample-efficient learning approach*** is to make the model quickly adapt to the new topology by using meta-learning or incorporating the sensitivity information into the model. In [37], with the historical dataset under different topologies, the meta-learning approach is utilized to find a good initialization weight for DNN. Then, the weight vector is used to fine-tune the model. In [38], a sensitivity-informed DNN is proposed, which aims not only to learn the mapping between the input and OPF solution but also match the relationship between the input and the Jacobian matrix.

Overall, the traditional data-driven OPF methods can provide OPF solutions in real time. However, when the system's topology changes, existing methods would face a dilemma: either ***retraining*** [38] or ***fine-tuning*** [33] [34] [37] the model. If

model retraining is utilized to accommodate this phenomenon, a large volume of training samples needs to be obtained under the new topology. Conversely, if model fine-tuning is utilized, the transfer route (transfer from one of the historical topologies to the new topology) needs to be determined carefully. An inappropriate transfer route would make the model provide sub-optimal OPF solution.

To address this dilemma, we propose a two-stage approach for topology change-aware data-driven OPF. The key contributions of this paper are summarised as follows:

- ***New framework (in Stage 1):*** Propose a topology transfer framework to enable efficient data-driven model training. It is achieved by augmenting the predicted OPF solutions to the input and also designing a specific kernel function according to the correlation analysis. Such changes can ensure the transfer framework not only learn the mapping between the power demand and the OPF solutions but also capture the nonlinear correlation between the new and predicted OPF solutions. In addition, any type of regression model can be utilized in our framework, which can be flexibly used for the power grid to obtain the OPF solution under the new topology.
- ***New ensemble strategy (in Stage 2):*** On the basis of Stage 1, develop a dynamic ensemble learning strategy based on the results of undersampled region identification. Even though the ensemble learning method has been widely utilized in the power system's application, to best of our knowledge, it is the first time to utilizing ensemble learning method in the data-driven OPF. In particular, the weights are calculated based on each topology transfer model's performance on a small validation set. Once the uncertainty of all the topology transfer model's output is larger than the threshold, the weights will be updated. In addition, the covariance is considered in the variance functions ensemble, which can improve the accuracy of the uncertainty quantification.

The rest of the paper is organized as follows: Section II gives the basic problem description of OPF and two questions this paper wants to answer. Section III describes the proposed two-stage approach. Section IV shows the evaluation metrics. Section V applies the proposed approach to the modified IEEE 14- and TAS 97-bus systems. Section VI draws conclusions.

II. Problem Statement

The OPF problem considering topology changes can be formulated as follows. Note that, for a practical power grid, the cost objective is influenced by different factors like line losses or fuel supplies. Following the existing paper [10] [32], to simplify the problem, the impact of such factors is ignored in our paper.

$$\min \sum_{i\in\Omega_G}\left(a_{2,i}\cdot(PG_i)^2+a_{1,i}\cdot PG_i+a_{0,i}\right) \tag{1a}$$

$$PG_i - PD_i = \sum_{j\in i} P_{ij}\left(\boldsymbol{V},\boldsymbol{\theta},\boldsymbol{A}^l,\boldsymbol{Z}^l\right), \forall i\in\Omega_B, l\in\Omega_T \tag{1b}$$

$$QG_i - QD_i = \sum_{j\in i} Q_{ij}\left(\boldsymbol{V},\boldsymbol{\theta},\boldsymbol{A}^l,\boldsymbol{Z}^l\right), \forall i\in\Omega_B, l\in\Omega_T \tag{1c}$$

$$PG_i^{\min}\le PG_i\le PG_i^{\max}, \forall i\in\Omega_G \tag{1d}$$

$$QG_i^{\min}\le QG_i\le QG_i^{\max}, \forall i\in\Omega_G \tag{1e}$$

$$V_i^{\min}\le V_i\le V_i^{\max}, \forall i\in\Omega_B \tag{1f}$$

$$\sqrt{P_{ij}^2+Q_{ij}^2}\le PL_{ij}^{\max}, \forall i,j\in\Omega_B \tag{1g}$$

where (1a) represents the objective function; $\Omega_G, \Omega_B, \Omega_T$ are the generator set, bus set and topology set, respectively; PG_i, QG_i are the active and reactive power generation at bus i, respectively; (1b) to (1c) represent the power flow equation; PD_i, QD_i are the active and reactive power demand at bus i, respectively; $\boldsymbol{V}, \boldsymbol{\theta}$ are the voltage magnitude and voltage angle vector, respectively; $\boldsymbol{A}^l, \boldsymbol{Z}^l$ represents the incidence matrix and impedance matrix, respectively; l is the index to represent the topology number of the system; P_{ij}, Q_{ij} are active and reactive power flow for branch connected bus i and bus j, respectively; (1d) to (1g) represent the power, voltage magnitude, and power flow constraint; $V_i^{\min}, V_i^{\max}$ are minimum and maximum voltage magnitude for bus i; $PG_i^{\min}, PG_i^{\max}$ are minimum and maximum active power generation for generator i; $QG_i^{\min}, QG_i^{\max}$ are minimum and maximum reactive power generation for generator i; $PL_{ij}^{\max}$ is the maximum power flow limit for branch ij.

To clarify the problem statement, we define $\boldsymbol{D}^l = \left\{\left(\boldsymbol{X}^l,\boldsymbol{Y}^l\right)\right\} = \left\{\left(\boldsymbol{X}_k^l,\boldsymbol{Y}_k^l\right)\right\}_{k=1}^{n_{\text{Train}}^l}$ as the historical dataset obtained under the topology $\boldsymbol{A}^l$. $\boldsymbol{X}_k^l \in \mathcal{R}^{1\times 2\|\Omega_B\|}$ contains the power demand and $\boldsymbol{Y}_k^l \in \mathcal{R}^{1\times(2\|\Omega_B\|+2\|\Omega_G\|-2)}$ contains the OPF solution such as active power generation, reactive power generation, voltage magnitude, and voltage angle. n_{Train}^l is the number of training samples and $l\in\{1,2,\cdots,s-1\}$, where $s-1$ is the number of historical topologies. $\|\Omega_B\|, \|\Omega_G\|$ are the number of buses and generators in the system, respectively.

With the acquired historical dataset $\boldsymbol{D}^l$, the data-driven OPF methods aim to learn the mapping from $\boldsymbol{X}^l$ to $\boldsymbol{Y}^l$:

$$\boldsymbol{Y}^l = f^l\left(\boldsymbol{X}^l;\boldsymbol{\beta}^l\right) \tag{2}$$

where $\boldsymbol{\beta}^l$ denotes the parameter of the mapping.

A well-trained $f^l(\cdot)$ provides OPF solutions in real time by inputting the current power demand. However, when a new topology h is occurring, it is necessary to determine whether model retraining or fine-tuning will be utilized to accommodate the topology change issue. Two questions can be answered to make the data-driven OPF methods applicable to the new topology:

1) How to design a sample-efficient training framework under the new topology to get topology transfer models?
2) How to avoid artificial transfer route selection to obtain a "high quality" OPF solution?

III. Methodology

Before introducing the proposed two-stage approach, we would like to give some definitions. Throughout this paper, we define $\boldsymbol{D}^l$ and $\boldsymbol{D}^h$ as the historical dataset (has enough samples) and new dataset (has limited samples), respectively; define $\hat{f}^l(\cdot)$ as the well-trained historical model; define $f^h(\cdot)$ as the new model. In particular, we define $\hat{f}^l\left(\boldsymbol{X}^h\right)$ as the predicted OPF solution under the topology A^l.

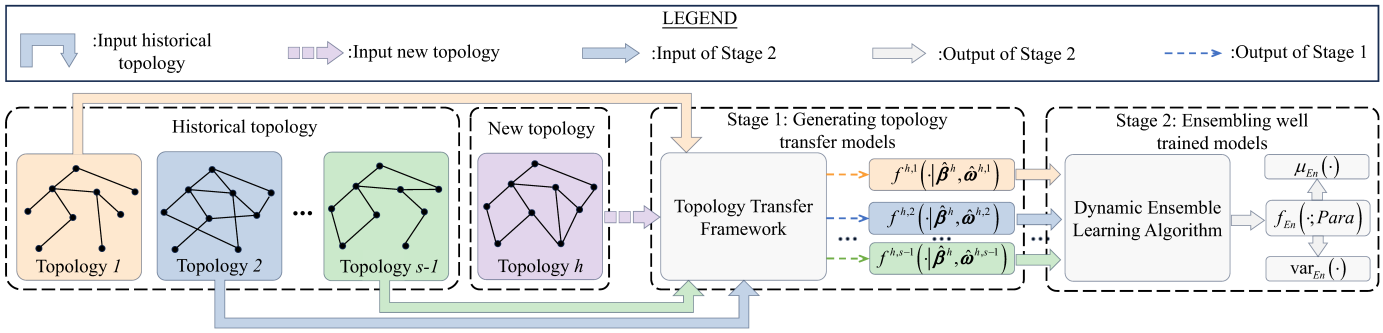

Fig. 1. Proposed two-stage approach

Fig. 1 provides an overview of the proposed two-stage approach. Specifically, the subsection III-A introduces Stage 1, which aims to construct a topology transfer framework by capturing the non-linear correlation between the new and predicted OPF solutions to answer question 1). On the basis of Stage 1, subsection III-B introduces Stage 2, which aims to develop a dynamic ensemble learning strategy to ensemble the topology transfer models to answer question 2).

A. Topology Transfer Framework based-OPF

Under the limited training sample scenario, it is easy to come up with the idea that we can learn $f^h(\cdot)$ by using an efficient learning-type method. Since GPR has been proven to be efficient in the presence of limited training samples, it is suitable in this scenario, which can be formulated as (3). Note that all the GPR used in the following are batch-independent multi-output Gaussian processes.

$$f^h\left(\boldsymbol{X}^h;\boldsymbol{\beta}^h\right)=G_1\left(\boldsymbol{X}^h\right)^T\cdot\boldsymbol{\beta}^h \tag{3}$$

where $\boldsymbol{\beta}^h\sim\mathcal{N}\left(\boldsymbol{\theta},\boldsymbol{\Sigma}^h\right)$ is the weight vector with Gaussian prior; $\boldsymbol{\Sigma}^h$ is the covariance matrix; $G_1(\cdot)$ is the basis function.

Since for every positive definite covariance function $Ke(\cdot,\cdot)$, there exists an infinite expansion in terms of basis functions, an equivalent formulation of (3) is,

$$f^h\left(\boldsymbol{X}^h;\sigma^h,\boldsymbol{\Lambda}^h\right)\sim\mathcal{N}\left(\boldsymbol{\theta},\left(\sigma^h\right)^2\cdot Ke\left(\boldsymbol{X}^h,\boldsymbol{X}^h;\boldsymbol{\Lambda}^h\right)\right) \tag{4}$$

where $\sigma^h,\boldsymbol{\Lambda}^h$ are the parameters that need to be estimated, and $Ke\left(\cdot,\cdot;\boldsymbol{\Lambda}^h\right)$ represents the covariance function.

A common choice of covariance function is the RBF, which can be formulated as,

$$Ke\left(\boldsymbol{X}_b^h,\boldsymbol{X}_c^h;\boldsymbol{\Lambda}^h\right)=\exp\left[-\frac{1}{2}\left(\boldsymbol{X}_{b-c}^h\right)\left(\boldsymbol{\Lambda}^h\right)^{-1}\left(\boldsymbol{X}_{b-c}^h\right)^T\right] \tag{5}$$

where $b,c\le n_{\text{Train}}^h$; $\boldsymbol{X}_{b-c}^h=\boldsymbol{X}_b^h-\boldsymbol{X}_c^h$; $\boldsymbol{\Lambda}^h=diag\left(\boldsymbol{\gamma}^h\right)$; $\boldsymbol{\gamma}^h=\left[\gamma_1^h,\cdots,\gamma_{2\|\Omega_B\|}^h\right]$ represents the parameter of the RBF.

When the model $f^h\left(\boldsymbol{X}^h;\sigma^h,\boldsymbol{\Lambda}^h\right)$ is well trained, the posterior distribution at the test input $\boldsymbol{X}^*$ is,

$$f^h\left(\boldsymbol{X}^*;\hat{\sigma}^h,\hat{\boldsymbol{\Lambda}}^h\right)\Big|\,\boldsymbol{D}^h\sim\mathcal{N}\left(\mu^h\left(\boldsymbol{X}^*\right),\text{var}^h\left(\boldsymbol{X}^*\right)\right) \tag{6a}$$

$$\mu^h\left(\boldsymbol{X}^*\right)=Ke^{*,h}\cdot\left[Ke^{h,h}\right]^{-1}\cdot\boldsymbol{Y}^h \tag{6b}$$

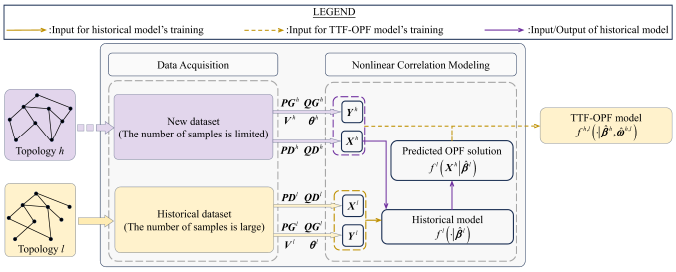

Fig. 2. The structure of TTF-OPF

$$\text{var}^h\left(\boldsymbol{X}^*\right)=\left(\hat{\sigma}^h\right)^2\cdot\left[Ke^{*,*}-Ke^{*,h}\cdot\left[Ke^{h,h}\right]^{-1}\cdot Ke^{h,*}\right] \tag{6c}$$

where $\hat{\sigma}^h,\hat{\boldsymbol{\Lambda}}^h$ are well trained parameters; $Ke^{B,C}$ is short for $Ke\left(\boldsymbol{X}^B,\boldsymbol{X}^C;\hat{\boldsymbol{\Lambda}}^h\right)$.

Even though the equations (6b) and (6c) can be used to obtain the OPF solution and quantify the uncertainty of this output under the new topology, this model may be over-fitted and provide sub-optimal OPF solutions under this limited training sample scenario.

According to Fig. 5 and Table II (refer to the correlation analysis in section V), one may observe that the new and predicted OPF solutions show a linear or nonlinear correlation, which means that the model's training efficiency can be enhanced once their correlation is well modeled. In this way, the modified data-driven model can achieve similar prediction accuracy with fewer training samples than a single GPR.

Based on this finding, the structure of the proposed framework can be visualized as shown in Fig 2, where the data acquisition block is utilized to obtain the dataset and the nonlinear correlation modeling block is utilized to capture the correlation between the new and predicted OPF solution. Note that l can be anyone in the historical set.

Given that our aim is to enable the regression model efficient training under the limited training sample scenario, the GPR is utilized to model our framework. The motivation for using GPR is threefold. First, the data efficiency of GPR has been proven in many research areas, like robotics, aircraft, etc. Second, the GPR can not only provide the mean prediction but can also quantify the uncertainty of such output. It is important since uncertainty quantification can be used to identify the undersampled area and can also be used to judge if we can use the OPF solution obtained from the topology transfer

model in our further applications. Moreover, the GPR has been utilized to learn the OPF mapping in [39] [40]. Such successful implementation motivates us in using GPR under the scenario.

Based on the (3), the mathematical formulation of TTF-OPF can be derived as follows:

$$f^{h,l}\left(\boldsymbol{X}^{h,l};\boldsymbol{\beta}^{h},\boldsymbol{\omega}^{h,l}\right)=\left(\hat{W}^{l}\left(\boldsymbol{X}^{h}\right)\right)^{T}\cdot\boldsymbol{\omega}^{h,l}+f^{h}\left(\boldsymbol{X}^{h};\boldsymbol{\beta}^{h}\right) \tag{7}$$

where $\boldsymbol{X}^{h,l}=\left(\boldsymbol{X}^{h},\hat{f}^{l}\left(\boldsymbol{X}^{h}\right)\right)$ is defined as the augment input; $\boldsymbol{\omega}^{h,l}\sim\mathcal{N}\left(\boldsymbol{\mu}_{\omega}^{h,l},\boldsymbol{\Sigma}^{h,l}\right)$; $\hat{W}^{l}\left(\boldsymbol{X}^{h}\right)=G_{2}\left(\hat{f}^{l}\left(\boldsymbol{X}^{h}\right)\right)$; $G_2(\cdot)$ represents the other basis function; $\hat{f}^{l}\left(\boldsymbol{X}^{h}\right)$ is short for $f^{l}\left(\boldsymbol{X}^{h};\hat{\boldsymbol{\beta}}^{l}\right)$; $f^{l}\left(\cdot;\hat{\boldsymbol{\beta}}^{l}\right)$ represents the well-trained data-driven model based on the historical dataset $\boldsymbol{D}^{l}$, and any regression model can be used for $f^{l}\left(\cdot;\hat{\boldsymbol{\beta}}^{l}\right)$.

To simplify the derivation, we assume $\hat{W}^{l}\left(\boldsymbol{X}^{h}\right)$ and $\boldsymbol{\mu}_{\omega}^{h,l}$ are independent in the following pages. Define the augmented dataset as $\bar{\boldsymbol{D}}^{h,l}=\left\{\left(\boldsymbol{X}^{h,l},\boldsymbol{Y}^{h}\right)\right\}$, and assume the independent property between the first term and the second term in the right part of (7), the equivalent formulation of (7) is,

$$f^{h,l}\left(\boldsymbol{X}^{h,l};\boldsymbol{Pa}\right)\sim\mathcal{N}\left(\mathbb{E}\left(\hat{W}^{l}\left(\boldsymbol{X}^{h}\right)\right)^{T}\cdot\boldsymbol{\mu}_{\omega}^{h,l},\left(\sigma^{h,l}\right)^{2}\cdot\boldsymbol{R}^{h,h}\right) \tag{8}$$

$$\begin{aligned}\boldsymbol{R}_{b,c}^{h,h}&=Ke\left(\boldsymbol{X}_{b}^{h},\boldsymbol{X}_{c}^{h};\boldsymbol{\Lambda}^{\rho}\right)\cdot Ke\left(\hat{f}^{l}\left(\boldsymbol{X}_{b}^{h}\right),\hat{f}^{l}\left(\boldsymbol{X}_{c}^{h}\right);\boldsymbol{\Lambda}^{h,l}\right)\\&+\rho\cdot Ke\left(\hat{f}^{l}\left(\boldsymbol{X}_{b}^{h}\right),\hat{f}^{l}\left(\boldsymbol{X}_{c}^{h}\right);\boldsymbol{\Lambda}^{h,l}\right)+Ke\left(\boldsymbol{X}_{b}^{h},\boldsymbol{X}_{c}^{h};\boldsymbol{\Lambda}^{h}\right)\end{aligned} \tag{9}$$

where the bc_{th} entries of $\boldsymbol{R}^{h,h}$ is defined as (9), which aims to not only learn the mapping between the $\boldsymbol{X}^{h}$ and $\boldsymbol{Y}^{h}$ but also capture the nonlinear correlation between the $\hat{f}^{l}\left(\boldsymbol{X}^{h}\right)$ and $\boldsymbol{Y}^{h}$; $\boldsymbol{Pa}=\left\{\boldsymbol{\mu}_{\omega}^{h,l},\sigma^{h,l},\boldsymbol{\Lambda}^{\rho},\boldsymbol{\Lambda}^{h,l},\boldsymbol{\Lambda}^{h},\rho\right\}$ are parameters will be determined based on the dataset $\bar{\boldsymbol{D}}^{h,l}$; $\mathbb{E}(\cdot)$ is the function to calculate the expectation of the variable.

Since the kernel function design is also a "hyper-parameter" [41], the equation (9) is constructed based on the correlation analysis in Section V. Take the second term as an example, one may observe that this term will tend towards to 1 when $\hat{f}^{l}\left(\boldsymbol{X}_{b}^{h}\right),\hat{f}^{l}\left(\boldsymbol{X}_{c}^{h}\right)$ are very close and $\hat{\rho}=1$. In this case, the new and predicted OPF solutions show a highly linear relationship, which means that an accurate new OPF solution can be easily obtained based on the predicted OPF solutions.

Since the effectiveness of using radial basis function to learn the mapping of OPF problem is proven in [39] [40] and meanwhile the rational quadratic or matern kernel are equivalent to the RBF in some setting [42], we choose RBF to construct the final kernel function (9).

To better understand (7) and interpret why the additional term can help to achieve efficient training under limited training samples scenario, suppose the $f^{h,l}(\cdot)$ and $f^{l}(\cdot)$ are well-trained based on the dataset $\bar{\boldsymbol{D}}^{h,l}$ and $\boldsymbol{D}^{l}$, respectively.

For a test input $\boldsymbol{X}^{*}$ under the new topology, the joint distribution can be formulated as,

$$\begin{pmatrix}\boldsymbol{Y}^{h}\\ \hat{f}^{h,l}\left(\boldsymbol{X}^{*,l}\right)\end{pmatrix}\sim\mathcal{N}\left(\begin{bmatrix}\mathbb{E}\left(\hat{W}^{l}\left(\boldsymbol{X}^{h}\right)\right)^{T}\\ \mathbb{E}\left(\hat{W}^{l}\left(\boldsymbol{X}^{*}\right)\right)^{T}\end{bmatrix}\cdot\hat{\boldsymbol{\mu}}_{\omega}^{h,l},\boldsymbol{R}\right) \tag{10a}$$

$$\boldsymbol{R}=\begin{bmatrix}\boldsymbol{R}^{h,h}&\boldsymbol{R}^{h,*}\\ \boldsymbol{R}^{*,h}&R^{*,*}\end{bmatrix} \tag{10b}$$

where $\hat{f}^{h,l}\left(\boldsymbol{X}^{*,l}\right)$ is short for $f^{h,l}\left(\boldsymbol{X}^{*,l};\widehat{\boldsymbol{Pa}}\right)$; $\boldsymbol{X}^{*,l}=\left(\boldsymbol{X}^{*},\hat{f}^{l}\left(\boldsymbol{X}^{*}\right)\right)$; $\widehat{\boldsymbol{Pa}}=\left\{\hat{\boldsymbol{\mu}}_{\omega}^{h,l},\hat{\sigma}^{h,l},\hat{\boldsymbol{\Lambda}}^{\rho},\hat{\boldsymbol{\Lambda}}^{h,l},\hat{\boldsymbol{\Lambda}}^{h},\hat{\rho}\right\}$ represents the well-trained parameters.

According to the properties of conditional multivariate normal distribution, the posterior distribution at the test input $\boldsymbol{X}^{*}$ is,

$$f^{h,l}\left(\boldsymbol{X}^{*,l};\widehat{\boldsymbol{Pa}}\right)\Big|\,\bar{\boldsymbol{D}}^{h,l}\sim\mathcal{N}\left(\mu^{h,l}\left(\boldsymbol{X}^{*,l}\right),\mathrm{var}^{h,l}\left(\boldsymbol{X}^{*,l}\right)\right) \tag{11a}$$

$$\begin{aligned}\mu^{h,l}\left(\boldsymbol{X}^{*,l}\right)=&\underbrace{\mathbb{E}\left(\hat{W}^{l}\left(\boldsymbol{X}^{*}\right)\right)^{T}\cdot\hat{\boldsymbol{\mu}}_{\omega}^{h,l}}_{Nonlinear-Term}+\underbrace{\boldsymbol{R}^{*,h}\cdot\left[\boldsymbol{R}^{h,h}\right]^{-1}\cdot\boldsymbol{Y}^{h}}_{Linear-Trem}\\&\underbrace{-\boldsymbol{R}^{*,h}\cdot\left[\boldsymbol{R}^{h,h}\right]^{-1}\cdot\left[\mathbb{E}\left(\hat{W}^{l}\left(\boldsymbol{X}^{h}\right)\right)^{T}\cdot\hat{\boldsymbol{\mu}}_{\omega}^{h,l}\right]}_{Correction-Term}\end{aligned} \tag{11b}$$

$$\mathrm{var}^{h,l}\left(\boldsymbol{X}^{*,l}\right)=\left(\hat{\sigma}^{h,l}\right)^{2}\cdot\left[R^{*,*}-\boldsymbol{R}^{*,h}\cdot\left[\boldsymbol{R}^{h,h}\right]^{-1}\cdot\boldsymbol{R}^{h,*}\right] \tag{11c}$$

Compared to (6b), instead of solely getting the OPF prediction from the linear combination of observation data $\boldsymbol{Y}^{h}$, three items are combined to provide the OPF mean prediction in (11b). The ***Nonlinear-Term*** captures the nonlinear correlation between the new and predicted OPF solutions to enhance the prediction accuracy. The role of the ***Linear-Term*** is similar to the (6b). The ***Correction-Term*** is used to correct the error by using the limited training samples.

One thing needs to be declared in (11b): If $\hat{f}^{l}(\cdot)$ is modeled by a deterministic regression method, the expectation symbol can be deleted in this equation, and the $\hat{W}^{l}(\cdot)$ could be seen as the nonlinear function of predicted output $\hat{f}^{l}(\cdot)$. However, if the $\hat{f}^{l}(\cdot)$ is modeled by a probabilistic regression method like GPR, the expectation symbol cannot be deleted directly. In this scenario, since the RBF is equivalent to the Hermite polynomials basis function, the term $\mathbb{E}\left(\hat{W}^{l}(\cdot)\right)$ is also the polynomials function of the expectation of $\hat{f}^{l}(\cdot)$. That means the interpretation above is also satisfied.

For the variance part, the formulation of (11c) is similar to (6c). However, since the correlation is captured by using a more complicated kernel function, the uncertainty quantification could be more accurate than only using a single GPR.

B. Dynamic Ensemble Learning Algorithm

The subsection III-A introduces a TTF-OPF based on the transfer route from historical topology l to the new topology h. Since we assume $s-1$ historical topologies with their training dataset are known, $s-1$ TTF-OPF models can be constructed step by step. However, since the training samples and their number in each historical dataset are different, the $s-1$ TTF-OPF models may show different performance even when tested on the same dataset. It means that selecting an appropriate transfer route is the key step to obtaining "good enough" OPF solutions. However, since there is no general method to obtain

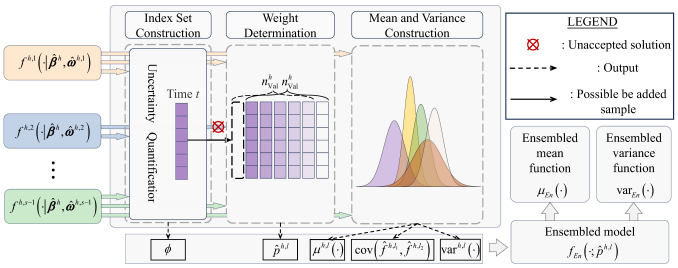

Fig. 3. The structure of dynamic ensemble learning strategy

the optimal transfer route, a biased OPF solutions may be obtained if the transfer route is selected manually.

To address this issue, a dynamic ensemble learning strategy is proposed in this subsection, which can be visualised as shown in Fig. 3. In Fig. 3, the index set construction block is utilized to quantify the uncertainty of each TTF-OPF model's output at time t and finally provide the active model set. Based on this set, the weight determination block is utilized to calculate the weights of each sub-model, where the validation dataset in this block is dynamically changed (once the active model set is empty, the test sample at time t will be solved using traditional solver and will be added to the validation dataset). Note that, to control the time costs in this block, the number of the samples in the validation dataset is set as a constant. That means, once a test sample is added to the validation dataset, then the last sample in this dataset will be removed. Then, the mean and variance construction block is utilized to obtain the mean, variance, and covariance terms. Using the output from these three blocks, one can finally obtain the ensembled model. Note that the trained parameter $\widehat{\boldsymbol{Pa}}$ is not shown in the following equations.

Following the structure in Fig. 3, the ensemble learning strategy can be formulated as follows,

$$f_{En}\left(\cdot; p^{h,l}\right) = \sum_{l\in\phi} \hat{p}^{h,l} \cdot \hat{f}^{h,l}(\cdot) \tag{12}$$

where ϕ represents the active model set and $p^{h,l}$ represents the weight of each sub-model.

Since the output of each $\hat{f}^{h,l}(\cdot)$ obeys the Gaussian distribution, the weighted linear combinations also obey the Gaussian distribution, which can be formulated as equation (13a).

$$f_{En}\left(\cdot; p^{h,l}\right) \sim \mathcal{N}\left(\mu_{En}\left(\cdot; p^{h,l}\right), \mathrm{var}_{En}\left(\cdot; p^{h,l}\right)\right) \tag{13a}$$

$$\mu_{En}\left(\cdot; p^{h,l}\right) = \sum_{l\in\phi} p^{h,l} \cdot \mu^{h,l}\left(\cdot\right) \tag{13b}$$

$$\begin{aligned}\mathrm{var}_{En}\left(\cdot; p^{h,l}\right) &= \sum_{l\in\phi} \left(p^{h,l}\right)^2 \cdot \mathrm{var}^{h,l}\left(\cdot\right) \\ &+ \sum_{\substack{l_1, l_2 \in \phi \\ l_1 \neq l_2}} p^{h,l_1} \cdot p^{h,l_2} \cdot \mathrm{cov}\left(\hat{f}^{h,l_1}(\cdot), \hat{f}^{h,l_2}(\cdot)\right)\end{aligned} \tag{13c}$$

where (13b) and (13c) represent the mean and variance function of the ensembled model, respectively; $\mathrm{cov}\left(\cdot,\cdot\right)$ calculate the covariance between two random variables.

The model (12) can be regarded as the ensemble of $\|\phi\|$ sub-models, and the idea is consistent with the basic ensemble learning. However, there are three main differences between our and the basic ensemble learning methods.

1) The number of ensembled models is fixed for the basic ensemble learning method. However, the number is dynamically changed for our method; the set ϕ is defined to construct the active model set, which is determined based on the undersampled area identification. In other words, once the uncertainty quantification of the sub-model is larger than the pre-defined threshold, then the OPF solution obtained from this sub-model cannot be utilized.

2) The value of $p^{h,l}$ is constant and not changed for the basic ensemble learning method. However, for our method, the value of $p^{h,l}$ is determined based on the sub-model's performance on a small validation dataset, which is dynamically changed. The reason is easy to understand. When the load condition changed (e.g. from light load to heavy load), if we still use fixed weights, the prediction accuracy of the ensembled model may be compromised.

3) Only the mean prediction is ensembled for the basic ensemble learning method. However, for our method, the variance function is also ensembled. Specifically, we do not assume that the outputs from different TTF-OPF models are independent, which could provide more accurate uncertainty quantification results.

Given the three differences, we will show the details of how to determine ϕ, $\hat{p}^{h,l}$, and how to ensemble the variance function.

To determine the set ϕ, a threshold Tr is utilized, which can be formulated as follows,

$$\phi = \left\{l : \mathrm{var}^{h,l}\left(\boldsymbol{X}^{*,l}\right) \leq Tr, l \in \{1, \cdots, s-1\}\right\} \tag{14}$$

To determine the value of $p^{h,l}$, we split the dataset $\boldsymbol{D}^h$ as $\left\{\boldsymbol{D}^h_{\text{Train}}, \boldsymbol{D}^h_{\text{Val}}\right\}$, where the $\boldsymbol{D}^h_{\text{Train}}$ is utilized for model training and $\boldsymbol{D}^h_{\text{Val}}$ is utilized to validate the performance of each sub-model. The number of data points in $\boldsymbol{D}^h$ can be denoted as $n^h = n^h_{\text{Train}} + n^h_{\text{Val}}$ according to the splitting of the dataset. And in this paper, we set $n^h_{\text{Val}} = 1/5 \cdot n^h$. Then, the three metrics $OFE\left(\cdot\right)$, $PFEE\left(\cdot\right)$, and $MVE\left(\cdot\right)$ are used to calculate the $p^{h,l}$, which can be formulated as,

$$\hat{p}^{h,l} = \frac{1}{OFE\left(\boldsymbol{D}^h_{\text{Val}}\right) + PFEE\left(\boldsymbol{D}^h_{\text{Val}}\right) + MVE\left(\boldsymbol{D}^h_{\text{Val}}\right)} \tag{15}$$

where the definition of these three metrics can be found in section IV.

Specifically, the weight of each sub-model needs to be normalized after all the values of $\hat{p}^{h,l}$ are calculated.

To ensemble the variance function, the equation (13c) can be utilized. Specifically, since the value of $\hat{p}^{h,l}$ is calculated based on (15) and $\mathrm{var}^{h,l}\left(\cdot\right)$ can be easily obtained from (11c), the key step is to calculate the covariance term. In this paper, since the distribution of $\hat{f}^{h,l}(\cdot)$ can be derived from (11), the covariance term can be approximated by using sample covariance, which can be formulated as follows,

$$\frac{\sum_{k=1}^{n_{Sam}} \left[\Xi\left(\hat{f}^{h,l_1}(\cdot)\right)_k - \mu^{h,l_1}\left(\cdot\right)\right] \left[\Xi\left(\hat{f}^{h,l_2}(\cdot)\right)_k - \mu^{h,l_2}\left(\cdot\right)\right]}{n_{Sam} - 1} \tag{16}$$

where n_{Sam} is the sample number, and $\Xi(\cdot)_k$ is the k_{th} data that is sampled from the corresponding distribution.

Combining the (14), (15) and (16), the final ensembled model can be obtained.

The implementation of the proposed method (including training procedure and execution procedure) is shown in Algorithm 1 and Algorithm 2.

Algorithm 1 Implementation of the proposed method (Training procedure)

Input: Historical topology set $\left\{\boldsymbol{A}^1, \boldsymbol{A}^2, \cdots, \boldsymbol{A}^{s-1}\right\}$, New topology $\boldsymbol{A}^h$

1: Initialize $l \leftarrow 1$
2: **repeat**
3: Data acquisition $\boldsymbol{D}^l \leftarrow \left\{\left(\boldsymbol{X}^l, \boldsymbol{Y}^l\right)\right\}$
$\boldsymbol{D}^h \leftarrow \left\{\left(\boldsymbol{X}^h, \boldsymbol{Y}^h\right)\right\}$
4: Historical regression model training via (2)
5: Predicted OPF solutions acquisition $\hat{f}^l\left(\boldsymbol{X}^h\right)$
6: Augmented dataset construction
$\bar{\boldsymbol{D}}^{h,l} \leftarrow \left\{\left(\boldsymbol{X}^{h,l}, \boldsymbol{Y}^h\right)\right\}$, $\boldsymbol{X}^{h,l} \leftarrow \left(\boldsymbol{X}^h, \hat{f}^l\left(\boldsymbol{X}^h\right)\right)$
7: Topology transfer model training via (8) and (9)
8: Final model acquisition $\hat{f}^{h,l}(\cdot)$
9: $l \leftarrow l+1$
10: **until** $l > s-1$

Output: Different topology transfer models
$\left\{\hat{f}^{h,1}(\cdot), \hat{f}^{h,2}(\cdot), \cdots, \hat{f}^{h,s-1}(\cdot)\right\}$
Different historical regression models
$\left\{\hat{f}^1(\cdot), \hat{f}^2(\cdot), \cdots, \hat{f}^{s-1}(\cdot)\right\}$

Algorithm 2 Implementation of the proposed method (Execution procedure)

Input: Test input $\boldsymbol{X}^*$ under the new topology

1: Get $\left\{\hat{f}^{h,1}(\cdot), \hat{f}^{h,2}(\cdot), \cdots, \hat{f}^{h,s-1}(\cdot)\right\}$ and get $\left\{\hat{f}^1(\cdot), \hat{f}^2(\cdot), \cdots, \hat{f}^{s-1}(\cdot)\right\}$ through **Algorithm 1**
2: Predicted OPF solutions acquisition
$\left\{\hat{f}^1\left(\boldsymbol{X}^*\right), \hat{f}^2\left(\boldsymbol{X}^*\right), \cdots, \hat{f}^{s-1}\left(\boldsymbol{X}^*\right)\right\}$
3: Augmented dataset construction
$\left\{\boldsymbol{X}^{*,1}, \boldsymbol{X}^{*,2}, \cdots, \boldsymbol{X}^{*,s-1}\right\}$
4: Individual OPF solutions acquisition
$\left\{\mu^{*,1}\left(\boldsymbol{X}^{*,1}\right), \mu^{*,2}\left(\boldsymbol{X}^{*,2}\right), \cdots, \mu^{*,s-1}\left(\boldsymbol{X}^{*,s-1}\right)\right\}$
5: Variance quantification
$\left\{\mathrm{var}^{*,1}\left(\boldsymbol{X}^{*,1}\right), \mathrm{var}^{*,2}\left(\boldsymbol{X}^{*,2}\right), \cdots, \mathrm{var}^{*,s-1}\left(\boldsymbol{X}^{*,s-1}\right)\right\}$
6: Active model set construction via (14)
7: **if** $\phi = \emptyset$ **then**
8: Solve this sample via traditional solver and replace the oldest sample in the validation set
9: **else**
10: Calculate the weight for active sub-model via (15)
11: Calculate the covariance for pairs sub-models via (16)
12: Obtain the final solutions via (13b) and its variance quantification via (13c)
13: **end if**

Output: Final OPF solutions $\mu_{En}\left(\boldsymbol{X}^*; \hat{p}^{h,l}\right)$
Variance quantification $\mathrm{var}_{En}\left(\boldsymbol{X}^*; \hat{p}^{h,l}\right)$

IV. Evaluation Metric

Seven metrics are designed from two perspectives: mean prediction evaluation and variance quantification evaluation. Specifically, the first five metrics are designed to examine the mean prediction performance, and the last two are designed to examine the variance quantification performance. Define $\boldsymbol{D}^h_{\text{Test}}$ as the test dataset and n^h_{Test} as the number of testing samples.

A. Metric for Mean Prediction Performance Validation

1) OFE: This metric $OFE\left(\boldsymbol{D}^h_{\text{Test}}\right)$ examines the optimality, which can be formulated as,

$$\frac{1}{n^h_{\text{Test}}}\sum_{k=1}^{n^h_{\text{Test}}}\left(\left|cost\left(\widehat{\boldsymbol{PG}}_k\right) - cost\left(\boldsymbol{PG}_k\right)\right| \Big/ cost\left(\boldsymbol{PG}_k\right)\right) \quad (17)$$

where $cost(\cdot)$ is the objective function (1a); $\widehat{\boldsymbol{PG}}_k, \boldsymbol{PG}_k$ denotes the estimated and real active power output from the generation at k_{th} testing sample, respectively.

2) PFEE: This metric $PFEE\left(\boldsymbol{D}^h_{\text{Test}}\right)$ examines the equality feasibility:

$$\frac{1}{n^h_{\text{Test}}}\sum_{k=1}^{n^h_{\text{test}}}\left|PF\left(\widehat{\boldsymbol{PG}}_k, \widehat{\boldsymbol{QG}}_k, \hat{\boldsymbol{V}}_k, \hat{\boldsymbol{\theta}}_k\right)\right| \quad (18)$$

where $\hat{\boldsymbol{V}}_k, \hat{\boldsymbol{\theta}}_k, \widehat{\boldsymbol{PG}}_k, \widehat{\boldsymbol{QG}}_k$ denotes the estimated OPF solution obtained from the method; $PF(\cdot,\cdot,\cdot,\cdot)$ denotes the power flow equations shown as (1b) to (1c).

3) MVR: This metric $MVR\left(\boldsymbol{D}^h_{\text{Test}}\right)$ examines the inequality feasibility:

$$\frac{\max\left\{g_1\left(\widehat{\boldsymbol{PG}}\right), g_1\left(\widehat{\boldsymbol{QG}}\right), g_1\left(\hat{\boldsymbol{V}}\right), 0\right\}}{\left\{\|\Omega_B\|, \|\Omega_G\|\right\}\cdot n^h_{\text{Test}}} \quad (19a)$$

$$g_1\left(\widehat{\boldsymbol{M}}\right) = \sum_{i\in\{\Omega_B,\Omega_G\}}\sum_{k=1}^{n^h_{\text{Test}}}\begin{bmatrix} g\left(VEU_{k,i}\right) \\ +g\left(VED_{k,i}\right)\end{bmatrix} \quad (19b)$$

$$VEU_{k,i} = \widehat{\boldsymbol{M}}_{k,i} - M_i^{\max}, VED_{k,i} = M_i^{\min} - \widehat{\boldsymbol{M}}_{k,i} \quad (19c)$$

where $g(\cdot)$ denotes the step function.

4) MVE: This metric $MVE\left(\boldsymbol{D}^h_{\text{Test}}\right)$ examines the inequality violation error:

$$\frac{\max\left\{g_2\left(\widehat{\boldsymbol{PG}}\right), g_2\left(\widehat{\boldsymbol{QG}}\right), g_2\left(\hat{\boldsymbol{V}}\right), 0\right\}}{\left\{\|\Omega_B\|, \|\Omega_G\|\right\}\cdot n^h_{\text{Test}}} \quad (20a)$$

$$g_2\left(\widehat{\boldsymbol{M}}\right) = \sum_{i\in\{\Omega_B,\Omega_G\}}\sum_{k=1}^{n^h_{\text{Test}}}\begin{bmatrix} VEU_{k,i}\cdot g\left(VEU_{k,i}\right) \\ +VED_{k,i}\cdot g\left(VED_{k,i}\right)\end{bmatrix} \quad (20b)$$

5) TSU: This metric $TSU\left(\boldsymbol{D}^h_{\text{Test}}\right)$ examines the speedup:

$$time\left(Pypower(\cdot)\right)/time\left(f_*(\cdot)\right) \quad (21)$$

where $time\left(Pypower(\cdot)\right), time\left(f_*(\cdot)\right)$ denote the time cost to solve the OPF problem by using the traditional solver (Pypower) and the given method, respectively.

6) BST: This metric $BST\left(\boldsymbol{D}_{\text{Test}}^{h}\right)$ examines the time spent in each block for the dynamic ensemble learning method:

$$\{time\left(ISC\right), time\left(WD\right), time\left(MVC\right)\} \tag{22}$$

where ISC, WD, MVC are short for "Index Set Construction", "Weight Determination", and "Mean and Variance Construction".

B. Metric for Variance Quantification Performance Validation

1) AQS: This metric $AQS\left(\boldsymbol{D}_{\text{Test}}^{h}\right)$ examines the average quantile score:

$$\frac{1}{n_{\text{Test}}^{h}\cdot\left\{\left\|\Omega_{B}\right\|,\left\|\Omega_{G}\right\|\right\}\cdot n_{\tau}}\sum_{k=1}^{n_{\text{Test}}^{h}}\sum_{i\in\{\Omega_{B},\Omega_{G}\}}\sum_{\tau=1}^{n_{\tau}}Q_{\tau}\left(\hat{\boldsymbol{y}}_{k,i},\boldsymbol{y}_{k,i}\right) \tag{23}$$

where $\boldsymbol{y}\in\{\boldsymbol{PG},\boldsymbol{QG},\boldsymbol{V},\boldsymbol{\theta}\}$; n_{τ} denotes the number of percentile (in this paper $n_{\tau}=9$, from 0.1 quantile to 0.9 quantile); $Q_{\tau}\left(\cdot,\cdot\right)$ denotes the τ quantile score.

2) AWS: This metric $AWS\left(\boldsymbol{D}_{\text{Test}}^{h}\right)$ examines the prediction interval:

$$\frac{1}{n_{\text{Test}}^{h}\cdot\left\{\left\|\Omega_{B}\right\|,\left\|\Omega_{G}\right\|\right\}}\sum_{k=1}^{n_{\text{Test}}^{h}}\sum_{i\in\{\Omega_{B},\Omega_{G}\}}WS_{\alpha}\left(\hat{\boldsymbol{y}}_{k,i},\boldsymbol{y}_{k,i}\right) \tag{24}$$

where $WS_{\alpha}\left(\cdot,\cdot\right)$ denotes the winkler score for α prediction intervals.

V. Numerical Tests

In this section, a modified 14- and 97-bus system is utilized to validate the effectiveness and applicability of the proposed two-stage approach.

Note that all simulations below are performed using GPyTorch on a computer CPU Intel(R) Xeon(R) W-3335 CPU 3.40 GHz, GPU NVIDIA GeForce RTX 3080 Ti, GPU memory 12G, Boost clock speed 1.67 GHz.

A. Numerical Test on the Modified 14-bus System

System Description: The single line diagram is shown in Fig. 4. The network parameters, topology, and base load values of each bus are obtained from PyPower.

Data Generation: For the 14-bus system, the load data is generated based on the load generation part in [43]. Meanwhile, three wind farms are integrated into buses 10, 11, and 14, respectively. The power output of each wind farm is generated based on the parameters and Weibull distribution in [44].

Topology Scenarios Generation: To simulate the phenomenon of topology changes, five topology scenarios are generated as shown in Table I, where $Sw_{i,j}$ denotes the switch between bus i and bus j. Topology 2 is set as the new topology, with only 70 samples obtained. For the testing dataset, 200 instances are involved.

Comparison Methods: The performance of the proposed framework is compared with six methods, including 1) purely train the model on the new dataset; 2) transfer learning, which is trained on one of the historical datasets and then fine-tuned on the new dataset; and 3) meta learning [37], which is trained on the historical dataset to find a good initial weight vector, then fine-tuning the model based on the new dataset and weight vector; 4) DSS [35], which aims to learn a solver that generalizes to a given distribution of one optimization problem in weak supervised learning manner; 5) TCA-GNN [48], which aims to achieve fast transferring by using GNN with Laplace matrix embedded; and 6) SOCP-OPF, which uses second order cone to relax the nonconvex power flow equation.

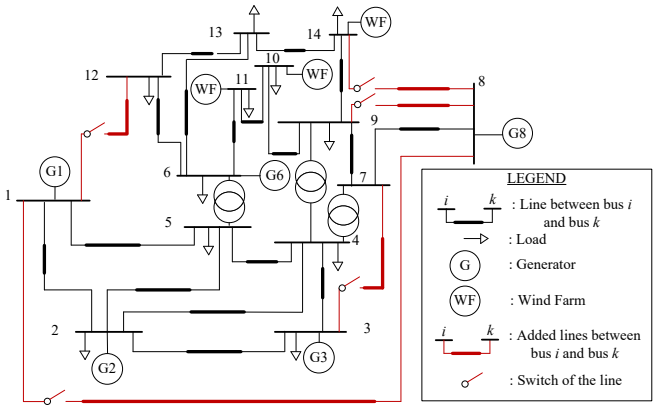

Fig. 4. Single line diagram of modified IEEE 14-bus system

TABLE I
Details of Five Topologies

Scenarios	Active switches	Number of training data
Topology 1	$Sw_{8,14}, Sw_{1,12}$	500
Topology 2	$Sw_{8,14}$	56
Topology 3	$Sw_{8,14}, Sw_{1,12}, Sw_{3,7}$	500
Topology 4	$Sw_{8,14}, Sw_{1,12}, Sw_{3,7}, Sw_{1,8}$	500
Topology 5	$Sw_{8,14}, Sw_{8,9}$	500

1) Correlation analysis: We present some simulation results on the 14-bus system to visualise the correlation between the new and predicted OPF solutions. Note that the new and predicted OPF solutions are obtained under topologies 2 and 5, respectively. By varying the active load on bus 3 and bus 4 and fixing the other parameters, the joint plot for reactive power generation is shown in Fig. 5. The results in Fig. 5 demonstrate that some of the correlation is linear (left sub-figure), while others is non-linear (right sub-figure).

To further analyze the correlation between other new and predicted OPF solutions (e.g., active power), the MPCC and MMIC are utilized. The correlation coefficients of each type OPF solution are shown in Table II. The results in Table II also confirm the findings in the last paragraph.

2) Framework effectiveness validation: Eight cases are proposed to validate the effectiveness of the proposed framework. The performance comparisons for mean prediction and variance quantification are shown in Table III and IV, respectively. In Table IV, the Q in front of the "RF", "NN", "NN-Transfer", and "Meta Learning" represents the corresponding quantile-type methods. Note that all the results in Table III and IV are based on the transfer route from topology 5 to topology 2.

Comparing "GPR" and "TTF-GPR", one may observe that

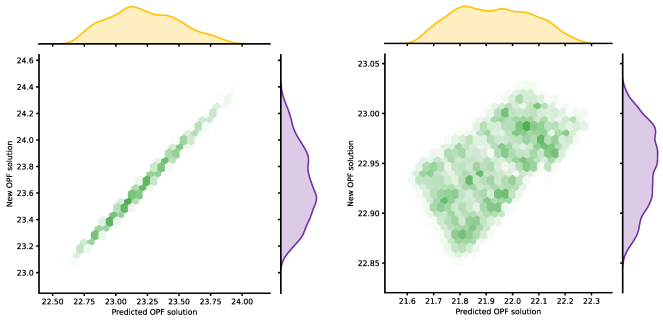

Fig. 5. Correlation analysis between new and predicted OPF solutions for (left) reactive power generation at bus 3 (right) reactive power generation at bus 2

TABLE II
THE MPCC AND MMIC BETWEEN NEW AND PREDICTED OPF SOLUTION

Metric	PG	QG	V	θ
MPCC	0.9967	0.5663	0.3376	0.9986
MMIC	0.9619	0.7058	0.5948	0.6765

the Optimality (OFE) and the Equality Feasibility (PFEE) could be improved by applying the proposed framework, which means the proposed framework could adapt to the new topology quickly compared to "GPR". However, the proposed framework may have a negative impact on Inequality Feasibility (MVR) and Speedup (TSU). For MVR, about a 4% increase can be obtained because of the prediction error from the GPR model. However, according to the MVE, one may observe that a lower value can be obtained by applying the proposed framework, which means that the 4% increase in MVR can be accepted. For TSU, the proposed framework could slow down the inferring process by a factor of about eight. This is due to the complex kernel in the framework and the additional time cost from the data-driven model's inferring (to obtain the predicted OPF solution). For the variance quantification validation, one may observe that a solid improvement to AQS and AWS can be achieved compared to the GPR, which means the proposed framework could identify the undersampled area accurately. Meanwhile, the comparisons between "RF" and "TTF-RF" and between "NN" and "TTF-NN" could also confirm the above findings. We also notice that the values of TSU for "RF" and "TTF-RF" are close. It is because "RF" requires much time for inferring compared to the time cost of applying the proposed framework.

Comparing "NN-Transfer", "Meta Learning", "TCA-GNN" and methods applying the proposed framework, one may observe that optimality enhanced and tight equality constraints satisfied OPF results can be obtained from the proposed framework, which proves that the proposed framework could adapt to the new topology more quickly than the existing sample-efficient learning approach. In addition, for variance quantification validation, one may obtain a more narrow prediction interval by applying the proposed framework, which means that a more reliable OPF solution can be obtained. Note that, since the active power generation is calculated based on the predicted LMPs in "TCA-GNN", the inequality constraints

TABLE III
PERFORMANCE COMPARISON FOR MEAN PREDICTION (14-BUS SYSTEM)

Method	OFE	PFEE	MVR	MVE	TSU
GPR	1.57E-02	5.92E-02	0.23%	2.65E-02	1777
TTF-GPR	**3.25E-03**	**8.53E-03**	4.30%	1.20E-03	209
RF	3.42E-02	1.84E-02	0.14%	3.46E-02	73
TTF-RF	6.41E-03	1.09E-02	6.40%	1.28E-02	62
NN	4.71E-02	2.62E-02	0.51%	7.17E-02	**3200**
TTF-NN	1.41E-02	1.43E-02	4.00%	3.78E-02	396
NN-Transfer	5.79E-02	2.94E-02	0.73%	5.33E-02	3000
Meta Learning	1.69E-02	2.16E-02	0.45%	4.09E-02	2666
DSS	1.77E-01	2.55E-01	**0.00%**	**0.00**	4
DSS-Sup	1.49E-01	2.66E-01	**0.00%**	**0.00**	4
TCA-GNN	8.54E-02	3.55E-02	4.11%	1.48E-05	600
SOCP-OPF	2.88E-04	1.13E-01	**0.00%**	**0.00**	18

for active power will be fully satisfied. Only the voltage magnitude inequality constraints may be violated, which is the reason why the MVE value is so small. Even though the "Meta Learning" method does not require to select the transfer route, it can be considered as the special "transfer learning" method which aims to transfer the model from one "virtual topology" to new topology. Thus, the performance of this method highly depend on the "virtual transfer route". In addition, it has been proved that "Meta Learning" requires a large number of training samples to achieve satisfying model performance [33] [47].

Comparing "DSS", "DSS-Sup" and methods applying the proposed framework, one may observe that the weak supervised learning methods or supervised learning methods fail to learn an accurate OPF mapping under this limited training sample scenario. This is mainly because there are so many parameters that need to be updated for this model, and 500 samples for each historical topology and 56 samples for the new topology are not enough. The other drawback of the "DSS" and "DSS-Sup" is long-time model training, which is also not suitable for real power grids.

Comparing "SOCP-OPF" and methods applying the proposed framework, one may observe that our method can provide equality constraints enhanced OPF solutions at a faster pace. The reason is the SOCP-OPF can provide exact OPF solutions under the radial networks. For the mesh networks, the estimation error for the voltage angle is large, which leads to a large PFEE value.

Overall, even though "NN-Transfer", "Meta Learning", "GPR", "TCA-GNN", and "NN" could provide the OPF solution in real time, they may tend to provide sub-optimal OPF solutions. In contrast, we could offer a "good enough" OPF solution with acceptable time spent on problem-solving. Note that the OPF solutions that don't satisfy the inequality constraints can be acceptable since the MVE is small for "TTF-GPR", "TTF-NN", and "TTF-RF". In addition, accurate variance quantification results can be obtained by applying the proposed framework, which can help us construct a reliable ensembled model to obtain optimality and equality feasibility-enhanced OPF solution.

3) The impact of transfer route selection: To demonstrate the impact of transfer route selection, the "TTF-GPR" is chosen. Note that the 500 training samples for each historical

TABLE IV
PERFORMANCE COMPARISON FOR VARIANCE QUANTIFICATION (14-BUS SYSTEM)

Method	AQS (PG)	AWS (PG, 80%PI)	AWS (PG, 60%PI)
GPR	8.46E-01	6.08	4.25
TTF-GPR	**1.50E-01**	**1.17**	**0.72**
QRF	2.18E+00	14.01	10.01
TTF-RF	2.53E-01	1.85	1.18
QNN	1.81E+00	9.79	7.94
TTF-NN	3.22E-01	2.67	1.58
QNN-Transfer	2.02E+00	13.58	9.45
QMeta Learning	1.17E+00	10.53	6.84

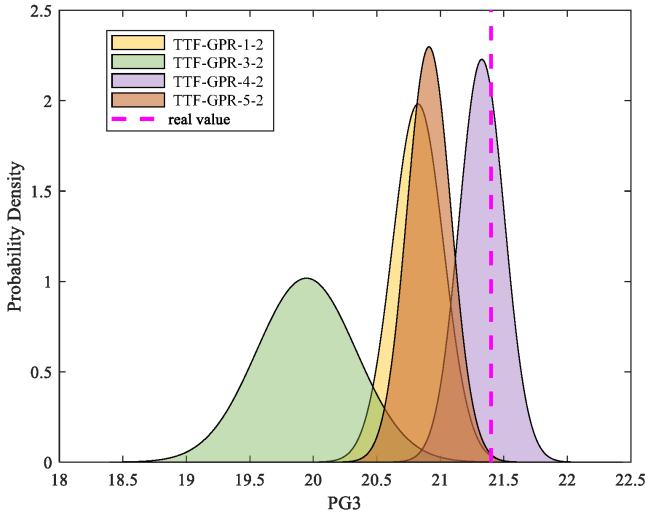

Fig. 6. "TTF-GPR" model's output comparison for generator 3 at the 50 th testing sample

dataset are set at the same load conditions in this scenario. The performance comparison is shown in Table V.

In terms of the mean prediction performance, the "TTF-GPR" trained on the transfer route from Topology 5 to Topology 2 can achieve the lowest OFE, PFEE, and MVE compared with the "TTF-GPR" methods trained on the other three transfer routes. It is mainly because topology 5 is the most similar to topology 2 according to the status of switches. All the results in Table V prove that the performance of the model may be compromised if it is trained based on an inappropriate transfer route.

In terms of the variance quantification performance, a similar finding can be obtained, which supports that selecting an appropriate transfer route is necessary.

4) Ensemble learning strategy effectiveness validation: We first plot different "TTF-GPR" model's output to demonstrate the necessity of dynamic ensemble learning, which is shown in Fig. 6. The label "TTF-GPR-x-y" represents the "TTF-GPR" model trained on the transfer route from topology "x" to topology "y". According to Fig. 6, one may observe that the variance quantification of "TTF-GPR-3-2" is larger than any other model's. Thus, the OPF solution obtained from the "TTF-GPR-3-2" cannot be utilized in the ensemble learning process; otherwise, the performance of the ensembled model may be compromised.

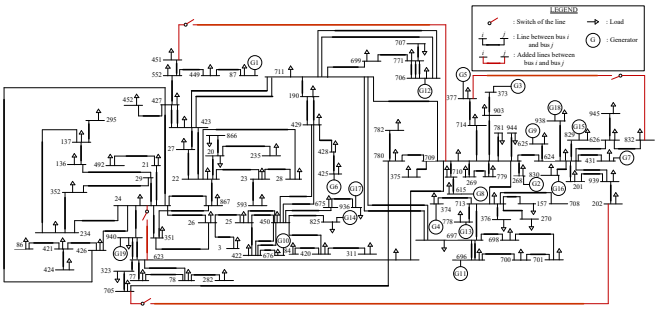

Fig. 7. Single line diagram of the modified TAS 97-bus system

To validate the effectiveness of the dynamic ensemble learning strategy, three cases are proposed. The performance comparison for different ensembled models is shown in Table VI. The "average" denotes the average ensemble learning strategy that uses average weights; "portrait divergence" denotes the weighted ensemble learning strategy where the weights are determined based on the topology similarity. Specifically, the details of how to calculate the portrait divergence between two topologies can be found in [45]. Note that all the ensemble learning strategies in Table VI are based on "TTF-GPR". To calculate the covariance, we set $n_{Sam} = 5$ in this paper.

Comparing the results shown in Table V and VI, one may observe that the performance of the ensembled model may outperform that of some "TTF-GPR" models. However, since the weights of "average" and "portrait divergence" are static, their performance may be worse than the "TTF-GPR-5-2" model. For our dynamic ensembled model, it can provide an optimality-enhanced and equality function-satisfied OPF solution, which can validate the effectiveness of the proposed ensemble learning strategy. One thing need to be noticed is that, due to the additional procedure to calculate the covariance terms (Mean and Variance Construction block: 0.06s) and to solve a few testing samples in a traditional solver (Weight Determination block: 0.30s), the proposed ensemble methods are slower than the single "TTF-GPR" models.

B. Numerical Test on the Modified TAS 97-bus System

System Description: The single line diagram is shown in Fig. 7. The network parameters, topology, and half-year's load profile of each bus can be obtained in [46].

Topology Scenarios Generation & Data Usage: Eight topology scenarios are generated, as shown in Table VII. Note that topology 5 is set as the new topology, and the other seven topologies are set as the historical. The details of data usage for each topology are shown in Fig. 8.

1) Framework effectiveness validation: Similar to the modified 14-bus system, eight cases are proposed to validate the effectiveness of the proposed framework. Due to the space limitation, we only show the performance comparison for mean prediction, as shown in Table VIII. Note that all the results in Table IV are based on the transfer route from topology 3 to topology 5. All the results confirm that the proposed framework can be used to the practical power system.

TABLE V
PERFORMANCE COMPARISON FOR "TTF-GPR" BASED ON DIFFERENT TRANSFER ROUTES

Transfer From	Transfer To	OFE	PFEE	MVR	MVE	AQS (PG)	AWS (PG, 80%PI)	AWS (PG, 60%PI)
Topology 1	Topology 2	4.13E-03	8.88E-03	5.00%	6.17E-03	1.82E-01	1.44E+00	8.70E-01
Topology 3	Topology 2	6.50E-03	9.50E-03	4.50%	6.26E-03	2.53E-01	1.80E+00	1.12E+00
Topology 4	Topology 2	9.50E-03	1.01E-02	**4.10%**	2.23E-02	5.07E-01	4.01E+00	2.66E+00
Topology 5	Topology 2	**3.25E-03**	**8.53E-03**	4.30%	**1.20E-03**	**1.36E-01**	**9.94E-01**	**6.45E-01**

TABLE VI
PERFORMANCE COMPARISON FOR DIFFERENT ENSEMBLE LEARNING METHOD (14-BUS SYSTEM)

Ensemble Method	OFE	PFEE	MVR	MVE	TSU	BST	AQS (PG)	AWS (PG, 80%PI)	AWS (PG, 60%PI)
Average	4.80E-03	6.35E-03	3.90%	5.72E-03	**169**	-	1.98E-01	8.82E-03	4.84E-03
Portrait divergence	4.02E-03	6.44E-03	5.40%	6.12E-03	**169**	-	2.10E-01	1.01E-02	5.62E-03
Proposed	**2.83E-03**	**5.96E-03**	**4.10%**	**3.90E-03**	74	{0.29 ,0.30, 0.06}	**1.43E-01**	**6.86E-03**	**3.54E-03**

TABLE VII
DETAILS OF SEVEN TOPOLOGIES

Scenarios	Active switches	Number of training data
Topology 1	$Sw_{451,709}, Sw_{377,832}$	2000
Topology 2	$Sw_{377,832}$	2500
Topology 3	$Sw_{451,709}, Sw_{377,832}$ $Sw_{24,623}$	3000
Topology 4	$Sw_{451,709}, Sw_{24,623}$ $Sw_{202,705}$	1500
Topology 5	$Sw_{24,623}, Sw_{202,705}$	200
Topology 6	$Sw_{24,623}$	3000
Topology 7	$Sw_{451,709}, Sw_{24,623}$ $Sw_{377,832}, Sw_{202,705}$	1500
Topology 8	$Sw_{202,705}$	1500

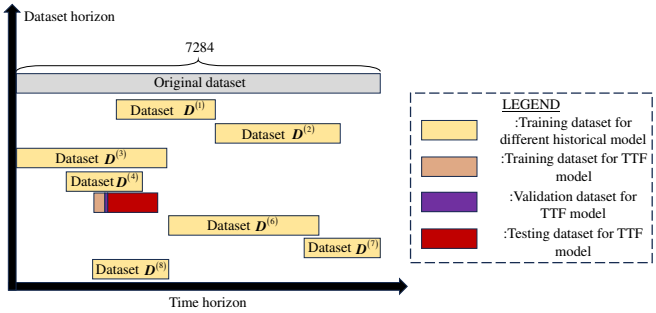

Fig. 8. Data usage for different topologies

2) Ensemble learning strategy effectiveness validation: Three cases are proposed to validate the effectiveness of the proposed ensemble learning strategy. The performance comparison for mean prediction is shown in Table IX. The sample number n_{Sam} is set as 5 in this case. All the results confirm that the proposed ensemble learning strategy can provide a "good enough" OPF solution with an acceptable time cost.

C. Discussion

The number of training samples has a large impact on the proposed framework. Enough training samples may improve the performance of the topology transfer model but may require a lot of time to construct the dataset and to train the model. To assess the concept and also quantify the interval of "limited training sample", the impact of training samples to our proposed method is studied.

According to Fig. 9, one may observe that the quantification interval of "limited training sample" highly depends on the type of historical model. For example, if we use "GPR" as

TABLE VIII
PERFORMANCE COMPARISON FOR MEAN PREDICTION (97-BUS SYSTEM)

Method	OFE	PFEE	MVR	MVE	TSU
GPR	1.27E-02	4.72E-02	0.99%	1.66E-02	337
TTF-GPR	**1.46E-03**	**2.12E-02**	0.85%	9.69E-03	138
RF	3.47E-02	2.42E-02	1.04%	5.74E-02	39
TTF-RF	2.16E-03	2.67E-02	1.06%	2.01E-02	22
NN	4.23E-02	4.67E-02	1.21%	5.43E-02	**2424**
TTF-NN	2.06E-03	2.62E-02	0.70%	4.03E-03	213
NN-Transfer	1.40E-02	4.79E-02	0.64%	3.22E-02	2024
Meta Learning	1.08E-02	4.70E-02	0.60%	3.14E-02	1948
DSS	3.30E-01	2.52E-01	5.28%	2.42E-01	7
DSS-Sup	3.06E-01	4.04E-01	7.18%	1.21E-01	7
TCA-GNN	9.58E-02	4.56E-02	4.31%	5.84E-05	457
SOCP-OPF	1.92E-04	1.48E+00	**0.00%**	**0.00**	10

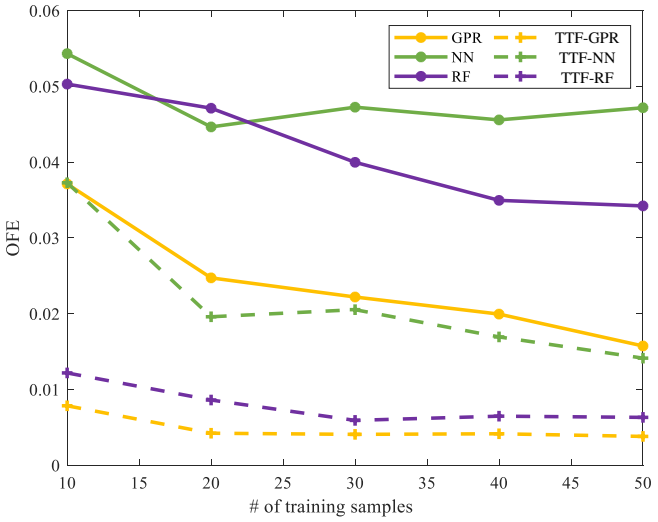

Fig. 9. The influence of training samples number for the proposed topology transfer framework

TABLE IX
PERFORMANCE COMPARISON FOR DIFFERENT ENSEMBLE LEARNING METHOD (97-BUS SYSTEM)

Ensemble Method	OFE	PFEE	MVR	MVE	TSU	BST
Average	1.30E-03	1.97E-02	0.84%	5.98E-03	**127**	-
Portrait divergence	2.03E-03	2.41E-02	0.78%	7.17E-03	**127**	-
Proposed	**1.24E-03**	**1.90E-02**	**0.72%**	**5.82E-03**	95	{3.13, 0.60, 0.45}

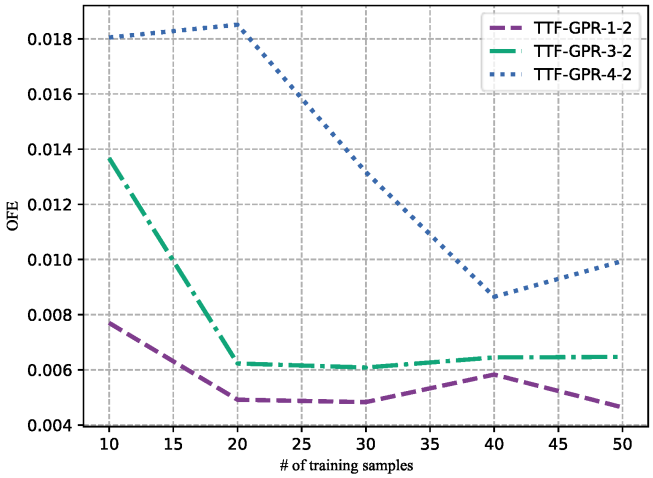

Fig. 10. The influence of samples for model training considering different transfer routes

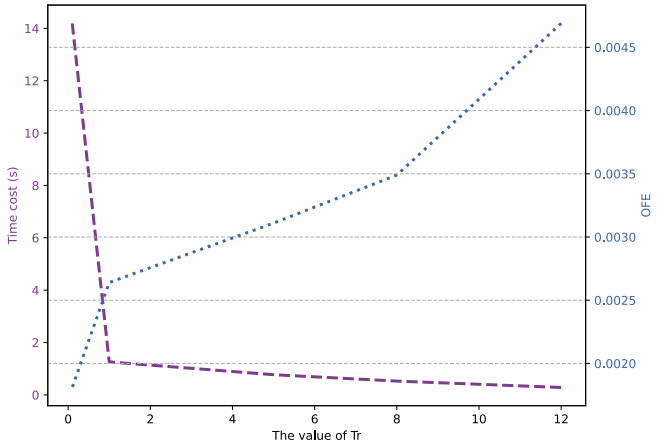

Fig. 11. The impact of threshold Tr for the proposed dynamic ensemble learning algorithm

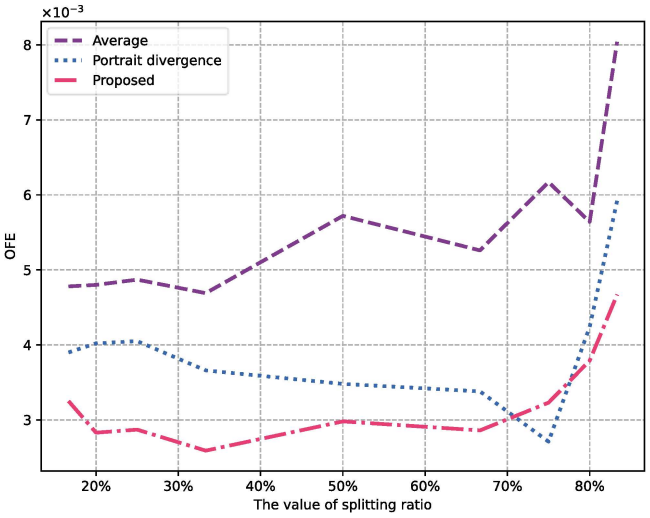

Fig. 12. The impact of threshold Tr for the proposed dynamic ensemble learning algorithm

a historical model to obtain the predicted OPF solution, then "TTF-GPR" can show satisfying performance with only 20 training samples, and even if we use more training samples, the performance of "TTF-GPR" may not be improved so much. However, if we use "NN" as a historical model, 50 training samples are not enough for the "TTF-NN".

Meanwhile, we conduct a simulation test on the same system to study the impact of the transfer route on the quantification interval of "limited training sample", which can be visualized as Fig. 10. One may observe that the quantification interval of "limited training sample" also depends on the transfer route. For example, 20-50 samples are enough for model TTF-GPR-3-2 training, but 50 samples are not even enough for model TTF-GPR-1-2 training.

Even though the quantification interval of "limited training sample" is difficult to derive from one theory, we found that 20-40 samples are suitable for our method's training since almost every sub-model shows good performance compared to other data-driven methods.

Note that the dimension of the feature vector would also have an impact on the quantification interval of "limited training sample". In this paper, we did not consider the case of islands or node outages caused by topology changes. Thus, the quantification interval of "limited training sample" on one specific system and fixed transfer route will not be influenced by the complexity of the feature vector.

In addition to finding the quantification interval of "limited training samples", we also test the impact of threshold Tr for the proposed dynamic ensemble learning algorithm. Note that all the results in Fig. 11 are based on the modified 14-bus system.

According to Fig. 11, one may observe that the larger the value of the threshold, the less time cost and the higher OFE for our ensemble model. Meanwhile, it is recommended to set Tr between 2 and 8, which can balance the time cost and performance of the ensemble model

The impact of splitting ratio for the ensemble learning algorithm is also tested, as shown in Fig. 12. Note that all the results are based on the modified 14-bus system, and Tr is set to 2 to balance the time cost and the model's performance.

According to Fig. 12, one may observe that our ensemble

learning method is robust towards the hyper-parameter "splitting ratio" (between 20% and 70%) compared to the other two ensemble methods. It is mainly because when the "splitting ratio" is set to higher than 70%, the number of training samples cannot support us in training the sub-models well.

D. Future Work

To avoid manually selecting the transfer route, another promising way is to find the optimal transfer route. Generally, finding the optimal transfer route requires measuring the similarity (new and historical topology, new and historical dataset, etc.) and solving a meaningful optimization problem. Meanwhile, how to theoretically prove this transfer route is optimal may also be important. It will be part of our future work to address this question.

It should be noted that all the simulation results are obtained based on the homothetic OPF Dataset. However, the "Generators cost varies" or "Generators outages" [6] may occur in the real power grid, which can provide non-homothetic OPF samples. To adapt our method to the non-homothetic scenario, we will try to combine the proposed method and classification neural networks in our future work.

According to the simulation results, one may observe that the feasibility of solutions cannot be guaranteed by the proposed method. It is important to obtain a feasible OPF solution for a practical power grid. Thus, we will try to combine the proposed framework and optimization layer to achieve this goal.

It should be noted that our method belongs to the supervised learning-type method, which highly relies on the training dataset generation. To alleviate this time-consuming process, studying how to construct an unsupervised learning or reinforcement learning model to achieve topology change-aware OPF problem solving will be part of our future work.

VI. Conclusions

We propose a two-stage approach to address a dilemma that the traditional data-driven model will face when the system's topology changes. In Stage 1, a topology transfer framework is proposed, where the correlation between the new and predicted OPF solutions can be fully captured. In Stage 2, a dynamic ensemble learning strategy is proposed to avoid obtaining biased OPF solutions. Numerical experiments based on the modified IEEE 14-system demonstrate that the proposed framework can provide an optimality-enhanced and equality function-satisfied OPF solution compared to the other data-driven methods like "NN-Transfer" and "Meta Learning". Numerical experiments based on the modified TAS 97-bus systems demonstrate the proposed framework and ensemble learning strategy can be used in the practical power system. In addition, we also find that the minimum training sample requirement for the proposed framework depends on the type of historical model used.

References

[1] X. Pan, M. Chen, T. Zhao and S. H. Low, "DeepOPF: A Feasibility-Optimized Deep Neural Network Approach for AC Optimal Power Flow Problems," IEEE Syst. J., vol. 17, no. 1, pp. 673-683, March 2023.

[2] A. S. Zamzam and K. Baker, "Learning Optimal Solutions for Extremely Fast AC Optimal Power Flow," 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Tempe, AZ, USA, 2020, pp. 1-6.

[3] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao and H. Yu, "Data-Driven Optimal Power Flow: A Physics-Informed Machine Learning Approach," IEEE Trans. Power Syst., vol. 36, no. 1, pp. 346-354, Jan. 2021.

[4] T. Pham and X. Li, "Reduced Optimal Power Flow Using Graph Neural Network," 2022 North American Power Symposium (NAPS), Salt Lake City, UT, USA, 2022, pp. 1-6.

[5] K. Yang, W. Gao and R. Fan, "Optimal Power Flow Estimation Using One-Dimensional Convolutional Neural Network," 2021 North American Power Symposium (NAPS), College Station, TX, USA, 2021, pp. 1-6.

[6] N. Popli, et al. "On the robustness of machine-learnt proxies for security constrained optimal power flow solvers." Sustainable Energy Grids Networks, 37 (2024): 101265.

[7] W. Huang, X. Pan, M. Chen and S. H. Low, "DeepOPF-V: Solving AC-OPF Problems Efficiently," IEEE Trans. Power Syst., vol. 37, no. 1, pp. 800-803, Jan. 2022.

[8] F. Ferdinando, T. W. K. Mak, and P. V. Hentenryck. "Predicting ac optimal power flows: Combining deep learning and lagrangian dual methods," Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 1, 2020.

[9] R. Nellikkath, and S. Chatzivasileiadis, "Physics-Informed Neural Networks for AC Optimal Power Flow," Electr. Power Syst. Res., Vol. 212, pp. 108412, November 2022.

[10] K. Baker, "Learning Warm-Start Points For Ac Optimal Power Flow," 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), Pittsburgh, PA, USA, 2019, pp. 1-6.

[11] X. Pan, T. Zhao, M. Chen and S. Zhang, "DeepOPF: A Deep Neural Network Approach for Security-Constrained DC Optimal Power Flow," IEEE Trans. Power Syst., vol. 36, no. 3, pp. 1725-1735, May 2021.

[12] M. Li, S. Kolouri and J. Mohammadi, "Learning to Solve Optimization Problems With Hard Linear Constraints," IEEE Access, vol. 11, pp. 59995-60004, 2023.

[13] M. Kim and H. Kim, "Projection-aware deep neural network for dc optimal power flow without constraint violations," 2022 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), 2022, pp. 116–121.

[14] P. Donti, D. Rolnick, and J. Z. Kolter, "DC3: A learning method for optimization with hard constraints," International Conference on Learning Representations, 2021.

[15] K. Hongseok, "Self-supervised Equality Embedded Deep Lagrange Dual for Approximate Constrained Optimization," arXiv preprint arXiv:2306.06674 (2023).

[16] E. Liang, M. Chen, and S. Low. "Low Complexity Homeomorphic Projection to Ensure Neural-Network Solution Feasibility for Optimization over (Non-) Convex Set," in ICML, 2023.

[17] T. W. Mak, F. Fioretto, and P. Van Hentenryck. "Load Encoding for Learning AC-OPF." arXiv preprint arXiv:2101.03973 (2021).

[18] S. Park, W. Chen, T. W. K. Mak and P. Van Hentenryck, "Compact Optimization Learning for AC Optimal Power Flow," IEEE Trans. Power Syst., vol. 39, no. 2, pp. 4350-4359, March 2024.

[19] M. Li, S. Kolouri and J. Mohammadi, "Learning to Optimize Distributed Optimization: ADMM-based DC-OPF Case Study," 2023 IEEE Power & Energy Society General Meeting (PESGM), Orlando, FL, USA, 2023, pp. 1-5.

[20] T. W. K. Mak, M. Chatzos, M. Tanneau and P. V. Hentenryck, "Learning Regionally Decentralized AC Optimal Power Flows With ADMM," IEEE Trans. Smart Grid, vol. 14, no. 6, pp. 4863-4876, Nov. 2023.

[21] F. Hasan, A. Kargarian and J. Mohammadi, "Hybrid Learning Aided Inactive Constraints Filtering Algorithm to Enhance AC OPF Solution Time," IEEE Trans. Ind. Appl., vol. 57, no. 2, pp. 1325-1334, March-April 2021.

[22] M. Sidhant, L. Roald, and Y. Ng. "Learning for constrained optimization: Identifying optimal active constraint sets." INFORMS Journal on Computing 34.1 (2022): 463-480.

[23] N. Popli, E. Davoodi, F. Capitanescu and L. Wehenkel, "Machine Learning Based Binding Contingency Pre-Selection for AC-PSCOPF Calculations," IEEE Trans. Power Syst., vol. 39, no. 2, pp. 4751-4754, March 2024.

[24] J. Wang and P. Srikantha, "Fast Optimal Power Flow With Guarantees via an Unsupervised Generative Model," IEEE Trans. Power Syst., vol. 38, no. 5, pp. 4593-4604, Sept. 2023.

[25] Y. Li, C. Zhao, C. Liu, Model-informed generative adversarial network (mi-gan) for learning optimal power flow, arXiv preprint arXiv:2206.01864 (2022).

[26] Z. Yan and Y. Xu, "Real-Time Optimal Power Flow: A Lagrangian Based Deep Reinforcement Learning Approach," IEEE Trans. Power Syst., vol. 35, no. 4, pp. 3270-3273, July 2020.
[27] A. R. Sayed, C. Wang, H. I. Anis and T. Bi, "Feasibility Constrained Online Calculation for Real-Time Optimal Power Flow: A Convex Constrained Deep Reinforcement Learning Approach," IEEE Trans. Power Syst., vol. 38, no. 6, pp. 5215-5227, Nov. 2023.
[28] D. Cao et al., "Deep Reinforcement Learning Based Approach for Optimal Power Flow of Distribution Networks Embedded with Renewable Energy and Storage Devices," J. Mod. Power Syst. Clean Energy, vol. 9, no. 5, pp. 1101-1110, September 2021.
[29] S. Babaeinejadsarookolaee, B. Park, B. Lesieutre and C. L. DeMarco, "Transmission Congestion Management via Node-Breaker Topology Control," IEEE Syst. J., vol. 17, no. 3, pp. 3413-3424, Sept. 2023.
[30] S. Babaeinejadsarookolaee, B. Park, B. Lesieutre and C. L. DeMarco, "Effective Congestion Management via Network Topology Optimization Based on Node-Breaker Representations," 2022 North American Power Symposium (NAPS), Salt Lake City, UT, USA, 2022, pp. 1-6.
[31] Y. Jia, X. Bai, L. Zheng, Z. Weng and Y. Li, "ConvOPF-DOP: A Data-Driven Method for Solving AC-OPF Based on CNN Considering Different Operation Patterns," IEEE Trans. Power Syst., vol. 38, no. 1, pp. 853-860, Jan. 2023.
[32] M. Zhou, M. Chen and S. H. Low, "DeepOPF-FT: One Deep Neural Network for Multiple AC-OPF Problems With Flexible Topology," IEEE Trans. Power Syst., vol. 38, no. 1, pp. 964-967, Jan. 2023.
[33] M. Gao, J. Yu, Z. Yang and J. Zhao, "A Physics-Guided Graph Convolution Neural Network for Optimal Power Flow," IEEE Trans. Power Syst., vol. 39, no. 1, pp. 380-390, Jan. 2024.
[34] T. Falconer and L. Mones, "Leveraging Power Grid Topology in Machine Learning Assisted Optimal Power Flow," IEEE Trans. Power Syst., vol. 38, no. 3, pp. 2234-2246, May 2023.
[35] D. Balthazar, et al. "Deep statistical solvers." Advances in Neural Information Processing Systems, 33 (2020): 7910-7921.
[36] D. Balthazar, et al. "Neural networks for power flow: Graph neural solver." Electr. Power Syst. Res, 189 (2020): 106547.
[37] Y. Chen, S. Lakshminarayana, C. Maple and H. V. Poor, "A Meta-Learning Approach to the Optimal Power Flow Problem Under Topology Reconfigurations," IEEE Open Access J. Power Energy, vol. 9, pp. 109-120, 2022.
[38] M. K. Singh, V. Kekatos and G. B. Giannakis, "Learning to Solve the AC-OPF Using Sensitivity-Informed Deep Neural Networks," IEEE Trans. Power Syst., vol. 37, no. 4, pp. 2833-2846, July 2022.
[39] P. Pareek and H. D. Nguyen, "Gaussian Process Learning-Based Probabilistic Optimal Power Flow," IEEE Trans. Power Syst., vol. 36, no. 1, pp. 541-544, Jan. 2021.
[40] M. Jalali, M. K. Singh, V. Kekatos, G. B. Giannakis and C. -C. Liu, "Fast Inverter Control by Learning the OPF Mapping Using Sensitivity-Informed Gaussian Processes," IEEE Trans. Smart Grid, vol. 14, no. 3, pp. 2432-2445, May 2023.
[41] J. Liu and P. Srikantha, "Kernel Structure Design for Data-Driven Probabilistic Load Flow Studies," IEEE Trans. Smart Grid, vol. 13, no. 4, pp. 2679-2689, July 2022.
[42] D. David. "Automatic model construction with Gaussian processes". Doctoral dissertation, 2014.
[43] M. Chatzos, T. W. K. Mak and P. V. Hentenryck, "Spatial Network Decomposition for Fast and Scalable AC-OPF Learning," IEEE Trans. Power Syst., vol. 37, no. 4, pp. 2601-2612, July 2022.
[44] Y. Li, W. Li, W. Yan, J. Yu and X. Zhao, "Probabilistic Optimal Power Flow Considering Correlations of Wind Speeds Following Different Distributions," IEEE Trans. Power Syst, vol. 29, no. 4, pp. 1847-1854, July 2014.
[45] J. P. Bagrow, E. M. Bollt, "An information-theoretic, all-scales approach to comparing networks," Appl. Network Sci., vol. 4, no. 1, pp. 1-15, July 2019.
[46] T. H. A. Cheung, M. Zhou, M. Chen, "Learning-based AC-OPF Solvers on Realistic Network and Realistic Loads," arXiv preprint arXiv: 2205.09452 (2022).
[47] T. Petru, et al. "Transfer learning with gaussian processes for bayesian optimization." International Conference on Artificial Intelligence and Statistics. PMLR, 2022.
[48] S. Liu, C. Wu and H. Zhu, "Topology-Aware Graph Neural Networks for Learning Feasible and Adaptive AC-OPF Solutions," IEEE Trans. Power Syst., vol. 38, no. 6, pp. 5660-5670, Nov. 2023.

Yixiong Jia (Member, IEEE) received the B.S. degree from North West Agriculture and Forestry University in June 2019, and the M.S. degree from ShanghaiTech University in June 2022. He is currently working toward the Ph.D. degree with The University of Hong Kong. His research interests include data-driven optimal power flow and power flow linearization in transmission systems.

Xian Wu is currently an undergraduate student with the Department of Weiyang College, Tsinghua University. His research interests include power system optimization and DER aggregation.

Zhifang Yang (Senior Member, IEEE) received his Ph.D. degree in electrical engineering from Tsinghua University in 2018. He currently works as a Full Professor at Chongqing University. His research interests include power system optimization and electricity market.

Yi Wang (Senior Member, IEEE) received the B.S. degree from Huazhong University of Science and Technology in June 2014, and the Ph.D. degree from Tsinghua University in January 2019. He was a visiting student with the University of Washington from March 2017 to April 2018. He served as a Postdoctoral Researcher in the Power Systems Laboratory, ETH Zurich from February 2019 to August 2021.

He is currently an Assistant Professor with the Department of Electrical and Electronic Engineering, The University of Hong Kong. His research interests include data analytics in smart grids, energy forecasting, multi-energy systems, Internet-of-things, cyber-physical-social energy systems.