IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Digital Object Identifier 10.48550/arXiv.2204.08557

PIDGeuN: Graph Neural Network-Enabled Transient Dynamics Prediction of Networked Microgrids Through Full-Field Measurement

Yin Yu[1], Xinyuan Jiang[2], (Member, IEEE), Daning Huang[1], (Member, IEEE), Yan Li[2], (Senior Member, IEEE), Meng Yue[3], (Member, IEEE), and Tianqiao Zhao[3], (Member, IEEE)

[1]Department of Aerospace Engineering, The Pennsylvania State University, University Park, PA 16802, USA (e-mail: daning@psu.edu)
[2]Department of Electrical Engineering, The Pennsylvania State University, University Park, PA 16802, USA (e-mail: yql5925@psu.edu).
[3]Interdisciplinary Science Department, Brookhaven National Laboratory, Upton, NY 11973, USA.

Corresponding author: Yan Li (e-mail: yql5925@psu.edu).

This work is supported by National Science Foundation under the award DMS-2229435.

ABSTRACT

A Physics-Informed Dynamic Graph Neural Network (PIDGeuN) is presented to accurately, efficiently and robustly predict the nonlinear transient dynamics of microgrids in the presence of disturbances. The graph-based architecture of PIDGeuN provides a natural representation of the microgrid topology. Using only the state information that is practically measurable, PIDGeuN employs a time delay embedding formulation to fully reproduce the system dynamics, avoiding the dependency of conventional methods on internal dynamic states, e.g., of controllers. Based on a judiciously designed message passing mechanism, the PIDGeuN incorporates two physics-informed techniques to improve its predictive performance, including a physics-data-fusion approach to determining the inter-dependencies between buses, and a loss term to enforce the known physical law of the power system, i.e., the Kirchhoff's law, to ensure the feasibility of the model prediction. Extensive tests show that PIDGeuN can provide accurate and robust prediction of transient dynamics for nonlinear microgrids over a long-term time period. Therefore, the PIDGeuN offers a potent tool for the modeling of large scale networked microgrids (NMs), with potential applications to predictive or preventive control in real time applications for the stable and resilient operations of NMs.

INDEX TERMS Graph Neural Network (GNN), networked microgrids (NMs), transient dynamics, prediction, distributed energy resources (DERs)

I. INTRODUCTION

MODERNIZATION of electric power grid is critical for improving the system's resiliency and reducing power outages , e.g., Manhattan blackout. To solve this problem, microgrids have been recognized as a promising archetype by integrating Distributed Energy Resources (DERs), such as wind and photovoltaic (PV). To further enhance the flexible and resilient operations of low- or medium-voltage distribution networks, networked microgrids (NMs) are currently under development. Microgrids and NMs are sensitive and vulnerable to disturbances such as PV fluctuations, leading to frequent transient dynamics. To improve the power capability of individual microgrids, it has been proposed to interconnect multiple microgrids to form Networked Microgrids. Interconnection enables microgrids to mutually support and provide ancillary services to the entire power grid. Moreover, interconnection propagates dynamics like photovoltaic output fluctuation-induced disturbances. In particular, when the system is disconnected from the main grid, considering networked microgrids' inertia is significantly reduced due to the high penetration of distributed energy resources with power-electronic interfaces, this interconnected system is sensitive to disturbances [1]. Therefore, studying and improving the transient dynamics of such an interconnected system is becoming a challenging scientific problem.

Power system stability has seen extensive efforts in the past decade [2]–[4]. This paper studies the transient dynamics of networked microgrids, which refers to the ability of power generation unit such as distributed energy resources or individual microgrid in the interconnected system to remain synchronism under credible disturbances. Although extensive research effort has been made on the microgrids' transient

behavior [5], it is still a challenge to study and stabilize the system's transients. First, detailed modeling is usually required for studying the transients; however, the wide integration of DERs results in a high-dimensional system, increasing the difficulty of efficiently analyzing the transient behavior. Second, microgrids are typical nonlinear systems, which is an inherent feature stemming from power loads and dynamics of DERs; and thus, the existing model may not be sufficiently accurate to fully represent the nonlinear dynamical system. Third, the operations of microgrids or NMs keep changing due to the fluctuations of DERs and/or the changes of system topology caused by the join or disconnection of microgrids or DERs. With the wide deployment of the advanced metering infrastructure (AMI), the nonlinear dynamical system is more observable than ever before. Hence, one inspiring solution of transient dynamics is: *to develop a data-driven approach to precisely and efficiently model and predict the system's transient dynamics, so that predictive or preventive control can be performed to stabilize microgrids and NMs.*

There are several existing data-driven approaches to identify the transient dynamics model of a nonlinear system through its operating data, which can fall into two major categories [6], namely linear models and data-driven nonlinear methods. First, the linear models are well-established and commonly-used system identification methods [7], which are relatively easy to implement and guaranteed to converge given sufficient system responses. However, these methods do not extrapolate due to their nature of local linearization, therefore are not suitable to identify microgrid systems that are typically nonlinear for the entire operating envelope. Second, several data-driven methods have also been developed to identify a nonlinear system to capture the global transient dynamics over the entire state space [8]. These methods can theoretically identify an accurate model if appropriate nonlinear terms are used. However, selecting the correct nonlinear terms is not trivial, and the required number of terms grows exponentially as the system size increases. Moreover, these methods usually involve system's state variables that are impractical to measure. Therefore, general nonlinear system identification methods may become intractable when applied to the identification of practical nonlinear systems such as microgrids.

Microgrids can be defined on buses and their pairwise connections, i.e., a graph, and the graph topology may significantly impact the microgrid transient dynamics. However, such topological information is not utilized in many aforementioned data-driven modeling approaches. The Graph Neural Network (GNN), a recent variant of deep learning models [9], [10], has emerged as a powerful tool for the modeling of data defined on graphs, and thus a promising candidate for the transient dynamics modeling of microgrids.

In general, the GNN-based methods have gained tractions for many complex dynamical physics simulations that can benefit from the graph representation of the underlying systems [11], [12], where the temporal dependency in the transient problems can be tackled with additional recurrent architecture [13]. The superior modeling capabilities of GNN has incurred interests in its application to power systems including microgrids, that have a natural graph structure. Most of the GNN-oriented studies on power systems focus on static problems, e.g., optimal power flow (OPF) problem [14], power flow approximations [15], [16], state estimation [17], [18], and anomaly detection [19], [20]. Fewer efforts based on the GNN methods have been devoted to the modeling of transient dynamics in power systems; relevant work include the short-term power prediction of DERs [21], [22], and transient stability assessment (TSA) [23]–[25]. However, these problems are considered on a relatively slow timescale, where the transient responses in microgrids are not well resolved. Therefore, in the literature, it still remains an open question whether a GNN-based data-driven model can be developed to capture and resolve the transient dynamics of a power system, esp. the NMs.

Most physical systems are governed by well-developed algebraic or differential equations that can be used to "inform" the data-driven model of the physics [26]. For power systems, Kirchhoff's law equation can be directly applied as the training objective in the power flow approximations to ensure that the network predictions are physically feasible [15]; the grid impedance is also commonly used as the edge weights of the underlying graph to indicate the strength of correlation between buses [14]. However, these approaches have been only applied to static problems, which are inherently governed by an algebraic equation. This paper focuses on transient problems of power systems, which are nonlinear dynamics governed by coupled differential-algebraic equations. Specifically, in dynamical modeling, using only the power flow equation as the objective function may overlook valuable time series data. In addition, the fixed and predefined edge weights are not necessarily good representations of time-dependent correlations between buses.

To bridge the gap identified above, we develop a novel Physics-Informed Dynamic Graph Neural Network (PIDGeuN), which is a data-driven approach for accurate, efficient, robust and time-resolved prediction of microgrid transient dynamics. The PIDGeuN incorporates the underlying physical laws of power system in two aspects, with the following motivations and novelties. *First*, long-term forecasting becomes a challenge in many time-series problems due to prediction error accumulation; in the context of microgrids, the prediction error is likely to manifest in the form of nonphysical loss or gain of conserved energy. To account for the conservation of energy, a partial set of the power system equations is implemented into the model training, so as to produce a PIDGeuN model that respects the physical laws during prediction. *Second*, the strengths of correlation between buses, characterized by the pairwise edge weights in the graph, are formulated to dynamically adjust based on the bus states and admittance matrix, so as to capture the time-varying inter-dependencies between buses and enhance the predictive accuracy of transient dynamics. The incorporation of the two physics-informed techniques improves the

network's training efficiency and prediction accuracy, and enables the unprecedented capability of accurate long-term prediction for microgrid transient dynamics. Such capability paves the way for performing real-time control in microgrids, which is the authors' next work.

The remainder of this paper is organized as follows: Section II poses the mathematical problem for the graph-based dynamic modeling of networked microgrids. Section III introduces the presented PIDGeuN method for transient dynamics prediction. In Section IV, the PIDGeuN model is benchmarked against existing methods using numerical examples to demonstrate the feasibility and effectiveness of PIDGeuN. Conclusions are drawn in Section V.

II. PROBLEM STATEMENT

A. NETWORKED MICROGRID SYSTEMS

Networked microgrids consist of several distributed and independent microgrids to provide local energy generation and delivery. Each microgrid is a group of DERs and loads within clearly defined electrical boundaries, which acts as a single controllable entity and can connect to or disconnect from NMs.

Assume in a NM system, there are $\mathbb{G}$ DERs, $\mathbb{L}$ power loads, and N buses $\{B_i\}_{i=1}^N$. The connection of buses is depicted by the admittance matrix $Y \in \mathbb{C}^{N\times N}$. Each B_i are described by the following quantities: active power $P_i \in \mathbb{R}$, reactive power $Q_i \in \mathbb{R}$, voltage $V_i\angle\delta_i \in \mathbb{C}$, and current $I_i\angle\theta_i \in \mathbb{C}$, which are measurable. These variables define a vector $\mathbf{x}_{gi} \in \mathbb{R}^6$,

$$\mathbf{x}_{gi} = [P_i, Q_i, \mathrm{Re}(V_i\angle\delta_i), \mathrm{Im}(V_i\angle\delta_i), \mathrm{Re}(I_i\angle\theta_i), \mathrm{Im}(I_i\angle\theta_i)] \tag{1}$$

Note that the dynamics of microgrids are determined by several factors such as controller of DERs, power loads, network topology, etc.. In this work, only the measurable variables such as $\{\mathbf{x}_{gi}\}_{i=1}^N$ are utilized to identify the system dynamics. It removes the dependence of conventional data-driven methods on the internal states of DERs that are hard to measure, making it feasible for real world applications.

At the steady state, DERs produce power to satisfy the consumption of power loads. When disturbance occurs, the outputs of dispatchable DERs are adjusted accordingly as well as power loads to compensate for the disturbance until a new equilibrium is reached. Our goal is to accurately predict the transients of the system in between two equilibrium points.

Assuming buses are measured locally, we collect the measurements within a period of time T to identify the dynamical system. To model the transient dynamics of microgrids subject to disturbances, the bus B_i at the time instance k is then characterized by an extended state vector,

$$\mathbf{x}_i^{(k)} = [\mathbf{x}_{gi}, dP_i, dQ_i, \gamma, \beta_i]^{(k)} \in \mathbb{R}^{10}, \tag{2}$$

where the first six variables correspond to the standard states $\mathbf{x}_{gi}$. The new variables are introduced below.

The power disturbances are parametrized by dP and dQ,

$$dP_i^{(k)} = P_i^{(k+1)} - P_i^{(k)}, \tag{3a}$$
$$dQ_i^{(k)} = Q_i^{(k+1)} - Q_i^{(k)}, \tag{3b}$$

and the values of dP and dQ are non-zero *only* in the buses where the disturbance occurs such as load or DER buses. A Boolean variable γ is introduced to indicate if disturbances occur in *any* of the buses, i.e., $\gamma^{(k)} = 1$ means a disturbance occurred in the system at the k^{th} time step (though not necessarily at bus B_i), and $\gamma^{(k)} = 0$ otherwise. β_i is the type index, meaning the type of the bus B_i, where,

$$\beta_i = \begin{cases} 0 & \text{Empty} \\ 0.5 & \text{Loads} \\ 1.0 & \text{DERs (w/ or w/o Loads)} \end{cases} \tag{4}$$

At the time step $t = t_k$, denote the collection of the extended states as $\mathbf{X}^{(k)} = \{\mathbf{x}_i^{(k)}\}_{i=1}^N \in \mathbb{R}^{N\times 10}$ and the standard states as $\mathbf{X}_g^{(k)} = \{\mathbf{x}_{gi}^{(k)}\}_{i=1}^N \in \mathbb{R}^{N\times 6}$. The transient dynamics modeling and prediction of microgrids is stated as follows: Given a sequence of states of C steps, $\mathcal{X}_C^{(k)} = [\mathbf{X}^{(k)}, \mathbf{X}^{(k-1)}, \cdots, \mathbf{X}^{(k-C+1)}] \in \mathbb{R}^{N\times 10\times C}$, predict the system states $\mathbf{X}_g^{(k+1)}$ at the future time $t = t_{k+1}$. Motivated by the time-delayed embedding technique [27], the use of consecutive time steps compensates for the partial knowledge of microgrids obtained through AMI and facilitate the complete reconstruction of the microgrid dynamics.

B. GRAPH REPRESENTATION

Power systems including microgrids can be represented by a graph, where the nodes are the buses of microgrids and the edges are the connections between buses. The weights of edges, i.e. the edge attributes, describe how correlated the states $\mathbf{x}_{gi}$ of two buses are. Formally, let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ be a graph with a set of N nodes, $\mathcal{V}$, a set of edges $\mathcal{E} \subseteq \mathcal{V}\times\mathcal{V}$ and the edge weights $\mathcal{W}$. The graph for microgrids is undirected, meaning that if $(i,j) \in \mathcal{E}$ then $(j,i) \in \mathcal{E}$. Conventionally the edge weights are computed as given in (5) [14],

$$w_{ij} = \begin{cases} \exp\left(-k|Y_{ij}|^2\right) & (i,j) \in \mathcal{E} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where k is tunable parameter chosen so that $\mathcal{W}$ is in the range of $[0.2, 1]$. Since the admittance matrix Y is symmetric, $w_{ij} = w_{ji}$. The adjacency matrix of a graph that describes the connections between the nodes is defined by $\mathbf{A} \in \mathbb{R}^{N\times N}$ where $[\mathbf{A}]_{ij} = w_{ij}$ if $(i,j) \in \mathcal{E}$ and 0 otherwise. Therefore, the diagonal terms in the adjacency matrix $\mathbf{A}$ are always zero, whereas those in the admittance matrix $\mathbf{Y}$ are not. The graph is more conveniently represented by a normalized Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, where $\mathbf{D}$ is a diagonal degree matrix with $[\mathbf{D}]_{ii} = \sum_j [\mathbf{A}]_{ij}$.

Data defined on a graph reside on a non-Euclidean space and often comes with a variable size of unordered nodes without a fixed spatial locality. The learning task on graph therefore poses challenges to the conventional machine learning algorithms, e.g., convolutional neural networks (CNNs).

The convolution operation used by CNN is defined on structured grid, and does not apply in the unstructured graph data. Additionally, the edges in graphs contain connection information among nodes. Such information can only be treated as node features in standard neural networks, which is inefficient and often inaccurate [28]. Graph neural networks generalize the convolution operation from multi-dimensional data arrays to data on irregular topology, i.e., on graphs, and serve as an effective filter that extracts localized features from graph data. Utilizing GNNs, microgrid transient dynamics is characterized by the following differential equation,

$$\dot{\mathbf{X}}_g^{(k)} = \mathbf{F}(\mathcal{X}_C^{(k)}, \mathcal{G}; \mathbf{\Theta}), \tag{6}$$

where a GNN $\mathbf{F}$, parametrized by $\mathbf{\Theta}$, maps the sequence of C consecutive extended state vectors $\mathcal{X}_C^{(k)}$ to the rate of change $\dot{\mathbf{X}}_g^{(k)}$ for the states on all buses at the current time step k, given the graph structure $\mathcal{G}$ of the system. Then the states of the buses at the next time step can be numerically calculated. (7) gives an example when the explicit integration is adopted.

$$\mathbf{X}_g^{(k+1)} = \mathbf{X}_g^{(k)} + \Delta t \dot{\mathbf{X}}_g^{(k)}, \tag{7}$$

where Δt is the time step size.

III. FORMULATION OF PIDGEUN

In this section, we present the key components and salient features of the proposed PIDGeuN architecture. The architecture builds upon the message passing (MP) mechanism of GNNs and judiciously chooses a hybrid form of two MP implementations as its building blocks to capture the dynamical system's transient dynamics. Furthermore, the known physical knowledge of microgrids is infused into both network architecture and loss function, in order to improve the expressiveness and training efficiency of the network.

A. MESSAGE PASSING MECHANISM

The message passing mechanism is the corner stone for many GNN architectures, which consists of multiple consecutive MP steps. Consider an input graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ of N nodes, and each node $v \in \mathcal{V}$ has a node feature vector $\mathbf{h}_v \in \mathbb{R}^D$ and a set of neighbor nodes $u \in \mathcal{N}(v)$. At the j^{th} MP step, the new feature of node v is computed using its previous feature and information from its neighbors as [29],

$$\mathbf{m}_{\mathcal{N}(v)}^j = \text{AGGREGATE}\left(\{\mathbf{h}_u^j \mid u \in \mathcal{N}(v)\}, \mathcal{W}\right), \tag{8a}$$

$$\mathbf{h}_v^{j+1} = \text{UPDATE}\left(\mathbf{h}_v^j, \mathbf{m}_{\mathcal{N}(v)}^j, \mathcal{W}\right), \tag{8b}$$

where AGGREGATE and UPDATE are nonlinear mappings, e.g., neural networks, and $\mathbf{m}_{\mathcal{N}(v)}$ denotes the information aggregated from the neighbors of node v. One MP step corresponds to the information exchange between 1-hop neighbors, i.e., the nodes that directly connected. It is possible to stack different forms of aggregators over k MP steps, and the feature vector of a node is influenced not only by its 1-hop neighbors, but also by the more distant k-hop neighbors.

In PIDGeuN for microgrids, using the MP mechanism, the change of states of each node is determined by the input states of itself and its neighbors through a sequence of neural network modules; the long-range interaction between the buses during a disturbance is captured via a stack of MP layers.

B. TWO TYPICAL MP IMPLEMENTATIONS

1) Graph Convolutional Layers (GCLs)

The GCLs generalize the convolution from multi-dimensional data arrays to data on graphs, and serve as an effective filter that extracts localized features from graph data. Under the framework of graph Fourier Transform (GFT), the GCLs performs the filtering on the spectrum of the Laplacian matrix [30].

An efficient implementation of the GCL is the ChebConv network [31], which performs the MP aggregation and updating over all nodes simultaneously. Let the input be $\mathbf{H}^j = \{\mathbf{h}_i^j\}_{i=1}^N \in \mathbb{R}^{N\times D}$, the ChebConv-based graph convolution is defined as

$$\mathbf{H}^{j+1} = \sigma\left(\sum_{k=0}^{K} T_k(\tilde{\mathbf{L}})\mathbf{H}^j\mathbf{\Theta}_k^j\right), \tag{9}$$

where σ is a nonlinear activation function, $\{\mathbf{\Theta}_k^j\}_{k=0}^K$ are learnable parameters, and $T_k(\tilde{\mathbf{L}})$ is a series of Chebyshev polynomicals of a scaled Laplacian $\tilde{\mathbf{L}} = (2/\lambda_{\max})\mathbf{L} - \mathbf{I}$ with $\lambda_{\max}$ the largest eigenvalue of $\mathbf{L}$. The ChebConv network avoids the direct computation of GFT and approximates the spectral filtering through the truncated series of Chebyshev polynomials T_k up to K^{th} order, which are equivalent to performing K MP steps.

2) Graph Attention Layers (GALs)

The GAL can be viewed as a nonlinear form of graph convolution where the adjacency matrix is dynamically adjusted according to the node features using the attention mechanism [32]. One GAL effectively performs just one MP step, but introduces stronger nonlinearity when compared to GCL.

In the general attention mechanism, for node u and its neighbor v, with feature vectors $\mathbf{h}_u^j$ and $\mathbf{h}_v^j$, one may compute m attention coefficients $\{\alpha_{uv}^j\}_{k=1}^m$ as

$$\alpha_{uv}^j = \frac{\exp(\text{LeakyReLU}(\mathbf{f}(\mathbf{h}_u^j, \mathbf{h}_v^j; \mathbf{\Theta}_f^j)))}{\sum_{w\in\mathcal{N}(u)}\exp(\text{LeakyReLU}(\mathbf{f}(\mathbf{h}_u^j, \mathbf{h}_w^j; \mathbf{\Theta}_f^j)))}. \tag{10}$$

In the attention calculation (10), $\mathbf{f}_k$ is nonlinear function, such as a neural network, which characterizes the correlation between two feature vectors; LeakyReLU is a nonlinear activation function $f(x) = \max(-\epsilon x, x)$, where $\epsilon = 0.2$ as a typical choice; the sum-of-exp formulation normalizes the correlation to produce $\alpha_{uv}^j \in [0, 1]$. Subsequently, defining a set of new adjacency matrices, $[\mathbf{A}_\alpha^j]_{uv} = \alpha_{uv}^j$, the node features are updated as

$$\mathbf{H}^{j+1} = \sigma(\mathbf{A}_\alpha^j\mathbf{H}^j\mathbf{\Theta}_\alpha^j) \in \mathbb{R}^{N\times D}. \tag{11}$$

Note that in general $\mathbf{A}_\alpha^j$ is not symmetric. It is possible that a node u is strongly influenced by its neighbor v, quantified by a large α_{uv}, but not vice versa.

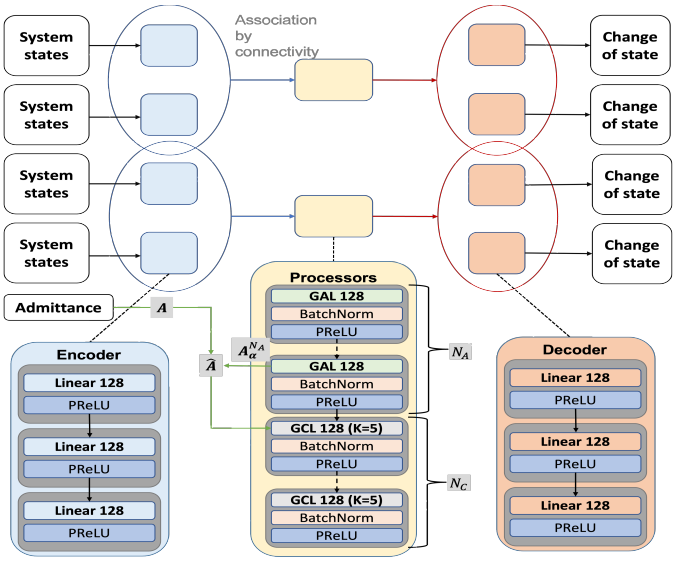


Figure 1: The PIDGeuN architecture.

C. PIDGEUN METHODOLOGY

The PIDGeuN architecture, shown in Fig. 1, uses an encoder-processor-decoder architecture and adopts a combination of graph convolutional and attention layers with physics-informed techniques. The key components of the PIDGeuN architecture are detailed in the following.

1) Encoder

First, the encoder is applied to each individual node. It maps microgrid state vectors at a node $\mathbf{x}_i$, which consists of both continuous and discrete variables, to a latent vector $\mathbf{h}_i^0 \in \mathbb{R}^D$. The latent vector is a set of high-dimensional nonlinear features that provide a continuous representation of the states on each bus, which is amenable for NN computations. For the i^{th} node at time step k, the encoder $\mathbf{f}_E$ is

$$\mathbf{h}_i^0 = \mathbf{f}_E(\mathbf{x}_i^{(k)}, \mathbf{x}_i^{(k-1)}, \cdots, \mathbf{x}_i^{(k-C+1)}; \boldsymbol{\Theta}^0), \tag{12}$$

where $\mathbf{f}_E$ is implemented as a standard fully-connected NN (FCNN) of N_M layers with a set of trainable parameters $\boldsymbol{\Theta}^0$. After the encoding, the latent vectors of all the nodes are denoted $\mathbf{H}^0 = \{\mathbf{h}_i^0\}_{i=1}^N \in \mathbb{R}^{N\times D}$.

2) Processor

Subsequently, a stack of $N = N_A + N_C$ graph MP layers serve as processors that successively aggregate the latent features from each node and its neighbors and update the latent vectors at each node. Formally, the j^{th} processor step is written as

$$\mathbf{H}^{j+1} = \mathbf{f}_P^j(\mathbf{H}^j; \boldsymbol{\Theta}^j), \tag{13}$$

where $\mathbf{f}_P^j$ is either a GCL or a GAL, with parameter $\boldsymbol{\Theta}^j$.

Specifically, starting from the encoded latent vector $\mathbf{H}^0$, the PIDGeuN first uses N_A GAL layers in the processor to successively generate a series of latent vectors $\mathbf{H}^1, \cdots, \mathbf{H}^{N_A}$, as well as the attention-based adjacency matrix $\mathbf{A}_\alpha^{N_A}$, using (10) and (11).

Next, to incorporate the physical knowledge of the microgrid into the network, a new *physics-infused adjacency matrix* $\hat{\mathbf{A}}$ is formed by combining the attention-based matrix $\mathbf{A}_\alpha^{N_A}$ and the admittance-based matrix $\mathbf{A}$ in (5),

$$\hat{\mathbf{A}} = \frac{1}{2}\left(\mathbf{A}_\alpha^{N_A} + \mathbf{A}\right) \tag{14}$$

Note that $\hat{\mathbf{A}}$ maintains the same graph topology as the admittance-based $\mathbf{A}$ in (5), but with different non-symmetric weights.

The processing step is finalized with N_C K^{th}-order GCL layers that use the normalized Laplacian $\hat{\mathbf{L}}$ computed from $\hat{\mathbf{A}}$, and generate a series of the latent vectors $\mathbf{H}^{N_A+1}, \cdots, \mathbf{H}^N$ using (9). The last output $\mathbf{H}^N$ is sent to the subsequent decoding step.

Over the entire processing step, the total number of effective MP step performed is $N_{MP} = N_A + KN_C$. In the special case that $N_A = 0$, the GCLs directly employ the symmetric admittance-based $\mathbf{A}$ as the adjacency matrix; while when $N_C = 0$, the GALs outputs the last latent vector $\mathbf{H}^{N_A}$ for the next step and the attention coefficients are not used.

3) Decoder

Finally, the decoder maps the latent vector of each node to the desired output, i.e. the rate of change,

$$\tilde{\dot{\mathbf{X}}}_g^{(k)} = \mathbf{f}_D(\mathbf{H}^N; \boldsymbol{\Theta}^{N+1}), \tag{15}$$

where $\mathbf{f}_D$ is a FCNN of N_M layers with trainable parameters $\boldsymbol{\Theta}^{N+1}$.

4) Loss Function

The network parameters $\boldsymbol{\Theta}$ need to be trained using a loss function, a typically choice of which is the Frobenius-norm between the predicted and true rate of change over the training sequence of N_t steps,

$$L_1(\boldsymbol{\Theta}) = \sum_{k=1}^{N_t} \left\| \tilde{\dot{\mathbf{X}}}_g^{(k)}(\boldsymbol{\Theta}) - \dot{\mathbf{X}}_g^{(k)} \right\|_F^2 \tag{16}$$

However, leveraging known physical principles, at each time step the predicted states of the grid, computed using (7), should satisfy the *physical constraints*, i.e. the Kirchhoff's law, at each node, or power flow computation for the system. This fact motivates the inclusion of an additional term in the loss function to penalize the violation of the Kirchhoff's law in the prediction at each node and at each time step,

$$L_2(\boldsymbol{\Theta}) = \sum_{i=1}^{N_k}\sum_{k=1}^{N_t} \left| \tilde{P}_i^{(k)} + i\tilde{Q}_i^{(k)} - (\tilde{V}_i^{(k)}\angle\tilde{\delta}_i^{(k)})\overline{(\tilde{I}_i^{(k)}\angle\tilde{\theta}_i^{(k)})} \right|^2 \tag{17}$$

(17) is expanded to a formulation that contains only real values in the implementation so that the network training does not involve any complex arithmetics. Combining (16) and (17), the loss function used to train the PIDGeuN network is

$$L(\boldsymbol{\Theta}) = L_1(\boldsymbol{\Theta}) + \nu L_2(\boldsymbol{\Theta}) \tag{18}$$

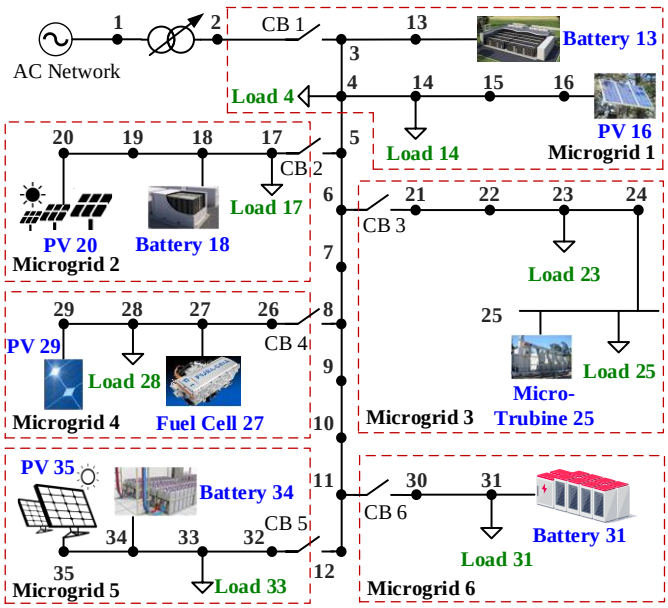


Figure 2: Networked microgrid test system.

where ν is a factor to control the penalty on physical violation; in this study $\nu = 1$ is used, which assigns both loss terms equal weights.

5) Activation Functions

In this study, the nonlinear activation function is designed to have two components to ensure model convergence during the training. First, a batch-normalization (BN) layer [33] is used to reduce the potential differences in the latent vectors caused by the differences due to disturbances. Second, the Parametric Rectified Linear Unit (PReLU) function is applied to compute the new latent vectors. From numerical experiments, the PReLU performed more robustly than the commonly used ReLU function, which suffered from the dying neuron problem and caused premature convergence in the training.

IV. NUMERICAL EXAMPLES

In this section, the PIDGeuN architecture is applied to model and predict the transient dynamics of a typical 33-bus networked microgrid system [1], as shown in Fig. 2, to demonstrate its accuracy, robustness and versatility in modeling and predicting dynamics on graph. Circuit Breaker 1 is open and others are closed, so the NM system is in the islanded operation. In this operation mode, the DER unit connected to Bus 13 is under voltage-frequency (Vf) control, and all other generators are under constant power output (PQ) control. The power loads in the system are constant power loads with the instantaneous load demands being the input disturbances to the system.

A. DESCRIPTION OF THE NUMERICAL EXAMPLE

1) Datasets

The dataset for training and testing the PIDGeuN models contains the transient responses of the system starting from different initial conditions with a step load change applied to randomly chosen nodes at the start of simulation. The magnitude of the load change is in the range of $\pm 10\%$ of the nominal value of each load, which is beyond the regime of linear analysis. The electromechanical DAE model of the Vf and PQ control of each generator connected with their individual phase-locked loop and the electrical network is first built and simulated to provide synthetic data for PIDGeuN. Two types of responses are generated: (1) *complete transient response* that starts from an equilibrium point and ends when the system reaches a new equilibrium point; (2) *initial transient response* during which random load changes are added every 0.01s so that the system is always away from equilibrium and shows transient dynamics. The training dataset consists of 90 complete transient responses and initial transient responses with 5000 load changes. The test dataset contains 30 complete transient responses and initial transient responses with 500 load changes.

2) Evaluation Metrics

The performance of the data-driven models are quantified using three types of metrics:

a) One-step root mean squared error (RMSE): The error in the rates of change of all buses is defined as

$$E_1 = \sqrt{\frac{1}{TN}\sum_{k=1}^{T}\left\|\tilde{\dot{\mathbf{X}}}^{(k)} - \dot{\mathbf{X}}^{(k)}\right\|_F^2}, \tag{19}$$

where T is the total number of time steps in the time series for prediction, and $\tilde{\dot{\mathbf{X}}}^{(k)} = \mathbf{F}(\mathcal{X}^{(k)}, \mathcal{G})$ is the predicted rate of change using the current *true* bus states. The one-step RMSE is equivalent to the loss term in (16) except that it is applied to the test dataset.

b) Cumulative RMSE (C-RMSE): The difference between the predicted and true dynamics of bus states is defined as

$$E_2 = \sqrt{\frac{1}{TN}\sum_{k=1}^{T}\left\|\tilde{\mathbf{X}}^{(k)} - \mathbf{X}^{(k)}\right\|_F^2}, \tag{20}$$

where $\tilde{\mathbf{X}}^{(k)}$ is the predicted bus states at time step k that are evaluated iteratively using (7) given only the initial condition $\mathcal{X}^{(0)}$. The C-RMSE accounts for the accumulation of prediction error in the time-series prediction and thus is the major metric for assessing the model performance.

c) Number of parameters, which measures the complexity of each model. More trainable parameters give the neural network more expressive power but may result in the overfitting issue and increased computational cost for training and prediction.

3) Implementation Details of Nominal PIDGeuN Model

The PIDGeuN architecture is implemented using PyTorch Geometric (PyG) [34], an open-source PyTorch-based machine learning framework for Graph Networks. The hyperparameters used for the nominal PIDGeuN model are: $N_M = 3$,

$N_A = N_C = 5$, $K = 5$, $D = 128$, and $C = 3$. During the training, the states as well as the rate of change are normalized to a range of $[0, 1]$. The loss is minimized using the standard Adam optimizer with an exponential decay of learning rate from 10^{-3} to 10^{-7}.

B. COMPARISON WITH BASELINE METHODS

1) Baseline Methods

The PIDGeuN model is benchmarked with a number of baseline methods that are specialized in time-series prediction, ranging from conventional data-driven models that do not account for graph topology to various forms of STGNNs. These methods are widely used and benchmarked, and are listed as follows:

a) Subspace identification [35]: A linear state-space system identification method using only the measured states.
b) Long Short-Term Memory (LSTM) [36]: A type of Recurrent Neural Network (RNN) that have been widely applied for time-series prediction. We used 5 stacked layers each with a hidden size of 128.
c) Graph Convolutional Recurrent Network (GCRN) [13]: A type of STGNN that uses a Chebyshev GCN (i.e., GCL) in space and a GRU in time for data correlation over a larger spatiotemporal scale. We used 5 stacked layers each with a hidden size of 128 and $K = 5$ for the GCL.
d) Spatial-Temporal GCN (STGCN) [37]: A type of STGNN that uses a GCL in space and 1D convolution instead of a RNN in time, which eliminates the usage of recurrent architecture and allows for faster training with fewer parameters. We used 2 temporal convolution layers, and 5 GCL layers with $K = 5$, all with a hidden size of 128.

The LSTM model is implemented with the PyTorch package, and the two recurrent GNN models are implemented using the PyTorch Geometric Temporal package [38]. The LSTM does not utilize the graph structure, therefore at the time step $t = t_k$, the extended node states $\mathbf{X}^{(k)} \in \mathbb{R}^{N\times 10}$ are stacked into a $\mathbb{R}^{10N}$ vector as input to the network.

2) Results and Discussion

In this experiment, we compare the performance of PIDGeuN against other baseline methods in the predictions of transient response of the test system.

The evaluation metrics are detailed in Table 1, including the one-step RMSE's for the training and test dataset, and the C-RMSE's for 200 and 700 time steps. The two C-RMSE's are chosen to quantify the short-term and long-term predictive capabilities of the models. Note that, when compared to the complete response cases, the dynamics of the initial response is more complex due to the frequently introduced disturbances, and thus the C-RMSE of initial responses is expected to be higher than that of complete responses.

Overall, the PIDGeuN outperforms the baseline methods by a significant margin. First, all the models achieve low training and test one-step RMSE's, showing that they are sufficiently complex and expressive to predict the rate of change if given the true states, and generalize to unseen inputs. Yet the PIDGeuN achieved the lowest training and test errors, highlighting its superior expressiveness and generalizability over other models. Second, the high C-RMSE's show that most of the baseline models fail to produce accurate predictions over a long time horizon; particularly the STGCN quickly diverges beyond 200 time steps. The best baseline model is in fact the subspace model, a linear method. On the contrary, the PIDGeuN consistently achieved the lowest C-RMSE's in all cases and the slowest growth in the error, which demonstrates its robustness in time series prediction. Finally, note that the superior performance of the PIDGeuN is achieved only using an amount of parameters that is comparable to the smallest and worst learning-based baseline model, i.e., the STGCN.

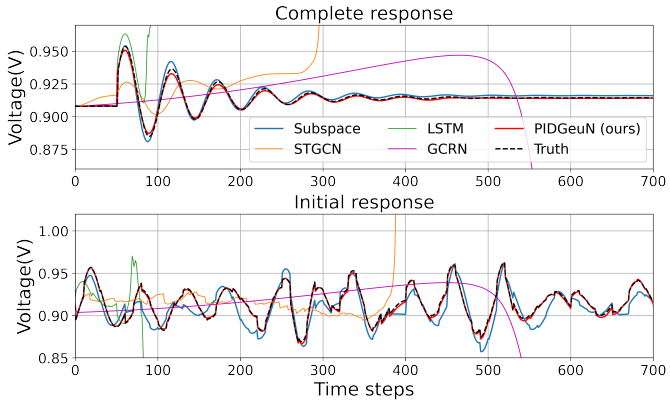


Figure 3: Complete and initial voltage dynamic response of a DER bus.

Subsequently, a typical voltage response of a DER bus is closely examined in Fig. 3. In the complete response case, a large load change is introduced in the grid at time $t_k = 50$ causing an initial step change in the DER voltage, followed by a damped oscillatory response until a new equilibrium in the system is reached. The complete response resembles that of a linear system to a step input. In the initial response case, smaller load changes are introduced every 10 time steps, and the system never reaches equilibrium in between the load changes. As a result, the system dynamics in the initial response case is more dynamic and nonlinear, and thus more challenging to predict. It also mimics the applications of PIDGeuN in the real world when the system is under frequent disturbances.

As visualized in Fig. 3, in the complete response case, the PIDGeuN reproduces the DER's response to the load change by accurately predicting first the initial step voltage of the DER and then the decay of signal oscillations that matches the true dynamics in both magnitude and frequency. The subspace method performs well overall except for not capturing the peak voltages in each oscillation and the final equilibrium voltage. The LSTM and STGCN are able to follow the first few periods of oscillation but both diverged because of the modeling error accumulation, resulting in large error. The GCRN model does not capture any oscillatory response and also diverged. In the initial response case, the PIDGeuN

Table 1: Comparison of PIDGeuN and baseline models

Model	Training one-step RMSE($\times 10^{-3}$)	Test one-step RMSE($\times 10^{-3}$)	C-RMSE 200 Complete/Initial	C-RMSE 700 Complete/Initial	# of parameters
Subspace	–	–	2.018/9.978	4.815/36.791	**7595**
LSTM	0.167	1.282	12.192/304.777	45.183/1348.707	1830088
GCRN	0.988	2.203	6.045/26.191	20.894/99.472	5000726
STGCN	0.338	3.426	19.953/35.719	$4.886E51/1.042E52$	805518
PIDGeuN	**0.093**	**0.167**	**0.986/1.276**	**2.311/11.911**	986006

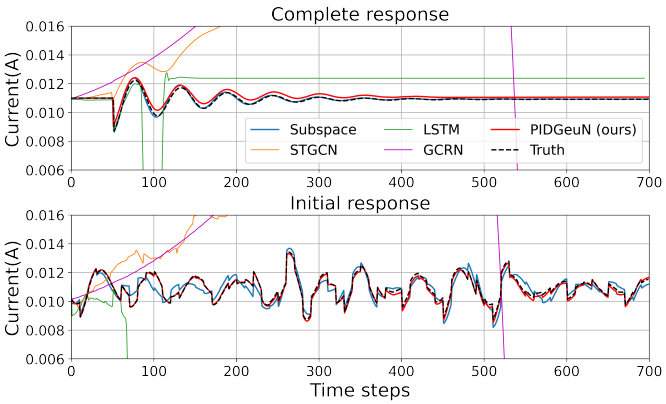


Figure 4: Complete and initial current dynamic response of a DER bus.

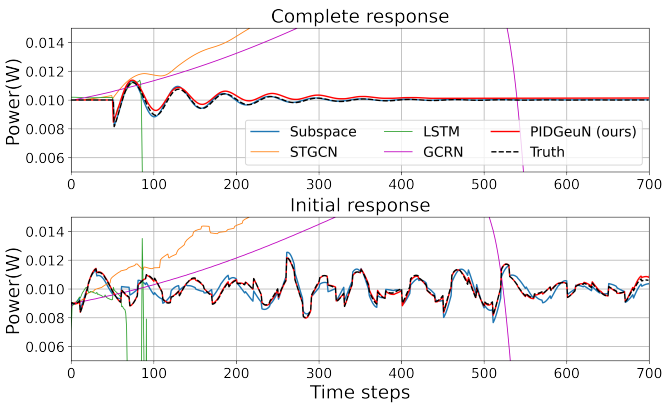


Figure 5: Complete and initial power dynamic response of a DER bus.

captures the transient dynamics almost perfectly despite the frequent introduction of load changes. The subspace method performs worse than the previous case and misses most of the peak voltages. The other three models show similar trend as in the previous test case, and do not make any useful prediction. The current and power responses of the same DER bus are respectively shown in Figs. 4 and 5, where PIDGeuN considerably outperforms the baseline methods in predictions again.

Lastly, a brief comparison in time complexity is provided. Leveraging the sparse graph structure, the cost in prediction for STGCN, GCRN and PIDGeuN is only $O(N)$ when compared to (N^2) for the subspace method and LSTM. Among the learning-based methods, the cost in training for PIDGeuN is $O(K)$, when compared to $O(K^2)$ for the LSTM, GCRN, and STGNN that all employed a recurrent network architecture. Therefore, PIDGeuN has the lowest time complexities in prediction and training among the models considered.

From the comparison of the evaluation metrics and typical response cases, it is clear that the PIDGeuN model significantly outperforms the baseline models, including a classical system identification method (subspace), a learning-based method without graph (LSTM), and two learning-based methods with graph information (STGCN and GCRN), in terms of the generalizability, predictive accuracy, robustness in long-term prediction, and time complexity.

C. ABLATION STUDY

To explain the effectiveness of the PIDGeuN and study how its components affect the performance, we conducted an ablation study where a number of hyperparameters are varied one by one while holding others at the nominal value. Specifically, the composition of the processors, the sizes of the MP layers, and the inclusion of physics-informed loss are examined. The complete list of tested models is provided in Table 2, where in each group the varying parameters are highlighted and the nominal model is labelled as C1. In addition to the hyperparameters listed in Table 2, we also studied the effects of varing the batch size of training data, number of MLP layers in the encoder and decoder, training learning rate, and number of attention heads in GALs. These hyperparameters did not have a significant impact on the model performance, and therefore not listed for concision. For the same reason, only the test case using complete response is presented.

Table 2: Ablation study

Model	N_A	N_C	PI-loss	K	D	C	C-RMSE 200	C-RMSE 700
C1	**5**	**5**	**True**	**5**	**128**	**3**	0.986	**2.311**
C2	3	5	True	5	128	3	1.087	2.634
C3	1	5	True	5	128	3	**0.941**	2.825
C4	0	5	True	5	128	3	2.494	$1.850E3$
C5	0	10	True	5	128	3	2.310	6.231
C6	5	3	True	5	128	3	1.115	3.108
C7	5	1	True	5	128	3	3.359	$6.941E61$
C8	5	0	True	–	128	3	3.736	$1.552E51$
C9	10	0	True	–	128	3	2.493	$3.591E42$
C10	3	3	True	5	128	3	0.985	2.648
C11	1	1	True	5	128	3	28.823	NaN
C12	5	5	False	5	128	3	1.563	3.732
C13	5	5	True	7	128	3	1.102	3.101
C14	5	5	True	3	128	3	1.484	2.869
C15	5	5	True	1	128	3	7.385	$3.717E6$
C16	5	5	True	5	64	3	1.054	2.787
C17	5	5	True	5	32	3	1.470	$2.614E3$
C18	5	5	True	5	256	3	1.010	2.489
C19	5	5	True	5	128	1	7.080	$4.472E31$
C20	5	5	True	5	128	2	1.045	2.581
C21	5	5	True	5	128	5	1.338	2.782

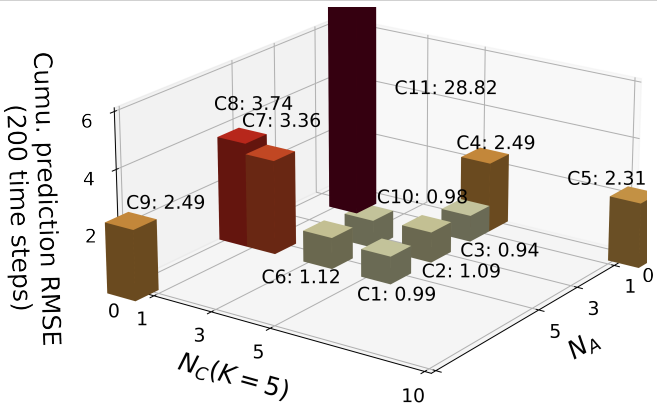


Figure 6: Impact of processor layers on the model performance.

1) Processor layers

To examine how the composition of the processor affects the model performance, the models C2-C11, having different combinations of GAL and GCL layers, are trained and tested. The differences among the models are two-fold. First, when $N_A = 0$ (GCL-only) or $N_C = 0$ (GAL-only), the PIDGeuN model no longer uses a hybrid architecture; the GCL-only models (C4, C5) and GAL-only models (C8, C9) use only the physics-based and data-driven adjacency matrices, respectively. Second, while the rest models all use a hybrid architecture, they differ significantly in the number of MP steps, which determines the capability of the network to propagate information between distant nodes, as noted in Sec. III-A.

The comparison of C-RMSE's for these cases are listed in Table 2 and the C-RMSE's for 200 steps are visualized in Fig. 6. First, the nominal PIDGeuN model (C1) and the other hybrid models outperforms the models with only GCLs (C4, C5) or only GALs (C8, C9) by a significant margin. The difference indicates the performance gain in microgrid dynamics prediction is facilitated by the proposed physics-data-infusion strategy that combines the physics-based admittance information and the data-driven attention coefficients in the adjacency matrix.

Next, focusing on the models with hybrid architecture, a strong correlation between prediction accuracy and number of MP steps is identified. The hybrid models with relatively fewer MP steps (C4, C7-C9, C11) tend to accumulate large errors during time series prediction and diverge in some test cases. Particularly, for models C7 and C11, the lack of MP steps limits the long-range information propagation between the nodes and the prediction performance is even worse than the GCL-only and GAL-only models. The rest hybrid models (C1-C3, C6, C10) achieved similar performance in the 200-step prediction, with C3 being the best. But in the 700-step prediction, model C1, which has the most MP steps, consistently produced the lowest prediction error for all test cases. The comparison indicates the importance of using sufficient number of MP steps to achieve high accuracy and robustness in the long-term prediction.

For the microgrid problem, since a load disturbance on the selected few load buses triggers a dynamical response in the whole grid, the network needs sufficient MP steps to ensure the global effects are captured. The lack of MP steps may also explain the poor performance of GCRN and STGCN in Sec. IV-B2. However, increasing the number of layers in GCRN and STGCN makes them vulnerable to over-smoothing issue in GNNs [39], and incurs prohibitive computational cost in training.

Lastly, the frequency and damping ratio of the dynamic responses are examined in further detail. Three models are selected for analysis: (1) C1, the nominal and best model; (2) C6, which is less accurate than C1; (3) C11, which diverges in long-term prediction. The frequencies f and damping ratios ζ of the first three dominating oscillation modes are extracted from the voltage response using the auto-regressive moving average (ARMA) method [40], and compared against the true values obtained from the analytical DAE model. The results are plotted in Fig. 7, where each data point corresponds to a node in the system and the shaded region illustrates the spread of the identified eigenvalues. In the true model, most of the nodes share similar frequencies but have different damping ratios. The eigenvalues of the PIDGeuN models differ drastically. Model C1 accurately captures most of the frequencies and damping ratios on different nodes for all three modes, which aligns with its low prediction errors. The model C6 over-predicts many damping ratios in the third mode and misses most of the frequencies, which leads to premature convergence to the equilibrium voltage. The model C11 under-predicts most of the damping ratios, and shows positive damping on some nodes, which can explain its early divergence in many short-term predictions.

2) Physics-informed Loss Function

Next, the effect of the physics-informed loss term (17), based on the Kirchhoff's law, is examined using model C12, where the PI loss is removed during the training process. Comparing the losses in Table 2, it is clear that the PI loss positively contributed to model performance. A further comparison of training loss of C1 and C12 is shown in Fig. 8. For C12, the training of network depends solely on the RMSE loss, and the additional loss term is only computed for recording purpose. During the training process, the RMSE of both cases decreased at a similar rate. However, the model prediction of C12 violates the Kirchhoff's law one order of magnitude more than that of C1, resulting in a higher total loss in training. As a result, in the actual prediction, the C12 model may produce responses that prone to violate the Kirchhoff's law, which explains its higher C-RMSE's in both 200-step and 700-step predictions than C1. Further examination of the predicted dynamics, though not shown in paper due to space limit, reveals that C12 can only capture the first few oscillations accurately and start to diverge after around 400 steps. Mathematically speaking, the PI loss term limits the learnable parameter space where the optimizer searches during the training, and results in more feasible model prediction in the tests.

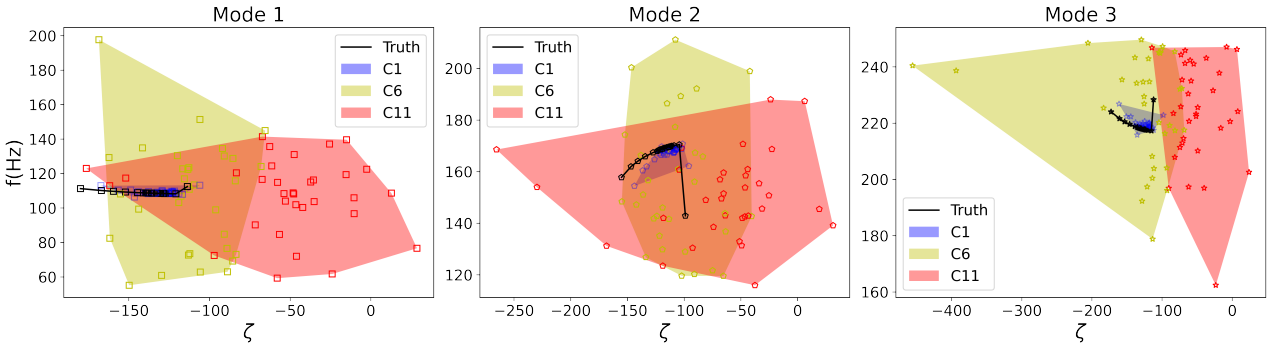


Figure 7: Frequencies and damping ratios of the first three modes in voltage dynamics, and the corresponding model predictions.

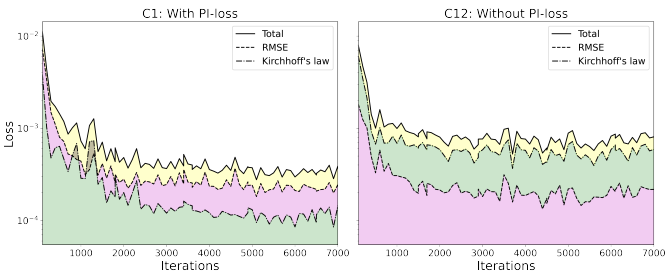


Figure 8: Comparison of loss function components in training.

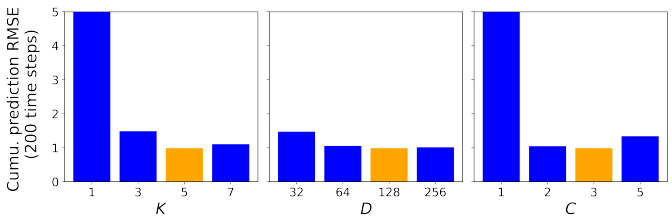


Figure 9: The effects of different hyperparameter choices on prediction performance.

3) Other hyperparameters

Finally, we study the effects of three main hyperparameters in the PIDGeuN network through a series of models: the order of Chebyshev polynomials K (C13-C15), the latent size of each hidden layer D (C16-C18), and the number of steps to include in the input C (C19-C21). The results are provided in Table 2 and visualized in Fig. 9. In these tests, the number of GALs and GCLs are kept the same as C1, i.e., $N_A = N_G = 5$.

As found earlier, a sufficient number of MP steps is critical in the microgrid prediction problem, which can be achieved by increase the polynomial order K in the GCLs. In the parametric study, increasing K from 1 to 3 rapidly decreases the prediction error as expected, but improvement becomes marginal beyond that. Once sufficient MP is reached, the model performance does not benefit from a larger K.

The size of hidden layer D determines the size of a network. The performance of PIDGeuN turns out to be less sensitive to D, and $D = 128$ achieves a good balance between network size and prediction accuracy.

The number of input steps C decides the amount of previous information the network can access during prediction. When $C = 1$, the network can only access the bus state of the current step, which is insufficient to reconstruct the dominating dynamics, resulting in poor performance. The performance is improved immediately when another step of states is included in the input ($C = 2$), but the improvement becomes marginal as more steps of previous states are included. This is likely due to the nature of the current problem where a long-term temporal dependency is not significant, and two steps of measured states already form sufficient time delay embedding to fully describe the system.

V. CONCLUSION

In this paper, we presented the Physics-Informed Dynamic Graph Neural Network, PIDGeuN, for accurate, efficient and robust prediction of transient dynamics in microgrids. The PIDGeuN model exploits its graph-based architecture to incorporate the topological information of microgrids. Furthermore, based on a judiciously designed message passing mechanism, the PIDGeuN incorporates two physics-informed techniques to improve its predictive performance. First, the PIDGeuN dynamically learns and adjusts the underlying graph representation of the system by combining the data-driven attention-based weights and physics-informed admittance-based weights, and thus better represents the inter-dependencies between buses. Second, the PIDGeuN includes the known equation of physical law of the power system in the loss function that ensure the feasibility of the predictions.

The PIDGeuN is demonstrated using transient response data of microgrids due to load changes that contain complete transient responses, and initial transient responses. The results show that the PIDGeuN can accurately and robustly predict the dynamics of the microgrid using initial states and load changes in the system, and outperforms a number of baseline methods in the transient predictions. Specifically, in the complete response cases, the PIDGeuN accurately captures the frequencies and damping ratios of the system as well as the new equilibrium states after the system stabilizes. In the initial response cases, the PIDGeuN is capable of capturing the transient and nonlinear dynamics due to the frequent load changes. The physics-informed techniques are proven to significantly contribute to the predictive accuracy of the model.

As a new data-driven methodology, the PIDGeuN architecture may be further extended and tested towards its real-world application. Currently the model assumes a high level of availability and accuracy in the measurement data. Yet, message passing and time-delay embedding can be generalized to account for data that is missing spatially and temporally; denoising and anomaly detection modules can be added to preprocess potentially polluted inaccurate data. Also, the PIDGeuN architecture shall be tested on larger systems with more complex operation scenarios having, e.g., un-balanced loads, and single and double phase laterals, and with more load characteristics, e.g., daily variation and dynamic motor models.

In sum, the results establish initial capability of the PIDGeuN to be applied to large scale networked microgrids, and show its potential as an online predictive tool to enable predictive or preventive control in real time applications, which is crucial to the stable operations of the NMs.

References

[1] Y. Li, *Cyber-Physical Microgrids*. Springer, 2021.

[2] P. Kundur, J. Paserba, V. Ajjarapu, G. Andersson, A. Bose, C. Canizares, N. Hatziargyriou, D. Hill, A. Stankovic, C. Taylor, T. Van Cutsem, and V. Vittal, "Definition and classification of power system stability ieee/cigre joint task force on stability terms and definitions," *IEEE Transactions on Power Systems*, vol. 19, 2004.

[3] M. Farrokhabadi, C. A. Cañizares, J. W. Simpson-Porco, E. Nasr, L. Fan, P. A. Mendoza-Araya, R. Tonkoski, U. Tamrakar, N. Hatziargyriou, D. Lagos, R. W. Wies, M. Paolone, M. Liserre, L. Meegahapola, M. Kabalan, A. H. Hajimiragha, D. Peralta, M. A. Elizondo, K. P. Schneider, F. K. Tuffner, and J. Reilly, "Microgrid stability definitions, analysis, and examples," *IEEE Transactions on Power Systems*, vol. 35, no. 1, pp. 13–29, 2020.

[4] N. Hatziargyriou, J. Milanovic, C. Rahmann, V. Ajjarapu, C. Canizares, I. Erlich, D. Hill, I. Hiskens, I. Kamwa, B. Pal, P. Pourbeik, J. Sanchez-Gasca, A. Stankovic, T. Van Cutsem, V. Vittal, and C. Vournas, "Definition and classification of power system stability – revisited & extended," *IEEE Transactions on Power Systems*, vol. 36, 2021.

[5] A. R. Sobbouhi and A. Vahedi, "Transient stability prediction of power system; a review on methods, classification and considerations," *Electric Power Systems Research*, vol. 190, p. 106853, 2021.

[6] H. Natke, "Recent trends in system identification," in *Structural dynamics*. Routledge, 2022, pp. 283–289.

[7] J. W. Pierre, D. Trudnowski, M. Donnelly, N. Zhou, F. K. Tuffner, and L. Dosiek, "Overview of system identification for power systems from measured responses," *IFAC Proceedings Volumes*, vol. 45, 2012.

[8] L. Fu and P. Li, "The research survey of system identification method," in *2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2. IEEE, 2013, pp. 397–401.

[9] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, 2005.

[10] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[11] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, "Learning to simulate complex physics with graph networks," *arXiv:2002.09405*, 2020.

[12] A. Sanchez-Gonzalez, V. Bapst, K. Cranmer, and P. Battaglia, "Hamiltonian graph networks with ODE integrators," *arXiv:1909.12790*, 2019.

[13] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," *arXiv:1612.07659*, 2018.

[14] D. Owerko, F. Gama, and A. Ribeiro, "Optimal Power Flow Using Graph Neural Networks," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2020-May, pp. 5930–5934, 2020.

[15] V. Bolz, J. Rueß, and A. Zell, "Power flow approximation based on graph convolutional networks," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 2019, pp. 1679–1686.

[16] A. B. Jeddi and A. Shafieezadeh, "A physics-informed graph attention-based approach for power flow analysis," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021, pp. 1634–1640.

[17] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State Estimation in Electric Power Systems Leveraging Graph Neural Networks," *arXiv:2201.04056*, 2022.

[18] L. Pagnier and M. Chertkov, "Physics-informed graphical neural network for parameter & state estimations in power systems," *arXiv:2102.06349*, 2021.

[19] D. Owerko, F. Gama, and A. Ribeiro, "Predicting power outages using graph neural networks," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2018, pp. 743–747.

[20] M. MansourLakouraj, R. Hossain, H. Livani, and M. Ben-Idris, "Application of graph neural network for fault location in PV penetrated distribution grids," in *2021 North American Power Symposium (NAPS)*, 2021, pp. 01–06.

[21] A. M. Karimi, Y. Wu, M. Koyuturk, and R. H. French, "Spatiotemporal graph neural network for performance prediction of photovoltaic power systems," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 17, pp. 15 323–15 330, May 2021.

[22] J. Simeunovic, B. Schubnel, P. J. Alet, and R. E. Carrillo, "Spatio-temporal graph neural networks for multi-site PV power forecasting," *IEEE Transactions on Sustainable Energy*, p. 1–1, 2021.

[23] P. Sun, L. Huo, S. Liang, and X. Chen, "Fast transient stability prediction using grid-informed temporal and topological embedding deep neural network," *arXiv2201.09245*, 2022.

[24] X. Zhou, X. Guan, D. Sun, H. Jiang, J. Peng, Y. Jin, and Y. Zhang, "Transient stability assessment based on gated graph neural network with imbalanced data in internet of energy," *IEEE Internet of Things Journal*, pp. 1–1, 2021.

[25] T. Zhao, M. Yue, and J. Wang, "Structure-informed graph learning of networked dependencies for online prediction of power system transient dynamics," *IEEE Transactions on Power Systems*, vol. 37, no. 6, pp. 4885–4895, 2022.

[26] M. Rafiq, G. Rafiq, and G. S. Choi, "Dsfa-pinn: Deep spectral feature aggregation physics informed neural network," *IEEE Access*, vol. 10, pp. 22 247–22 259, 2022.

[27] F. Takens, "Detecting strange attractors in turbulence," in *Dynamical systems and turbulence, Warwick 1980*. Springer, 1981, pp. 366–381.

[28] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, pp. 4–24, 2021.

[29] W. L. Hamilton, *Graph Representation Learning: Foundations, Methods, Applications and Systems*, 2020, vol. 14, no. 3.

[30] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv:1312.6203*, 2013.

[31] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2017.

[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv:1502.03167*, 2015.

[34] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

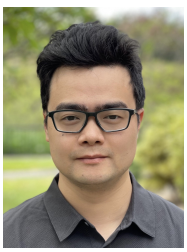
[35] P. Van Overschee and B. De Moor, *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

[36] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, p. 3104–3112.

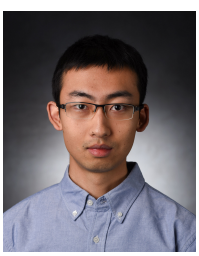
[37] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 3634–3640.

[38] B. Rozemberczki, P. Scherer, Y. He, G. Panagopoulos, A. Riedel, M. Astefanoaei, O. Kiss, F. Beres, , G. Lopez, N. Collignon, and R. Sarkar, "PyTorch Geometric Temporal: Spatiotemporal Signal Processing with

Neural Machine Learning Models," in *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, 2021, p. 4564–4573.
[39] Q. Li, Z. Han, and X.-m. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018.
[40] M. Smail, M. Thomas, and A. Lakis, "ARMA models for modal analysis: Effect of model orders and sampling frequency," *Mechanical Systems and Signal Processing*, vol. 13, no. 6, pp. 925–941, 1999.



YIN YU received his dual bachelor's degree in Aerospace Engineering from the University of Michigan, Ann Arbor in 2014, in Mechanical Engineering from Shanghai Jiao Tong University, Shanghai, China in 2015, and earned his master's degree in Aerospace Engineering from U of M. in 2016. He is currently working toward his Ph.D. degree in Aerospace Engineering at the Pennsylvania State University, University Park, PA, USA. His research interests include geometric deep learning for aerospace and generic engineering applications, with an emphasis on dynamical system modeling.



XINYUAN JIANG received the B.S. degrees in electrical engineering from the North China Electric Power University, Baoding, China, and Illinois Institute of Technology, Chicago, IL, USA, both in 2017. He is currently working toward his Ph.D. degree in electrical engineering at the Pennsylvania State University, University Park, PA, USA. His research interests include data-driven modeling and distributed analysis of microgrid systems.

DANING HUANG received his Ph.D. degree in Aerospace Engineering from the University of Michigan, Ann Arbor in 2019. He is currently an assistant professor of aerospace engineering at the Pennsylvania State University. His research interests include data-driven modeling, analysis, control and optimization of multi-disciplinary dynamical systems.

YAN LI received her Ph.D. degree in Electrical Engineering from the University of Connecticut, Storrs, CT, U.S., in 2019. She also holds a Ph.D. degree in Electrical Engineering from Tianjin University, Tianjin, China, in 2013. She is currently an assistant professor at the School of Electrical Engineering and Computer Science at the Pennsylvania State University, University Park, PA, U.S. Her research interests include cyber-physical microgrids, data-driven modeling and control, quantum computing, stability and resilience analysis, cybersecurity, reachability, software-defined networking, etc.

MENG YUE received the Ph.D. degree in electrical engineering from Michigan State University, East Lansing, MI, USA. He is currently with Interdisciplinary Science Department, Brookhaven National Laboratory, Upton, NY, USA. His research interests include power system dynamics and control, uncertainty modeling and probabilistic applications in grid planning and operation, and data analytic in damage forecasting and anomaly detection

TIANQIAO ZHAO received the B.Eng. degree in automatic control from North China Electric Power University, Beijing China, in 2013, and the Ph.D. degree in electrical and electronic engineering from the University ofManchester, Manchester, U.K., in 2019. From September 2018 to August 2019, he was a Postdoctoral Associate with the Department of Electrical and Electronic Engineering, University of Manchester. From September 2019 to February 2021, he was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, Southern Methodist University, Dallas, TX, USA. He is currently working with Brookhaven National Laboratory, Upton, NY, USA. His research interests include machine learning applications, distributed optimization and control, smart grid, and energy storage systems.