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

SUBMISSION DATE / POSTED DATE

21-02-2023 / 23-02-2023

CITATION

Li, Yufei; Liao, Yicheng; Chen, Minjie; Wang, Xiongfei; Nordström, Lars; Mittal, Prateek; et al. (2023): Few-Shot Machine Learning at the Grid-Edge: Data-Driven Impedance Models for Model-Free Smart Inverters. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.22137905.v1

DOI

10.36227/techrxiv.22137905.v1

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

Yufei Li[1], Yicheng Liao[2], Minjie Chen[1*], Xiongfei Wang[3*], Lars Nordström[3], Prateek Mittal[1] and H. Vincent Poor[1]

[1]Electrical and Computer Engineering, Princeton University, 86 Olden Street, Princeton, 08544, NJ, USA.
[2]Energinet, Tonne Kjærsvej 65, Fredericia, 7000, Danmark.
[3]Electrical Engineering, KTH Royal Institute of Technology, Lindstedtsvägen 3, Stockholm, 10044, Sweden.

*Corresponding author(s). E-mail(s): minjie@princeton.edu; xiongfei@kth.se;
Contributing authors: yl5385@princeton.edu; yli@energinet.dk; larsno@kth.se; pmittal@princeton.edu; poor@princeton.edu;

Abstract

The future electric grid will be pervasively supported by a large number of smart inverters distributed at the grid edge, whose dynamics are critical for grid stability and resiliency. The operating conditions of these inverters may vary across a wide range, leading to various impedance patterns and complicated grid-inverter interaction behaviors. Existing analytical impedance models require thorough and precise understandings of system parameters and make numerous assumptions to reduce the system complexity. They can hardly capture complete electrical behaviors of physical systems when inverters are controlled with sophisticated algorithms or performing complex functions. Real-world impedance acquisitions across multiple operating points through simulations or measurements are expensive and impractical. Leveraging the recent advances in artificial intelligence and machine learning, we present the InvNet, a few-shot machine learning framework that is capable of characterizing inverter impedance patterns across a wide operation range when only limited impedance data for each inverter is available. The InvNet is capable of extrapolating from physics-based models to real-world models and from inverters to inverters. Comprehensive evaluations were conducted to verify the effectiveness of the proposed approach in various application scenarios. All data and models were open-sourced. We showcase machine learning and neural networks as powerful tools for modeling black-box characteristics of sophisticated grid-edge energy systems and their capabilities of analyzing behaviors of larger-scale systems that cannot be described via traditional analytical methods.

1 Introduction

As the global energy system is witnessing a paradigm shift, where renewable energy resources are rapidly penetrating modern power systems, the power electronic converters at the grid-edge are playing more important roles than ever before [1, 2]. Grid-edge inverters are ubiquitously needed to connect renewable energy resources to the future electric grid. The dynamics of grid-edge inverters, commonly characterized as *impedances* under small-signal perturbations, are critical for grid stability analysis and control [3, 4]. The impedance describes the converters' frequency-dependent behaviors under external small-signal disturbances [5, 6]. The operating conditions of these inverters may change across a wide range, leading to different impedance patterns requiring precise modeling for system-level analyses [7]. Physics-based small-signal models derived from circuit analyses can capture the impedances of grid-edge inverters when all physical parameters for simplified operating conditions are known and when inverters are controlled by classic control strategies [8, 9].

However, the physical parameters of grid-edge inverters usually change with particular hardware and software implementations and operating conditions. Different power stage designs and different control frameworks may lead to different physical parameters and nonidealities, limiting the applicability of physics-based analytical models when significant nonidealities and/or non-modellable nonlinearities exist. Example 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-edge inverters are often controlled by nonlinear controllers when performing smart and sophisticated functions (e.g., low-frequency ride through, active power sharing, black-start). Therefore, impedance modeling of grid-edge inverters via analytical models are prone to be unreliable and inaccurate [8], especially when system parameters are unknown, or sometimes kept confidential by power converter manufacturers due to security and intellectual property considerations [1]. In a future electric grid with a large number of inverters at the edge, it is impractical to assume that physical parameters of inverters are known to system operators, let alone analytical models.

The inverter impedance can change significantly depending on the operating conditions. Measuring or simulating the impedances of clusters of inverters in real-time and real-world is possible, but expensive and impractical. The data acquisition process of real-world impedance data across multiple operating points (OPs, including voltage - V, frequency - f, real power - P, reactive power - Q) via experiments or electromagnetic transient (EMT) simulations usually require tremendous computational resources and cannot satisfy real-time needs. Only a small amount of measurement data is available for each type of inverters.

Leveraging the recent advances in artificial intelligence and machine learning, we present the InvNet, a simple but effective physics-informed machine learning framework that is capable of characterizing inverter impedance patterns across a wide operation range when only a small amount of measurement data is available [10]. As shown in Fig. 1, InvNet is an end-to-end framework comprising automatic data acquisition, scalable model training, and comprehensive model validation. It is modular, scalable, and flexible for modeling a large number of smart inverters at the grid-edge with sophisticated control structures. Leveraging transfer learning, we substantially reduced the InvNet's demand for large-scale experimental data while maintaining precise impedance predictions by pre-training the model with 1) data from simplified/incomplete analytical models, or 2) data collected from other similar inverter implementations. The InvNet framework will further advance system-level studies for grid-edge inverter systems and pave the way toward more data-driven approaches for multi-scale, multi-physics modeling of a future electric grid with a high percentage of renewable energy resources and power electronic converters at the edge.

2 Machine learning methods for inverter impedance characterization

As shown in Fig. 1a, the key principle of this approach is to model the impedance of each grid-edge inverter across a wide operation range as a data-driven digital twin, such as a neural network (NN). The grid-edge inverters can be implemented as different topologies ranging from two-level inverters to multilevel inverters [11], adopting very different control strategies (e.g. grid-forming or grid-following [12], current source or voltage source behaviors [13], etc.), and performing different functions (e.g., black start, low frequency 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, where 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-lock-loop (PLL) implementation (Components with Number 1 through 6 in Fig. 1a). The inverter admittances, 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 (Fig. 1b), i.e., the conductance and susceptance.

In this work, the data used for NN training is obtained from: 1) analytical models, 2) electromagnetic transient (EMT) simulations, and 3) experimental measurements. Fast Fourier transform (FFT) is utilized to transform the voltage and current to frequency-dependent impedances (complex values on the left of Fig. 1b). The surrogate model is obtained through training a small-scale feedforward neural network (FNN) with OP information (V, f, P, Q) as its inputs and admittances as outputs (Fig. 1c). State-of-the-art NN hyperparameter tuning tools and optimizers, such as Optuna [14] and Adam [15], are used to optimize the NN structure and parameters (Fig. 1c). We have also demonstrated the strength of transfer learning [16, 17] with InvNet. The size of the data needed to obtain a high performance NN model can be greatly reduced by leveraging data created from existing simplified/incomplete analytical models, or data obtained from measuring similar inverters but with different parameters. With InvNet, one can take a few quick snapshots of impedances of a new inverter at a few OPs and rapidly predict the behaviors of this new inverter across a wide operation range.

3 Results

3.1 Performance evaluation under unknown parameter and no analytical model circumstances

We demonstrate the advantages of the InvNet framework over traditional physics-based analytical methods under the following two scenarios: 1) when unknown information (not captured by analytical models) exists, and 2) when analytical models do not exist, or are incomplete. We collect data from analytical calculations, EMT simulations, and experimental measurements. Each type of data is randomly partitioned into a training set (70% of total dataset), a validation set (15% of total dataset), and a test set (15% of 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 training processes.

To facilitate the performance evaluation, four different grid-edge inverters with parameters demonstrated in Table 1 were exemplified throughout this article. Two datasets collected through analytical calculations and automatic EMT simulations (using the simulation platform of PLECS integrated with MATLAB Simulink), i.e., CalcData and SimData, respectively, were applied to train the FNN, resulting in two NNs, i.e., CalcNN and SimNN, respectively.

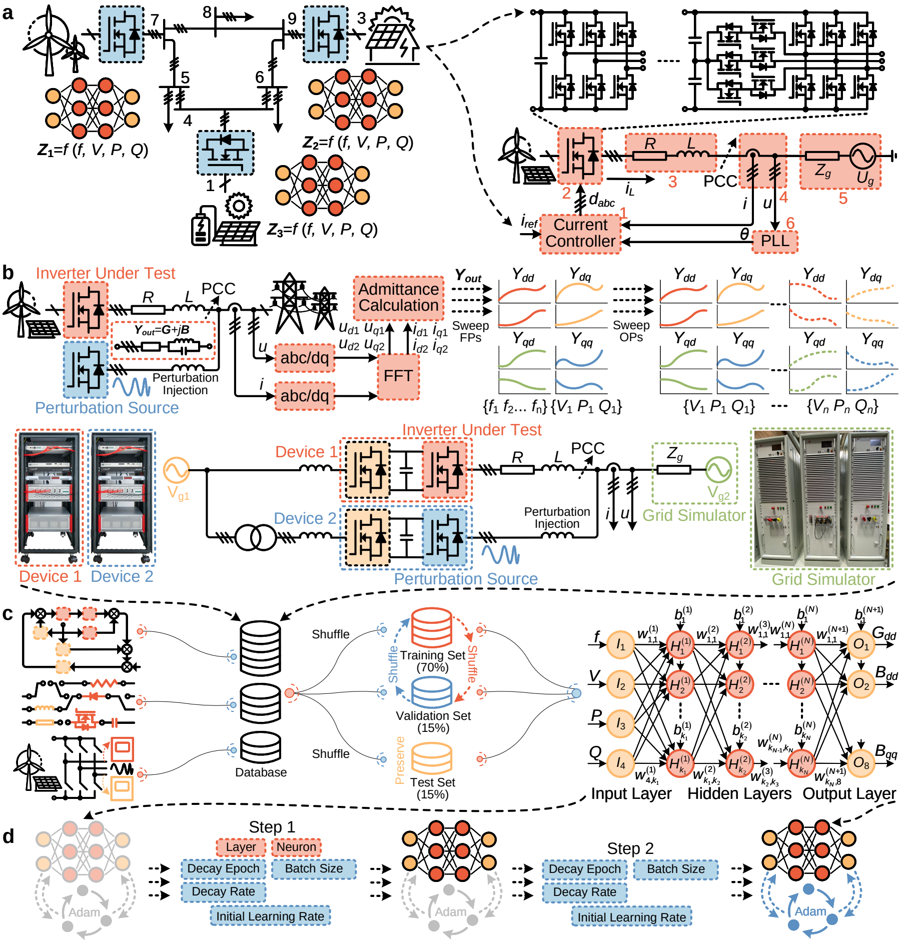


Fig. 1 Overview of the InvNet framework and methodologies. (a) **Key Principle.** Left: Modeling inverter impedances as NNs for stability analyses of distribution networks comprising a variety of inverters. Right: The system diagram of typical grid-edge inverters with PLL-based current control, where nonidealities and nonlinearities in the physical and control systems (such as sampling errors, parameter discrepancies, and converter dead-time, etc.) make precise impedance modeling highly challenging. (b) **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 experimental platform for impedance data acquisition. (c) **Model Training.** Training an FNN in the TensorFlow platform with data collected trough either analytical models, electromagnetic transient (EMT) simulation models in MALTAB+PLECS, or real-world experiments, where the OPs serve as inputs and admittances as outputs. (d) **Hyperparameter Tuning.** Hyperparameter tuning process for the InvNet using the Optuna framework.

The two datasets were constructed in a similar 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

Table 1 Parameters of Inverters

	Inverter 1	**Inverter 2**	**Inverter 3**	**Inverter 4**
PLL Bandwidth (B_{pll})	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 mΩ			
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.

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 has 800 (20×40) $\{f, V, P, Q\}$ data points in total. Each data point comprises an admittance matrix $[Y_{dd}, Y_{dq}; Y_{qd}, Y_{qq}]$. The EMT-simulation-generated SimData was visulaized in the Extended Data Fig. 2 as an example of the database. Both CalcData 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 CalcNN and SimNN.

Traditional analytical impedance models need thorough and comprehensive understandings of both control and system parameters, such as circuit component parameters, bandwidths and intrinsic structures of the PLL, parameters and architectures of the control system, etc., such that the impedance model can be analytically derived [9] (Methods). However, these information are often unknown or even kept confidential from manufacturers. Even though the analytical impedance model exists and the complete system information can be known, to date, precise impedance modeling is still being throttled by the difficulties in modeling the nonideal-switching impact on inverter impedances, such as the dead-time impacts [18]. Many existing and emerging inverter implementations adopt nonlinear controllers, such as the model predictive control (MPC) [19], for which there are no analytical models to describe the impedance behaviors thus far, due to their intrinsically sophisticated functions and behaviors.

In this work, to mimic a real-world scenario when dealing with a wide range of different inverters with unknown information, we applied deadtime to the switching events in the EMT simulation. The deadtime 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 the analytical model may use incorrect system parameters (SimData was directly

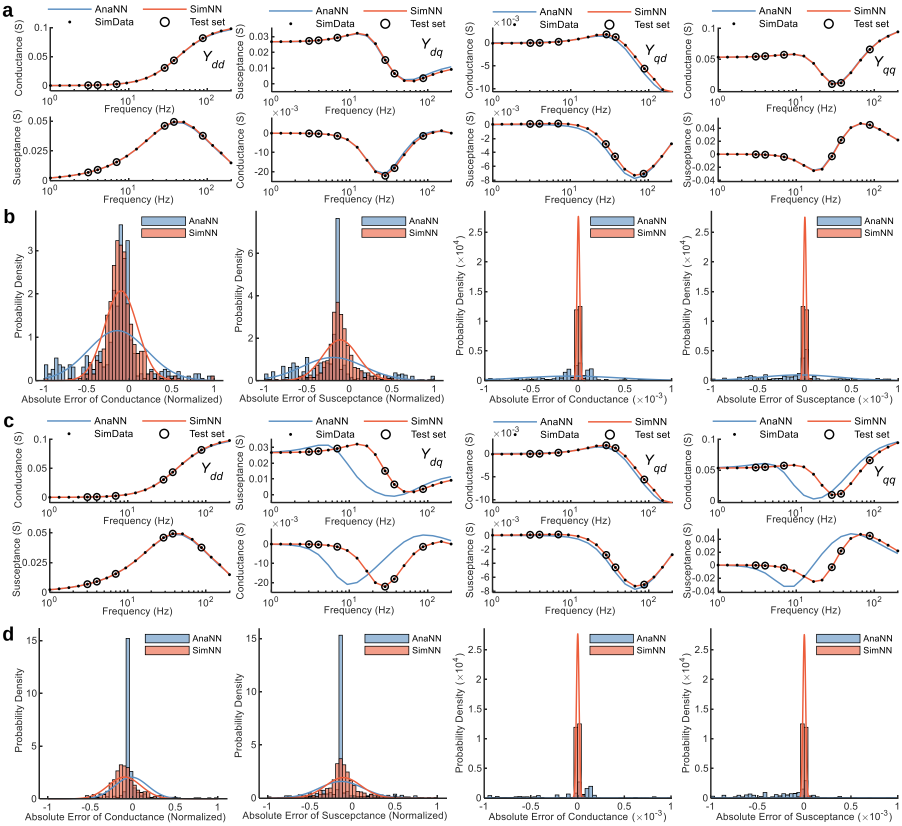


Fig. 2 Performance evaluation under unknown parameter circumstances. (a) Prediction results of the CalcNN and the SimNN compared to the test set at the same example OP (the No. 23 OP in both the CalcData and SimData, where V, P, and Q are $\{1.1, -1, 0.5\}$), and the SimNN captures the dead-time effect while the CalcNN does not. The SimNN achieves better accuracy than the CalcNN. (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 CalcNN and SimNN compared to the test set at the same example OP (the No. 23 OP in both the CalcData and SimData, where V, P, and Q are $\{1.1, -1, 0.5\}$), and the SimNN captures the complete information of the real system while the CalcNN, unaware of system parameters, fails to establish an accurate admittance model. The SimNN achieves better accuracy than the CalcNN. (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.

captured from EMT simulations without the necessity of recognizing system parameters). Similar methods can be used to study the impact of other system parameters on inverter impedances.

We first showcase 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, the existing analytical model fails to complete an accurate modeling for not capturing the dead-time effect, thus, the

CalcData does not comprise dead-time information. However, as the EMT simulation is able to establish the dead-time in each power switch of the inverter, the SimData contains the dead-time information, which cannot be captured by the analytical model. As depicted in Fig. 2a, the SimNN curve differs slightly from the CalcNN curve in Y_{dd} and Y_{qq}, while reveals obvious deviations from the CalcNN in Y_{dq} and Y_{qd}, for both conductance and susceptance curves, which implies the inaccuracy of the analytical model. The test set, stochastically extracted from the SimData, is in line with the SimNN curve, while inconsistent with the CalcNN 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 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. 2b, the probability densities of the normalized absolute errors of the SimNN are more concentrated around zero than those of the CalcNN for both the conductance and susceptance, which indicates that admittances predicted by the SimNN are more accurate than the ones predicted by the CalcNN. Further, we also visualized the probability density histograms of the absolute errors with their normal distribution fits in the right two graphs of Fig. 2b. The probability densities of the real absolute errors of the SimNN are in sharp contrast to the ones of the CalcNN, the former ones are remarkably concentrated around zero while the later ones are flattened, which confirms the superiority of the SimNN over the CalcNN.

Next, to further demonstrate the advantages of the InvNet when dealing with classified or uncertain information, we proceed to use Inverter 1 as the CalcData source and Inverter 2 as the SimData source, assuming that the analytical model has no perceptions on actual inverter parameters. Similarly, as illustrated in Fig. 2c, the SimNN unfolds apparent discrepancies with the CalcNN in Y_{dq}, Y_{qd}, and Y_{qq}, while differs slightly from the CalcNN in Y_{dd}. The normalized probability densities of the absolute errors are depicted in the left two graphs of Fig. 2d, where the histograms of SimNN are more like the normal distributions. The actual probability densities of the absolute errors are given in the right two graphs of Fig. 2d, where the probability densities of the absolute errors of the SimNN are also in stark contrast to the ones of the CalcNN, the former ones are mostly located near zero while the later ones span across a wide range which are almost close to the zero baseline. This further confirms the effectiveness of the InvNet in dealing with scenarios when system parameters are unknown.

We proceeded to demonstrate the capability of the InvNet in coping with no analytical model circumstances. Replacing the PI current controller (Fig. 1a) with the nonlinear multi-vector MPC (Methods) makes the entire grid-edge inverter system non-modellable–there is no analytical model that is able to describe the impedance pattern of the inverter. We established the automatic

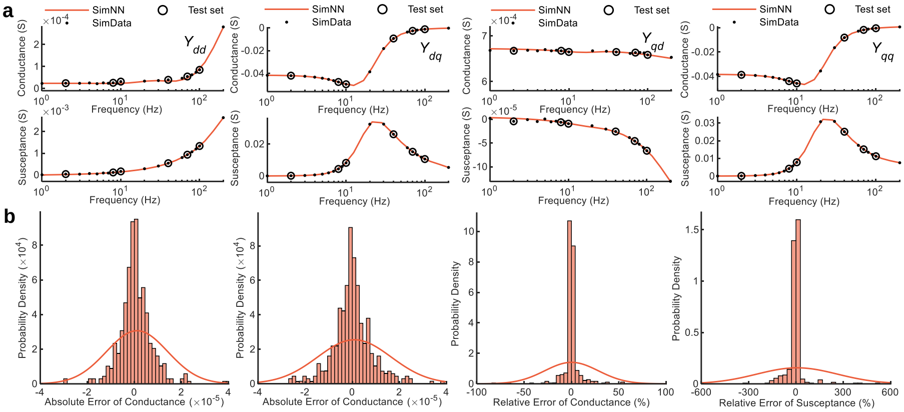


Fig. 3 Performance evaluation under no analytical model circumstances. (a) The SimNN is trained on the SimData collected from the multi-vector-MPC-controlled inverter with the same PLL bandwidth as Inverter 2. Prediction results of the SimNN are compared with the test set at the same example OP (the No. 23 OP in the SimData, where V, P, and Q are $\{1.1, -1, 0.5\}$). The SimNN are well-matched with the test set. (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.

EMT simulations to collect the admittance data (SimData) for an inverter with the same PLL bandwidth as Inverter 2, but controlled by multi-vector MPC. Here, we picked up 19 FPs as ones, tens, and hundreds in the range of $f \in [1, 200]$, which are $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 60, 70, 80, 90, 100, 200\}$. The test set was also randomly selected and reserved from the SimData by 15% and used for model evaluations. As shown in Fig. 3a, at an arbitrary OP, the test set data points all align with the SimNN curve, which proves 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 follow a normal distribution and are mostly concentrated around an extremely narrow band around zero (left two graphs of Fig. 3b), which confirms the accuracy of SimNN predictions. Moreover,the probability density histograms of the relative errors are also largely concentrated around a narrow band around zero (right two graphs of Fig. 3b), which further confirms the accuracy of the predictions.

3.2 Few-shot experiments and model extrapolation using transfer learning

Usually, to achieve comprehensive understandings of inverter impedances across wide OP ranges, it is necessary to train the NN on a comparatively large amount of data. However, acquiring large database is never a trivial process especially when it comes to EMT simulations and real-world experiments. 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 FPs and injects perturbations twice for each FP, then sweep all investigated OPs. Moreover, the collected voltage and current data needs to undergo the FFT to obtain the impedance data (Fig. 1b). Even though the pseudorandom binary sequence (PRBS) signal method [20], 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 based on Silicon Carbide (SiC) or Gallium Nitride (GaN) power devices [21], making impedance data collections even more challenging.

To this end, we leverage transfer learning to reduce the database volume and empower the InvNet framework with the ability of cross-inverter extrapolation (Fig. 4a). We first demonstrate the extrapolation from an FNN trained on data from analytical models, which may not perceive correct impedance information due to insufficient parameter information from the inverter system, to an FNN understanding the impedance knowledge from the real-world experiments. The FNN was firstly 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. An experimental rig (Fig. 1b) was constructed in our lab to collect the real-world impedance data. A power amplifier to the right with an attached inductive impedance Z_g works as a grid simulator and a power source V_{g1} to the left mimics the renewable energy sources. The inverter under test has the same parameters as Inverter 1 in Table 1. There are two devices ("Microgrid Tech Bench" from Imperix) with a back-to-back converter structure in each one of them, where one of the converters works 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 are connected through an inductive impedance Z_g to a power amplifier ("DM 45000/APS" from Spitzenberger & Spies) which works as a grid simulator to mimic the power grid. The power source V_{g1} to the left of Fig. 1b mimics the renewable energy sources and is also connected through interfacing inductors to the devices, where Device 2 is connected through a transformer to achieve galvanic isolation between Device 1 and 2. The inverter in Device 1 is the one under test (has the same parameters as Inverter 1 in Table 1) for admittance measurements while the inverter in Device 2 serves as a perturbation source injecting perturbations for admittance measurements. To comparatively expedite the admittance data collection process, the PRBS method was applied here [20]. 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 contains 43 OPs, which constructs a database with 817 (19 × 43) $\{f, V, P, Q\}$ data points in total and is referred to as the ExData. This ExData was also randomly partitioned into a 70% training set, a 15% validation set used for monitoring training process, and a 15% test set reserved for final model evaluations.

We pre-trained the model on the CalcData generated from analytical models for Inverter 2, 3, and 4 (Table 1). Due to the simplicity of generating data from analytical models, we were able to construct a large database as CalcData. Similarly to Fig. 2, 20 frequency points 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 × 1084) data points in total. We then fine-tuned the model on the ExData (from Inverter 1) using 5, 10, 30, 50, 100, 150, 200, 300, 400, and 550 data points. To achieve fair comparisons, we also trained the model from scratch on the ExData using the same number of data points as well. We repeatedly trained each model for ten times with a different random seed each time and recorded each evaluation result at the end of each training for further analyses. As seen from the left two figures in Fig. 4b, the transfer learned models reveal remarkably low mean squared error (MSE) even when trained on only 30 data points, for both conductances and susceptances. More astonishingly, the MSE of the transfer learned models even approaches the zero line at only 100 data points. Moreover, the transfer learned models considerably outperformed the models trained from scratch. From the MSE perspective, the models trained from scratch can only reach the 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 the used data points were fewer than 50. We also evaluated the probability densities of the absolute errors (right two figures in Fig. 4b). Similar to the studies in Fig. 2, the probability density histograms together with a normal distribution fit for each histogram (when using 30 data points) are also shown. The probability densities of absolute errors of the transfer learned models are more concentrated around zero than those of the model trained from scratch, which indicates that admittances predicted by the transfer learned models are more accurate than the ones predicted by the model trained from scratch.

In addition to the aforementioned model extrapolation from analytical data-based model to experimental data-based model, we also performed the cross-extrapolations, i.e., from all other three inverters to one inverter in Table 1 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. 4 (with each training process repeated ten times).

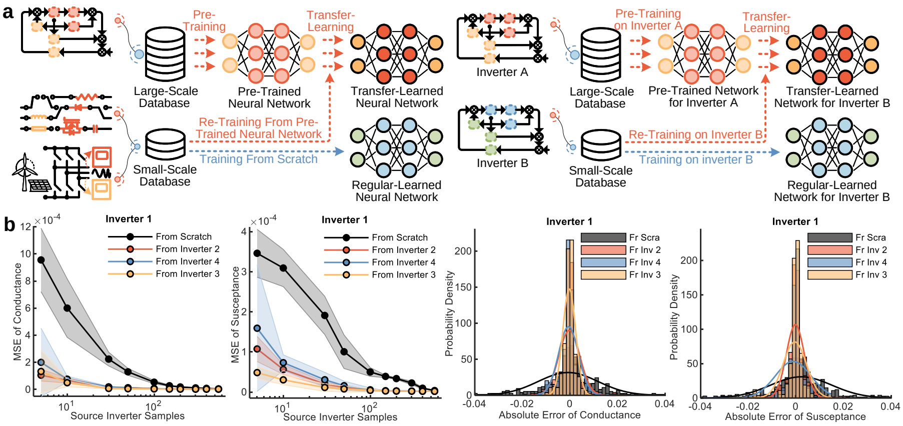

a
Pre-Training
Transfer-Learning
Large-Scale Database
Pre-Trained Neural Network
Transfer-Learned Neural Network
Re-Training From Pre-Trained Neural Network
Training From Scratch
Small-Scale Database
Regular-Learned Neural Network
Inverter A
Inverter B
Pre-Training on Inverter A
Pre-Trained Network for Inverter A
Transfer-Learning
Transfer-Learned Network for Inverter B
Re-Training on Inverter B
Training on inverter B
Regular-Learned Network for Inverter B
b
Inverter 1
From Scratch
From Inverter 2
From Inverter 4
From Inverter 3
MSE of Conductance
MSE of Susceptance
Source Inverter Samples
Probability Density
Fr Scra
Fr Inv 2
Fr Inv 4
Fr Inv 3
Absolute Error of Conductance
Absolute Error of Susceptance

Fig. 4 The transfer learning framework and performance evaluations for the transfer learning from analytical data-based model to experimental data-based model. (a) Transfer learning framework. The InvNet uses 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). (b) Performance evaluations for the transfer learning. The MSE curves of the transfer learned models are 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-time trainings.) and the probability densities of the absolute errors of the transfer learning are more concentrated than those of the models learned from scratch (when using 30 data points), indicating the superior performance of the transfer learning.

All the 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 performances in terms of MSE (Fig. 5). 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 begin to approach zero, which is remarkably lower than those of the models trained from scratch. Furthermore, the probability densities constructed in the same way as Fig. 4 (when using 30 data points) all reflect the transfer learning's superior performance over the learning from scratch and its extremely mild reliance on database size.

We conclude that leveraging transfer learning enables the InvNet's less dependency on large databases and empowers the InvNet 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 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-edge inverters, outperforming the models trained from scratch. We expect that this approach could greatly reduce the

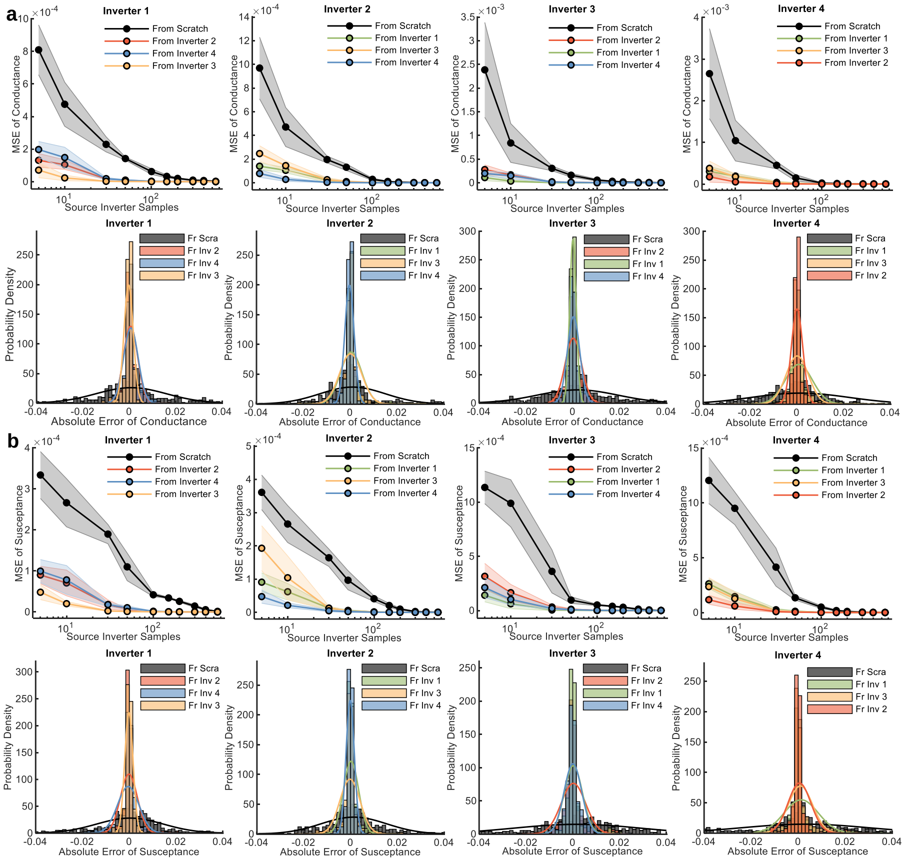


Fig. 5 Performance evaluations for cross-inverter extrapolations using transfer learning. (a) Conductance evaluations: The MSE curves were visualized by adding standard deviations as margins of the mean MSE value for the ten-time trainings. The transfer learned models outperformed the models learned 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 conductance 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).

database needed to achieve excellent modeling accuracy, especially when considering the extremely time-consuming data acquisition process for real-world experiments and simulations, and could also save computational resources for cross-inverter model extrapolations using very small-scale databases.

4 Discussion and conclusions

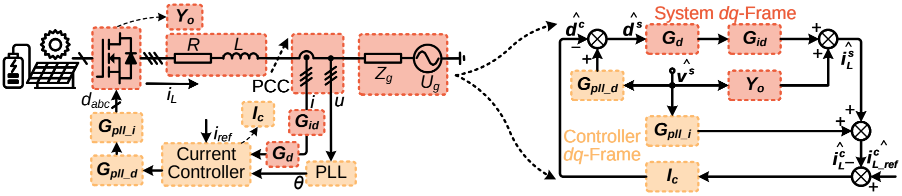
Using analytical models for the impedance modeling of grid-edge inverters are sometimes unreliable, inaccurate, and not scalable, especially when system parameters are unknown or kept confidential for cybersecurity or intellectual property protection purposes. A number of practical issues hinder the application of analytical models: 1) The dead-time effects and high frequency switching actions of power devices cannot be effectively modelled, making analytical models inaccurate in capturing the full-spectrum characteristics of the inverter impedances. 2) The inverter system parameters are often unknown, which makes the application of analytical models impractical. 3) The recent advances in inverter control usher in several nonlinear control strategies such as the MPC, for which no mature analytical models have been derived thus far. 4) The acquisition of high-quality real-world impedance data via experiments or simulations is always expensive and impractical, especially when dealing with multiple OP circumstances or a large number of geographically distributed inverters, which therefore, require massive computational efforts and enormous manpower resources. Measuring or simulating the impedances of clusters of inverters in real-world for stability evaluations is even more impractical. To address such issues, we proposed a data-driven InvNet framework for grid-edge inverter impedance modeling, which not only solves the hardship in multiple-OP impedance modeling, but also address the problem of large database requirements for the high-performance NN construction. We showcased a data-driven NN framework with standard FNN models that can replace the physics-based impedance model in system-level stability analyses for grid-edge inverters, and can promptly predict the output impedances based on the operating conditions of inverters. This approach is parameter-agnostic and completely data-driven, and thus does not rely on precise understandings of the inverter hardware or software implementations. In addition, leveraging transfer learning, the InvNet is able to extrapolate from analytical data-based models to real-world data-based models using a small-scale real-world database, and also able to cross-extrapolate among various inverters using small-scale databases.

In summary, we present a machine learning framework to rapidly construct impedance models for grid-edge inverters across wide ranges of OPs. The generated models can be further used for system-level stability studies for grid-edge inverters. Researchers can train the models on small-scale databases thanks to the use of transfer learning, and promptly establish the desired impedance models at specific OPs. We confirm that this proposed framework reduces the need for large-scale databases and improves both accuracy and efficiency of impedance modelings for grid-edge inverters. We expect that the proposed framework can further advance system-level studies for grid-edge inverter system and pave the way toward more data-driven approaches.

5 Methods

5.1 Analytical impedance models

There are many different ways of implementing grid-edge inverters. Different hardware and software implementations lead to different complexities that are



Extended Data Fig. 1 The small-signal impedance model of grid-edge inverters with current control loop and PLL. The control parameters are assumed to be unknown and may vary from inverters to inverters and from manufacturers to manufacturers.

hard to be captured by analytical models. grid-edge inverters are typically controlled as current sources by current controllers with a phase-locked loop (PLL) [6], also known as grid-following inverters [22] (Fig. 1a).

Small-signal linearized models for grid-edge inverters operating at specific equilibrium points are well-studied [9]. The time delay matrix $\boldsymbol{G_d}$ of the digital control system, as shown in Fig. 1, is given by

$$\boldsymbol{G_d} = \begin{bmatrix} e^{-1.5T_s s} & 0 \\ 0 & e^{-1.5T_s s} \end{bmatrix}, \tag{1}$$

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

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

where ω is the fundamental angular frequency of the system, L and R are the inductance and resistance in Fig. 1a. The decoupled current controller matrix $\boldsymbol{I_c}$ can be expressed as

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

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 operating point yield

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

and

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

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 $\boldsymbol{G_{pll\_i}}$ from the system voltage to the current in the controller d-q frame and the small-signal perturbation path matrix $\boldsymbol{G_{pll\_d}}$ from the system voltage to the duty cycle in the controller d-q frame are, respectively, given by

$$\boldsymbol{G_{pll\_i}} = \begin{bmatrix} 0 & I_q G_{pll} \\ 0 & -I_d G_{pll} \end{bmatrix}, \tag{6}$$

$$\boldsymbol{G_{pll\_d}} = \begin{bmatrix} 0 & -D_q G_{pll} \\ 0 & D_d G_{pll} \end{bmatrix}. \tag{7}$$

The open-loop output admittance without the PLL is derived by forcing the perturbations of the duty ratio and dc voltage to zero [9], thus

$$\boldsymbol{Y_o} = \begin{bmatrix} \frac{Ls+R}{(Ls+R)^2+(\omega L)^2} & \omega L \\ -\omega L & \frac{Ls+R}{(Ls+R)^2+(\omega L)^2} \end{bmatrix}. \tag{8}$$

The output admittance of the grid-edge inverter system is

$$\boldsymbol{Y_{out}} = \left[\boldsymbol{I} - \boldsymbol{G_d G_{id} I_c}\right]^{-1} \cdot \left\{\boldsymbol{G_d G_{id}}\left[\boldsymbol{G_{pll\_d}} + \boldsymbol{I_c G_{pll\_i}}\right] + \boldsymbol{Y_o}\right\}. \tag{9}$$

$\boldsymbol{Y_{out}}$ can be typically represented by a 2×2 matrix with four complex elements:

$$\boldsymbol{Y_{out}} = \begin{bmatrix} Y_{dd} & Y_{dq} \\ Y_{qd} & Y_{qq} \end{bmatrix}. \tag{10}$$

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-edge inverter at a particular operating point 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 [18]. This analytical model requires precise knowledge of the circuit and control parameters of inverters, which may not be accurate, or may even kept confidential in practical scenarios.

5.2 Database construction

Fig. 1b illustrates the admittance acquisition process. 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 ($\boldsymbol{G}$) and a four-element susceptance matrix ($\boldsymbol{B}$):

$$\boldsymbol{G} = \begin{bmatrix} G_{dd} & G_{dq} \\ G_{qd} & G_{qq} \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} B_{dd} & B_{dq} \\ B_{qd} & B_{qq} \end{bmatrix}. \tag{11}$$

In this work, we used three types of data for performance evaluations, i.e., the CalcData generated by analytical models, the SimData generated by EMT simulations, and the ExData collected through real-world experiments. To facilitate the data acuisition 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. For the SimData used in Fig. 2 that comprised 40 OPs and 20 FPs, it took approximately 40 hours to complete all iterations on a PC with Intel 11th Gen i7-11700 processor. The ExData, by using the method of admittance measurements presented in [20], was collected through experiments conducted in the testing platform demonstrated in Fig. 1b, the entire process of which took us more than one week to construct a database comprising 817 data points.

5.3 Hyperparameter optimization

The optimization of NN hyperparameters were 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 (Fig. 1d). First, the NN struture, namely, the layer and neuron numbers and the Adam optimizer [15] 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 the 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 performed in this work implied that the obtained hyperparameters did not lead to overfit and achieved optimal performances.

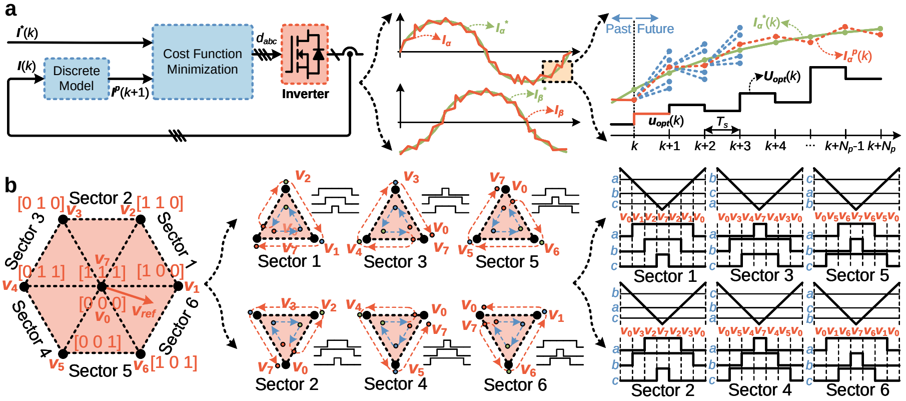
5.4 Neural network training

We developed FNNs to model the admittances of a grid-edge inverter in this work. Fig. 1c 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 ($\boldsymbol{G}$) and the susceptance ($\boldsymbol{B}$) of the output admittance, i.e., G_{dd}, B_{dd}, G_{dq}, B_{dq}, G_{qd}, B_{qd}, G_{qq}, and B_{qq}.

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. For the studies in Fig. 2, 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 [23] 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 that generalizes well. We trained each model for 3000 epochs, which took approximately 8 minutes for each training process on the same PC. For studies in Fig. 3, the FNN structure was the same as the previous one which also had three hidden layers with 15 neurons in each layer. We also trained the model in Matlab using Bayesian regularization optimizer for 1000 epochs, which took roughly less than 3 minutes. For the transfer learning evaluations in Figs. 4 and 5, we trained the models in the TensorFlow platform using the Adam optimizer, the adopted FNN comprised 5482 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.

5.5 Transfer learning

For the extrapolation from analytical data-based model to experimental data-based model, we first pre-trained the model on the CalcData generated from analytical models (from Inverter 2, 3, and 4) with 21680 (20×1084, 1084 OPs, 20 FPs) data points in total for 500 epochs. 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. For comparison purposes, we also trained the model from scratch on the ExData using the same number of data points for 500 epochs as well. We repeated each training for 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. In this regard, the total training count was: 3 (source inverter count)×1 (target Inverter count)×10 (repeating 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). For the cross-inverter extrapolations, the performance evaluations were conducted in similar ways. We first pre-trained the model on the CalcData generated from analytical models of source inverters (from Inverters 1, 2, 3, and 4) with 21680 (20×1084, 1084 OPs, 20 FPs) data points in total for 500 epochs. We then fine-tuned the model on the other CalcData from traget 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 CalcData from target inverters using

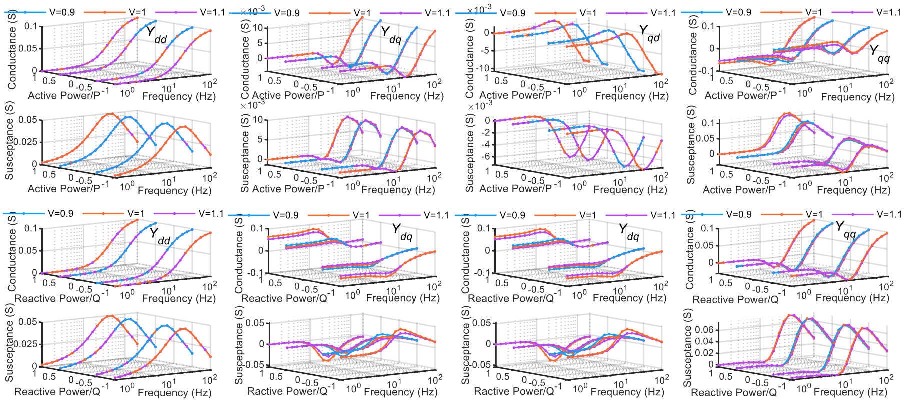


Extended Data Fig. 2 The control diagrams for MPC. (a) The principle of MPC. The selection process for optimal switching actions is highly nonlinear that can hardly be modeled through traditional small-signal approaches. (b) The control diagram of the multi-vector MPC. The current tracking is realized 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 [19]. 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-edge inverters controlled by such kind of controllers.

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)×4 (target Inverter count)×10 (repeating times)×10 (number of used data points)=1600, which took approximately 160 hours to complete the all the training.

5.6 Multi-vector model predictive control

Traditionally, grid-edge 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, the MPC is a promising alternative to control power electronic converters, which has been applied to almost all kinds of converters over the past few years [24]. 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-edge inverters, due to the inherent nonliearities 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 (left of Extended Data Fig. 2a). The optimization process is a highly nonlinear one that can hardly be modeled through traditional small-signal modeling approaches (right of Extended Data Fig. 2a). In this



Extended Data Fig. 3 The SimData 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-edge inverters is more susceptible to active and reactive power than to the output voltage.

work, we adopted the multi-vector MPC [19], which can not only achieve constant switching frequency but also retain the MPC's fast dynamic response characteristic.

The implementation process is illustrated in Extended Data Fig. 2. First, the physical model of the inverter system in the $\alpha\beta$-frame can be given by

$$\boldsymbol{u} = L\frac{d\boldsymbol{i}}{dt} + \boldsymbol{i}R \tag{12}$$

where $\boldsymbol{u} = [u_\alpha \, u_\beta]^T$ and $\boldsymbol{i} = [i_\alpha \, i_\beta]^T$, which are converter voltage and current vectors in the $\alpha\beta$-frame respectively. Then the discrete-time model can be obtained by applying Euler Forward Approximation as

$$\boldsymbol{i}\left(k+1\right) = \frac{T_s}{L}\boldsymbol{u}\left(k\right) + \left(1 - \frac{RT_s}{L}\right)\boldsymbol{i}\left(k\right). \tag{13}$$

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

$$\boldsymbol{u}^*\left(k\right) = \frac{L}{T_s}\boldsymbol{i}^*\left(k+1\right) + \left(R - \frac{L}{T_s}\right)\boldsymbol{i}\left(k\right). \tag{14}$$

where $\boldsymbol{u}^*\left(k\right)$ represents the reference voltage vector that forces the actual current to ideally track the reference current $\boldsymbol{i}^*\left(k+1\right)$.

Then, the calculated reference vector is rapidly located in the 120° 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. Next, the current tracking is prioritized through duty cycle optimization of the selected adjacent vectors. Finally, the optimal switching sequence is generated through an external modulator which follows the symmetric pulse pattern of seven segments (Extended Data Fig. 2b). As expected, nonlinear features of the multi-vector MPC in the current tracking process make the inverter impedance model hard to derive, therefore, the proposed model-agnostic data-driven methods can be used to establish impedance models.

6 Code availability

A TensorFlow implementation of the InvNet framework and the data supporting the findings of this work are pubicly available at https://github.com/superrabbit2023/InvNet.

References

[1] Sajadi, A., Kenyon, R.W., Hodge, B.M.: Synchronization in electric power networks with inherent heterogeneity up to 100% inverter-based renewable generation. Nature Communications **13**, 2490 (2022)

[2] Williamson, S.S.: The success of electric mobility will depend on power electronics. Nature Electronics **5**, 14–15 (2022)

[3] Chen, M., Poor, H.V.: High-frequency power electronics at the grid edge: a bottom-up approach toward the smart grid. IEEE Electrification Magazine **8**(3), 6–17 (2020)

[4] Wang, X., Blaabjerg, F.: Harmonic stability in power electronic-based power systems: concept, modeling, and analysis. IEEE Transactions on Smart Grid **10**(3), 2858–2870 (2019)

[5] Li, C., Burgos, R., Wen, B., Tang, Y., Boroyevich, D.: Analysis of statcom small-signal impedance in the synchronous d-q frame. IEEE Journal of Emerging and Selected Topics in Power Electronics **8**(2), 1894–1910 (2020)

[6] Wang, S., Liu, Z., Liu, J., Boroyevich, D., Burgos, R.: Small-signal modeling and stability prediction of parallel droop-controlled inverters based on terminal characteristics of individual inverters. IEEE Transactions on Power Electronics **35**(1), 1045–1063 (2020)

[7] Wen, B., Dong, D., Boroyevich, D., Burgos, R., Mattavelli, P., Shen, Z.: Impedance-based analysis of grid-synchronization stability for three-phase paralleled converters. IEEE Transactions on Power Electronics **31**(1), 26–38 (2016)

[8] Wang, X.: Unified impedance model of grid-connected voltage-source converters. IEEE Transactions on Power Electronics **33**(2), 1775–1787 (2018)

[9] Wen, B., Boroyevich, D., Burgos, R., Mattavelli, P., Shen, Z.: Analysis of d-q small-signal impedance of grid-tied inverters. IEEE Transactions on Power Electronics **31**(1), 675–687 (2016)

[10] Wright, L.G., Onodera, T., Stein, M.M., Wang, T., Schachter, D.T., Hu, Z., McMahon, P.L.: Deep physical neural networks trained with backpropagation. Nature Electronics **601**, 549–555 (2022)

[11] Li, Y., Wang, Y., Li, B.Q.: Generalized theory of phase-shifted carrier pwm for cascaded h-bridge converters and modular multilevel converters. IEEE Journal of Emerging and Selected Topics in Power Electronics **4**(2), 589–605 (2016)

[12] Rosso, R., Wang, X., Liserre, M., Lu, X., Engelken, S.: Grid-forming converters: control approaches, grid-synchronization, and future trends—-a review. IEEE Open Journal of Industry Applications **2**, 93–109 (2021)

[13] Wang, X., Taul, M.G., Wu, H., Liao, Y., Blaabjerg, F., Harnefors, L.: Grid-synchronization stability of converter-based resources—-an overview. IEEE Open Journal of Industry Applications **1**, 115–134 (2020)

[14] Akiba, T., Sano, S., et al.: Optuna: a next-generation hyperparameter optimization framework. Paper presented at the 2019 Knowledge Discovery and Data Mining (KDD 2019) (2019)

[15] Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. https://arxiv.org/abs/1412.6980 (2014)

[16] Weiss, K., Khoshgoftaar, T.M., Wang, D.: A survey of transfer learning. Journal of Big Data **3**(9) (2016)

[17] Choudhury, S., Moret, M., Salvy, P., Weilandt, D., Hatzimanikatis, V., Miskovic, L.: Reconstructing kinetic models for dynamical studies of metabolism using generative adversarial networks. Nature Machine Intelligence **4**, 710–719 (2022)

[18] Liao, Y., Wang, X.: Stationary-frame complex-valued frequency-domain modeling of three-phase power converters. IEEE Journal of Emerging and Selected Topics in Power Electronics **8**(2), 1922–1933 (2020)

[19] Li, Y., Diao, F., Zhao, Y.: A generic multivector model predictive control with symmetric pulse pattern for hybrid multilevel converters. IEEE Transactions on Industrial Electronics **68**(12), 12185–12195 (2021)

[20] Gong, H., Wang, X., Yang, D.: Dq-frame impedance measurement of three-phase converters using time-domain mimo parametric identification. IEEE Transactions on Power Electronics **36**(2), 2131–2142 (2021)

[21] Zhang, Y., Udrea, F., Wang, H.: Multidimensional device architectures for efficient power electronics. Nature Electronics **5**, 723–734 (2022)

[22] Huang, L., Wu, C., Zhou, D., Blaabjerg, F.: A double-plls-based impedance reshaping method for extending stability range of grid-following inverter under weak grid. IEEE Transactions on Power Electronics **37**(4), 4091–4104 (2022)

[23] Blalock, T., Ortiz, J., Frankle, J., Guttag, J.: What is the state of neural network pruning? Published in Proceedings of Machine Learning and Systems 2020 (MLSys 2020) (2020)

[24] Vazquez, S., Rodriguez, J., Rivera, M., Franquelo, L.G., Norambuena, M.: Model predictive control for power converters and drives: advances and trends. IEEE Transactions on Industrial Electronics **64**(2), 935–947 (2017)