# Spatio-Temporal Graph Convolutional Neural Networks for Physics-Aware Grid Learning Algorithms

Tong Wu, *Member, IEEE*, Ignacio Losada Carreño, *Member, IEEE*, Anna Scaglione, *Fellow, IEEE*, Daniel Arnold, *Member, IEEE*,

***Abstract*—This paper proposes novel architectures for spatio-temporal graph convolutional and recurrent neural networks whose structure is inspired by the physics of power systems. The key insight behind our design consists in deriving the so-called graph shift operator (GSO), which is the cornerstone of Graph Convolutional Neural Network (GCN) and Graph Recursive Neural Network (GRN) designs, from the power flow equations. We demonstrate the effectiveness of the proposed architectures in two applications: in forecasting the power grid state and in finding a stochastic policy for foresighted voltage control using deep reinforcement learning. Since our design can be adopted in single-phase as well as three-phase unbalanced systems, we test our architecture in both environments. For state forecasting experiments we consider the single phase IEEE 118-bus case systems; for voltage regulation, we illustrate the performance of deep reinforcement learning policy on the unbalanced three-phase IEEE 123-bus feeder system. In both cases the physics based GCN and GRN learning algorithms we propose outperform the state of the art.**

***Index Terms*—GCN, Deep Reinforcement Learning, Cyber-Physical Attacks.**

## NOMENCLATURE

*Abbreviation*

| | |
|---|---|
| GCN | Graph convolutional neural network. |
| GRN | Graph recurrent neural network. |
| GSO | Graph shift operator. |
| GS | Graph signal. |
| GSP | Graph signal processing. |
| FNN | Fully connected neural network. |
| CNN | Convolutional neural network. |
| RNN | Recurrent neural network. |
| DRL | Deep reinforcement learning. |
| PSSE | Power system state estimation. |
| PSSF | Power system state forecasting. |

*Sets*

| | |
|---|---|
| $\mathcal{N}, \mathcal{E}$ | Sets of grid buses $\mathcal{N} = \{1, \cdots, N\}$ and lines. |
| $\mathcal{N}_s$ | subset of single-phase buses with smart inverters. |
| $\mathcal{P}_{mn}$ | Phases of line $(m, n) \in \mathcal{E}$. |
| $\mathcal{P}_m$ | Phases of node $n \in \mathcal{N}$. |

*Variables*

| | |
|---|---|
| $\boldsymbol{v}_n$ | $\boldsymbol{v}_n = [v_{n_\phi} \vert \phi \in \mathcal{P}_n] \in \mathbb{C}^{\vert\mathcal{P}_n\vert \times 1}$ with phase $\angle v_{n_\phi}$ and magnitude $\vert v_{n_\phi}\vert$. |
| $\boldsymbol{v}, \boldsymbol{i}$ | Vectors of all voltage current injections. |
| $\boldsymbol{s}$ | Vector of all apparent power injections, $\boldsymbol{s} = \boldsymbol{p} + \mathrm{j}\boldsymbol{q}$. |

*Operators*

| | |
|---|---|
| $\boldsymbol{A}^T, \boldsymbol{A}^H$ | The transpose and Hermitian of matrix $\boldsymbol{A}$. |
| $D(\boldsymbol{A})$ | The vector of the diagonal elements of $\boldsymbol{A}$. |
| $diag(\boldsymbol{a})$ | A diagonal matrix with diagonal entries from $\boldsymbol{a}$. |
| $\boldsymbol{A} \circ \boldsymbol{B}$ | Hadamard product (entry by entry product). |
| $(\boldsymbol{A})^*$ | Conjugate of a complex vector or matrix. |

Tong Wu, Ignacio Losada Carreño and Anna Scaglione are with the Department of Electrical and Computer Engineering, Cornell Tech, Cornell University, New York City, NY, 10044 USA (e-mail: {tw385, il244, as337}@cornell.edu). Daniel Arnold is with Lawrence Berkeley National Laboratory (e-mail: dbarnold@lbl.gov). Preliminary work was invited to be presented in [1] in the 58th Annual Allerton Conference, 2022.

This research was supported in part by the National Science Foundation (NSF) under Grant NSF ECCS # 2210012; and in part by the Director, Cybersecurity, Energy Security, and Emergency Response, Cybersecurity for Energy Delivery Systems program, of the U.S. Department of Energy, under contract DE-AC02-05CH11231. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsors of this work.

## I. INTRODUCTION

### A. Background and Motivation

The access to high-quality grid sensors data, and phasor measurement units (PMUs) in particular, has prompted a lot of interest in applying advanced learning algorithms to address grid inference and control problems [2]. The main advantage of learning techniques, when compared to regression or optimization problems that purely rely on the physics, lies in their ability to internalize statistical patterns in the training data that are not captured by the physical constraints only. On the other hand, while many *black-box* learning approaches respond well to the challenge, accounting for the physical equations explicitly, rather than learning them as a pattern, reduces the number of parameters in the model, mitigating overfitting problems. This is why not all Neural Networks (NN) have the same architecture and, in particular, both time-series and images are best processed by Convolutional or Recursive NN, leveraging the shift or state invariance of the data to reduce the parameter space and increase generalization ability.

The framework of Graph Signal Processing (GSP) has emerged as the the most promising approach to generalize these architectures to data that have the irregular support of a network. GSP provides a natural representation for both the data (node attributes) and the underlying structure (edge attributes) [3]. Its application to grid measurements has recently spurred significant interest [4]. However, the linear models of Graph filters have limited capability to learn the possibly complex mappings that are needed for classification, forecasting and for the approximation of optimum control policies when compared to neural network models. In fact, on several applications that include high-dimensional data from a network structure, GCNs have shown to have the best generalization capabilities [5]. Motivated by the promise of AI applications to power system data, the overarching goal of this paper is to develop state of the art GCN architectures for inference and decision making that best capture the spatial and temporal features of grid data that derive from the AC power-flow constraints.

### B. Related Works

We first highlight the related prior research on grid GSP and GCN and then the literature on the two applications

we consider to showcase the performance benefits of the proposed schemes, namely: power system state estimation and forecasting, and reinforcement learning for voltage control.

*1) Graph Convolutional Neural Network:* A handful of papers have so far successfully applied GCN to distribution systems' management, considering applications that include fault localization [6], distribution system state estimation [7], and synthetic data generation [8]. GCNs are a generalization of CNN, aimed at capturing the impact that the network connectivity has on the patterns of the data associated to the network nodes. The foundation of GCN lies in the GSP definition of graph filters and of what is referred to as the Graph Shift Operator (GSO). In [6], the GSO is defined as the weighted adjacent matrix, constructed based on the physical distance between nodes. In [7] the authors prune the weights of a conventional Fully connected Neural Network (FNN) based on the power grid topology, without considering the grid lines admittances. In the paper [8] the GSO is constructed as an adjacent matrix that captures the correlation among historical data. All these approaches are not directly considering the electrical characteristics of the overhead power lines. In prior work [4] we provided ample evidence that the right framework to apply GSP for grid signals should be rooted in the basic network analysis that has been used to model power systems signals for decades [9]. In fact, Ohm's law is the obvious driver of the correlation in the state vector of the grid, and the GSO can be derived from first principles from the grid physics, which suggests that the right GSO is the admittance matrix itself. While [4] has shown the benefits of using complex graph filtering methods to address a number of inference problems, physics-based Graph Neural Networks architectures are still missing. We note that Pytorch and Tensorflow [10, 11] operate on real valued nodal data, which are not compatible with the GSO derivation in [4].

This paper addresses two gaps left by the prior art. First, we derive a physics inspired GSO based on the power flow equations in the real domain. Using that we defined two efficient spatio-temporal graph neural network architectures: Graph Convolutional Neural Networks (GCN) and Graph Recurrent Neural networks (GRN). Second, we extend these architectures to the unbalanced three-phase power systems, unleashing the power of physics inspired AI methods to distribution systems.

*2) Power System State Estimation and Forecasting:* There is a vast literature on Power System State Estimation (PSSE) [12]. The use of graph neural networks is very recent [13, 14]. Neither papers derived the GSO from the power flow equations (e.g. adjacency matrix used in [13]). Also, [13, 14] ignored the temporal dependencies of the voltage phasors and focused on state reconstruction, not forecasting [1]. Power Systems State Forecasing (PSSF) has so far been pursued via a single-hidden-layer NN in [15, 16]. Because the number of FNN parameters grows linearly with the length of the input sequences, the proposed methods are prone to overfitting. In Fig. 5 of Section V, we show that our method attains about an order of magnitude improvement in accuracy over the FNN in [15, 16].

*3) Reinforcement Learning for Voltage Control:* Voltage control problems can be modeled as mixed-integer nonlinear programs that include an optimal power flow; they are *nonconvex* and *NP-hard* [17] and, therefore, impractical for a real-time implementation. In recent years several authors have explored Deep Reinforcement Learning (DRL) methods as alternatives to brute force optimization, to search via training approximately optimal policy functions, parametrized as a neural network. Existing DRL methods for Volt-VAR control in distribution grids are broadly classified as value-based [18–21] and policy-based RL algorithms [22–24]. Unfortunately, DRL algorithms can become unstable when combining function approximation, off-policy learning, and bootstrapping (a combination referred to as the deadly triad [25]). Many authors resorted to FNN or CNN architectures which, as we discussed previously, are over-parametrized in their feature extraction layers and, therefore, likely to trigger the *deadly triad* of DRL, i.e., ending up in a lot of instabilities, or no convergence [17].

Similar conclusions apply to the adversarial DRL approach for Volt-VAR control proposed in [26] for distribution grids. Very recently, [27, 28] considered the graph correlation of voltage phasors in their DRL design, but ignored the temporal correlation of their time series. Note that [27, 28] require the full system state. In fact, all aforementioned algorithms require access to measurements of the full state of the system as an input [26, 29]. Even when the state is observable, it is hard to scale these methods to work with large-scale networks with high-dimensional features [17, 30]. This motivates the derivation of a reduced GSO that contracts the network to the buses that are controlled.

### C. Contributions and Organization

Our main contributions are summarized next:

- We develop novel physics-aware Graph Convolutional Neural networks and Graph Recursive Neural networks architectures that are applicable to single and three-phase unbalanced power systems. Architecturally, the main novelty is in deriving a real-valued GSO from the physics of power flow equations.
- For the GCN design, in each layer the parameters of Chebyshev graph-temporal filters capture the spatio-temporal features of the grid signals. In terms of the GRN design, we combine the feedback structure of RNN with the graph filters to consider the spatio-temporal dependence of the grid signals.
- To deal with sparse deployments of PMUs, or simply for scalability we show that one can use the kron-reduced network GSO instead of the full GSO.
- We demonstrate in two case studies how the proposed architectures outperform the state of the art in PSSE and PSSF, as well as in Volt-Var control. Specifically, for the PSSE application, GCN and GRN has an order of magnitude smaller MSE than other benchmarks, i.e., FNN, CNN, RNN and GNN. For the Volt-Var control, GCN and GRN do enhance the stability of DRL, whereas FNN and CNN trigger the deadly triad of DRL.

[1]Power system state estimation (reconstruction) estimates the *current* the system state, i.e. the voltage phasors at all buses, based on the available measurements. Power system state forecasting, instead predicts the *future* the state based on current and past measurements.

The rest of the paper is organized as follows. In Section II, we briefly review the key notions of GSP, setting the stage in Section III where we derive the physics inspired GSO and introduce our graph neural networks architectures. In Section IV, we describe two applications of the proposed GCN and GRN frameworks that are tested numerically in Section V. Finally, we conclude the paper in Section VI.

## II. A Brief Review of Graph Signal Processing

GSP is a vibrant branch of signal processing research whose aim is to generalize digital signal process (DSP) notions [31]. The cornerstone of GSP is the definition of Graph Shift Operator (GSO). The vast majority of GSP-based algorithms uses real-valued GSOs and considers real-valued graph signals (GS) (see e.g. the surveys [3, 32]). Having selected the GSO, one can define graph-filters; the most popular graph filter model is the Chebyshev filter [31, 33, 34].

To make the paper self-contained, we first review the basic theory of GSP (more details can be found e.g. in [4]) for a general graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, with vertex set $\mathcal{V}$ and edge set $\mathcal{L}$. The concepts defined here will be applied to three-phase distribution network whose graph topology is $\mathcal{D} = (\mathcal{N}, \mathcal{E})$, where the node $n_\phi$ of bus $n$ on phase $\phi$ in $\mathcal{D}$ corresponds to the $i^{th} \in \mathcal{V}$ node of $\mathcal{G}$ and the edges are the transmission lines connecting the buses. A graph signal $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{V}|}$ (which, in the grid, is mainly the state vector) is a vector indexed by the network nodes. The set $\mathcal{N}_i$ denotes the subset of nodes connected to node $i$, i.e. node $i$'s neighborhood. A GSO is a matrix $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ that linearly combines graph signal neighbors' values. Almost all operations including filtering, transformation and prediction are directly related to the GSO [4] which generalizes the $s$ variable representing the derivative in the Laplace domain for signals in time. Consistent with the intuition that it should operate as a differential operator, the GSO, denoted by $\mathbf{S} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, is usually chosen as a graph weighted Laplacian:

$$[\mathbf{S}]_{ij} = \begin{cases} \sum_{k\in\mathcal{N}_i} S_{i,k}, & i = j, \\ -S_{i,j}, & i \neq j. \end{cases} \tag{1}$$

In this work, we focus on real symmetric GSOs, i.e., $\mathbf{S} = \mathbf{S}^\top$ that are appropriate for power grid applications. A graph filter is a linear matrix operator $\mathcal{H}(\mathbf{S})$, function of the GSO, that operates on graph signals as follows

$$\boldsymbol{w} = \mathcal{H}(\mathbf{S})\boldsymbol{x}. \tag{2}$$

What defines the dependency of $\mathcal{H}(\mathbf{S})$ on the GSO is that $\mathcal{H}(\mathbf{S})$ must be shift-invariant (like a linear time invariant filter in the time domain), i.e. $\mathcal{H}(\mathbf{S})\mathbf{S} = \mathbf{S}\mathcal{H}(\mathbf{S})$. This is possible if and only if $\mathcal{H}(\mathbf{S})$ is a matrix polynomial [2]:

$$\mathcal{H}(\mathbf{S}) = \sum_{k=0}^{K} h_k \mathbf{S}^k. \tag{3}$$

Let the eigenvalue decomposition be $\mathbf{S} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$, where $\mathbf{\Lambda}$ is a diagonal matrix with eigenvalues $\lambda_1 \leq \cdots \leq \lambda_{|\mathcal{V}|}$. Since the GSO $\mathbf{S}$ is symmetric, $\mathbf{U}$ is unitary and the basis for Graph Fourier Transform (GFT). The GFT of a graph signal is, therefore, $\tilde{\boldsymbol{x}} = \mathbf{U}^\top \boldsymbol{x}$ and the eigenvalues $\lambda_\ell, \ell = 1, \ldots, |\mathcal{V}|$ are the *graph frequencies*. From (3) it follows that:

$$\mathcal{H}(\mathbf{S}) = \mathbf{U}\Big(\sum_{k=0}^{K} h_k \Lambda^k\Big)\mathbf{U}^{-1}. \tag{4}$$

The matrix $\sum_{k=0}^{K} h_k \Lambda^k$ is a diagonal, with $i^{th}$ entry $\tilde{h}(\lambda_i) \triangleq \sum_{k=0}^{K} h_k \lambda_i^k$. Hence, $\tilde{\boldsymbol{h}} = [\tilde{h}(\lambda_1), \ldots, \tilde{h}(\lambda_{|\mathcal{V}|})]$ is the transfer function for graph filters. In the GFT domain this yields:

$$\boldsymbol{w} = \mathcal{H}(\mathbf{S})\boldsymbol{x} \iff \tilde{\boldsymbol{w}} = \tilde{\boldsymbol{h}} \circ \tilde{\boldsymbol{x}}. \tag{5}$$

For time series of graph signals $\{\boldsymbol{x}_t\}_{t\geq 0}$ one can use graph temporal filters models:

$$\boldsymbol{w}_t = \sum_{\tau=0}^{t} \mathcal{H}_{t-\tau}(\mathbf{S})\boldsymbol{x}_\tau \qquad \mathcal{H}_t(\mathbf{S}) = \sum_{k=0}^{K} h_{k,t}\mathbf{S}^k, \tag{6}$$

and harness DSP tools, defining a combined GFT and $z-$transform for their analysis:

$$\mathbf{X}(z) = \sum_{t=0}^{T-1} \boldsymbol{x}_t z^{-t}, \quad \tilde{\mathbf{X}}(z) = \mathbf{U}^\top \mathbf{X}(z), \tag{7}$$

where $T$ is the length of the graph signal time series. In particular, for filter of order $T$, $\mathbf{S} \otimes z$ is the graph temporal GSO and a graph Spatio-Temporal filter is defined as follows:

$$\mathcal{H}(\mathbf{S} \otimes z) = \sum_{k=0}^{K} H_k(z)\mathbf{S}^k, \quad H_k(z) = \sum_{t=0}^{T-1} h_{k,t} z^{-t} \tag{8}$$

i.e. $H_k(z)$ is the $z-$transform of the filter coefficients $h_{k,t}$. In the $z$-domain the input output relationship is:

$$\mathbf{W}(z) = \mathcal{H}(\mathbf{S} \otimes z)\mathbf{X}(z), \tag{9}$$

The graph-temporal joint transfer function is:

$$\mathbb{H}(\mathbf{\Lambda}, z) = \sum_{k=0}^{K}\sum_{t=0}^{T-1} h_{k,t}\mathbf{\Lambda}^k z^{-t}. \tag{10}$$

which is a diagonal matrix. Denoting by $\tilde{\mathbf{X}}(z) = \mathbf{U}^\top \mathbf{X}(z)$, the input-output relationship in the combined GFT-$z$-domain is:

$$\tilde{\mathbf{W}}(z) = \mathbb{H}(\mathbf{\Lambda}, z)\tilde{\mathbf{X}}(z). \tag{11}$$

## III. Graph Convolutional Neural Networks Architectures for Power Grid Signals

In Fig. 1, we illustrate differences of GCN, CNN and FNN applied to the same graph signal with specific neuron structures for each time instant. Observe that GCNs are generalizations of CNNs where time-series filters are replaced by application-dependent graph temporal graph filters. In fact, in a GCN, the weights $h_{k,t}$ are the parameters learnt during training in the feature extraction layers [33]. In FNNs, instead, one may train matrices with arbitrary weights, and therefore the parameters grow in the order of the number of nodes and filter memory squared. Next, we will introduce the physics-based derivation of the GSO for the grid GCN.

[2]Note that the graph filter order $K$ can be infinite.

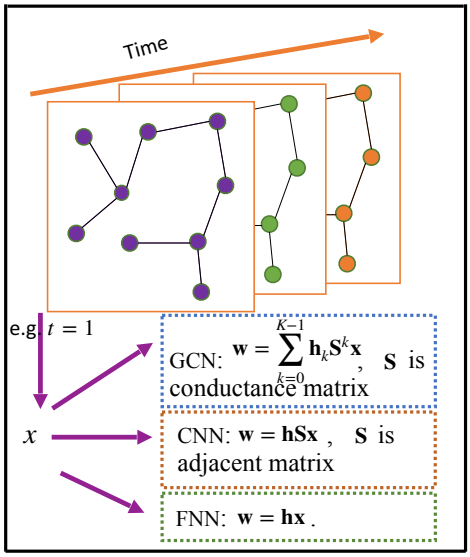



Fig. 1: Information flow of the power systems.

### A. Real-Valued Grid Graph System Operator

A physics-inspired framework for Grid-GSP was proposed in [4] to provide an interpretation for the spatio-temporal properties of voltage phasor measurements by utilizing the admittance matrix as graph filters. GCNs tools currently takes only real-valued inputs [35]. Next we show how to derive a real-valued, physics inspired, GSO from the power flow equations that can be used to capture the features of the real graph signal components represented by the pairs of voltage magnitude $|v_{n_\phi}|$ and re-centered voltage phase $\varphi_{n_\phi}$ vectors. Let $\boldsymbol{s}=\boldsymbol{p}+\mathrm{j}\boldsymbol{q}$ be the vector of net apparent power at buses ($\boldsymbol{s}=[\boldsymbol{s}_1^\top,\cdots,\boldsymbol{s}_{|\mathcal{N}|}^\top]^\top$), with the $n^{th}$ entry $\boldsymbol{s}_n=\boldsymbol{p}_n+\mathrm{j}\boldsymbol{q}_n, \boldsymbol{s}_n\in\mathbb{C}^{|\mathcal{P}_n|\times 1}$. Further, let $\boldsymbol{v}$ and $|\boldsymbol{v}|$ be the vectors of bus voltage phasors and magnitudes, respectively, with $\boldsymbol{v}\in\mathbb{C}^{\sum_{n\in\mathcal{N}}|\mathcal{P}_n|\times 1}$ and $|\boldsymbol{v}|\in\mathbb{R}_+^{\sum_{n\in\mathcal{N}}|\mathcal{P}_n|\times 1}$, and let $\boldsymbol{i}\in\mathbb{C}^{\sum_{n\in\mathcal{N}}|\mathcal{P}_n|\times 1}$ and $|\boldsymbol{i}|\in\mathbb{R}_+^{\sum_{n\in\mathcal{N}}|\mathcal{P}_n|\times 1}$ be the vectors of net bus current phasors and magnitudes, respectively:

$$v_{n_\phi}=|v_{n_\phi}|\,e^{\mathrm{j}\angle v_{n_\phi}},\ i_{n_\phi}=|i_{n_\phi}|\,e^{\mathrm{j}\angle i_{n_\phi}},\ \forall n\in\mathcal{N},\phi\in\mathcal{P}_n,$$
$$\boldsymbol{i}=\mathbf{Y}\boldsymbol{v}, \tag{12}$$

where $\mathbf{Y}$ in (12) is a block matrix of dimensions $\sum_{n\in\mathcal{N}}|\mathcal{P}_n|\times\sum_{n\in\mathcal{N}}|\mathcal{P}_n|$ and $\mathbf{B}$ is the susceptance matrix. More specifically, the blocks in $\mathbf{Y}$ are:

1) the matrices $\mathbf{Y}_{mn}$, occupying the $|\mathcal{P}_{mn}|\times|\mathcal{P}_{mn}|$ off-diagonal block corresponding to line $(m,n)\in\mathcal{E}$; and,
2) the $|\mathcal{P}_{mn}|\times|\mathcal{P}_{mn}|$ diagonal block corresponding to node $n\in\mathcal{N}$ with $\mathcal{N}_m=\{n|(m,n)\in\mathcal{E}\}$:

$$[\mathbf{Y}]_{\mathcal{P}_n,\mathcal{P}_n}=\sum_{m\in\mathcal{N}_n}\left(\frac{1}{2}\mathbf{Y}_{mn}^s+\mathbf{Y}_{mn}\right) \tag{13}$$

$\mathbf{Y}_{mn}^s$ is the shunt element. As first noted in [36], Ohm's law allows us to view voltage as the output *low-pass* filter by $\boldsymbol{v}=\mathbf{Y}^{-1}\boldsymbol{i}$ (an integrator), implying that $\mathbf{Y}$ is an appropriate GSO. The first step to obtain the GSO in the real domain, is to express the three-phase power flow equations as follows:

$$\boldsymbol{i}_n=\sum_{m\in\mathcal{N}_n}\left[\left(\frac{1}{2}\mathbf{Y}_{mn}^s+\mathbf{Y}_{mn}^{(n)}\right)\boldsymbol{v}_n+\mathbf{Y}_{mn}^{(m)}\boldsymbol{v}_m\right]$$

Note that, in general, for distribution lines $\mathbf{Y}_{mn}^{(m)}=-\mathbf{Y}_{mn}^{(n)}$ with the exception of transformer or regulators. In the following analysis, we omit the influence of transformer or regulators and assume $\mathbf{Y}_{mn}^{(m)}=-\mathbf{Y}_{mn}^{(n)}$. The power flowing from bus $n\in\mathcal{N}$ to bus $m\in\mathcal{N}$ is:

$$\boldsymbol{s}_n=D\left(\boldsymbol{v}_n\boldsymbol{i}_n^H\right),\qquad\forall\phi\in\mathcal{P}_n,n\in\mathcal{N}. \tag{14}$$

Assuming that the susceptance $\mathbf{B}$ dominates over the conductance, denoting the imaginary parts of the matrices $\mathbf{Y}_{mn}^s$, $\mathbf{Y}_{mn}^{(n)}$ and $\mathbf{Y}_{mn}^{(m)}$ respectively by $\mathbf{B}_{mn}^s$, $\mathbf{B}_{mn}^{(n)}$ and $\mathbf{B}_{mn}^{(m)}$ (which are symmetric), we have:

$$\boldsymbol{s}_n\approx\sum_{m\in\mathcal{N}_n}-\mathrm{j}D\left(\boldsymbol{v}_n\boldsymbol{v}_n^H\left(\frac{1}{2}\mathbf{B}_{mn}^s+\mathbf{B}_{mn}^{(n)}\right)+\boldsymbol{v}_n\boldsymbol{v}_m^H\mathbf{B}_{mn}^{(m)}\right). \tag{15}$$

From (15) we obtain an approximation of the power flow equations that describes the dependence between the active and reactive power on $|\boldsymbol{v}_n|$ and $\angle\boldsymbol{v}_n$ by approximating quadratic terms $\boldsymbol{v}_n\boldsymbol{v}_n^H$ and $\boldsymbol{v}_n\boldsymbol{v}_m^H$ as follows:

**Lemma 1** *With a first order expansion of the phase term the quadratic term $\boldsymbol{v}_n\boldsymbol{v}_m^H$ can be approximated as:*

$$\boldsymbol{v}_n\boldsymbol{v}_m^H\approx\left(\mathbb{1}\mathbb{1}^\top+\mathrm{j}(\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}_m^\top)\right)\circ\left(diag(|\boldsymbol{v}_n|)\Gamma diag(|\boldsymbol{v}_m|)\right)$$

*where $\boldsymbol{\varphi}_n$ is re-centered by $\boldsymbol{\varphi}_n^\top=[\varphi_{n_a},\varphi_{n_b},\varphi_{n_c}]\triangleq[\angle v_{n_a},\angle v_{n_b}+\frac{2\pi}{3},\angle v_{n_c}-\frac{2\pi}{3}]$, and $\Gamma$ is expressed as:*

$$[\Gamma]_{kn}=e^{\mathrm{j}\frac{2(k-n)\pi}{3}}=[\Gamma_c]_{kn}+\mathrm{j}[\Gamma_s]_{kn},\ k,n\in\{0,1,2\}$$

1) *In the active power equation the dominant term is:*

$$\boldsymbol{v}_n\boldsymbol{v}_m^H\approx\mathrm{j}(\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}_m^\top) \tag{16}$$

2) *In the reactive power equation the dominant term is:*

$$\boldsymbol{v}_n\boldsymbol{v}_m^H\approx diag(|\boldsymbol{v}_n|)\Gamma diag(|\boldsymbol{v}_m|) \tag{17}$$

*Replacing $m$ with $n$ in (16) and (17), we have the approximation for $\boldsymbol{v}_n\boldsymbol{v}_n^H$ for the active and reactive power injections.*

**Proof 1** *See the proof in the appendix.*

Applying the approximations in Lemma 1 to the expression in (15) we obtain the physics in inspired GSO introduced in the the following Proposition:

**Proposition 1** *Let us define the following matrices:*

$$\hat{\mathbf{B}}_{mn}^s\triangleq\Gamma_c\circ\mathbf{B}_{mn}^s,\qquad\hat{\mathbf{B}}\triangleq((\mathbb{1}\mathbb{1}^\top)_N\otimes\Gamma_c)\circ\mathbf{B}, \tag{18}$$

$$\boldsymbol{p}_n^{cst}\triangleq\tfrac{1}{2}D\left(\Gamma_s\mathbf{B}_{mn}^s\right),\quad\boldsymbol{q}_n^{cst}\triangleq-\frac{1}{2}D(\hat{\mathbf{B}}_{mn}^s) \tag{19}$$

*where $(\mathbb{1}\mathbb{1}^\top)_N$ is the all-ones matrix with dimension $N$ and $\otimes$ is Kronecker product. Let $\boldsymbol{p}$, $\boldsymbol{q}$, $\boldsymbol{p}^{cst}$, $\boldsymbol{q}^{cst}$, $|\boldsymbol{v}|$ and $\boldsymbol{\varphi}$ be the vectors stacking all the sub-vector corresponding to the multi-phase grid buses. The following approximations holds:*

$$\begin{bmatrix}\boldsymbol{p}\\\boldsymbol{q}\end{bmatrix}-\begin{bmatrix}\boldsymbol{p}^{cst}\\\boldsymbol{q}^{cst}\end{bmatrix}=\overbrace{\begin{bmatrix}\hat{\mathbf{B}}&\mathbf{0}\\\mathbf{0}&\hat{\mathbf{B}}\end{bmatrix}}^{\mathbf{S}:\ GSO}\overbrace{\begin{bmatrix}\boldsymbol{\varphi}\\|\boldsymbol{v}|\end{bmatrix}}^{\boldsymbol{x}:\ GS}=\mathbf{S}\boldsymbol{x} \tag{20}$$

*where $\boldsymbol{x}$ and $\mathbf{S}$ are the graph signal and the Graph Shift Operator (GSO) in our problem, and $\boldsymbol{p}^{cst}$ and $\boldsymbol{p}^{cst}$ represent*

*the constant bias, which does not change with GSO and GS changing.*

**Proof 2** *See the proof in the appendix.*

Note that the GSO in (20) is a valid Laplacian matrix. We emphasize that this linearization is different from the existing DC linearization [37] for the three-phase unbalanced power flow equations and that the Laplacian matrix $\hat{\mathbf{B}}$ is a modified susceptance matrix $\mathbf{B}$.

### *B. GCN and GRN*

Power systems are dynamic systems with time-varying voltage phasors. In order to fuse features from both spatial

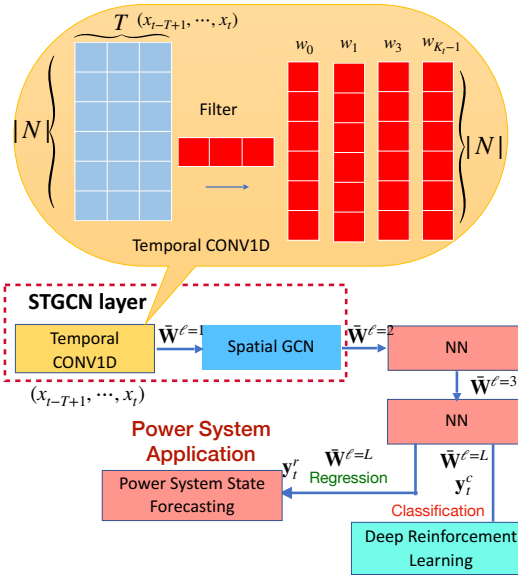



Fig. 2: GCN structure achieved by the CNN and GCN blocks.

*1) GCN:* As shown in Fig. 2, the temporal convolutional layer contains a 1-D CNN with a width-$T$ kernel with $K_t$ output channels. The convolutional kernel $\mathbf{\Gamma} \in \mathbb{R}^{T\times K_t}$ is designed to map the input $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times T}$ into a output graph signal with $C_t$ channels $\bar{\mathbf{X}} \in \mathbb{R}^{|\mathcal{V}|\times K_t}$.

Based on (10), the graph signal $\boldsymbol{w}_t^c$ from the feature extraction layer is:

$$\boldsymbol{w}_t^c = \mathrm{ReLU}\left[\sum_{\tau=0}^{K_t-1}\left(\sum_{k=0}^{K} h_{k,\tau}\mathbf{S}^k\bar{\boldsymbol{x}}_{t-\tau}\right)\right], \tag{21}$$

where $h_{k,\tau}$ is a trainable parameter, which is a scalar. Here, $\mathbf{S}$ in (21) refers to GSO in (20) and $\bar{\boldsymbol{x}}_{t-\tau}$ in (21) refers to GS in (20). Followed by the feature extraction layer (21), the remaining hidden layers $\ell \in \{1, \cdots, L-1\}$ are analogous to those of a fully connected neural network:

$$\boldsymbol{w}_{t,\ell+1}^c = \mathrm{ReLU}\left(\Theta_\ell \cdot \boldsymbol{w}_{t,\ell}^c\right), \quad L-1 \geq \ell \geq 1. \tag{22}$$

For the output layer $L$, the regression samples are:

$$\boldsymbol{y}_t = \tanh\left(\Theta_L \cdot \boldsymbol{w}_{t,L}^c\right), \tag{23}$$

where $\boldsymbol{y}_t$ is the regression targets and $\Theta_\ell, \forall \ell = 1, \cdots, L$ is the trainable matrix. Finally, the multi-layer GCN learning function is:

$$\boldsymbol{y}_t = \Phi^c(\mathbf{X}_t, \mathbf{S}, \theta), \tag{24}$$

where $\theta \triangleq \{(\Theta_\ell, \Theta_k, h_{k,t})|\forall \ell, \forall k\}$ represent the trainable parameters and $\mathbf{X}_t = [\boldsymbol{x}_{t-T+1}, \cdots, \boldsymbol{x}_t]$. Here, we have omitted the bias term to unburden the notation, but they are present in the trainable model we use.

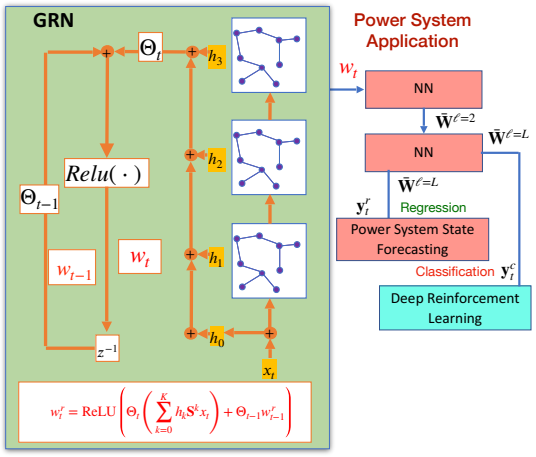



Fig. 3: GRN structure achieved by the RNN and GCN blocks.

*2) GRN:* RNNs are systems that exploit recurrence to learn dependencies in sequences of variable length. Next, we adapt the operations performed by RNNs to take the graph structure into account when dealing with graph processes as follows:

$$\boldsymbol{w}_t^r = \mathrm{ReLU}\left[\Theta_t\left(\sum_{k=0}^{K} h_k \mathbf{S}^k \boldsymbol{x}_t\right) + \Theta_{t-1}\boldsymbol{w}_{t-1}^r\right], \tag{25}$$

where we also omit the biased term to unburden the notation. Likewise, $\mathbf{S}$ in (25) refers to GSO in (20) and $\bar{\boldsymbol{x}}_{t-\tau}$ in (25) refers to GS in (20). The remaining hidden and output layers layers of GRN are:

$$\begin{aligned} \boldsymbol{w}_{t,\ell+1}^r &= \mathrm{ReLU}\left(\Theta_\ell \cdot \boldsymbol{w}_{t,\ell}^r\right), \quad L-1 \geq \ell \geq 1, \\ \boldsymbol{y}_t &= \tanh\left(\Theta_L \cdot \boldsymbol{w}_{t,L}^r\right), \end{aligned} \tag{26}$$

where $\boldsymbol{y}_t = \Phi^r(\mathbf{X}_t, \mathbf{S}, \theta)$ is defined as in the multi-layer GRN, $\theta \triangleq \{(\Theta_\ell, \Theta_t, h_{k,t})|\forall \ell, \forall k\}$ represent the trainable parameters. Similar to our description of GCN, we illustrate the proposed GRN with $K = 3$ on the middle side of Fig. 3. It shows that the graph signals are processed through GCN to capture spatial features of the signal, and then processed by an RNN to capture the temporal correlation. Both GCN and GRN architectures capture the spatiotemporal correlation of voltage phasors, but they have unique advantages. In (23), the proposed GCN model handles the time convolutions via a CNN that allows to use GPUs to accelerate the computations during training. In contrast, the GRN has long memory built in due to the feedback connections with $\boldsymbol{w}_t^r$ and $\boldsymbol{w}_{t-1}^r$ in (25). Hence, GRN is best suited for environments driven by state equations.

### *C. GSO for Partial Observation*

It is very useful for GCN architectures to be able to accept an input that does not include the complete information about the state, because of lacking measurements or for better scalability. In this subsection, we provide the correct GSO for a down-sampled graph signal as an input of a reduced order GCN. Let $\boldsymbol{v}_{\mathcal{M}}$ (time index $t$ is ignored for simplicity) be the

down-sampled voltage graph signal where $\mathcal{M} \in \mathcal{N}$ in the set of node indices of the corresponding buses. We leverage the result in [4], which is summarized in the following lemma:

**Lemma 2 ([4, Lemma 1])** *To define the GSO with respect to the reduced-graph of $\mathcal{M}$, denoted by $\mathbf{S}_{red,\mathcal{M}}$, let us partition the grid GSO $\mathbf{S}$ as follows:*

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{\mathcal{M},\mathcal{M}} & \mathbf{S}_{\mathcal{M},\mathcal{M}^c} \\ \mathbf{S}_{\mathcal{M},\mathcal{M}^c}^\top & \mathbf{S}_{\mathcal{M}^c,\mathcal{M}^c} \end{bmatrix}. \tag{27}$$

*From Ohm's law, it follows that the samples of the state $\boldsymbol{v}_{\mathcal{M}}$ are such that:*

$$\boldsymbol{v}_{\mathcal{M}} = \mathcal{H}(\mathbf{S}_{red,\mathcal{M}})\boldsymbol{\kappa}, \tag{28}$$

*where $\mathbf{S}_{red,\mathcal{M}}$ is the Schur complement of the block $\mathbf{S}_{\mathcal{M}^c\mathcal{M}^c}$ (which is the Kron-reduction of $\mathbf{S}$), i.e.,*

$$\mathbf{S}_{red,\mathcal{M}} = \mathbf{S}_{\mathcal{M}\mathcal{M}} - \mathbf{S}_{\mathcal{M}\mathcal{M}^c}\mathbf{S}_{\mathcal{M}^c\mathcal{M}^c}^{-1}\mathbf{S}_{\mathcal{M}\mathcal{M}^c}^\top.$$

**Proof 3** *The statement follows from the observation that* (28) *holds since from Ohm's law:*

$$\boldsymbol{v}_{\mathcal{M}} = \overbrace{\mathbf{S}_{red,\mathcal{M}}^{-1}}^{\mathcal{H}(\mathbf{S}_{red,\mathcal{M}})} \overbrace{[\mathbb{I}_{|\mathcal{M}|} - \mathbf{S}_{\mathcal{M}\mathcal{M}^c}\mathbf{S}_{\mathcal{M}^c\mathcal{M}^c}^{-1}](\mathbf{S}(\mathbf{S}^{-1})\boldsymbol{i})}^{\boldsymbol{\kappa}}, \tag{29}$$

*where $\mathbb{I}$ is an identity matrix. The equation establishes a generative graph filter model, with GSO $\mathbf{S}_{red,\mathcal{M}}$ for the decimated voltage phasors, supporting such GSO choice.*

## IV. GCN and GRN Applications

Through two applications, in this section we illustrate how our framework can be used to advance AI for grid data.

### A. Power System State Estimation and Forecasting

An important contribution of our design is its capability to take inputs that contain only a subset $\mathcal{M}$ of state variables. It is natural to expect that the performance of the PSSE and PSSF is affected by the subset $\mathcal{M}$ where PMUs are installed. This is why we provide an optimized criterion to select $\mathcal{M}$ leveraging GSP sampling theory. Let the GFT basis corresponding to the first dominant $k$ graph frequencies be $\mathbf{U}_{\mathcal{K}}$. As shown in [4] the best $\mathcal{M}$ is one-to-one with the subset of rows of $\mathbf{U}_{\mathcal{K}}$ with minimum correlation. Let $\mathcal{F}_{\mathcal{M}}$ be what is called the *vertex limiting operator* i.e. the matrix such that $\mathcal{F}_{\mathcal{M}} = \boldsymbol{\mathcal{Q}}_{\mathcal{M}}\boldsymbol{\mathcal{Q}}_{\mathcal{M}}^\top$, where $\boldsymbol{\mathcal{Q}}_{\mathcal{M}}$ has columns that are the coordinate vectors pointing to each vertex/node in $\mathcal{M}$. Mathematically, the optimal placement can be sought by maximizing the smallest singular value, $\max_{\mathcal{F}_{\mathcal{M}}} \varpi_{\min}(\mathcal{F}_{\mathcal{M}}\mathbf{U}_{\mathcal{K}})$, of the matrix $\mathcal{F}_{\mathcal{M}}\mathbf{U}_{\mathcal{K}}$. Such choice amounts to the selection of rows of $\mathbf{U}_{\mathcal{K}}$ that are as uncorrelated as possible, because the resulting matrix $\mathcal{F}_{\mathcal{M}}\mathbf{U}_{\mathcal{K}}$ has the highest conditional number [38].

After choosing the best location of PMUs $\mathcal{M}$ by the aforementioned method, we have the sub-sampled measurement $\boldsymbol{z}_t$. Let $\mathcal{M}$ denote the set of available measurement buses and $\mathcal{U}$ denote the set of unavailable ones. Therefore, (12) can be written as:

$$\underbrace{\begin{bmatrix} \hat{\boldsymbol{i}}_{\mathcal{M}} \\ \hat{\boldsymbol{v}}_{\mathcal{M}} \end{bmatrix}}_{\boldsymbol{z}_t} = \underbrace{\begin{bmatrix} \boldsymbol{Y}_{\mathcal{M}\mathcal{M}} & \boldsymbol{Y}_{\mathcal{M}\mathcal{U}} \\ \mathbb{I}_{|\mathcal{M}|} & \mathbf{0} \end{bmatrix}}_{\boldsymbol{H}} \underbrace{\begin{bmatrix} \boldsymbol{v}_{\mathcal{M}} \\ \boldsymbol{v}_{\mathcal{U}} \end{bmatrix}}_{\boldsymbol{x}_t} + \boldsymbol{\varepsilon}_t, \tag{30}$$

where $\boldsymbol{\varepsilon}_t$ is a vector of measurement noise. Our task is to estimate the voltage phasors at the present time and forecast the future voltage phasors by the GCN and GRN methods. The time-series voltage phasor forecasting problem is modeled as predicting the most likely voltage phasors in the next $H$ time steps given the previous $T$ sub-sampled observation as

$$\boldsymbol{x}'_{t+H} = \arg\max_{\boldsymbol{x}_{t+H}} \log P\left(\boldsymbol{x}_{t+H} \mid \boldsymbol{z}_{t-T+1}, \ldots, \boldsymbol{z}_t\right),$$

where $\boldsymbol{z}_t \in \mathbb{R}^{2|\mathcal{M}|}$ is an observation vector of $|\mathcal{M}|$ measurements for both voltage and current phasors at time step $t$, each element of which records the historical observation for a bus.

*1) Sparse PMU Measurements:* First of all, we need to recover the voltage phasors $\boldsymbol{x}_t$ based on $\mathcal{M}$ measurements, i.e. $\boldsymbol{z}_t = \left[\hat{\boldsymbol{i}}_{\mathcal{M}}, \hat{\boldsymbol{v}}_{\mathcal{M}}\right]^\top$ by solving the regularized least square problem:

$$\min_{\boldsymbol{x}_t} \|\boldsymbol{z}_t - \boldsymbol{H}\boldsymbol{x}_t\|_2^2 + \mu_1(\boldsymbol{x}_t^H\mathbf{S}\boldsymbol{x}_t) \tag{31}$$

where $\mu_1$ is positive. Note that the regulation term $\mu_1(\boldsymbol{x}_t^H\mathbf{S}\boldsymbol{x}_t)$ can be used in such cases to not only improve the generalizability of the model, but also allows to uniquely determine the solution even though the number of variables in the linear system exceeds the number of observations. The closed-form solution of (31) is:

$$\hat{\boldsymbol{x}}_t = \left(\boldsymbol{H}^H\boldsymbol{H} + \mu_1\mathbf{S}\right)^\dagger \boldsymbol{H}^H\boldsymbol{z}_t, \tag{32}$$

where $\hat{\boldsymbol{x}}_t$ is the estimated voltage phasor and where $(\cdot)^\dagger$ denotes the pseudo-inverse. The complete algorithm is below:

**Algorithm 1:** Voltage Phasor Forecasting

1. We collect $T$ historical sub-sampled measurements $\boldsymbol{z}_{t-T+1}, \ldots, \boldsymbol{z}_t$;
2. We utilize (32) to obtain the estimated full observations $\hat{\mathbf{X}} = [\hat{\boldsymbol{x}}_{t-T+1}, \ldots, \hat{\boldsymbol{x}}_t]$;
3. The loss function of the GCN or GRN function for voltage phasor prediction is written as

$$\mathcal{L}(\Phi,\theta) = \sum_t \Big\{ \|\boldsymbol{y}_t - \boldsymbol{x}_{t+H}\|^2 + \mu_2 \left\| \hat{\boldsymbol{v}}_{t+H,\mathcal{M}} \circ \hat{\boldsymbol{i}}^*_{t+H,\mathcal{M}} - \left[\boldsymbol{y}_t \circ (\mathbf{S}\boldsymbol{y}_t)^*\right]_{\mathcal{M}} \right\|^2 \Big\}, \tag{33}$$

where $\boldsymbol{x}_{t+H}$ is the ground truth voltage phasor in the next $H$ time step, $(\cdot)^*$ denotes the conjugate operator and $\boldsymbol{y}_t = \Phi(\hat{\mathbf{X}}_t, \mathbf{S}, \theta)$ is the predicted target to approximate the ground-truth regression target $\Re(\boldsymbol{x}_{t+H})$ and $\Im(\boldsymbol{x}_{t+H})$, where $\Phi$ could be either $\Phi^c$ in (24) or $\Phi^r$ in (26). Note that $H = 0$ is the voltage phasor estimation, and $H \geq 1$ is the voltage phasor forecasting;

Note that the regularization term in (33) favors voltage phasor forecasts that minimize the sum of the absolute value of the apparent power injections. After training, we use $\Phi(\hat{\mathbf{X}}_t, \mathbf{S}, \theta)$ to forecast $\boldsymbol{x}_{t+H}$ given the observations $\boldsymbol{z}_{t-T+1}, \ldots, \boldsymbol{z}_t$. We also emphasize that Algorithm 1 requires the GCN or GRN retraining for different time windows $H$.

After obtaining $\boldsymbol{x}_{t+H}$, we can also map voltage phasors to power injections, and then estimate the total fuel costs. In particular, power injections $\boldsymbol{s}_{t+H}$ is estimated by $\boldsymbol{s}_{t+H} =$

$\boldsymbol{x}_{t+H} \circ (\mathbf{S}\boldsymbol{x}_{t+H})^*$. With the forecasting demand $\boldsymbol{p}^d_{t+H}$ at time $t+H$, we can obtain the power generators $\boldsymbol{p}^g_{t+H} = \Re(\boldsymbol{s}_{t+H}) + \boldsymbol{p}^d_{t+H}$. Therefore, the fuel costs are expressed by $(\boldsymbol{a} \circ \boldsymbol{a})^\top \boldsymbol{p}^g_{t+H} + \boldsymbol{b}^\top \boldsymbol{p}^g_{t+H} + \mathbb{1}^\top \boldsymbol{c}$, where $\boldsymbol{a}$, $\boldsymbol{b}$ and $\boldsymbol{c}$ are the fuel cost parameters (that are provided by Matpower [39].)

*2) PMU Measurements with Bad Data Injection:* In this part, we investigated the robustness of the proposed GCN and GRN against bad data injection. We consider bad data due to a stealth false data injection attack, which is hard to detect because it does not produce higher MSE residuals than compared to good measurements. In a stealth attack, the attacker manipulates both current and voltage phasor measurements on a subset buses $\mathcal{C}$ by introducing a perturbation:

$$\delta \boldsymbol{x}_t^\top = \begin{bmatrix} \delta \boldsymbol{x}_{\mathcal{C}}^\top & \mathbf{0}^\top_{|\mathcal{P}|+|\mathcal{U}|} \end{bmatrix}, \text{ such that } \boldsymbol{Y}_{\mathcal{P}\mathcal{C}} \delta \boldsymbol{x}_{\mathcal{C}} = \mathbf{0}, \mathcal{C} \subset \mathcal{M}$$

where $\mathcal{P}$ is the set of uncompromised sen
This requires special conditions and place
tall and does not have full column-rank fo
of attacker $\mathcal{C}$. Therefore, the received da
injection have the structure:

$$\tilde{\boldsymbol{z}}_t = \boldsymbol{z}_t + \overbrace{\boldsymbol{H}\delta \boldsymbol{x}_t}^{\delta \boldsymbol{z}_t} + \boldsymbol{\varepsilon}_t.$$

where $\tilde{\boldsymbol{z}}_t$ is the polluted PMU measuren
injection, $\boldsymbol{z}_t$ is original measurements, $\delta$
data, and $\boldsymbol{\varepsilon}_t$ is the measurement noise.
bad data measurements $\tilde{\boldsymbol{z}}_t$, we conduct t
Algorithm 1 to test the robustness of the
GRN against the bad measurements.

*3) Sparse AMI Measurements:* We
vanced Metering Infrastructure (AMI) m
are widely deployed in the power systems
measuring apparent power, voltage mag
magnitude are widely deployed in power
[40]. To refer to them, we use the following notation: $\boldsymbol{v}^2$ as the vector of voltage magnitudes squared, $\boldsymbol{i}^2$ as the vector of current magnitude squared, and $\boldsymbol{s}$ as the vector of net apparent power injection, which can be written as follows:

$$|\boldsymbol{v}|^2 = \text{diag}(\boldsymbol{v}\boldsymbol{v}^H), |\boldsymbol{i}|^2 = \text{diag}(\mathbf{S}\boldsymbol{v}\boldsymbol{v}^H\mathbf{S}^H), \boldsymbol{s} = \text{diag}(\boldsymbol{v}\boldsymbol{v}^H\mathbf{S}^H). \quad (36)$$

There is a preprocessing step to prepare the input to GCN or GRN, that extracts a coarse estimate of the voltage phasors $\boldsymbol{x}_t$ based on $\mathcal{M}$ AMI measurements, i.e. $\boldsymbol{z}_t = \left[[|\boldsymbol{v}_t|^2]_{\mathcal{M}}, [|\boldsymbol{i}_t|^2]_{\mathcal{M}}, [\boldsymbol{s}_t]_{\mathcal{M}}\right]^\top$ by training a simple fully connected neural network mapping from the AMI measurement to voltage phasors, i.e. $\hat{\boldsymbol{x}}_t = f_{NN}(\boldsymbol{z}_t)$. This step is to estimate the voltage phasors roughly, which is the pre-processing step of GCN and GRN. This step is analogous to that done for the PMU measurements, by training the GCN and GRN models to forecast $\boldsymbol{x}_{t+H}$ given the observations $\hat{\boldsymbol{x}}_{t-T+1}, \ldots, \hat{\boldsymbol{x}}_t$.

### B. Deep Reinforcement Learning Control of Smart Inverters

Following [17, 41] we consider reactive power support for smart inverters operating on a per-phase basis, where each smart inverter is installed at bus $n_\phi$. The reactive power support of the smart inverter depends on the nominal per-phase capacity, denoted by $s_{n_\phi}$. Specifically, the range of possible reactive power, $q_{n_\phi}$ for the smart inverter is:

$$|q_{n_\phi}| \leq \bar{q}_{n_\phi} \triangleq \sqrt{s^2_{n_\phi} - p^2_{n_\phi}} \quad (37)$$

where $\bar{q}_{n_\phi} = \sqrt{s^2_{n_\phi} - p^2_{n_\phi}}$ denotes the maximum reactive power of this smart inverter installed at bus $n_\phi$. Here, we define the control variable as $a_{n_\phi} \in [-1, 1]$, and the reactive power injected into the distribution network is $q_{n_\phi} = a_{n_\phi} \bar{q}_{n_\phi}$.

*1) Methodology:* The DRL training is aimed at learning the parameters of GCN (or GRN) encoding the optimum stochastic policy function for mapping the voltage phasors measurements onto the control variables $a_{n_\phi}$ that regulate the voltage magnitude. The interaction between the agent and

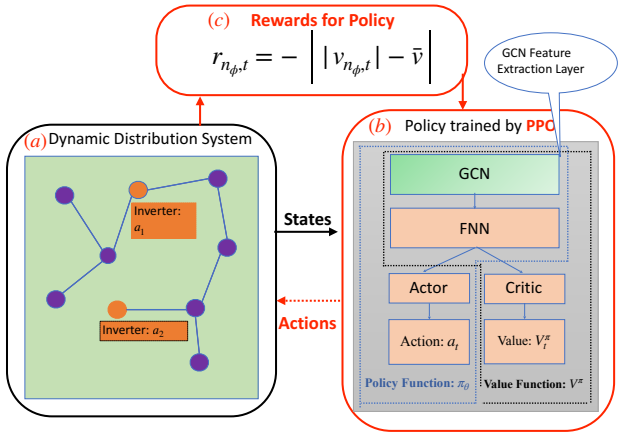



Fig. 4: Overview of GCN-based DRL for Voltage Control.

The overview of the problem statement and methodology is shown in Fig. 4. During the training stage, as illustrated in Fig. 4($a$), we perform realistic simulation of the distribution system with multiple smart inverters installed, to generate using this *digital twin of the system* the vector of states for training our DRL algorithm. The distribution system simulations provide the input to compute the instantaneous rewards of the policy (as shown in Fig. 4($b$)) during the training of the policy, computed with the neural network architecture in Fig. 4($b$). Moreover, both policy function and value function are trained by the Proximal Policy Optimization (PPO) algorithm, and share the same spatio-temporal GCN feature extraction layers. As a feedback, the policy gives an action to provide the reactive power support to the distribution system. After the policy is well trained, the parameters of the neural network in Fig. 4($b$) are fixed and the policy can be deployed in the real system.

*a) State and Action:* The tuple of actions for the Volt-VAR control of the inverters in the bus set $\mathcal{N}_s$ is denoted by:

$$\boldsymbol{a} = [a_1, \cdots, a_{n_\phi}, \cdots, a_{|\mathcal{N}_s|}], a_{n_\phi} \in [-1, 1], \quad (38)$$

and thus the corresponding reactive power injections are:

$$\boldsymbol{q}_{n_\phi} = [q_1, \cdots, q_{n_\phi}, \cdots, q_{|\mathcal{N}_s|}], \text{where } q_{n_\phi} = a_{n_\phi} \bar{q}_{n_\phi}, \quad (39)$$

where $|\mathcal{N}_s|$ represents the number of smart inverters, $a_i$ denotes the control action on the $i$th smart inverter. The vector $\boldsymbol{a}$, output of the GCN (or GRN) approximating the optimum policy, is a function of the three-phase voltages at all, or part, of the buses of the distribution system, which constitute the observation and are the input of the GCN-DRL or GRN-DRL. The state vector/observation is $\boldsymbol{x} = [\boldsymbol{\varphi}; |\boldsymbol{v}|]^\top$, where $\boldsymbol{\varphi}$ is the vector of re-centered voltage phases and $|\boldsymbol{v}|$ is the vector of voltage magnitudes. The output of GCN-DRL or GRN-DRL is the action vector $\boldsymbol{a}_t$ corresponding to $\boldsymbol{y}_t$ in (24) or (26).

*b) Reward:* The regret is defined as the magnitude of voltage deviation from the reference at bus $n_\phi$ at time $t$ as follows [19, 24]:

$$r_{n_\phi,t} = -\left||v_{n_\phi,t}| - \bar{v}\right|, n_\phi \in \mathcal{N}_s, \tag{40}$$

where $\bar{v}$ denotes the desired voltage magnitude (i.e., 1 p.u.), and $|v_{n_\phi,t}|$ is the measured voltage magnitude on phase $n_\phi$.

*c) Objectives of DRL:* In this application, $\boldsymbol{a}_t = \Phi(\mathbf{X}_t, \mathbf{S}, \theta)$ denotes a stochastic policy that models the probability distribution of $\boldsymbol{a}_t \in \mathcal{A}$ given a sequence of observations $\mathbf{X}_t$. The goal of each agent is to find a policy, which maximizes its expected discounted return:

$$\Phi(\mathbf{X}_t, \mathbf{S}, \theta) \in \arg\max J(\pi) = \mathbb{E}_{\varsigma\sim\Phi}\left[\sum_{t=0}^{T} \gamma^\top r_t\right], \tag{41}$$

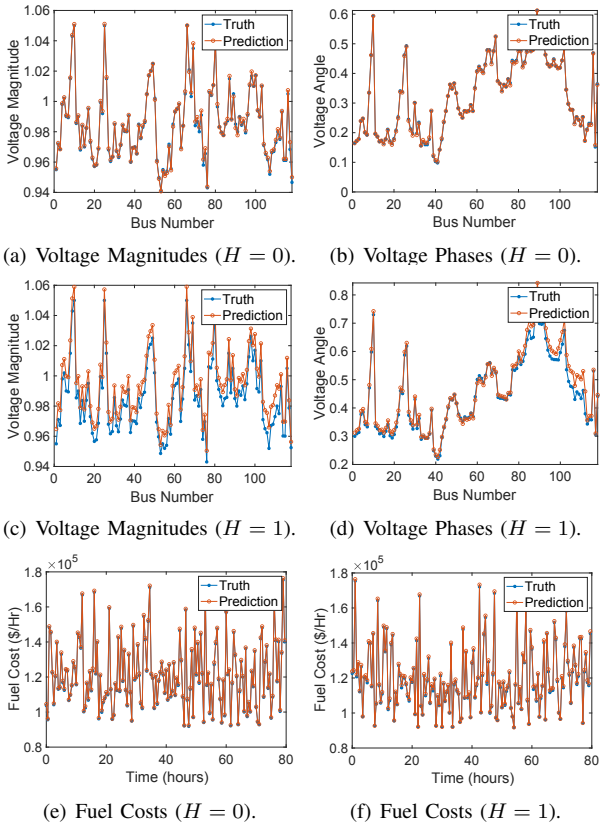
where $\varsigma$ is the trajectory generated by policy $\Phi(\mathbf{X}_t, \mathbf{S}, \theta)$, i.e., the action $\boldsymbol{a}_t$ is taken according to policy $\Phi(\mathbf{X}_t, \mathbf{S}, \theta)$, $r_t$ represents rewards at time $t$. The parameter $\gamma \in (0, 1)$ is the discounting factor, discounting future rewards.

## V. Experimental Results

In this section, we perform numerical experiments adopting the IEEE 118-bus transmission network and the 123-bus feeder distribution system to validate the proposed GCN and GRN frameworks for power system state estimation and forecasting, and DRL-based voltage control. In both cases, for the formulation and training of the GCN and GRN architectures we relied on PyTorch 1.10.0. All algorithms are executed on a 64-bit Windows with 2.6 GHz Six-Core Intel Core i7 and a total of 16 GB RAM, and NVIDIA GeForce RTX 2060.

### A. Power System State Estimation and Forecasting

*1) Experimental setup:* For the first application, we use realistic load time-series from the Texas grid and use Matpower to compute the optimal power flow solutions to obtain the voltage phasors for the IEEE 118-bus system. All the tests $T = 10$ hours as the historical time window, i.e. 10 observed data points, to forecast voltage phasors in the next hours $H = 1, 2, 3, 4, 5$ where $H = 0$ is a PSSE problem that estimates the complete voltage phasors $\boldsymbol{x}_t$ given $\boldsymbol{z}_{t-T+1}, \ldots, \boldsymbol{z}_t$. We compare the proposed GCN and GRN with other NNs. In particular, the benchmark algorithm FNN has 4 layers with 512 neurons each layer. Another benchmark algorithm CNN has three hidden layers with the 32, 64 and 32 output channels and one fully connected NNs, respectively. The third benchmark GNN [14, 42] is a 1st-order approximation Chebyshev GCN [33] with the adjacency matrix as GSO. The fourth benchmark RNN utilizes the RNN as the feature extraction layer, and then FNNs as the hidden and output layers.




(a) Voltage Magnitudes ($H = 0$). (b) Voltage Phases ($H = 0$).
(c) Voltage Magnitudes ($H = 1$). (d) Voltage Phases ($H = 1$).
(e) Fuel Costs ($H = 0$). (f) Fuel Costs ($H = 1$).

Fig. 5: An example of PSSE and PSSF by GCN for the IEEE 118-bus system.

TABLE I: PMU Installed Buses in the Transmission Network

| Systems | Bus Name |
|---|---|
| 118-bus with PMUs | 14, 117, 72, 86, 43, 67, 99, 87, 16, 33, 112, 28, 98, 111, 53, 97, 1 42, 107, 48, 22, 46, 13, 24, 101, 44, 73, 109, 29, 20, 91, 26, 84, 10 52, 57, 76, 115, 39, 74, 104, 93, 79, 35, 6, 18, 88, 60, 116, 55, 58 68, 64, 7, 50, 103, 75, 78, 83, 69. |

With the predicted voltage phasors, we further utilize the power flow solver, i.e. Matpower, to obtain the feasible power generations, and then calculate their corresponding fuel costs. The evaluation metrics for comparison includes mean square error (MSE) between the predicted and ground-truth voltage phasors and Mean Absolute Percentage Error (MAPE) between the predicted and optimal fuel costs. As shown in Table I, we choose the number of sensor placements $|\mathcal{M}| = 60$ and place them so as to maximize $\max_{\mathcal{F}_\mathcal{M}} \varpi_{\min}(\mathcal{F}_\mathcal{M}\mathbf{U}_\mathcal{K})$. Besides, through numerous simulations for the hyperparameter tuning, we choose $\mu_1$ = 1$e$-6 and $\mu_2$ = 1$e$-3 for all benchmarks.

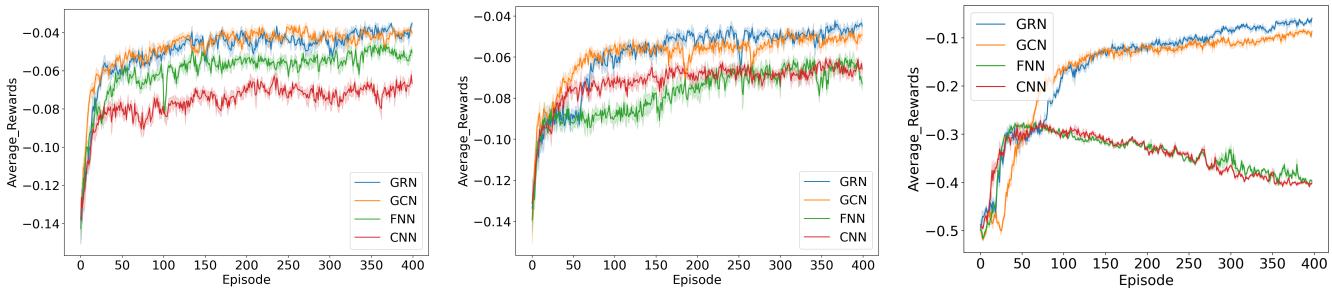
TABLE II: MSE: PSSF for Voltage Phasors in Transmission Networks

| Future (Hours) | $H = 0$ | $H = 1$ | $H = 2$ | $H = 3$ | $H = 4$ | $H = 5$ |
|---|---|---|---|---|---|---|
| FNN | 1.1217$e$-4 | 4.7697$e$-4 | 7.7697$e$-4 | 5.4458$e$-4 | 9.2263$e$-4 | 8.8066$e$-3 |
| CNN | 2.8069$e$-4 | 4.4070$e$-4 | 1.7169$e$-3 | 1.7238$e$-2 | 1.6815$e$-2 | 1.6758$e$-2 |
| RNN | 8.8034$e$-4 | 8.7639$e$-4 | 7.6329$e$-4 | 7.7802$e$-4 | 7.1659$e$-4 | 7.8581$e$-4 |
| $1^{st}$GNN[42] | 7.8849$e$-4 | 7.2899$e$-4 | 7.5874$e$-4 | 8.6065$e$-4 | 7.5357$e$-4 | 8.1019$e$-4 |
| GCN | **6.1381*e*-5** | **1.0080*e*-4** | **2.0157*e*-4** | 2.6714$e$-4 | 3.2469$e$-4 | 2.2308$e$-4 |
| GRN | 7.2153$e$-5 | 1.8058$e$-4 | 2.4738$e$-4 | **2.1372*e*-4** | **2.9829*e*-4** | **2.1137*e*-4** |

*2) PSSE and PSSF Results:* Tables II and III show the results of GCN and GRN and various baselines on the IEEE

This article has been accepted for publication in IEEE Transactions on Smart Grid. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TSG.2023.3239740






(a) Training curves of 3 small inverters based on full observations. (b) Training curves of 3 small inverters based on partial observations. (c) Training curves of 6 small inverters based on full observations.

Fig. 6: (a) and (b) have 3 smart inverters and (c) has 6 smart inverters. (a) and (b) illustrate the learning (training) curves of the GCN-DRL and GRN-DRL algorithms for voltage magnitude regulations with full and partial observations, respectively.

TABLE III: MAPE: PSSF for Fuel Costs in Transmission Networks

| Future (Hours) | $H=0$ | $H=1$ | $H=2$ | $H=3$ | $H=4$ | $H=5$ |
|---|---|---|---|---|---|---|
| FNN | 2.0602% | 2.1338% | 6.5671% | 4.5503% | 6.9320% | 7.0781% |
| CNN | 2.0683% | 9.1396% | 9.8635% | 16.4026% | 28.7120% | 58.2591% |
| RNN | 3.0049% | 1.9634% | 2.3701% | 2.7542% | 2.1048% | 2.4831% |
| $1^{st}$GNN[42] | 3.7144% | 2.8667% | 2.2925% | 2.3752% | 2.3347% | 4.7779% |
| GCN | **0.3838%** | **1.0809%** | **0.9991%** | **1.2734%** | 2.0542% | 2.5816% |
| GRN | 0.7065% | 0.7533% | 1.6357% | 1.6365% | **1.5054%** | **1.8188%** |

118-bus system experiments described above. The results illustrate that both GCN and GRN achieve the best performance. In particular, $H=0$ is the PSSE problem, and the MSE of (32) for estimation is 2.3708$e$-4. While this is a respectable outcome, the supervised GCN and GRN have much smaller error, i.e. 6.1381$e$-5 and 7.2153$e$-5, respectively. Another observation is that the voltage phasors predicted by GCN and GRN could approximate the OPF results with much smaller MAPE, e.g. 0.9991% and 1.6357% at $H=2$, compared with other methods, e.g. 6.5671% of FNN and 9.8635% of CNN. We illustrate examples by the GCN method for the IEEE 118-bus system in Fig. 5 to show the ground-truth fully-observed voltage phases and fuel costs with the predicted ones with $H=0,1$, which shows both the predicted voltage phases and fuel costs are very close to the ground-truth. We also observe that the performance of GRN and GCN are similar for short-time forecasting, while GRN outperforms GCN in the long-time forecasting task (e.g. 1.5054% compared with 2.0542% of MAPE for $H=4$). Here, we emphasize that the reason why GCN and GRN have very small MSE and MAPE for forecasting is that voltage phasors have a constrained (low variance) distribution, i.e., power flow constraints, which has the support of the graph. Therefore, the GCN is most effective at internalizing the distribution and approximating the Bayesian MSE estimator by capturing the spatiotemporal correlations.

*3) Computation Time:* In Fig. 7, we compare the training time for PSSE and PSSF with different neural network methods. The result shows that RNN-based methods, including RNN and GRN, still suffer from time-consuming iterations, gate mechanisms, and slow response to dynamic changes. The CNN-based methods, including CNN and GCN, allow for fast training, have a simpler structures, and no dependency constraints from previous steps. Besides, the proposed GCN model handles the time convolutions via a CNN that allows to use GPUs to accelerate the computations during training. On the other hand, although GRN requires a long time for training, this method is best suited for environments driven by state equations due to long memory built in.

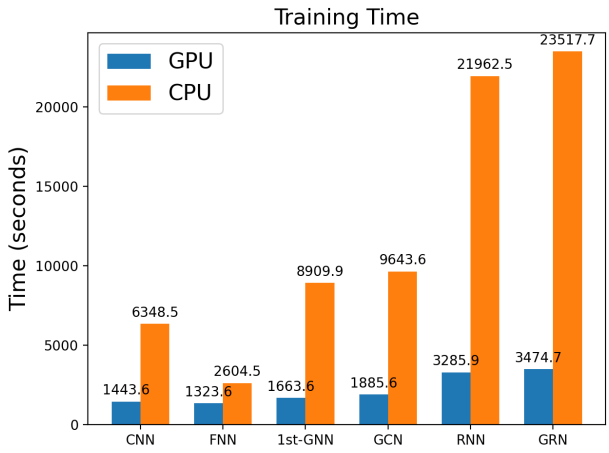
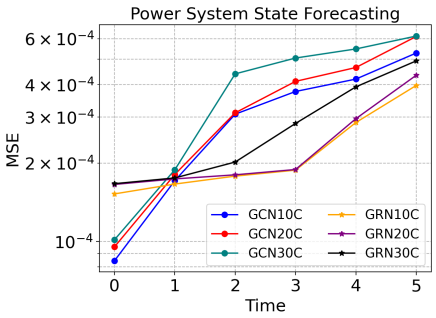



Fig. 7: Training Time Comparison.

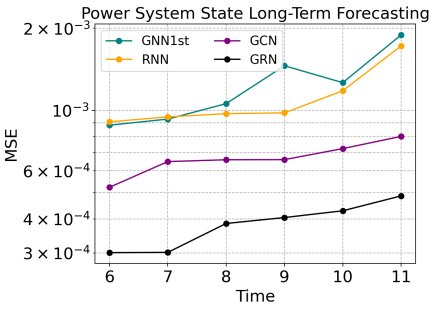
*4) Robustness to Bad Data:* In this simulation, we inject the bad data $\delta \boldsymbol{z}_t$ into the measurement $\tilde{\boldsymbol{z}}_t = \boldsymbol{z}_t + \delta \boldsymbol{z}_t + \boldsymbol{\varepsilon}_t$ in (32). Here, in order to test the proposed GCN and GRN for PSSE and PSSF, we choose the different number of buses attacked from 10 to 30, which is denoted by $|\mathcal{C}|$. The results are shown in Fig. 8, which shows that the proposed GCN is robust enough against the bad data or false data. This is because GCN and GRN has the low-pass GSO to filter the time-series voltage phasors. However, bad data and false data are the high-pass signals.

*5) Long-Time Forecasting Results:* In order to further investigate the long-time forecasting phenomenon, we consider the $H=6,7,8,9,10,11$ for the top four algorithms in Table II and III, i.e., RNN, $1^{st}$GNN, GCN and GRN. The results are shown in Fig. 8, which indicates that GRN is the best algorithm for the long-time forecasting algorithm for voltage phasors.

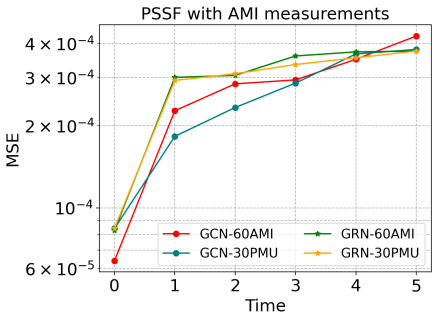
*6) AMI and Sparser PMU Measurements:* We show the results by AMI measurements in Fig. 10. The results show that the proposed strategy enables the GCN and GRN to work on both the *sparse AMI measurements* and *fewer PMU measurements* (only 30 PMUs) than the above cases with 60 PMUs, which have the similar performance with GCN and GRN on the sparse *PMU measurements* (60 PMUs).



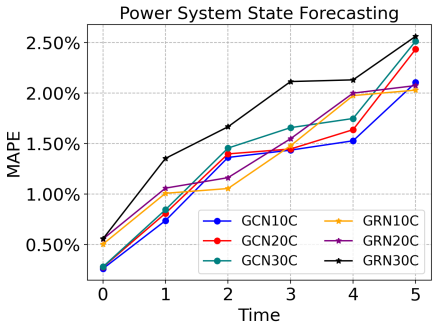

(a) MSE: PSSF for voltage phasors with polluted measurements.




(a) MSE: Power state long-term forecasting for voltage phasors.

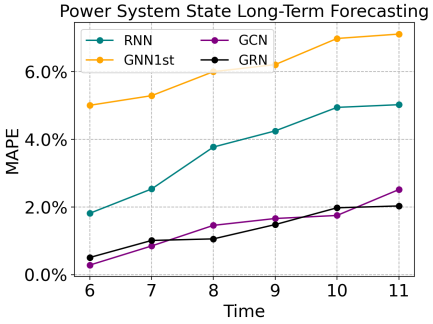



(a) MSE: PSSF for voltage phasors with AMI and fewer PMU measurements.



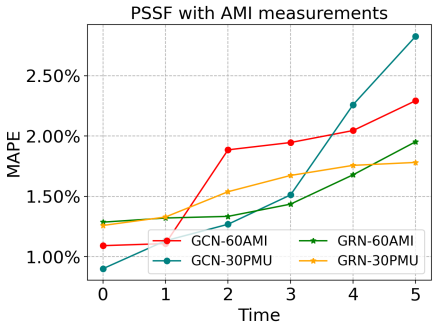

(b) MAPE: PSSF for fuel costs with polluted measurements.

Fig. 8: PSSF with polluted measurements in the IEEE 118-bus system.




(b) MAPE: Power state long-term forecasting for fuel costs.

Fig. 9: Power system state long-term forecasting in the IEEE 118-bus system.




(b) MAPE: PSSF for fuel costs with AMI and fewer PMU measurements.

Fig. 10: PSSF with AMI measurements in the IEEE 118-bus system.

TABLE IV: PMU Installed Buses in the Distribution Network

| Systems | Bus Name |
|---|---|
| 123-bus DG with PMUs | 1.1, 1.2, 1.3, 2.2, 3.3, 7.1, 7.2, 7.3, 4.3, 5.3, 6.3, 8.1, 8.2, 8.3, 10.1, 12.2 13.1, 13.2, 13.3, 9r.1, 14.1, 34.3, 18.1, 18.2, 18.3, 11.1, 15.3, 16.3, 17.3 9.1, 19.1, 150.1, 150.2, 150.3, 150r.1, 150r.2, 150r.3, 149.1, 149.2, 149.3 |

### *B. GCN-DRL and GRN-DRL for Voltage Control*

In this section, we compare the performance of voltage control DRL strategies using GCN and GRN that we propose with benchmark algorithms and study its learning stability in the training phase. In Fig. 6(b) we also validate the efficacy of the reduced GSO **Lemma 2** in Section III-C.

*1) Policy Training:* To validate the advantages of the proposed GCN and GRN over the state of the art, the DRL scheme we showcase is an instance of the popular PPO [43]. We compare the proposed GCN-DRL and GRN-DRL architecture with existing DRL methods for voltage regulation. As PPO outputs are discrete actions, we discretize the actions space $[-1, 1]$ with spacing $0.2$.

*2) Experiment Setup:* The DRL experiments are run on the 123-bus feeder distribution network test case. We use demand data from Austin in the OpenEI[3], and historical PV data for training and testing, with three PV smart inverters installed in the load buses (Buses 51, 53, 60) in Fig. 6(a) and six PV smart inverters (Buses 69, 51, 52, 82, 68, 94) in Fig. 6(c), respectively. We use OpenDSS to estimate the grid state. We set the desired voltage magnitude $\bar{v} = 1$ p.u. We test Lemma 2 in Section III-C and apply the proposed Kron-reduction to design the GSO (see Figure 6(b)). The measurement buses, shown in TABLE IV, are selected according to the algorithm in Section IV-A. In particular, we select 40 phases from 278 phases ($\approx$ 14% of the buses) in the three-phase 123-bus feeder system. Considering that real distribution feeders include thousands of buses, this would bring the cost for PMU measurements systems to reasonable levels. The DRL parameters are as follows. The learning rate is 0.0007. The discounted factor $\gamma$ is 0.99. The PPO clip parameter $\epsilon$ is 0.1, the entropy loss weight is 0.01 and value loss weight is 1. There are 10 spatial and temporal channels for both GCN and GRN layers performing the feature extraction, followed by 512 neurons in an FNN layer followed by the output layer.

*3) DRL regulation results:* The learning curves of GRN-DRL, GCN-DRL, FNN-DRL, and CNN-DRL with the full and partial observations are shown in Fig. 6(a) and Fig. 6(b), respectively. The two figures show the average training reward, where the bands represent the standard deviation over 5 runs. In particular, the results in Fig. 6(a) show that the voltage deviations of GRN and GCN, i.e., $\sum_{n_\phi \in \mathcal{N}_s} |r_{n_\phi}|$, are 0.0332 p.u. and 0.0398 p.u., which outperform FNN and CNN that have 0.0504 p.u. and 0.0657 p.u., respectively. Another observation is that GRN and GCN are competitive in convergence time and performance. With the partial observations (40 out of 278), the results in Fig.6(b) show that $\sum_{n_\phi \in \mathcal{N}_s} |r_{n_\phi}|$ of GCN and GRN converge into 0.0492 p.u. and 0.0469 p.u., respectively. However, $\sum_{n_\phi \in \mathcal{N}_s} |r_{n_\phi}|$ of FNN and CNN converge into 0.0784 p.u. and 0.0697 p.u., respectively. With 6 smart inverters, the test in Fig. 6(c) shows the learning curves of FNN and CNN decreases after 50 episodes, which indicates

[3]https://data.openei.org/data_lakes#Data-Lakes-Datasets

that they trigger the deadly triad of DRL. In contrast, GCN and GRN continue to have excellent performance, demonstrating that they do enhance the stability of DRL.

## VI. Conclusions

In this paper we proposed novel physics-aware GCN and GRN frameworks for single and three-phase power systems. The proposed frameworks are effective in capturing both temporal and spatial features of the voltage phasors data. One of the key ingredients of our method is to derive the GSO from the power flow equations. The proposed architectures are shown to be more effective than conventional NNs in extracting spatio-temporal features from the voltage phasors, in power system state forecasting and reinforcement learning for voltage control applications. We further utilize the kron-reduced network GSO to deal with sparse deployments of PMUs, or simply for scalability. Moreover, we show that even having roughly 14% of the state values measurements leads to excellent performance compared to other benchmarks, i.e., FNN, CNN, RNN and GNN for the aforementioned applications. Future work will extend the real-valued GCN and GRN to the complex-valued graph signals and explore the topic of scalability for large networks.

## Appendix

### A. Proof of Lemma 1

Here we obtain two decoupled real equations describing the dependence between the active and reactive power and the magnitude and phases of the state vector. To do so, we will be using the expansion $e^{jx} = 1 + jx$ for phase terms of the three-phase state sub-vectors in the products $\boldsymbol{v}_n\boldsymbol{v}_n^H$ and $\boldsymbol{v}_n\boldsymbol{v}_m^H$, after re-centering them around the phases of a balanced system. Let $\Psi_n^{(3)} \triangleq \text{diag}([1, e^{-\text{j}2\pi/3}, e^{\text{j}2\pi/3}]^\top)$, $c_c = \cos(\frac{2\pi}{3})$ and $c_s = \sin(\frac{2\pi}{3})$. $\mathbb{1}$ is the all-ones vector and $\mathbb{1}\mathbb{1}^\top$ is the all-ones matrix. In the following, we assume $\Psi_n^{(3)} = \Psi_m^{(3)}$ but we could similarly account for other shifts modeling specific electrical elements, such as transformers. We will make use of the following propositions. With $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{E}$ real square matrices, and $\boldsymbol{a}$ and $\boldsymbol{b}$ are real vectors, the following holds:

**Proposition 2** *(P2) If* $\mathbf{C}$ *and* $\mathbf{E}$ *are diagonal matrices, then* $\mathbf{C}(\mathbf{A}\circ\mathbf{B})\mathbf{E} = \mathbf{A}\circ(\mathbf{C}\mathbf{B}\mathbf{E})$.

**Corollary 1** *(C1) If* $\mathbf{C}$ *and* $\mathbf{E}$ *are diagonal matrices, then* $\mathbf{C}\mathbf{A}\mathbf{E} = \mathbf{A}\circ(\mathbf{C}(\mathbb{1}\mathbb{1}^\top)\mathbf{E}) = \mathbf{A}\circ(\textit{diag}(\mathbf{C})(\textit{diag}(\mathbf{E}))^\top)$.

**Proposition 3** *(P4)* $D(\mathbf{A}\mathbf{B}) = \sum_j(\mathbf{A}\circ\mathbf{B}^\top)_{ij}$.

**Proposition 4** *(P5) If* $\mathbf{B}$ *and* $\mathbf{C}$ *are symmetric,* $D((\mathbf{A}\circ\mathbf{B})\mathbf{C}) = D(\mathbf{A}(\mathbf{B}\circ\mathbf{C})) = D(\mathbf{A}(\mathbf{B}\circ\mathbf{C})^\top)$.

*proof:*

$$D((\mathbf{A}\circ\mathbf{B})\mathbf{C}) = \sum_i((\mathbf{A}\circ\mathbf{B})\circ\mathbf{C}^\top)_{ij} = \sum_i(\mathbf{A}\circ(\mathbf{B}\circ\mathbf{C}^\top))_{ij}$$
$$= \sum_i(\mathbf{A}\circ(\mathbf{B}^\top\circ\mathbf{C}^\top))_{ij} = \sum_i(\mathbf{A}\circ(\mathbf{C}^\top\circ\mathbf{B}^\top))_{ij}$$
$$= \sum_i(\mathbf{A}\circ(\mathbf{C}\circ\mathbf{B})^\top)_{ij} = D(\mathbf{A}(\mathbf{C}\circ\mathbf{B})) = D(\mathbf{A}(\mathbf{B}\circ\mathbf{C})).$$

**Proposition 5** *(P5)* $D(\boldsymbol{a}\boldsymbol{b}^T) = \textit{diag}(\boldsymbol{b})\boldsymbol{a}$ , $D(\mathbf{A}) = D(\mathbf{A}^\top)$.

Now, we are ready to introduce how to design the GSO. We will refer to the specific propositions or corollary in each equation, such as **P2** or **C1**, with blue color. By adding and subtracting from the phase angle in $\boldsymbol{v}_n$, we obtain:

$$\boldsymbol{v}_n = \Psi_n^{(3)}\text{diag}(|\boldsymbol{v}_n|)\begin{bmatrix} e^{\text{j}\varphi_{n_a}} \\ e^{\text{j}\varphi_{n_b}} \\ e^{\text{j}\varphi_{n_c}}\end{bmatrix} = \Psi_n^{(3)}\text{diag}(|\boldsymbol{v}_n|)e^{\text{j}\boldsymbol{\varphi}_n} \quad (42)$$

Therefore, the outer product $\boldsymbol{v}_n\boldsymbol{v}_m^H$ are:

$$\boldsymbol{v}_n\boldsymbol{v}_m^H = \Psi_n^{(3)}\text{diag}(|\boldsymbol{v}_n|)e^{\text{j}(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_m^\top)}\text{diag}(|\boldsymbol{v}_m|)(\Psi_m^{(3)})^H$$
$$= \overbrace{\text{diag}(|\boldsymbol{v}_n|)\Psi_n^{(3)}}^{\mathbf{C}\text{ in }\mathbf{C1}}\Big(\overbrace{\mathbb{1}\mathbb{1}^\top + \text{j}(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_m^\top)}^{\mathbf{A}\text{ in }\mathbf{C1}}\Big) \quad (43)$$
$$\Big[\overbrace{(\Psi_m^{(3)})^H\text{diag}(|\boldsymbol{v}_m|)}^{\mathbf{E}\text{ in }\mathbf{C1}}\Big] \overset{\mathbf{C1}}{=} \big(\mathbb{1}\mathbb{1}^\top + \text{j}(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_m^\top)\big) \quad (44)$$
$$\circ\Big(\text{diag}(|\boldsymbol{v}_n|)\Gamma\text{diag}(|\boldsymbol{v}_m|)\Big) \quad (45)$$

where $\Gamma \triangleq \Psi_n^{(3)}(\mathbb{1}\mathbb{1}^\top)(\Psi_m^{(3)})^H$ and can be expressed as:

$$[\Gamma]_{kn} = e^{\text{j}\frac{2(k-n)\pi}{3}} = [\Gamma_c]_{kn} + \text{j}[\Gamma_s]_{kn},\ k,n\in\{0,1,2\}$$

where it is easy to verify that $\Gamma_c = \Gamma_c^\top$.

### B. Proof of Proposition 1

*1) Active Power GSO:* Next we use the approximation in developing the component relative to the phase term we use the approximation[4] that $|\boldsymbol{v}_n| \approx \mathbb{1}$ and $|\boldsymbol{v}_m| \approx \mathbb{1}$ in (45). With this approximation, we substitute (45) in (15). Therefore, the real part of the first term in (15) is

$$\Re\Big\{-D\Big(\boldsymbol{v}_n\boldsymbol{v}_n^H\big(\frac{\text{j}}{2}\mathbf{B}_{mn}^s + \text{j}\mathbf{B}_{mn}^{(n)}\big)\Big)\Big\} = -D\Big(\Re\Big\{\Big(\big[\text{j}\mathbb{1}\mathbb{1}^\top$$
$$(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_n^\top)\big]\circ\Gamma\Big)\big(\frac{1}{2}\mathbf{B}_{mn}^s + \mathbf{B}_{mn}^{(n)}\big)\Big\}\Big) = D\Big(\Big(\mathbb{1}\mathbb{1}^\top$$
$$\circ\Gamma_s + (\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_n^\top)\circ\Gamma_c\Big)\big(\frac{1}{2}\mathbf{B}_{mn}^s + \mathbf{B}_{mn}^{(n)}\big)\Big) \quad (46)$$

We separate the biased part that does not involve in $(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_n^\top)$ from (46), and define it as:

$$\boldsymbol{p}_n^{inc} \triangleq D\Big(\Gamma_s\big(\frac{1}{2}\mathbf{B}_{mn}^s + \mathbf{B}_{mn}^{(n)}\big)\Big). \quad (47)$$

where $\mathbb{1}\mathbb{1}^\top\circ\Gamma_s = \Gamma_s$. The remaining part of (46) involving $(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_n^\top)$, denoted by $\boldsymbol{p}_n^{in}$, can be expressed as:

$$\boldsymbol{p}_n^{in} \triangleq D\Big(\Big(\overbrace{(\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_n^\top)}^{\mathbf{A}\text{ in }\mathbf{P4}}\circ\overbrace{\Gamma_c}^{\mathbf{B}\text{ in }\mathbf{P4}}\Big)\overbrace{\big(\frac{1}{2}\mathbf{B}_{mn}^s + \mathbf{B}_{mn}^{(n)}\big)}^{\mathbf{B}\text{ in }\mathbf{P4}}\Big) \quad (48)$$
$$\overset{\mathbf{P4}}{=} D\Big((\boldsymbol{\varphi}_n\mathbb{1}^\top - \mathbb{1}\boldsymbol{\varphi}_n^\top)\big(\frac{1}{2}\overbrace{\hat{\mathbf{B}}_{mn}^s}^{\triangleq\Gamma_c\circ\mathbf{B}_{mn}^s} + \overbrace{\hat{\mathbf{B}}_{mn}^{(n)}}^{\triangleq\Gamma_c\circ\mathbf{B}_{mn}^{(n)}}\big)^\top\Big) \quad (49)$$

[4]This is effective but not truly necessary since the multiplication with $\text{diag}(|\boldsymbol{v}_n|)$ could be used as part of the definition of the graph signal.

$$\overset{\text{P5}}{=} \text{diag}\Big(\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\mathbb{1}\Big)\boldsymbol{\varphi}_n-\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\boldsymbol{\varphi}_n. \tag{50}$$

As the Hadamard product commutes and $\Gamma_c$, $\mathbf{B}^{s}_{mn}$ and $\mathbf{B}^{(n)}_{mn}$ are symmetric, $\hat{\mathbf{B}}^{s}_{mn}$ and $\hat{\mathbf{B}}^{(n)}_{mn}$ are symmetric. Replacing $m$ with $n$ and $\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)$ with $\hat{\mathbf{B}}^{(m)}_{mn}$, this result applies also to $\boldsymbol{v}_n\boldsymbol{v}^H_m$. Thus, the real part of the $2^{nd}$ term in (15) is

$$\begin{aligned}&\Re\Big\{-D\Big(\boldsymbol{v}_n\boldsymbol{v}^H_m\ \mathrm{j}\mathbf{B}^{(m)}_{mn}\Big)\Big\}=\\&=D\Bigg(\Big(\Gamma_s+(\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}^\top_m)\circ\Gamma_c\Big)\mathbf{B}^{(m)}_{mn}\Bigg)\end{aligned} \tag{51}$$

Likewise, we separate the bias, that is not part of $(\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}^\top_m)$ from (51), and define it as:

$$\boldsymbol{p}^{ouc}_n\triangleq D\left(\Gamma_s\mathbf{B}^{(m)}_{mn}\right). \tag{52}$$

The remaining part of (51) that involves in $(\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}^\top_m)$, denoted by $\boldsymbol{p}^{in}_n$, can be expressed as:

$$\begin{aligned}&D\Bigg(\Big((\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}^\top_m)\circ\Gamma_c\Big)\mathbf{B}^{(n)}_{mn}\Bigg)\\&=\text{diag}\left(\hat{\mathbf{B}}^{(m)}_{mn}\mathbb{1}\right)\boldsymbol{\varphi}_n-\hat{\mathbf{B}}^{(m)}_{mn}\boldsymbol{\varphi}_m\end{aligned} \tag{53}$$

Finally, by excluding $\boldsymbol{p}^{inc}_n$ and $\boldsymbol{p}^{ouc}_n$ from GSO, we have

$$\begin{aligned}&\tilde{\boldsymbol{p}}_n\triangleq\boldsymbol{p}_n-\boldsymbol{p}^{inc}_n-\boldsymbol{p}^{ouc}_n=\boldsymbol{p}_n-\boldsymbol{p}^{cst}_n\\&\sum_{m\in\mathcal{N}_n}\Bigg(\Big(\text{diag}\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\mathbb{1}+\hat{\mathbf{B}}^{(m)}_{mn}\mathbb{1}\Big)\boldsymbol{\varphi}_n\\&-\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\boldsymbol{\varphi}_n-\hat{\mathbf{B}}^{(m)}_{mn}\boldsymbol{\varphi}_m\Bigg)\end{aligned} \tag{54}$$

To write it in a compact way, we have

$$\tilde{\boldsymbol{p}}=\hat{\mathbf{B}}\boldsymbol{\varphi}, \tag{55}$$

where $\tilde{\boldsymbol{p}}$, $\hat{\mathbf{B}}$ an $\boldsymbol{\varphi}$ are denoted by

$$\tilde{\boldsymbol{p}}=\left[\tilde{\boldsymbol{p}}^\top_1,\cdots,\tilde{\boldsymbol{p}}^\top_{|\mathcal{N}|}\right]^\top,\hat{\mathbf{B}}\triangleq((\mathbb{1}\mathbb{1}^\top)_N\otimes\Gamma_c)\circ\mathbf{B} \tag{56}$$

$$\boldsymbol{\varphi}=\left[\boldsymbol{\varphi}^\top_1,\cdots,\boldsymbol{\varphi}^\top_{|\mathcal{N}|}\right]^\top, \tag{57}$$

and $\hat{\mathbf{B}}$ has the same structure with $\mathbf{B}$ with replacing $\mathbf{B}^{s}_{mn}$, $\mathbf{B}^{(n)}_{mn}$ and $\mathbf{B}^{(m)}_{mn}$ with $\hat{\mathbf{B}}^{s}_{mn}$, $\hat{\mathbf{B}}^{(n)}_{mn}$ and $\hat{\mathbf{B}}^{(m)}_{mn}$, respectively.

*2) Reactive Power GSO:* The reactive power analysis is similar to the active power analysis. In particular, we use the approximation that $\boldsymbol{\varphi}_n\mathbb{1}^\top-\mathbb{1}\boldsymbol{\varphi}^\top_m\approx\mathbf{0}$ in (45), where $\mathbf{0}$ is the all-zeros matrix. Therefore, the first part of (15) is

$$\begin{aligned}&-D\Big(\boldsymbol{v}_n\boldsymbol{v}^H_n\big(\frac{\mathrm{j}}{2}\mathbf{B}^{s}_{mn}+\mathrm{j}\mathbf{B}^{(n)}_{mn}\big)\Big)\approx-D\Big(\mathbb{1}\mathbb{1}^\top\circ\\&\Big(\overbrace{\text{diag}(|\boldsymbol{v}_n|)}^{\text{C in C1}}\ \overbrace{\Gamma}^{\text{A in C1}}\ \overbrace{\text{diag}(|\boldsymbol{v}_n|)}^{\text{E in C1}}\Big)\big(\frac{\mathrm{j}}{2}\mathbf{B}^{s}_{mn}+\mathrm{j}\mathbf{B}^{(n)}_{mn}\big)\Big)\\&\overset{\text{C1}}{=}-D\Bigg(\Big((|\boldsymbol{v}_n||\boldsymbol{v}_n|^\top)\circ(\Gamma_c+\mathrm{j}\Gamma_s)\Big)\big(\frac{\mathrm{j}}{2}\mathbf{B}^{s}_{mn}+\mathrm{j}\mathbf{B}^{(n)}_{mn}\big)\Bigg)\end{aligned} \tag{58}$$

Then we take the imaginary part of (58) as:

$$-D\Bigg(\Big((|\boldsymbol{v}_n||\boldsymbol{v}_n|^\top)\circ\Gamma_c\Big)\big(\frac{1}{2}\mathbf{B}^{s}_{mn}+\mathbf{B}^{(n)}_{mn}\big)\Bigg) \tag{59}$$

$$=-D\Bigg((|\boldsymbol{v}_n||\boldsymbol{v}_n|^\top)\circ\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\Bigg) \tag{60}$$

$$\approx-D\Bigg((\mathbb{1}|\boldsymbol{v}_n|^\top)\circ\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\Bigg). \tag{61}$$

The process of transformation from (59) to (60) is similar to the transformation from (48) to (49). From (60) to (61), we relax $|\boldsymbol{v}_n||\boldsymbol{v}_n|^\top\approx\mathbb{1}|\boldsymbol{v}_n|^\top$ by $|\boldsymbol{v}_n|\approx\mathbb{1}$.

Replacing $m$ with $n$ this form applies also to $\boldsymbol{v}_n\boldsymbol{v}^H_m$. Therefore, the second imaginary part of (15) is

$$\Im\Big\{-D\left(\mathrm{j}\boldsymbol{v}_n\boldsymbol{v}^H_m\mathbf{B}^{(m)}_{mn}\right)\Big\}\approx-D\left(\big(\mathbb{1}|\boldsymbol{v}_m|^\top\big)\circ\hat{\mathbf{B}}^{(m)}_{mn}\right). \tag{62}$$

By summing (61) and (62) together, we have

$$\begin{aligned}\boldsymbol{q}_n\approx-D\Bigg(&(\mathbb{1}|\boldsymbol{v}_n|^\top\pm|\boldsymbol{v}_n|\mathbb{1}^\top)\circ\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}\\&+\hat{\mathbf{B}}^{(n)}_{mn}\big)\Bigg)-D\Bigg((\mathbb{1}|\boldsymbol{v}_m|^\top)\circ\hat{\mathbf{B}}^{(m)}_{mn}\Bigg),\end{aligned} \tag{63}$$

where we add and minus this item, i.e., $|\boldsymbol{v}_n|\mathbb{1}^\top$, in order to split (63) into three parts, i.e., $\tilde{\boldsymbol{q}}^{ouc}_n$, $\tilde{\boldsymbol{q}}^{inc}_n$ and $\tilde{\boldsymbol{q}}^{cst}_n$. Specifically:

$$\tilde{\boldsymbol{q}}^{inc}_n\triangleq-D\Bigg((\mathbb{1}|\boldsymbol{v}_n|^\top-|\boldsymbol{v}_n|\mathbb{1}^\top)\circ\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\Bigg) \tag{64}$$

$$=D\Bigg((|\boldsymbol{v}_n|\mathbb{1}^\top-\mathbb{1}|\boldsymbol{v}_n|^\top)\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)^\top\Bigg)\overset{\text{P4\&P5}}{=} \tag{65}$$

$$\begin{aligned}&\text{diag}\Big(\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\mathbb{1}\Big)|\boldsymbol{v}_n|-\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)|\boldsymbol{v}_n|,\end{aligned} \tag{66}$$

where the transformation from (65) to (66) is the same one from (49) to (50). Likewise, we could replace $m$ with $n$ and $\left(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\right)$ with $\hat{\mathbf{B}}^{(m)}_{mn}$, and have the second part:

$$\begin{aligned}&\tilde{\boldsymbol{q}}^{ouc}_n\triangleq D\Bigg((|\boldsymbol{v}_n|\mathbb{1}^\top)\circ\overbrace{\hat{\mathbf{B}}^{(m)}_{mn}}^{-\hat{\mathbf{B}}^{(n)}_{mn}}\Bigg)-D\left((\mathbb{1}^\top|\boldsymbol{v}_m|)\circ\hat{\mathbf{B}}^{(m)}_{mn}\right)\\&=\text{diag}\left(\hat{\mathbf{B}}^{(m)}_{mn}\mathbb{1}\right)|\boldsymbol{v}_n|-\hat{\mathbf{B}}^{(m)}_{mn}|\boldsymbol{v}_m|\end{aligned} \tag{67}$$

The remaining part of Eq. (63) is

$$\tilde{\boldsymbol{q}}^{cst}_n\triangleq-D\Bigg((|\boldsymbol{v}_n|\mathbb{1}^\top)\circ\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}\Bigg) \tag{68}$$

With $|\boldsymbol{v}_n|\approx\mathbb{1}$, $\boldsymbol{q}^{cst}_n$ could be relaxed as a biased part that does not involve in $|\boldsymbol{v}_n|\mathbb{1}^\top-\mathbb{1}|\boldsymbol{v}_m|^\top$:

$$\tilde{\boldsymbol{q}}^{cst}_n\approx\boldsymbol{q}^{cst}_n\triangleq-D\left(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}\right) \tag{69}$$

Finally, by excluding $\boldsymbol{q}^{cst}_n$ from the GSO, we have

$$\begin{aligned}&\boldsymbol{q}_n-\boldsymbol{q}^{cst}_n\approx\tilde{\boldsymbol{q}}^{inc}_n+\tilde{\boldsymbol{q}}^{ouc}_n\triangleq\tilde{\boldsymbol{q}}_n=\\&\sum_{m\in\mathcal{N}_n}\Bigg(\text{diag}\Big(\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)\mathbb{1}+\hat{\mathbf{B}}^{(m)}_{mn}\mathbb{1}\Big)|\boldsymbol{v}_n|\\&-\big(\frac{1}{2}\hat{\mathbf{B}}^{s}_{mn}+\hat{\mathbf{B}}^{(n)}_{mn}\big)|\boldsymbol{v}_n|-\hat{\mathbf{B}}^{(m)}_{mn}|\boldsymbol{v}_m|\Bigg)\end{aligned} \tag{70}$$

In the same way with active power injects, we have

$$\tilde{\boldsymbol{q}}=\hat{\mathbf{B}}|\boldsymbol{v}|, \tag{71}$$

$$\tilde{\boldsymbol{q}}=\left[\tilde{\boldsymbol{q}}^\top_1,\cdots,\tilde{\boldsymbol{q}}^\top_{|\mathcal{N}|}\right]^\top,|\boldsymbol{v}|=\left[|\boldsymbol{v}_1|^\top,\cdots,|\boldsymbol{v}_{|\mathcal{N}|}|^\top\right]^\top. \tag{72}$$

## REFERENCES


[1] T. Wu, A. Scaglione, and D. Arnold, "Reinforcement learning using physics inspired graph convolutional neural networks," *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2022.

[2] T. Wu, Y.-J. A. Zhang, Y. Liu, W. C. Lau, and H. Xu, "Missing data recovery in large power systems using network embedding," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 680–691, 2020.

[3] X. Dong and et al., "Graph signal processing for machine learning: A review and new perspectives," *IEEE Signal Process. Mag.*, 2020.

[4] R. Ramakrishna and A. Scaglione, "Grid-Graph Signal Processing (Grid-GSP): A Graph Signal Processing Framework for the Power Grid," *IEEE Trans. Signal Process.*, 2021.

[5] R. Jiang, T. Zahavy, Z. Xu, A. White, M. Hessel, C. Blundell, and H. Van Hasselt, "Emphatic algorithms for deep reinforcement learning," in *ICML*, 2021.

[6] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He, "Fault location in power distribution systems via deep graph convolutional networks," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 1, pp. 119–131, 2019.

[7] A. S. Zamzam and et al., "Physics-aware neural networks for distribution system state estimation," *IEEE Trans. Power Syst.*, 2020.

[8] M. Liang and et al., "Feedergan: Synthetic feeder generation via deep graph adversarial nets," *IEEE Trans. Smart Grid*, 2020.

[9] P. F. Ribeiro and et al., *Power systems signal processing for smart grids*. John Wiley & Sons, 2013.

[10] A. Paszke and et al, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[11] M. Abadi, A. Agarwal, and et al, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[12] L. Zhang and et al., "Real-time power system state estimation and forecasting via deep unrolled neural networks," *IEEE Trans. Signal Process.*, vol. 67, no. 15, pp. 4069–4077, 2019.

[13] O. Kundacina, M. Cosovic, and D. Vukobratovic, "State estimation in electric power systems leveraging graph neural networks," *arXiv preprint arXiv:2201.04056*, 2022.

[14] M. J. Hossain and M. Rahnamay-Naeini, "State estimation in smart grids using temporal graph convolution networks," in *2021 North American Power Symposium (NAPS)*. IEEE, 2021, pp. 01–05.

[15] M. B. Do Coutto Filho and et al., "Forecasting-aided state estimation—part i: Panorama," *IEEE Trans. Power Syst.*, 2009.

[16] ——, "Forecasting-aided state estimation—part ii: Implementation," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1678–1685, 2009.

[17] Y. Zhang and et al., "Deep reinforcement learning based volt-var optimization in smart distribution systems," *IEEE Trans. Smart Grid*, vol. 12, no. 1, pp. 361–371, 2021.

[18] J. G. Vlachogiannis and et al., "Reinforcement learning for reactive power control," *IEEE Trans. Power Syst.*, vol. 19, no. 3, 2004.

[19] H. Xu and et al., "Optimal tap setting of voltage regulation transformers using batch reinforcement learning," *IEEE Trans. Power Syst.*, 2019.

[20] Q. Yang and et al., "Two-timescale voltage control in distribution grids using deep reinforcement learning," *IEEE Trans. Smart Grid*, 2019.

[21] J. Duan and et al., "Deep-reinforcement-learning-based autonomous voltage control for power grid operations," *IEEE Trans. Power Syst.*

[22] S. Wang and et al., "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4644–4654, 2020.

[23] T. Haarnoja and et al., "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *ICML*, 2018.

[24] D. Cao and et al., "A multi-agent deep reinforcement learning based voltage regulation using coordinated pv inverters," *IEEE Trans. Power Syst.*, 2020.

[25] H. Van Hasselt and et al., "Deep reinforcement learning and the deadly triad," *arXiv:1812.02648*.

[26] H. Liu and et al., "Two-stage deep reinforcement learning for inverter-based volt-var control in active distribution networks," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2037–2047, 2020.

[27] T. Zhao and J. Wang, "Learning sequential distribution system restoration via graph-reinforcement learning," *IEEE Trans. Power Syst.*, 2021.

[28] X. Y. Lee and et al., "A graph policy network approach for volt-var control in power distribution systems," *arXiv:2109.12073*.

[29] Y. Gao and et al., "Consensus multi-agent reinforcement learning for volt-var control in power distribution networks," *IEEE Trans. Smart Grid*, 2021.

[30] X. Sun and et al., "Two-stage volt/var control in active distribution networks with multi-agent deep reinforcement learning method," *IEEE Trans. Smart Grid*, 2021.

[31] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs," *IEEE transactions on signal processing*, vol. 61, no. 7, pp. 1644–1656, 2013.

[32] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[33] M. Defferrard and et al., "Convolutional neural networks on graphs with fast localized spectral filtering," *NIPS*, 2016.

[34] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[35] L. Ruiz and et al., "Graph neural networks: Architectures, stability, and transferability," *Proc. IEEE*, vol. 109, no. 5, pp. 660–682, 2021.

[36] R. Ramakrishna and A. Scaglione, "On modeling voltage phasor measurements as graph signals," in *2019 IEEE DSW*.

[37] Y. Liu and et al., "Data-driven power flow linearization: A regression approach," *IEEE Trans. Smart Grid*, 2018.

[38] A. Anis and et al., "Efficient sampling set selection for bandlimited graph signals using graph spectral proxies," *IEEE Trans. Signal Process.*, 2016.

[39] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[40] S. S. Saha, A. Scaglione, R. Ramakrishna, and N. G. Johnson, "Distribution systems ac state estimation via sparse ami data using graph signal processing," *IEEE Transactions on Smart Grid*, 2022.

[41] R. R. Jha and et al., "Bi-level volt-var optimization to coordinate smart inverters with voltage control devices," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1801–1813, 2019.

[42] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR*, 2017.

[43] J. Schulman and et al., "Proximal policy optimization algorithms," *arXiv:1707.06347*.