

# Machine Learning at the Grid Edge: Data-Driven Impedance Models for Model-Free Inverters

Yufei Li, *Member, IEEE*, Yicheng Liao, *Member, IEEE*, Liang Zhao, *Student Member, IEEE*,  
Minjie Chen, *Senior Member, IEEE*, Xiongfei Wang, *Fellow, IEEE*, Lars Nordström, *Senior Member, IEEE*,  
Prateek Mittal, *Senior Member, IEEE*, H. Vincent Poor, *Fellow, IEEE*

**Abstract**—The future electric grid is supported by a vast number of smart inverters interfacing with distributed energy resources at the edge. These inverters' dynamics are typically characterized as impedances, which are crucial for ensuring grid stability and resiliency. However, the physical implementation of these inverters may change significantly from inverters to inverters and may be kept confidential. Existing analytical impedance models require a complete and precise understanding of system parameters. They can hardly capture the complete electrical behaviors when the inverters are performing complex functions. Online impedance measurements for many inverters across multiple operating points are not scalable. To address these issues, we present *InvNet*, a machine learning framework to systematically evaluate the effectiveness of data-driven methods for modeling inverter impedance patterns across a wide operation range, even with limited impedance data. Leveraging transfer learning, the *InvNet* can extrapolate from physics-based models to real-world ones and from one inverter to another with very limited data. This framework demonstrates machine learning as a powerful tool for modeling and analyzing black-box characteristics of grid-tied inverter systems that cannot be accurately described by traditional analytical methods, such as inverters under model predictive control. Comprehensive evaluations were conducted to verify the effectiveness of the *InvNet* in various scenarios. All data and models were open-sourced<sup>1</sup>.

**Index Terms**—Grid edge, impedance, model-free inverter, machine learning, transfer learning.

## I. INTRODUCTION

THE global energy system is witnessing a paradigm shift, where fossil fuels are being replaced by renewable energy resources in electricity production. Power electronics converters at the grid edge are playing more important roles than ever

before [1], [2]. These inverters at the grid edge are ubiquitously needed to integrate renewable energy resources, battery storage systems, loads, and more, to the future power grid. The grid-tied inverters' dynamics are digitally programmed with sophisticated control algorithms, and they tend to interact with one another or with other power system components, bringing unprecedented challenges to grid stability and security [3].

The dynamics of grid-tied inverters are commonly characterized as *impedances* under small-signal perturbations, which are essentially dynamic representations of inverters in the frequency domain [4], [5]. Such impedance representation is essential for system stability analysis, such as using the generalized Nyquist criterion [6]. The operating conditions of these inverters may change across a wide range, leading to various impedance patterns requiring precise modeling for system-level analyses. Physics-based small-signal models derived from circuit analyses can only capture inverter impedances when all physical parameters for simplified operating conditions are known and when inverters are controlled by classic control strategies [7], [8]. These traditional analytical models usually require a comprehensive understanding of both control and system parameters, such as parameters of circuit components, bandwidths and intrinsic structures of the PLL, and architectures of the control system, etc., such that the impedance model can be analytically derived [8].

However, the physical parameters of grid-tied inverters usually change with particular hardware and/or software implementations and operating conditions. Different power-stage designs and control structures may lead to physical parameter deviations and nonidealities, limiting the efficacy of physics-based analytical models when significant nonidealities and/or non-modellable nonlinearities exist. Examples of nonidealities and nonlinearities that cannot be captured by analytical models include variable frequency switching, control dead-time, quantization error, sampling error, digital delay, nonlinear control framework, and system parameter discrepancies, etc. In addition, grid-tied inverters are often equipped with nonlinear controllers when performing smart and sophisticated functions (e.g., low-voltage ride-through, active power sharing, etc.). Therefore, impedance modeling of grid-tied inverters via analytical models are prone to be unreliable and inaccurate [7], especially when system parameters are unknown, or sometimes kept confidential by power converter manufacturers due to security and intellectual property considerations. In a future power grid with a large number of inverters at the edge, it is impractical to assume that the physical parameters of inverters

This work was supported by the C3.ai Digital Transformation Institute as a collaboration between Princeton University and KTH Royal Institute of Technology. (Corresponding Author: Minjie Chen and Xiongfei Wang.)

Y. Li is with the State Key Laboratory of Electrical Insulation and Power Equipment, School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China, he was with the Department of Electrical and Computer Engineering and the Andlinger Center for Energy and the Environment, Princeton University, Princeton, NJ, 08544, USA (e-mail: yflee@xjtu.edu.cn).

Y. Liao is with Energinet, Fredericia, Denmark (e-mail: yli@energinet.dk).

L. Zhao is with the Department of Energy (AAU Energy), Aalborg University, Aalborg 9220, Denmark (e-mail: lzh@energy.aau.dk).

M. Chen, P. Mittal, and H. Vincent Poor are with the Department of Electrical and Computer Engineering and the Andlinger Center for Energy and the Environment, Princeton University, Princeton, NJ, 08544, USA (e-mail: minjie, pmittal, poor@princeton.edu).

X. Wang and L. Nordström are with the Department of Electrical Engineering, KTH Royal Institute of Technology, Stockholm, Sweden (e-mail: xiongfei, larsno@kth.se).

<sup>1</sup>The simulation and data acquisition platform, the TensorFlow implementation of the *InvNet* framework, and the database supporting the findings of this work are all publicly available at <https://github.com/superrabbit2023/InvNet>.

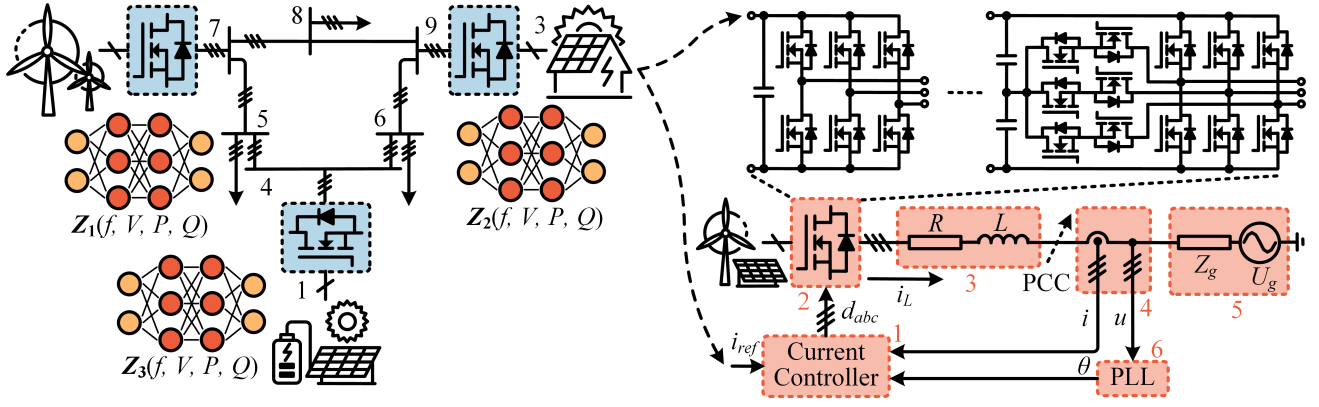


Fig. 1. Key Principles of the *InvNet* framework. Left: modeling inverter impedances as neural networks for stability analyses of distribution networks comprising a variety of inverters. Right: the system diagram of typical grid-tied inverters with PLL-based current control, where nonidealities and nonlinearities in the physical and control systems (such as sampling errors, parameter discrepancies, and dead-time, etc.) make precise impedance modeling highly challenging.

are known to system operators, let alone analytical models or control strategies. Even though the analytical impedance model exists and the complete system information is known, to date, precise impedance modeling is still being throttled by difficulties in modeling nonideal-switching impacts on inverter impedances, such as the dead-time impacts [9]. For many existing and emerging inverter implementations adopting nonlinear controllers, such as the model predictive control (MPC) [10], there are no analytical models to describe the impedance behaviors thus far, due to their intrinsically sophisticated functions and nonlinear behaviors.

In addition, the inverter impedance can change significantly depending on the operating conditions of power grid. Measuring or simulating the impedances of clusters of inverters in the real world is possible [8], but expensive and impractical. The acquisition process of real-world impedance data of thousands of inverters across a wide range of operating points (OPs, including voltage- $V$ , frequency- $f$ , real power- $P$ , reactive power- $Q$ ) via deterministic signal processing methods, such as experiments or electromagnetic transient (EMT) simulations, requires massive human efforts and computational resources. Multi-time-scale EMT simulation cannot fully capture the system dynamics ranging from 60 Hz to a few hundred kilohertz. Frequently measuring inverter impedances online may create stability and security concerns to the grid. Real-world impedance measurement needs to sweep all investigated frequency points (FP) and injects perturbations twice for each FP, then sweep all investigated OPs. Moreover, the collected voltage and current data need to undergo the fast Fourier transform (FFT) to obtain the impedance data. Even though the pseudo-random binary sequence (PRBS) signal method [11], containing multiple frequency components can be used to expedite the frequency sweep process to some extent, the process of experimental measurements can still be very tedious when considering multiple OPs. Future power electronics converters may operate at higher frequencies with sophisticated switching actions (hard or soft switching) based on Silicon Carbide (SiC) [12] or Gallium Nitride (GaN) power devices [13], making impedance data collections even more challenging.

Data-driven methods, which are model-agnostic from circuit

parameters and control structures, become an effective alternative in addressing the above challenges. Prior work includes feed-forward neural network (FNN) for modeling the inverter impedances [14] and physics-informed FNN for reducing the data demand [15]. However, these studies oversimplify the inverter system by characterizing the inverter impedances only along single OP variables. Our prior work [16] considers a wide range of OP variables, but no experimental verification was provided. All aforementioned studies did not consider the scenario when inverters are controlled by nonlinear controllers where no analytical impedance models are available (such as modeling inverters under model predictive control).

Leveraging the recent advances in artificial intelligence and machine learning [17], [18], we present *InvNet*, a simple but effective model-free machine learning framework that is capable of characterizing inverter impedance patterns across a wide operation range 1) without a thorough or precise understanding of the physical system, or 2) if only a small amount of measurement data is available. The key principle of this approach is shown in Fig. 1. The main contributions of this paper include:

- We demonstrated an end-to-end machine learning framework for modeling the impedance of grid-tied inverters with data acquisition, model training, and performance evaluation, for a variety of inverters with wide operation range. The data acquisition system includes analytical calculations, EMT simulations, and experimental measurements. The neural network (NN) training framework includes NN architecture design, parameter tuning, and transfer learning.
- We showcased that it was effective to model the impedance of grid-tied inverters with NN models when 1) the analytical model exists, but the model parameters are unknown, 2) the analytical model does not exist or is not mature, and 3) there is no sufficient data available for modeling a particular inverter.

The remainder of this article is organized as follows: Section II elaborates on the proposed approach in detail; Section III gives comprehensive performance evaluations of the proposed *InvNet* framework; Section IV concludes this article.

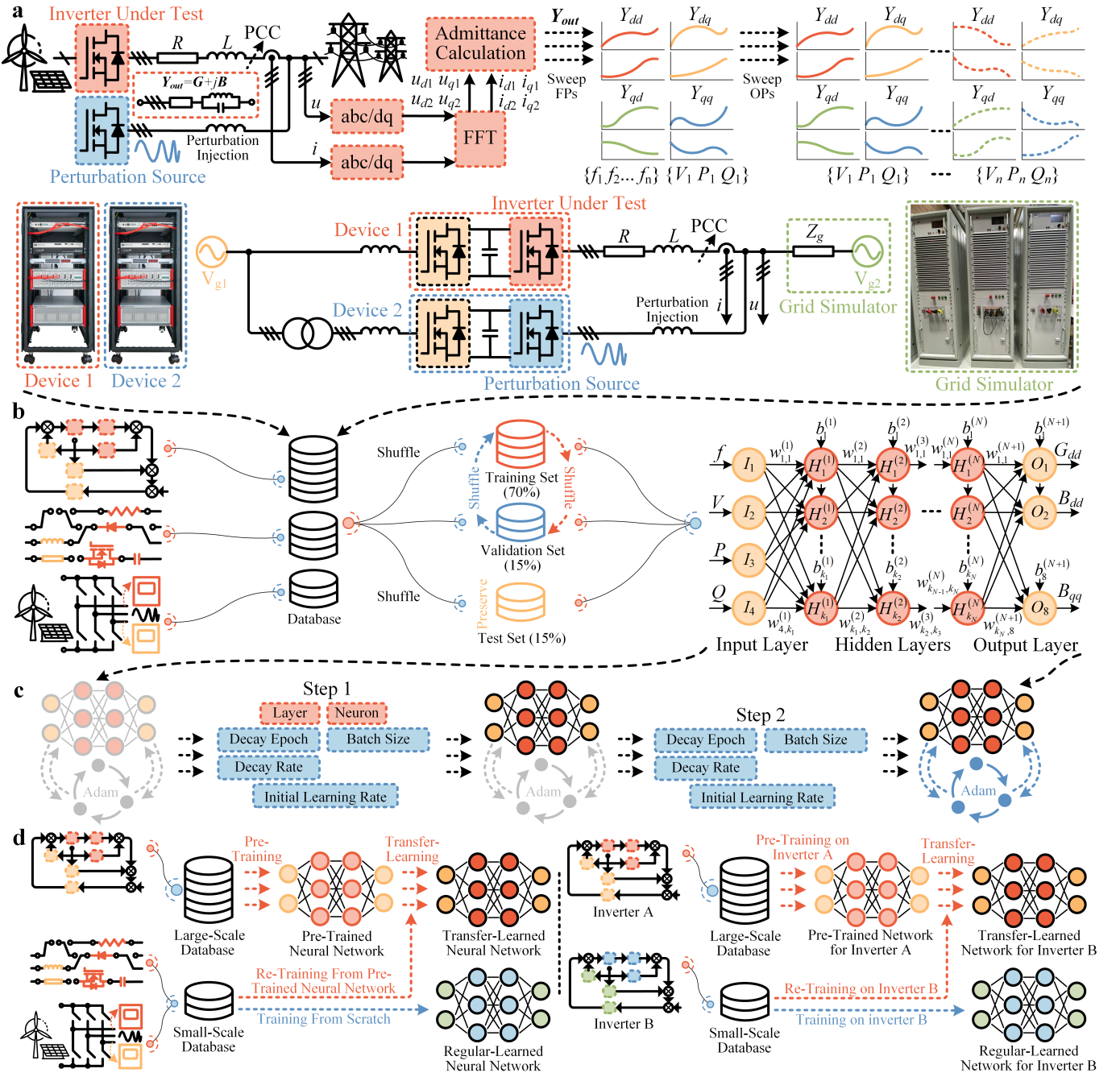


Fig. 2. Overview of the *InvNet* framework and methodologies. (a) **Data Acquisition.** Top: the implementation process of the admittance collection, which models the admittance as a function of the perturbation frequency  $f$ , inverter output voltage  $V$ , active power  $P$ , and reactive power  $Q$  and sweeps the FPs and OPs across a wide range. Bottom: an example of an experimental platform for impedance data acquisition. (b) **Model Training.** Training an FNN in the TensorFlow platform with data collected through either analytical models, EMT simulation models in MATLAB+PLECS framework, or real-world experiments, where the OPs serve as inputs and admittances as outputs. (c) **Hyperparameter Tuning.** Hyperparameter tuning process for the *InvNet* using the Optuna framework. (d) **Transfer Learning Framework.** The *InvNet* used models pre-trained by large-scale databases to extrapolate to real-world models based on small-scale databases (left) and pre-trained inverter models to extrapolate to other inverter models through transfer learning (right).

## II. MACHINE LEARNING FRAMEWORK FOR INVERTER IMPEDANCE CHARACTERIZATION

The grid-tied inverters can be implemented as different topologies ranging from two-level inverters to multilevel inverters [19] (shown in Fig. 1), adopting very different control strategies (e.g. grid-forming or grid-following [20], current source or voltage source behaviors [21], etc.), and performing

different functions (e.g., black start, low-voltage ride-through, reactive support, etc.). Various types of inverters with different implementations at different OPs usually reveal different output impedance patterns. We use a classic two-level grid-following inverter as a baseline example. The methodologies and modeling results presented in this work are applicable to other inverter topologies and control implementations as

well. There are six major factors that may influence inverter impedance patterns, i.e., controller parameters, circuit parameters, sampling process, grid conditions, and phase-locked-loop (PLL) implementation (Components #1 through #6 in Fig. 1), where PLL is used for grid-inverter phase synchronizations. The inverter admittance, typically embodied by a four-element ( $Y_{dd}$ ,  $Y_{dq}$ ,  $Y_{qd}$ , and  $Y_{qq}$ ) matrix at various frequency points (FPs) across multiple OPs, are represented by complex values with the real and imaginary parts ( $Y_{out} = G + jB$  in Fig. 2a, where  $G$  and  $B$  represent the conductance and susceptance, respectively).

As demonstrated in Fig. 2, the *InvNet* is an end-to-end framework comprising automatic data acquisition, scalable model training, and comprehensive model validation (Fig. 2). It is modular, scalable, and flexible for modeling a large number of smart inverters at the grid edge with sophisticated control structures. In this work, the data used for NN training was obtained from 1) analytical models, 2) EMT simulations, and 3) experimental measurements. FFT was utilized to transform the voltage and current to frequency-dependent impedances (complex values in Fig. 2a). The surrogate model was obtained through training a small-scale FNN with OP information ( $V$ ,  $f$ ,  $P$ ,  $Q$ ) as its inputs and admittances as outputs (Fig. 2b). State-of-the-art NN hyperparameter tuning tools and optimizers, such as Optuna [22] and Adam [23], were used to optimize the NN structure and parameters (Fig. 2c). As demonstrated in [24], the design of NN structure and parameters can also be conducted based on latent features of the inverter impedances. We also explored the strength of transfer learning [25]–[27] with the *InvNet*. Leveraging transfer learning, the size of the data needed to obtain a high-performance NN model could be greatly reduced by leveraging data created from existing simplified/incomplete analytical models, or data obtained from measuring similar inverters but with different parameters (Fig. 2d). With the *InvNet*, one can take a few quick snapshots of the impedances of a new inverter at a few OPs and rapidly predict the behaviors of this new inverter across a wide operation range.

#### A. Analytical impedance models

There are many different ways of implementing grid-tied inverters. Different hardware and software implementations lead to different complexities that are hard to be captured by analytical models. Grid-tied inverters are typically controlled as current sources by controllers with a phase-locked loop (PLL) [5], also known as grid-following inverters [1] (Fig. 1).

Small-signal linearized models for grid-tied inverters operating at specific equilibrium points are well-studied [8]. The time delay matrix  $G_d$  of the digital control system, as shown in Fig. 3, is given by

$$G_d = \begin{bmatrix} e^{-1.5T_s s} & 0 \\ 0 & e^{-1.5T_s s} \end{bmatrix}, \quad (1)$$

where  $T_s$  is the sampling period. The transfer function matrix  $G_{id}$  between the duty-ratio and inductor current vector is

$$G_{id} = \frac{-U_{dc}}{(Ls + R)^2 + (\omega L)^2} \begin{bmatrix} Ls + R & \omega L \\ -\omega L & Ls + R \end{bmatrix}, \quad (2)$$

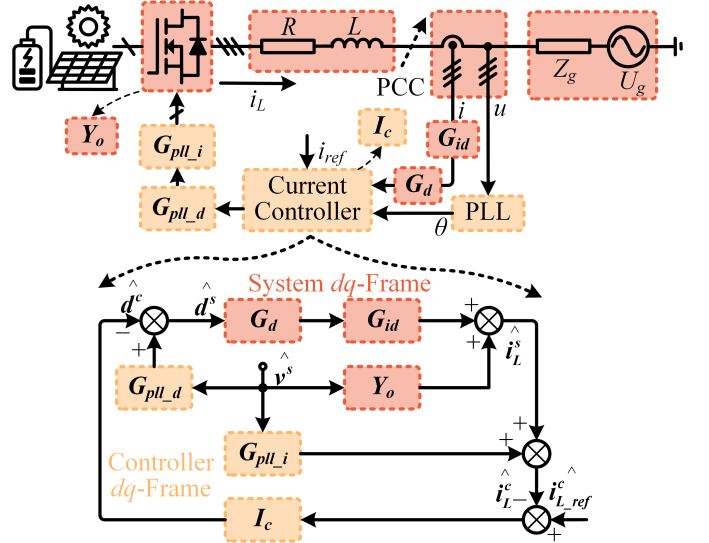


Fig. 3. Small-signal impedance model of grid-tied inverters with current control loop and PLL. The control parameters are assumed to be unknown and may vary from inverter to inverter and from manufacturer to manufacturer.

where  $\omega$  is the fundamental angular frequency of the system,  $L$  and  $R$  are the inductance and resistance in Fig. 2a. The decoupled current controller matrix  $I_c$  can be expressed as

$$I_c = \begin{bmatrix} k_p + \frac{k_i}{s} & -\omega L \\ \omega L & k_p + \frac{k_i}{s} \end{bmatrix}, \quad (3)$$

where  $k_p$  and  $k_i$  are the proportional-integral (PI) parameters, respectively, of the current controller. Defining  $\{I_d, I_q\}$  as the inductor currents and  $\{V_d, V_q\}$  as system voltages in the system  $d$ - and  $q$ - axes, respectively, defining  $\{D_d, D_q\}$  as the duty cycles in the system  $d$ - and  $q$ - axes at the steady-state OP yield

$$\begin{cases} D_d = \frac{1}{U_{dc}} (V_d - I_d R + \omega L I_q) \\ D_q = \frac{1}{U_{dc}} (V_q - I_q R - \omega L I_d), \end{cases} \quad (4)$$

and

$$G_{pll} = \frac{k_{p\_pll} s + k_{i\_pll}}{s^2 + k_{p\_pll} V_d s + k_{i\_pll} V_d}, \quad (5)$$

where  $k_{p\_pll}$  and  $k_{i\_pll}$  are the PI parameters of the PLL. To model the dynamic impact of the PLL, the small-signal perturbation path matrix  $G_{pll\_i}$  from the system voltage to the current in the controller  $d$ - $q$  frame and the small-signal perturbation path matrix  $G_{pll\_d}$  from the system voltage to the duty cycle in the controller  $d$ - $q$  frame are, respectively, given by

$$G_{pll\_i} = \begin{bmatrix} 0 & I_q G_{pll} \\ 0 & -I_d G_{pll} \end{bmatrix}, G_{pll\_d} = \begin{bmatrix} 0 & -D_q G_{pll} \\ 0 & D_d G_{pll} \end{bmatrix}. \quad (6)$$

The open-loop output admittance without the PLL is derived by forcing the perturbations of the duty ratio and dc voltage to zero [8], thus

$$Y_o = \begin{bmatrix} \frac{Ls + R}{(Ls + R)^2 + (\omega L)^2} & \frac{\omega L}{Ls + R} \\ -\omega L & \frac{Ls + R}{(Ls + R)^2 + (\omega L)^2} \end{bmatrix}. \quad (7)$$

The output admittance of the grid-tied inverter system is

$$\mathbf{Y}_{out} = [\mathbf{I} - \mathbf{G}_d \mathbf{G}_{id} \mathbf{I}_c]^{-1} \times \{ \mathbf{G}_d \mathbf{G}_{id} [\mathbf{G}_{pll_d} + \mathbf{I}_c \mathbf{G}_{pll_i}] + \mathbf{Y}_o \}. \quad (8)$$

$\mathbf{Y}_{out}$  can be typically represented by a  $2 \times 2$  matrix with four complex elements:

$$\mathbf{Y}_{out} = \begin{bmatrix} Y_{dd} & Y_{dq} \\ Y_{qd} & Y_{qq} \end{bmatrix}. \quad (9)$$

$Y_{dd}$  represents the current response in the  $d$  channel when  $d$  channel voltage is perturbed;  $Y_{qq}$  represents the  $q$  channel current response when  $q$  channel voltage is perturbed.  $Y_{dq}$  and  $Y_{qd}$  represent the  $d$ - $q$  coupling admittance.

Therefore, the output admittance of a grid-tied inverter at a particular OP in the  $d$ - $q$  frame can be expressed as a function of the perturbation frequency  $f$ , inverter output voltage  $V$ , active power  $P$ , and reactive power  $Q$  [9], [24]. This analytical model requires precise knowledge of circuits and control parameters of inverters, which may not be accurate, or may even be kept confidential in practical scenarios.

### B. Database construction

Fig. 2a illustrates the admittance acquisition process. In this work, the admittances were collected through iteratively sweeping the selected OPs and selected perturbation FPs. The inverter admittance at arbitrary OP is modeled as a four-element conductance matrix  $\mathbf{G}$  and a four-element susceptance matrix  $\mathbf{B}$ :

$$\mathbf{G} = \begin{bmatrix} G_{dd} & G_{dq} \\ G_{qd} & G_{qq} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} B_{dd} & B_{dq} \\ B_{qd} & B_{qq} \end{bmatrix}. \quad (10)$$

We used three types of data for performance evaluations, i.e., the *AnaData* generated by analytical models, the *SimData* generated by EMT simulations, and the *ExData* collected through real-world experiments. To facilitate the data acquisition process, we established an automatic EMT simulation model via the platform of PLECS Blockset integrated with MATLAB Simulink, to rapidly generate admittance data. We called the ‘‘AC sweep’’ block in PLECS from a MATLAB script to inject perturbations and collect the frequency responses throughout the range of selected OPs. Then, the admittance data, modeled as conductances and susceptances for each OP, are saved as a .csv file after each iteration. Through repeatedly sweeping the OPs and perturbation frequencies throughout the selected range, output admittances of inverters can be either calculated by analytical equations or by EMT simulations.

To exemplify the databases that were used in this work, the EMT simulation-generated *SimData* was visualized in Fig. 4 as an example of the database, where there are 40 OPs and 20 FPs in each OP, which therefore, constitutes a database with 800 ( $20 \times 40$ ) data points in total. It took approximately 40 hours to complete all iterations on a personal computer (PC) with Intel 11th Gen i7-11700 processor. The same PC was used throughout this work. The *ExData*, by using the method of admittance measurements presented in [11], was collected through experiments conducted in the testing

platform demonstrated in Fig. 2a, the entire process of which took us more than one week to construct a database comprising 817 data points for the *ExData*.

### C. Hyperparameter optimization

In this work, the optimization of NN hyperparameters was conducted via the Optuna framework (<https://optuna.org/>) embedded in the TensorFlow platform (<https://www.tensorflow.org/>). The optimization process was conducted in a two-step manner (as depicted in Fig. 2c). First, the NN structure, namely, the layer and neuron numbers and the Adam optimizer [23] parameters, i.e., decay epoch (specifying the epoch number that each decay of the learning rate takes), batch size, decay rate, and initial learning rate were all put into trials. We implemented 100 trials in total which took approximately 16 hours to complete. Then, we fixed the NN structure by using the optimized layer and neuron numbers and only conducted the optimization of the Adam optimizer parameters for another 100 trials which also took roughly 16 hours. Finally, we used the NN structure from the first step and Adam parameters from the second step as the final hyperparameters. The evaluations presented in Section III implied that the obtained hyperparameters did not lead to overfitting and achieve optimal performances.

### D. Neural network training

We developed FNNs to model the admittances of a grid-tied inverter in this work. Fig. 2b illustrates a typical FNN structure with one input layer,  $N$  hidden layers, and one output layer. The FNN has four input neurons, i.e.,  $I_1$ – $I_4$ , representing  $f$ ,  $V$ ,  $P$ , and  $Q$ , and eight output neurons, i.e.,  $O_1$ – $O_8$ , representing the conductance ( $\mathbf{G}$ ) and the susceptance ( $\mathbf{B}$ ) of the output admittance, i.e.,  $G_{dd}$ ,  $B_{dd}$ ,  $G_{dq}$ ,  $B_{dq}$ ,  $G_{qd}$ ,  $B_{qd}$ ,  $G_{qq}$ , and  $B_{qq}$ . As shown in Fig. 2b, the  $i^{th}$  hidden layer has  $k_i$  neurons, and the neuron values are subject to the following equation

$$\mathbf{H}^{(i)} = \sigma(\mathbf{w}^{(i)} \mathbf{H}^{(i-1)} + \mathbf{b}^{(i)}), \quad (11)$$

where  $\sigma$  is the activation function, e.g., the sigmoid activation function that frames the neuron values into probability values between 0 and 1, and  $\mathbf{H}^{(i)}$ ,  $\mathbf{w}^{(i)}$ , and  $\mathbf{b}^{(i)}$  are the matrices of neuron values, weights, and biases in the  $i^{th}$  hidden layer, which are given by

$$\mathbf{H}^{(i)} = [\mathbf{H}_1^{(i)} \mathbf{H}_2^{(i)} \cdots \mathbf{H}_{k_i}^{(i)}]^T, \quad (12)$$

$$\mathbf{b}^{(i)} = [b_1^{(i)} b_2^{(i)} \cdots b_{k_i}^{(i)}]^T, \quad (13)$$

$$\mathbf{w}^{(i)} = \begin{bmatrix} w_{1,1}^{(i)} & w_{2,1}^{(i)} & \cdots & w_{k_{i-1},1}^{(i)} \\ w_{1,2}^{(i)} & w_{2,2}^{(i)} & \cdots & w_{k_{i-1},2}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,k_i}^{(i)} & w_{2,k_i}^{(i)} & \cdots & w_{k_{i-1},k_i}^{(i)} \end{bmatrix}. \quad (14)$$

The neuron values of the first hidden layer are

$$\mathbf{H}^{(1)} = \sigma(\mathbf{w}^{(1)} \mathbf{I} + \mathbf{b}^{(1)}), \quad (15)$$

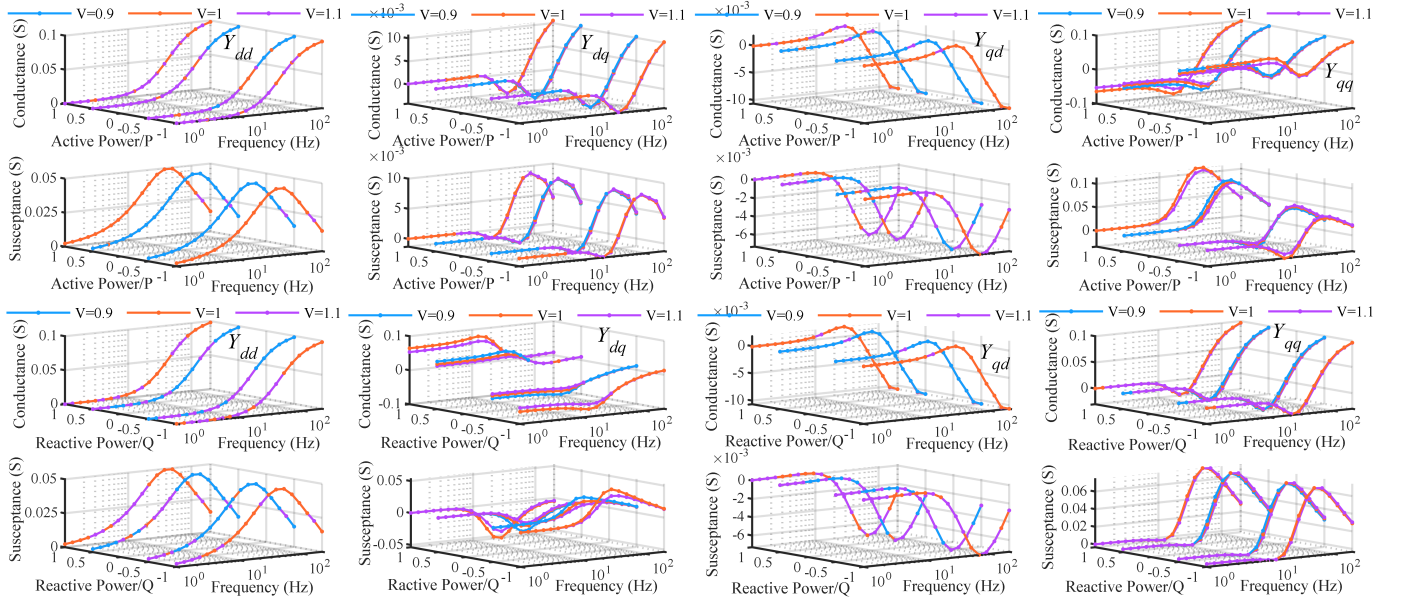


Fig. 4. *SimData* (used for performance evaluations in Section III) visualized in 3-D view. The conductance and susceptance are illustrated against both the active-power- $P$  and frequency when the reactive-power is zero and both the reactive-power and frequency when the active-power is zero, respectively. As shown, The output voltages 0.9, 1.0 and 1.1 (nominal value) have trivial impacts on the admittance, the admittance in  $Y_{qq}$  changes significantly along the active-power axis while the admittance in  $Y_{dq}$  varies significantly with the reactive-power, which imply that the admittance of grid-tied inverters is more susceptible to active and reactive power than to the output voltage.

where  $\mathbf{I} = [I_1, I_2, I_3, I_4]^T$ . The output layer is

$$\mathbf{O} = \mathbf{w}^{(N+1)} \mathbf{H}^{(N)} + \mathbf{b}^{(N+1)}, \quad (16)$$

where  $\mathbf{O} = [O_1, O_2, \dots, O_8]^T$  and

$$\mathbf{b}^{(N+1)} = [b_1^{(N+1)} \ b_2^{(N+1)} \ \dots \ b_8^{(N+1)}]^T, \quad (17)$$

$$\mathbf{w}^{(N+1)} = \begin{bmatrix} w_{1,1}^{(N+1)} & w_{2,1}^{(N+1)} & \dots & w_{k_N,1}^{(N+1)} \\ w_{1,2}^{(N+1)} & w_{2,2}^{(N+1)} & \dots & w_{k_N,2}^{(N+1)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{1,8}^{(N+1)} & w_{2,8}^{(N+1)} & \dots & w_{k_N,8}^{(N+1)} \end{bmatrix}. \quad (18)$$

The next step is to train the NN. As shown in Fig. 2b, we split the acquired database into training (70%), validation (15%), and test sets (15%), respectively. The test set was reserved for final evaluations, while the training and validation sets were randomly shuffled before each training process, such that every item had the same chance to be used for training. Then, a back-propagation algorithm with the mean squared error (MSE) loss function between the FNN prediction and the admittance data was used to train the FNN. Finally, the test sets were used to verify the performance of the FNN with multiple different types of figure-of-merits (e.g., MSE, mean average error (MAE), 95<sup>th</sup> percentile error, etc.).

#### E. Transfer learning

Transfer learning is a machine learning technique that involves leveraging knowledge gained from one task to improve the performance of another related task. In traditional machine learning, models are usually trained from scratch for each new task or problem. Transfer learning, however, takes a different

approach. Instead of starting from scratch, a pre-trained model that has been trained on a large and diverse dataset for a related task is used as a starting point. This pre-trained model has already learned useful features, representations, and patterns from the original task, which can be beneficial for the new task. It has been widely successful in various domains, including natural language processing, computer vision, audio analysis, and more [17], [18], [25]–[28].

It usually involves two main steps: 1) *Pre-training*: In this step, a model is trained on a large dataset and a specific task, often referred to as the source task. This training helps the model learn general features and patterns that are useful for various related tasks. Common pre-trained models include language models trained on a large corpus of text, image models trained on a large dataset of images, and more [27]; 2) *Fine-tuning*: After pre-training, the model is fine-tuned on the target task, which is the specific problem we want to solve. During fine-tuning, the model's parameters are adjusted using a smaller dataset specific to the target task. The knowledge gained during pre-training helps the model adapt and learn task-specific nuances more quickly and with less data.

Transfer learning offers several advantages: 1) *Faster training*: Since the model starts with knowledge from a related task, it often requires fewer iterations or epochs to achieve good performance on the target task; 2) *Less data dependency*: Transfer learning can mitigate the need for an extremely large dataset for the target task, as the model can generalize from the knowledge it gained during pre-training; 3) *Better generalization*: The learned features from the pre-trained model are often more robust and generalizable, leading to improved performance on the target task; 4) *Reduced computational resources*: Training a model from scratch can be computationally

TABLE I  
PARAMETERS OF FOUR INVERTER IMPLEMENTATIONS

|                                          | Inverter 1 | Inverter 2      | Inverter 3                     | Inverter 4                     |
|------------------------------------------|------------|-----------------|--------------------------------|--------------------------------|
| PLL Bandwidth ( $B_{pli}$ )              | 7 Hz       | 20 Hz           | 7 Hz                           | 20 Hz                          |
| PI Controller Coefficient ( $k_p, k_i$ ) | $K_p, K_i$ | $K_p, K_i$      | $\frac{K_p}{2}, \frac{K_i}{2}$ | $\frac{K_p}{2}, \frac{K_i}{2}$ |
| DC Voltage ( $U_{dc}$ )                  |            | 600 V           |                                |                                |
| Fundamental Frequency ( $f$ )            |            | 50 Hz           |                                |                                |
| Line-to-Line Voltage ( $V_{ll}$ )        |            | $110\sqrt{3}$ V |                                |                                |
| Rated Active Power ( $P$ )               |            | 2.3 kW          |                                |                                |
| Rated Reactive Power ( $Q$ )             |            | 2.3 kvar        |                                |                                |
| Interfacing Inductance ( $L$ )           |            | 2 mH            |                                |                                |
| Interfacing Resistance ( $R$ )           |            | 62.8 m $\Omega$ |                                |                                |
| Sampling Rate ( $f_s$ )                  |            | 10 kHz          |                                |                                |

Note:  $K_p = 10.5$  and  $K_i = 2741.6$ . The rated power is calculated based on the rated current amplitude of 10 A. The bandwidth of the PLL is usually relatively low such that the inverter system can be easily stabilized during both steady- and transient-state.

intensive, while transfer learning can significantly reduce the required resources [26], [27].

As demonstrated in Fig. 2d, we showcased the strength of transfer learning in two commonly useful scenarios: 1) *Model extrapolation from analytical data-based model to real-world data-based model*: Supposing that we have a pre-trained model that is trained on a large amount of *AnaData* from a particular type of Inverter A whose parameters and control structures are all available (note the *AnaData* cannot capture dead-time effect and other nonlinear or nonideal information), then, we can re-train the model based on a small amount of real-world data from this same Inverter A or another Inverter B (with different parameters but the same control framework) that is controlled as a black-box, i.e., its parameters are all unavailable. In this way, we can get a model with high accuracy for both Inverter A and B, since the pre-trained model are fine-tuned by real-world data. 2) *Model extrapolation from one inverter to another*: Supposing that we have a pre-trained model that is trained on a large-scale database from Inverter A (i.e., a large number of confidential impedance data available from the manufacturer), then, we can use a small-scale database from Inverter B to re-train the pre-trained model to get Inverter B's impedance model with high accuracy.

### III. MODEL PERFORMANCE EVALUATION

We evaluate the performance of the *InvNet* framework in the following three scenarios:

- When an analytical model exists, but the parameters are unknown or kept confidential.
- When an analytical model does not exist or is not mature.
- Lack of sufficient data for particular inverters.

We collected data from analytical calculations, EMT simulations, and experimental measurements. Each type of data was randomly partitioned into a training set (70% of the total dataset), a validation set (15% of the total dataset), and a test set (15% of the total dataset). In each training process, we monitored the model's performance on the validation set during training while the test set was preserved and referred to once the training was complete to evaluate the model performance. The final models were thoroughly evaluated on the test set that was never used during the training process.

We exemplified four different grid-tied inverters, each with parameters listed in Table I. Two datasets collected through analytical calculations and automatic EMT simulations (using the simulation platform of PLECS integrated with MATLAB Simulink), i.e., the *AnaData* and the *SimData*, respectively, were applied to train the FNN, resulting in two NNs, i.e., the *AnaNN* and the *SimNN*, respectively. The two datasets were constructed in the same way. The operation ranges of the inverters were normalized to the following ranges in per unit:  $V \in [0.9, 1.1]$ ,  $P \in [-1, 1]$ , and  $Q \in [-1, 1]$ . We selected 20 frequency steps evenly distributed in the logarithmic scale in the range of  $f \in [1, 200]$  and the steps of  $V$ ,  $P$ , and  $Q$  were selected as  $\{0.1, 0.5, 0.5\}$ . OPs leading to over-modulation (modulation index greater than 1) and over-current (inductor current greater than 110% of the rated current) conditions were excluded from all datasets. Therefore, the final dataset had 800 ( $20 \times 40$ )  $\{f, V, P, Q\}$  data points in total. Each data point comprised an admittance matrix  $[Y_{dd}, Y_{dq}; Y_{qd}, Y_{qq}]$ . The EMT simulation-generated *SimData* was visualized in Fig. 4 as an example of the database. Both the *AnaData* and *SimData* were split into three portions, i.e., 70% for training, 15% for validation, and 15% for testing. The same 15% test set of the *SimData* was used for evaluations of both the *AnaNN* and *SimNN*.

To mimic a real-world scenario when dealing with a wide range of different inverters with unknown information, we applied dead-time to the switching events in the EMT simulation. The dead-time effect is usually not captured by commonly used analytical methods. We also assigned the same synchronous reference frame PLL (SRF-PLL) architecture to both analytical models and EMT simulations, but with different PLL bandwidths or proportional-integral (PI) controller coefficients, based on the assumption that analytical models may use incorrect system parameters (The *SimData* was directly captured from EMT simulations without the necessity of recognizing system parameters). Similar methods can be used to study the impact of other system parameters.

An experimental rig (Fig. 2b) was constructed to collect real-world impedance data. A power amplifier to the right with an attached inductive impedance  $Z_g$  worked as a grid simulator and a power source  $V_{g1}$  to the left mimicked the renewable energy sources. The inverter under test had the same parameters as Inverter 1 in Table I. There were two devices ("Microgrid Tech Bench" from Imperix) with a back-to-back converter structure in each one of them, where one of the converters worked as a rectifier absorbing power from energy sources and the other as an inverter injecting power to the grid. To emulate the grid, all converters were connected through an inductive impedance  $Z_g$  to a power amplifier ("DM 45000/APS" from Spitzenberger & Spies) which worked as a grid simulator to mimic the power grid. The power source  $V_{g1}$  to the left of Fig. 2b emulated the renewable energy sources and was also connected through interfacing inductors to the devices, where Device 2 was connected through a transformer to achieve galvanic isolation between Device 1 and 2. The inverter in Device 1 was the one under test (had the same parameters as Inverter 1 in Table I) for admittance measurements while the inverter in Device 2 served as a perturbation source injecting

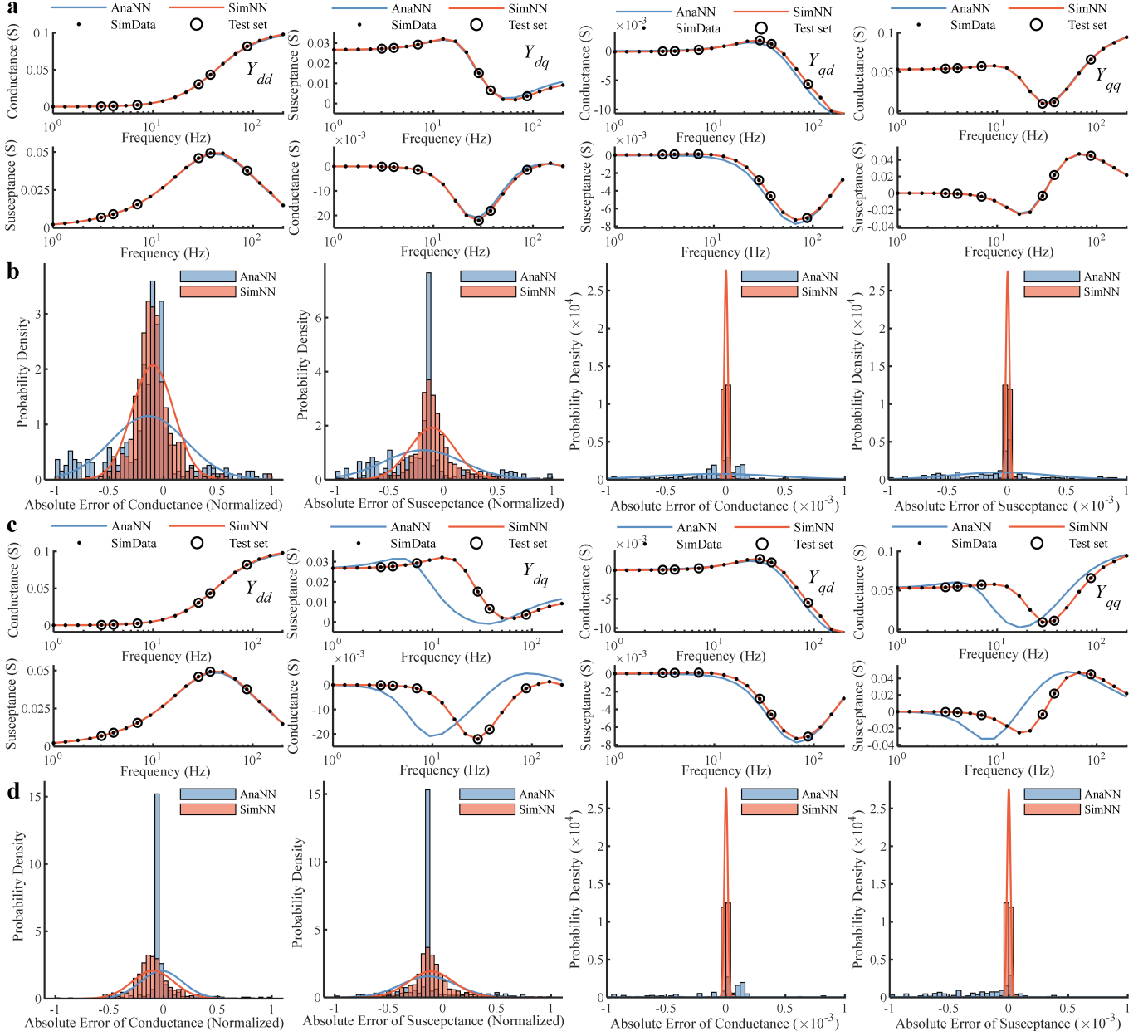


Fig. 5. Performance evaluation under unknown parameter circumstances. (a) Prediction results of the *AnaNN* and the *SimNN* compared to the test set at the same example OP (the No. 23 OP in both the *AnaData* and the *SimData*, where  $V$ ,  $P$ , and  $Q$  were  $\{1.1, -1, 0.5\}$ ). The *SimNN* managed to capture the dead-time effect while the *AnaNN* did not. Thus, the *SimNN* achieved better accuracy than the *AnaNN*. (b) The probability density histograms together with a normal distribution fit for each one of them. Left two: probability densities of the normalized absolute errors. Right two: probability densities of the actual absolute errors. (c) Prediction results of the *AnaNN* and the *SimNN* compared to the test set at the same example OP (the No. 23 OP in both the *AnaData* and the *SimData*, where  $V$ ,  $P$ , and  $Q$  were  $\{1.1, -1, 0.5\}$ ). The *SimNN* successfully captured the comprehensive information of the real system, while the *AnaNN*, which lacked awareness of system parameters, failed to establish an accurate admittance model. Therefore, the *SimNN* achieved better accuracy than the *AnaNN*. (d) The probability density histograms together with a normal distribution fit for each one of them. Left two: probability densities of the normalized absolute errors. Right two: probability densities of the actual absolute errors.

perturbations for admittance measurements. To comparatively expedite the admittance data collection process, the PRBS method was applied here [11]. In addition, to facilitate the FFT analysis for the experimentally measured voltage and current data, 19 FPs were also selected as ones, tens, and hundreds in the range of  $f \in [1, 200]$ , i.e.,  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 60, 70, 80, 90, 100, 200\}$ . To diversify the database, the step of  $V$  was selected as 0.1, while the steps of  $P$  and

$Q$  were selected by choosing the steps of active and reactive currents  $I_d$  and  $I_q$  as  $\{0.4, 0.4\}$ , where OPs resulting in over-modulation and over-current conditions were also excluded. In this way, the final experimental database was composed of 43 OPs, which constructed a database with 817 ( $19 \times 43$ )  $\{f, V, P, Q\}$  data points in total and was referred to as the *ExData*. This *ExData* was also randomly partitioned into a 70% training set, a 15% validation set, and a 15% test set.

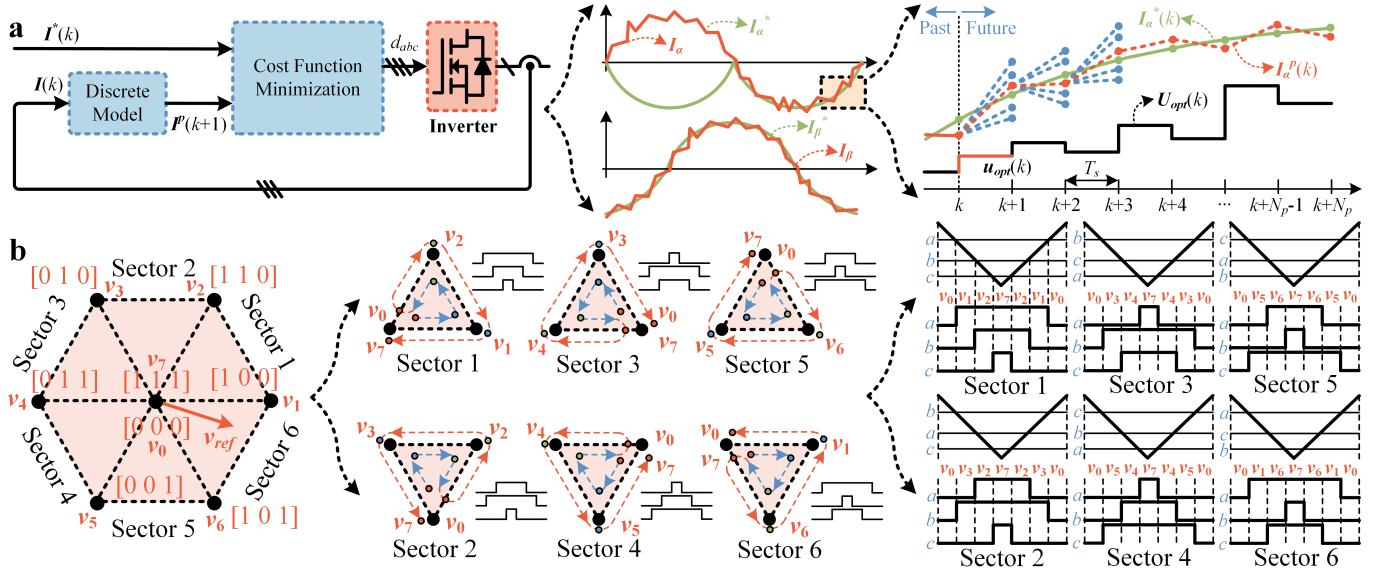


Fig. 6. Control diagrams of multi-vector model predictive control (MV-MPC) [10]. (a) The principle of MPC. The selection process for optimal switching actions is highly nonlinear that can hardly be modeled through traditional small-signal approaches. (b) The control diagram of the multi-vector MPC. The current tracking is realized through the use of MPC while also retaining the fixed switching frequency characteristics in traditional PI-controller-based space-vector pulse-width-modulation (PWM) strategies. Due to inherent nonlinearities of the MPC, the impedance models are unavailable to date. The Data-driven methods can be used to establish multi-OP impedance models for grid-tied inverters controlled by such kind of controllers.

#### A. Analytical model exists but model parameters are unknown

We first showcased the effectiveness of the *InvNet* in capturing the admittance features brought about by the dead-time effect, through the exemplification of Inverter 2. As aforementioned, existing analytical models cannot achieve sufficient accuracy for not capturing the dead-time effect, thus, the *AnaData* did not contain dead-time information. However, since the EMT simulation was able to implement the dead-time in each power switch of the inverter, the dead-time information was included in the *SimData* but could not be captured by the analytical model. As depicted in Fig. 5a, the *SimNN* curve differed slightly from the *AnaNN* curve in  $Y_{dd}$  and  $Y_{qq}$ , while revealing obvious deviations from the *AnaNN* curve in  $Y_{dq}$  and  $Y_{qd}$ , for both conductance and susceptance curves, which implied the inaccuracy of the analytical model. The test set, stochastically extracted from the *SimData*, was in line with the *SimNN* curve, while inconsistent with the *AnaNN* curve, especially in  $Y_{dq}$  and  $Y_{qq}$ , for both conductance and susceptance. The absolute errors were calculated by subtracting the predictions of NNs from the test set and classified into conductance and susceptance absolute errors, respectively. By normalizing the absolute errors into the range of  $[-1, 1]$  through the use of “MinMaxScaler”, we constructed the probability density histograms together with a normal distribution fit for each one of them. As shown in the left two graphs of Fig. 5b, the probability densities of the normalized absolute errors of the *SimNN* were more concentrated around zero than those of the *AnaNN* for both the conductance and susceptance, which indicated that admittances predicted by the *SimNN* were more accurate than the ones predicted by the *AnaNN*. Further, we also visualized the probability density histograms of the absolute errors with their normal distribution

fit in the right two graphs of Fig. 5b. The probability densities of the real absolute errors of the *SimNN* were in sharp contrast to the ones of the *AnaNN*, the former ones were concentrated around zero while the latter ones were more widespread, which confirmed the superiority of the *SimNN* over the *AnaNN*.

To further demonstrate the advantages of the *InvNet* in dealing with classified or uncertain information, we conducted experiments using Inverter 1 as the source of the *AnaData* and Inverter 2 as the source of the *SimData*, assuming that the analytical model had no prior knowledge of the actual inverter parameters. Likewise, as depicted in Fig. 5c, it exposed noticeable discrepancies with the *AnaNN* in  $Y_{dq}$ ,  $Y_{qd}$ , and  $Y_{qq}$ , whereas it only differed slightly from the *AnaNN* in  $Y_{dd}$ . The normalized probability densities of the absolute errors were depicted in the left two graphs of Fig. 5d, where the histograms of the *SimNN* were more like the normal distributions. The actual probability densities of the absolute errors were given in the right two graphs of Fig. 5d, where the probability densities of the absolute errors of the *SimNN* were also in stark contrast to the ones of the *AnaNN*, the former ones were mostly located near zero while the later ones span across a wide range which were almost close to zero.

For the studies in Fig. 5, the FNN consisted of three hidden layers that had 683 parameters in total: layer #1, 4 neurons; layer #2, #3, and #4, 15 neurons; layer #5, 8 neurons. Using smaller-scale NNs can enhance the computational efficiency, techniques such as network pruning [31] can be adopted to further reduce the model size. We trained the model in Matlab using the optimizer of Bayesian regularization back-propagation, which updates the weight and bias values according to Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights, and then determines the correct combination so as to produce a network

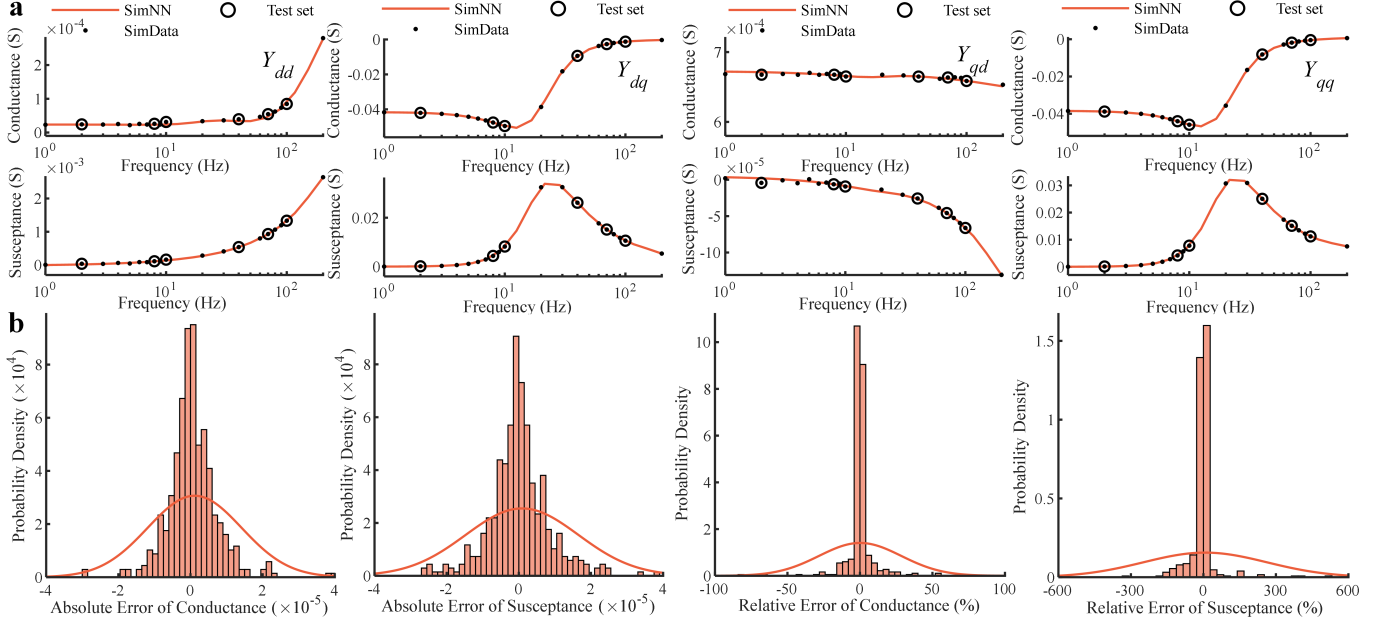


Fig. 7. Performance evaluation under no analytical model circumstances. (a) The *SimNN* was trained on the *SimData* collected from the multi-vector MPC-controlled inverter with the same PLL bandwidth as Inverter 2. Prediction results of the *SimNN* were compared with the test set at the same example OP (the No. 23 OP in the *SimData*, where  $V$ ,  $P$ , and  $Q$  were  $\{1.1, -1, 0.5\}$ ). (b) The probability density histograms together with a normal distribution fit for each one of them. Left two: probability densities of the absolute errors. Right two: probability densities of the relative errors.

that generalizes well. We trained each model for 3,000 epochs, which took approximately 8 minutes for each training process on the same PC (Intel 11th Gen i7-11700 processor).

#### B. Analytical model does not exist or is not mature

Traditionally, grid-tied inverters were mostly controlled by linear-controllers which can be analytically modeled straightforwardly. In recent years, applications of more advanced nonlinear controllers have become a future trend, among them, MPC is a promising alternative to control power electronic converters, which has been heavily explored over the past few years [29], [30]. Due to its distinctive advantages, such as fast dynamic response, straightforward implementation, compatibility with nonlinear constraints of converters, and the capability to simultaneously tackle multiple control objectives, MPC is much more powerful to address emerging challenges that modern power converters are facing than traditional control methods are. However, to date, there are no mature solutions for the modeling of MPC-controlled grid-tied inverters, due to the inherent nonlinearities of MPC. The MPC uses the discrete model of the system to minimize the cost function such that the optimal switching action is selected for the next control iteration (on the left of Fig. 6a). The optimization process is a highly nonlinear one that can hardly be modeled through traditional small-signal modeling approaches (on the right of Fig. 6a). Most existing MPC frameworks do not have a constant switching frequency behavior. Their stability cannot be solely described by frequency-domain impedance analysis. In this work, we adopted the multi-vector model-predictive-control (MV-MPC) [10], a subset of MPC control strategy which has a constant switching frequency behavior that can be described by frequency-domain impedance analysis.

Fig. 6 illustrates the implementation process of the MV-MPC. The physical model of the inverter system in the  $\alpha\beta$ -frame is

$$\mathbf{u} = L \frac{d\mathbf{i}}{dt} + \mathbf{i}R \quad (19)$$

where  $\mathbf{u} = [u_\alpha \ u_\beta]^T$  and  $\mathbf{i} = [i_\alpha \ i_\beta]^T$ , which are converter voltage and current vectors in the  $\alpha\beta$ -frame respectively.

The discrete-time model can be obtained by applying Euler Forward Approximation:

$$\mathbf{i}(k+1) = \frac{T_s}{L} \mathbf{u}(k) + \left(1 - \frac{RT_s}{L}\right) \mathbf{i}(k). \quad (20)$$

To determine the optimal voltage vector in terms of the current tracking performance at the time instant  $k$ , the following prediction model can be established:

$$\mathbf{u}^*(k) = \frac{L}{T_s} \mathbf{i}^*(k+1) + \left(R - \frac{L}{T_s}\right) \mathbf{i}(k). \quad (21)$$

Here  $\mathbf{u}^*(k)$  represents the reference voltage vector that forces the actual current to ideally track the reference current  $\mathbf{i}^*(k+1)$ . The calculated reference vector is rapidly located in the  $120^\circ$  oblique frame (transformed from the  $\alpha\beta$ -frame) and three adjacent voltage vectors are selected over one control (sampling) iteration, which replaces the computationally inefficient calculation or lookup table approaches to simple integer arithmetic. Current tracking is prioritized through duty cycle optimization of the selected adjacent vectors. Finally, the optimal switching sequence is generated through an external modulator that follows the symmetric pulse pattern of seven segments (Fig. 6b). Replacing the proportional-integral (PI) current controller (Fig. 2a) with a nonlinear MV-MPC made the entire grid-tied inverter system “model-free”.

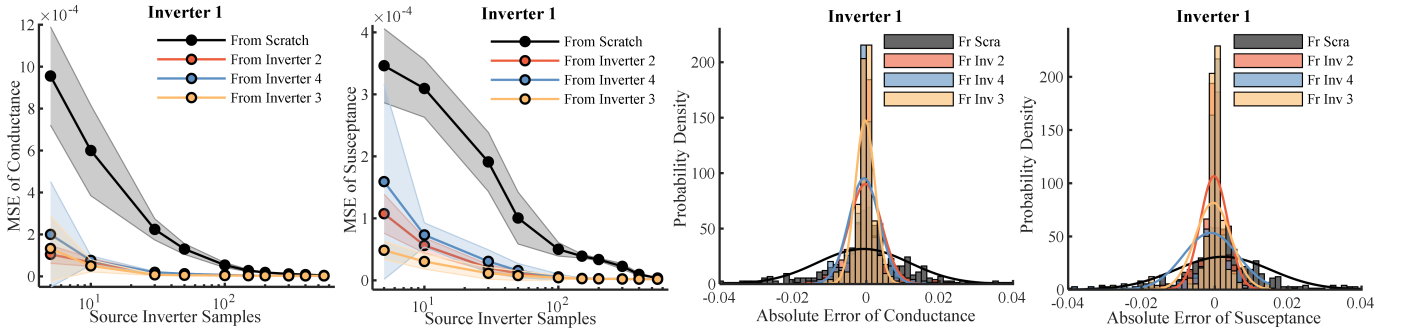


Fig. 8. Transfer learning performance evaluations for the extrapolations from analytical data-based model to experimental data-based model. The MSE curves of the transfer learned models were lower than the models learned from scratch even when training on only a few data points (the MSE curves were visualized by adding standard deviations as margins of the mean MSE value for the ten training cycles) and the probability densities of the absolute errors of the transfer learning were more concentrated than those of the models learned from scratch (when using 30 data points), indicating the superior performance of transfer learning.

We developed an automatic EMT simulation platform in PLECS to collect the admittance data (*SimData*) for an inverter with the same PLL bandwidth as Inverter 2, but controlled by MV-MPC. Here, we selected 19 FPs as ones, tens, and hundreds in the range of  $f \in [1, 200]$ , which were  $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 60, 70, 80, 90, 100, 200\}$ . A randomly selected test set was reserved from the *SimData* using a 15% split and used for model evaluations. As shown in Fig. 7a, at an arbitrary OP, the test set data points all aligned with the *SimNN* curve, which proved the *InvNet*'s power of re-constructing admittance models at the whole OP range. We also constructed the probability density histograms together with a normal distribution fit for each one of them. The probability densities of the absolute errors followed a normal distribution and were primarily concentrated around an extremely narrow band around zero (left two graphs of Fig. 7b), which confirmed the accuracy of the *SimNN*'s predictions. Moreover, the probability density histograms of the relative errors were also largely concentrated around a narrow band around zero (right two graphs of Fig. 7b).

For studies in Fig. 7, the FNN structure also had three hidden layers with 15 neurons in each layer. We also trained the model in Matlab using Bayesian regularization optimizer for 1,000 epochs, which took roughly less than 3 minutes.

### C. Insufficient data for modeling a particular inverter

To gain a comprehensive understanding of inverter impedances across a wide range of OPs, it is usually necessary to train the NN using a relatively large amount of data. We leveraged transfer learning (Fig. 2d) to reduce the database volume and empower the *InvNet* framework with the ability of cross-inverter extrapolation (Fig. 8a) when there is no sufficient data for a particular inverter.

The first demonstration includes extrapolating from an FNN trained on data from analytical models, which may not accurately capture impedance information due to insufficient parameter knowledge from the inverter system, to an FNN capable of understanding impedance knowledge from real-world experiments. The FNN was first pre-trained on a large amount of data generated from analytical models and then

fine-tuned by a small amount of data collected from real-world experiments.

We pre-trained the model on the *AnaData* generated from analytical models for Inverter 2, 3, and 4 (Table I) for 500 epochs. With data readily generated from analytical models, we were able to construct a large database as the *AnaData*. Similarly to Fig. 5, 20 FPs were also evenly selected in the logarithmic scale in the range of  $f \in [1, 200]$  but with the steps of  $V$ ,  $P$ , and  $Q$  selected as  $\{0.1, 0.1, 0.1\}$ . Excluding over-modulation and over-current conditions, it resulted in 1084 OPs and 21680 ( $20 \times 1084$ , 1,084 OPs, 20 FPs) data points in total. We then re-trained (fine-tuned) the model on the *ExData* (from Inverter 1) using 5, 10, 30, 50, 100, 150, 200, 300, 400, and 550 data points for also 500 epochs. To achieve a fair comparison, we also trained the model from scratch on *ExData* using the same number of data points as well. We repeatedly trained each model ten times with a different random seed each time and recorded the absolute errors and MSEs for both conductance and susceptance simultaneously at the end of each training cycle for further analyses. In this regard, the total training count was: 3 (source inverter count)  $\times$  1 (target Inverter count)  $\times$  10 (repeating times)  $\times$  10 (number of used data points) = 300, which took approximately 30 hours to complete on the same PC (Intel 11th Gen i7-11700 processor).

As seen from the left two figures in Fig. 8b, the transfer learned models' MSE dropped significantly even when trained with only 30 data points (note each data point is a set of time sequences at an operating point), for both conductances and susceptances. The MSE of the transfer learned models approached zero after only 100 data points. The transfer-learned models considerably outperformed the models trained from scratch. From the MSE perspective, the models trained from scratch could only reach a performance comparable to the transfer-learned models at approximately 200 data points for the conductance and 550 data points for the susceptance. Training from scratch led to significantly higher MSE when using fewer than 50 data points. We also evaluated the probability densities of the absolute errors (the right two figures in Fig. 8b). Similar to the studies in Fig. 5, the probability density histograms together with a normal distribution fit for each histogram (when using 30 data points) were also shown. The

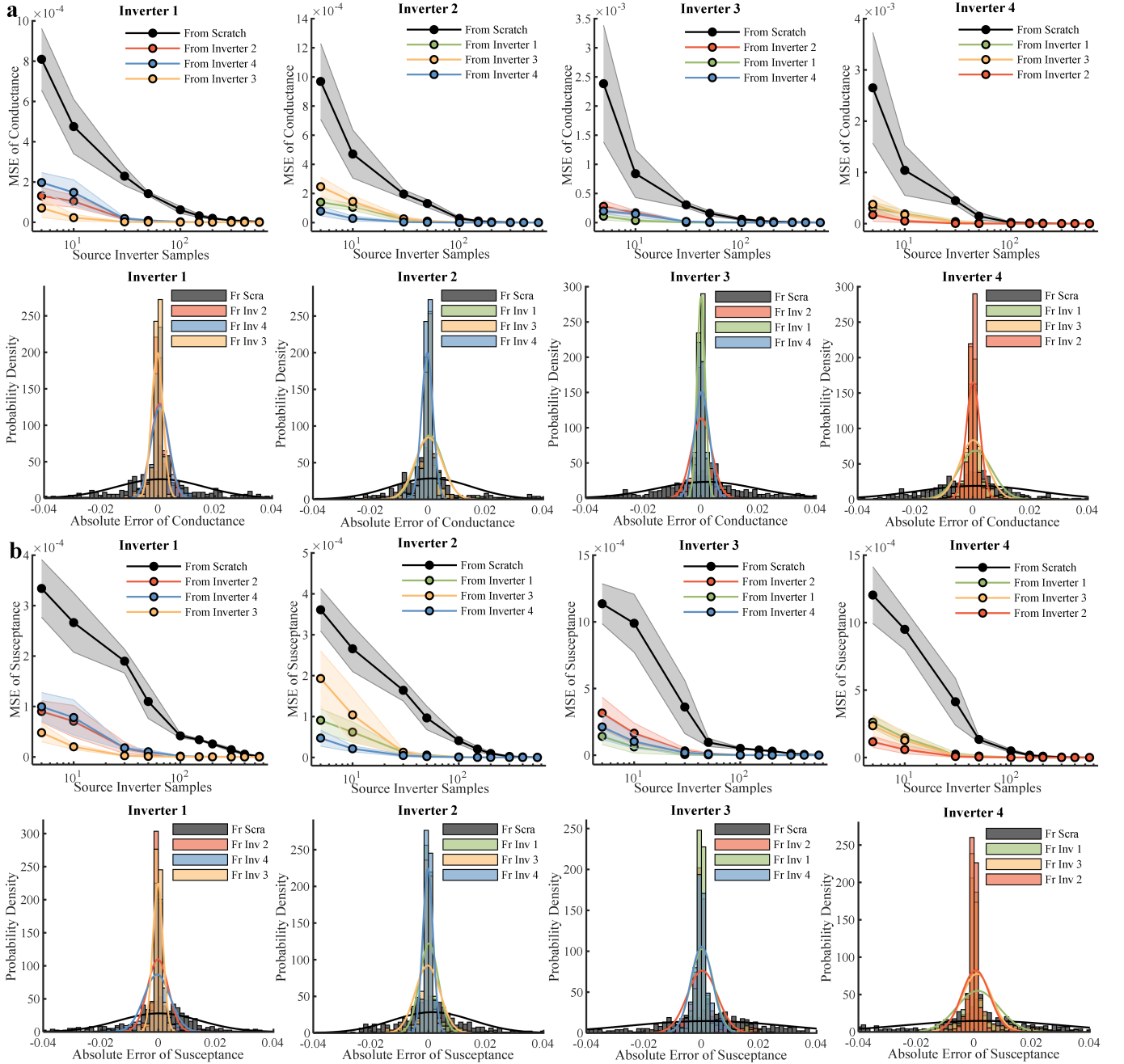


Fig. 9. Performance evaluations for cross-inverter extrapolations using transfer learning. (a) Conduction evaluations: The MSE curves were visualized by adding standard deviations as margins of the mean MSE value for the ten-time training. The transfer-learned models outperformed the models learned from scratch even when only a few data points were used. The probability densities of the absolute errors also confirmed the superiority of the transfer learning (when using 30 data points). (b) Susceptance evaluations: The MSE curves were visualized in similar ways to the conduction evaluations. The transfer-learned models also outperformed the models learned from scratch. The probability densities of the absolute errors also consolidated the superiority of the transfer learning (when using 30 data points).

probability densities of absolute errors of the transfer learned models were more concentrated around zero than those of the model trained from scratch, which indicated that admittances predicted by transfer learned models were more accurate than the ones predicted by the models trained from scratch.

In addition to the aforementioned model extrapolation from the analytical data-based model to the experimental data-based model, we also performed cross-extrapolations, i.e., from all other three inverters to one inverter in Table I using

transfer learning (for instance, from Inverter 2, 3, and 4 to Inverter 1). This time, to conduct a proof-of-concept for cross-extrapolations and facilitate the data acquisition process, we used analytical models to generate databases assuming that all analytical models were well-aware of all system parameters. The performance evaluations were conducted in the same way as studies demonstrated in Fig. 8 (with each training process repeated ten times). We first pre-trained the model on the *AnaData* generated from analytical models of source inverters

(from Inverters 1, 2, 3, and 4) with 21,680 ( $20 \times 1084$ , 1,084 OPs, 20 FP) data points in total for 500 epochs. We then fine-tuned the model on the other *AnaData* from target inverters (also from Inverters 1, 2, 3, and 4) using 5, 10, 30, 50, 100, 150, 200, 300, 400, and 550 data points for 500 epochs. Also, we trained the model from scratch on the *AnaData* from target inverters using the same numbers of data points for 500 epochs as well. We also repeated each training for ten times. Therefore, the total training count was: 4 (source inverter count)  $\times$  4 (target Inverter count)  $\times$  10 (repeating times)  $\times$  10 (number of data points) = 1,600, which took approximately 160 hours to complete all the training.

All transfer-learned models outperformed the models trained from scratch, and the transfer learning depended on considerably less data than the training from scratch did to achieve the same level of performance in terms of MSE (Fig. 9). The transfer-learned models even revealed promising extrapolation results when only 10 data points were used for fine-tuning. The transfer learning from Inverter 1, 2, and 3 to Inverter 4 performed exceptionally well, with close to zero MSE at only 5 data points. Subsequent to 30 data points, the MSEs of all transfer learned models began to approach zero, which were remarkably lower than those of the models trained from scratch. Furthermore, the probability densities constructed in the same way as Fig. 8 (when using 30 data points) all reflect the transfer learning's superior performance over training from scratch and its extremely mild reliance on database size. In Figs. 8 and 9, we trained the models in the TensorFlow platform using the Adam optimizer, the adopted FNN comprised 5,482 parameters in total (three hidden layers): layer #1, 4 neurons; layer #2, 43 neurons, layer #3, 56 neurons, layer #4, 43 neurons, and layer #5, 8 neurons.

Transfer learning mitigates the NN's dependency on large databases and empowers the NN with cross-extrapolation capability through the use of very small-scale databases from real-world simulations or measurements. This is extremely valuable for rapidly evaluating the stability of a future power grid with a large number of inverters with opaque black-box behaviors at the grid edge. It was shown that with the pre-trained models re-trained on only a few admittance data points, transfer-learned models captured the necessary information for re-constructing accurate admittance models of the grid-tied inverters, outperforming the models trained from scratch. We expect that this approach could greatly reduce the database required to achieve excellent modeling accuracy, especially when considering the unrealistic and expensive data acquisition process for simulations and real-world experiments, and could also save computational resources for cross-inverter model extrapolations using very small-scale databases.

#### IV. CONCLUSION

This paper presents an end-to-end machine-learning framework for data-driven modeling of model-free inverters. The machine learning framework spans from data acquisition (including analytical calculation, EMT simulation, and experimental measurement), model design, model training, to model evaluation. The modeling framework is evaluated and

validated through various test scenarios across a wide range of operating conditions. Transfer learning is found to be effective to reduce the data needed for training and for modeling particular inverters. These modeling tasks cannot be performed by traditional analytical modeling methods.

The future grid is a large-scale, sophisticated network comprising hundreds of millions of distributed passive and active electronics components functioning across a wide frequency spectrum. Their behaviors highly depend on the operating conditions (e.g., temperature, aging, power flow). The machine learning framework presented in this work can be extended to rapidly construct models for a large-scale power grid with many different components operating under sophisticated conditions and power flow. The transfer learning framework can reduce the need for large-scale databases and improves both the accuracy and efficiency of rapid modelings of black-box grid-tied energy systems.

#### REFERENCES

- [1] F. Blaabjerg, Y. Yang, K. A. Kim, and J. Rodriguez, "Power Electronics Technology for Large-Scale Renewable Energy Generation," *Proceedings of the IEEE*, vol. 111, no. 4, pp. 335–355, April 2023.
- [2] M. Chen and H. V. Poor, "High-frequency power electronics at the grid edge: a bottom-up approach toward the smart grid," *IEEE Electrific. Magazine*, vol. 8, no. 3, pp. 6–17, Sept. 2020.
- [3] X. Wang and F. Blaabjerg, "Harmonic stability in power electronic-based power systems: concept, modeling, and analysis," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 2858–2870, May 2019.
- [4] C. Li, R. Burgos, B. Wen, Y. Tang, and D. Boroyevich, "Analysis of STATCOM small-signal impedance in the synchronous d-q frame," *IEEE J. Emerg. Select. Topics Power Electron.*, vol. 8, no. 2, pp. 1894–1910, June 2020.
- [5] S. Wang, Z. Liu, J. Liu, D. Boroyevich, and R. Burgos, "Small-signal modeling and stability prediction of parallel droop-controlled inverters based on terminal characteristics of individual inverters," *IEEE Trans. Power Electron.*, vol. 35, no. 1, pp. 1045–1063, Jan. 2020.
- [6] B. Wen, D. Dong, D. Boroyevich, R. Burgos, P. Mattavelli, and Z. Shen, "Impedance-based analysis of grid-tied synchronization stability for three-phase paralleled converters," *IEEE Trans. Power Electron.*, vol. 31, no. 1, pp. 26–38, Jan. 2016.
- [7] X. Wang, L. Harnefors and F. Blaabjerg, "Unified impedance model of grid-connected voltage-source converters," *IEEE Trans. Power Electron.*, vol. 33, no. 2, pp. 1775–1787, Feb. 2018.
- [8] B. Wen, D. Boroyevich, R. Burgos, P. Mattavelli, and Z. Shen, "Analysis of D-Q small-signal impedance of grid-tied inverters," *IEEE Trans. Power Electron.*, vol. 31, no. 1, pp. 675–687, Jan. 2016.
- [9] Y. Liao and X. Wang, "Stationary-frame complex-valued frequency-domain modeling of three-phase power converters," *IEEE J. Emerg. Select. Topics Power Electron.*, vol. 8, no. 2, pp. 1922–1933, June 2020.
- [10] Y. Li, F. Diao, and Y. Zhao, "A generic multivector model predictive control with symmetric pulse pattern for hybrid multilevel converters," *IEEE Trans. Ind. Electron.*, vol. 68, no. 12, pp. 12185–12195, Dec. 2021.
- [11] H. Gong, X. Wang, and D. Yang, "DQ-frame impedance measurement of three-phase converters using time-domain MIMO parametric identification," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 2131–2142, Feb. 2021.
- [12] F. Diao, X. Du, Z. Ma, Y. Wu, F. Guo, Y. Li; Z. Zhao, and Y. Zhao, "A megawatt-scale Si/SiC hybrid multilevel inverter for electric aircraft propulsion applications," *IEEE J. Emerg. Select. Topics Power Electron.*, vol. 11, no. 4, pp. 4095–4107, Aug. 2023.
- [13] Y. Zhang, F. Udrea, and H. Wang, "Multidimensional device architectures for efficient power electronics," *Nature Electronics*, vol. 5, pp. 723–734, Nov. 2022.
- [14] M. Zhang, X. Wang, D. Yang, and M. G. Christensen, "Artificial neural network based identification of multi-operating-point impedance model," *IEEE Trans. Power Electron.*, vol. 36, no. 2, pp. 1231–1235, Feb. 2021.
- [15] M. Zhang, Q. Xu and X. Wang, "Physics-informed neural network based online impedance identification of voltage source converters," *IEEE Trans. Ind. Electron.*, vol. 70, no. 4, pp. 3717–3728, April 2023.

- [16] Y. Li, Y. Liao, X. Wang, L. Nordström, P. Mittal, M. Chen, and H. P. Poor, "Neural network models and transfer learning for impedance modeling of grid-tied inverters," *2022 IEEE 13th International Symposium on Power Electronics for Distributed Generation Systems (PEDG)*, Kiel, Germany, 2022, pp. 1-6.
- [17] Y. LeCun, Y. Bengio and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, May 2015.
- [18] L. G. Wright, T. Onodera, M. M. Stein, T. Wang, D. T. Schachter, Z. Hu, and P. L. McMahon, "Deep physical neural networks trained with backpropagation", *Nature Electronics*, vol. 601, pp. 549–555, Jan. 2022.
- [19] Y. Li, Y. Wang, and B. Q. Li, "Generalized theory of phase-shifted carrier PWM for cascaded H-bridge converters and modular multilevel converters," *IEEE J. Emerg. Select. Topics Power Electron.*, vol. 4, no. 2, pp. 589-605, June 2016.
- [20] R. Rosso, X. Wang, M. Liserre, X. Lu, and S. Engelken, "Grid-forming converters: control approaches, grid-synchronization, and future trends—a review," *IEEE Open Journal of Ind. Appl.*, vol. 2, pp. 93-109, 2021.
- [21] X. Wang, M. G. Taul, H. Wu, Y. Liao, F. Blaabjerg, and L. Harnefors, "Grid-synchronization stability of converter-based resources—an overview," *IEEE Open Journal of Ind. Appl.*, vol. 1, pp. 115-134, 2020.
- [22] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: a next-generation hyperparameter optimization framework," in *KDD*, 2019.
- [23] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization", 2014, <https://arxiv.org/abs/1412.6980>.
- [24] Y. Liao, Y. Li, M. Chen, L. Nordström, X. Wang, P. Mittal, and H. V. Poor, "Neural network design for impedance modeling of power electronic systems based on latent features," *IEEE Transactions on Neural Networks and Learning Systems*, 2023, to be published, doi: 10.1109/TNNLS.2023.3235806.
- [25] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning", *Journal of Big Data*, vol. 3, No. 9, May 2016.
- [26] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43-76, Jan. 2021..
- [27] S. Choudhury, M. Moret, P. Salvy, D. Weilandt, V. Hatzimanikatis, and L. Miskovic, "Reconstructing kinetic models for dynamical studies of metabolism using generative adversarial networks", *Nature Machine Intelligence*, vol. 4, pp. 710–719, 2022.
- [28] E. Dogariu, H. Li, D. Serrano López, S. Wang, M. Luo and M. Chen, "Transfer learning methods for magnetic core loss modeling," *2021 IEEE 22nd Workshop on Control and Modelling of Power Electronics (COMPEL)*, 2021.
- [29] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: advances and trends", *IEEE Trans. Ind. Electron.*, vol. 64, no. 2, pp. 935–947, Feb. 2017.
- [30] I. Harbi et al., "Model-predictive control of multilevel inverters: challenges, recent advances, and trends," *IEEE Trans. Power Electron.*, vol. 38, no. 9, pp. 10845-10868, Sept. 2023.
- [31] T. Blalock, J. Ortiz, J. Frankle, and J. Gutttag, "What is the state of neural network pruning? *Proceedings of Machine Learning and Systems 2020 (MLSys 2020)*, 2020.