Deep Reinforcement Learning with Graph ConvNets for Distribution Network Voltage Control

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

Abstract—This paper proposes a model-free Volt-VAR control (VVC) algorithm via the spatio-temporal graph ConvNet-based deep reinforcement learning (STGCN-DRL) framework, whose goal is to control smart inverters in an unbalanced distribution system. We first identify the graph shift operator (GSO) based on the power flow equations. Then, we develop a spatio-temporal graph ConvNet (STGCN), testing both recurrent graph ConvNets (RGCN) and convolutional graph ConvNets (CGCN) architectures, aimed at capturing the spatiotemporal correlation of voltage phasors. The STGCN layer performs the feature extraction task for the policy function and the value function of the reinforcement learning architecture, and then we utilize the proximal policy optimization (PPO) to search the action spaces for an optimum policy function and to approximate an optimum value function. We further utilize the low-pass property of voltage graph signal to introduce an GCN architecture for the the policy whose input is a decimated state vector, i.e. a partial observation. Case studies on the unbalanced 123-bus systems validate the excellent performance of the proposed method in mitigating instabilities and maintaining nodal voltage profiles within a desirable range.

Index Terms—Graph ConvNet, Deep Reinforcement Learning, Cyber-Physical Attacks.

I. INTRODUCTION

A. Background and Motivation

In distribution systems, voltage profiles are the most critical indicator of the system operating condition, whilst reliable and efficient energy management is the core task [1][4]. This is why Volt-VAR control (VVC) schemes have been developed and integrated into distribution systems to reduce network losses [2], avoid voltage violations [5] and mitigate cyber attacks [6]. However, the rapid growth of distributed energy resources makes it increasingly difficult to manage voltage profiles on active distribution networks.

Recently, many authors have studied reinforcement learning for a variety of distribution system optimization and control applications (see e.g. [7] for a review). Deep reinforcement learning methods utilize deep neural networks to approximate the optimal policy functions [8] and, thanks to their strong generalization capability in high-dimensional state spaces, they can address more complex tasks with lower prior knowledge, by learning different levels of abstractions from the data [9]. However, it is well known that reinforcement learning algorithms can become unstable when combining function approximation, off-policy learning, and bootstrapping — in fact, such combination is referred to as the deadly triad [10]. Recent advances in Graph Signal Processing (GSP) embedded in neural networks, called Graph Convolutional Neural Network (GCN), have opened a new way to learn better feature representational signals whose supports are large-scale networks, thereby significantly alleviating the instability of deep reinforcement learning [11]. The tenet of our work is that GCN is a key building block in the application of deep reinforcement learning (DRL) for electric power systems in general, where the control policies are driven by the system state, and for distribution grids in particular. Our paper showcases its performance when seeking a policy to select the set-points of inverters in a distribution grid. Next, we survey the related literature and then summarize our contributions.

B. Related Works

The literature on voltage control is vast. We limit our review to the most relevant literature pertaining DRL and GCN.

1) Deep Reinforcement Learning: Existing DRL methods for Volt-VAR control in distribution grids are broadly classified as value-based [12][15] and policy-based RL algorithms [16][18]. These methods have the following limitations for distributed system control. First, by ignoring the spatio-temporal correlations of the grid state, the fully connected neural network (NN) or convolutional neural networks (CNNs) architectures adopted in the literature are over-parametrized in their feature extraction layers and, therefore, likely to trigger the aforementioned deadly triad of DRL [4]. Second, for the most part, the DRL algorithms proposed take as an input the full state of the system [1][2]. Even when the state is observable, it is hard to scale these methods to work with large-scale network systems with high-dimensional features [3][4]. Some researchers have proposed adversarial DRL for Volt-VAR control in distribution grids. Unfortunately, this approach still suffers from poor efficiency and presents convergence issues [1]. Very recently, [19][20] considered the graph correlation of voltage phasors in their DRL design. However, the authors ignored the temporal correlation of their time series. Note that [19][20] require the full system state.

2) Graph Convolutional Neural Network: A handful of papers have so far successfully applied GCN to distribution systems’ management, considering applications that include fault localization [21], distribution system state estimation [22], and data generation [23]. GCNs are a generalization of convolutional neural networks (CNN), aimed at capturing...
the impact that the network connectivity has on the patterns of values that are associated to the network nodes. For this reason, one of the foundations of GCN lies in the Graph Signal Processing (GSP) definition of graph filters and of Graph Shift Operator (GSO). In \cite{21}, the GSO is defined as the weighted adjacent matrix, constructed based on the physical distance between nodes. Such filters are not suitable to extract relevant features from voltage phasors because the electrical characteristics of transmission lines are not accounted for. Related is also the work in \cite{22}, where the authors prune the weights of the NN based on the power grid topology, without considering the grid admittances. Instead of considering the physical power grid parameters, \cite{23} constructs an adjacent matrix that captures the correlation among historical data. Since Ohm’s law is the obvious driver of the correlation matrix that captures the correlation among historical data, considering the grid admittances. Instead of considering the power grid topology, without weighting the adjacent matrix, constructed based on the physical Graph Shift Operator (GSO). In \cite{21}, the GSO is defined as the weight adjacency matrix.

C. Contributions and Organization

To address the challenges mentioned above, in this paper we propose a Spatio-Temporal Graph ConvNet-based Deep Reinforcement Learning (STGCN-DRL) framework, in two architectures: Recurrent Graph ConvNets (RGCNs) and Conv1D Graph ConvNets (CGCNs). We embed the STGCN framework in the policy-gradient DRL, specifically, the Proximal Policy Optimization (PPO), used to train STGCN-DRL to learn policies that control smart inverters. Our main contributions are summarized next.

- We develop a novel STGCN-based DRL to train distribution network voltage control policies for smart inverters. We test both RGCN and CGCN architectures for the extraction of spatio-temporal features from the voltage phasors of the system. We show numerically that the proposed STGCN outperforms schemes where fully-connected neural networks and CNNs are trained.
- Since the state-of-the-art GCNs currently take only real-valued inputs, we derive a suitable graph shift operator defined as a function of the system susceptance matrix for the amplitude and phase of the voltage phasor.
- We introduce a DRL architecture with an input that is a decimated system state vector, to either deal with sparse deployments of Phasor Measurements Units, or simply for scalability. In this case, we show that one can use the kron-reduced network GSO instead of the full GSO.
- The proposed STGCN-DRL method targets the mitigation of oscillations of the voltage profile while, at the same time, maintaining nodal voltage profiles within a desirable range. The policy we propose is more versatile than others proposed in the literature, at it addresses effectively and rapidly a relatively complex objective, responding rapidly to undesired dynamics and voltage values.

The rest of the paper is organized as follows. In Section II, we introduce the problem formulation and system model. In Section III, we describe the spatio-temporal graph ConvNet-based deep reinforcement learning architecture in general. In Section IV, we specialize the proposed STGCN-DRL architecture to target voltage control of three-phase distribution networks. The case studies are given in Section V. Finally, we conclude the paper in Section VI.

D. Notation

The symbols and notations are summarized as follows. The grid has an associated graph with \( N \) nodes, whose set is denoted by \( \mathcal{N} = \{1, \ldots, N\} \) and a set of lines that are the grid edges \( \mathcal{E} \subseteq \mathcal{N} \times \mathcal{N} \) represent overhead or underground lines. \( \mathcal{N}_s \) denotes a set of the single phase of buses with smart inverters installed and \( |\mathcal{N}_s| \) denotes its cardinality. We denote \( \mathcal{P}_{mn} \in \{a_{mn}, b_{mn}, c_{mn}\} \) and \( \mathcal{P}_m \in \{a_m, b_m, c_m\} \) the phases of line \((m,n)\) in \( \mathcal{E} \) and node \( n \in \mathcal{N} \), respectively. Let \( v_{n0} \in \mathbb{C} \) be the complex line-to-ground voltage at node \( n \in \mathcal{N} \) of phase \( \phi \in \mathcal{P}_n \). Let \( \nu \) represent the vector of voltage phasors over all buses, i.e., \( \nu = [v_1^T, \ldots, v_N^T]^T \) with the \( n^{th} \) entry \( v_n^T = [v_{n0}\phi \in \mathcal{P}_n] \in \mathbb{C}|\mathcal{P}_n||x|^1 \) with voltage phase \( \theta_{n0} \) and voltage magnitude \( |v_{n0}| \). The vector sizes of the current injection phasors \( i \), apparent power injections \( p \), active and reactive power injections, respectively, are consistent with the size of \( \nu \). \( s = p + jq \) be the vector of net apparent power, where \( j = \sqrt{-1} \) represents the imaginary unit.

II. Problem Formulation

This section introduces the model of smart inverters using Volt-Var (VV) and Volt-Watt (VW) schemes. Then, we integrate such a model into the power flow calculation (e.g. OpenDSS) to evaluate the impacts of the status changes on distribution systems. Leveraging these changes, we propose a STGCN-DRL method to obtain control actions so as to realize effective voltage regulation.

A. Control of Smart Inverters

The power injection control functionality of smart inverters is defined by two VV and VW piece-wise functions of the voltage magnitude, referred as “droop” curves. The VV-VW curves are shown in Fig. \ref{fig:fig1}(c) and Fig. \ref{fig:fig1}(d); their set-points are tied to the five parameters associated to the segments of the piece-wise linear VV and VW curves, denoted by \( \eta = [\eta_1, \ldots, \eta_5]^T \in \mathbb{R}^5 \). Mathematically, they are:

\[
\begin{align}
    f_n^p(\nu_n) & \triangleq \begin{cases} 
    \bar{q} \left( \frac{n_2 - |v_{n0}|}{n_2 - n_1} \right) & |v_{n0}| \in [0, \eta_1] \\
    0 & |v_{n0}| \in (\eta_1, \eta_2] \\
    -\left( \frac{n_3 - |v_{n0}|}{n_3 - n_2} \right) \bar{q} & |v_{n0}| \in (\eta_2, \eta_3] \\
    -\bar{q} & |v_{n0}| \in (\eta_3, \eta_4) \\
    \end{cases} \\
    f_n^q(\nu_n) & \triangleq \begin{cases} 
    \bar{p} \left( \frac{n_5 - |v_{n0}|}{n_5 - n_4} \right) & |v_{n0}| \in [0, \eta_4] \\
    0 & |v_{n0}| \in (\eta_4, \eta_5] \\
    \end{cases}
\end{align}
\]

where \( \bar{p} \) and \( \bar{q} \) are the active and reactive powers injected into the system, in response to the estimated voltage amplitude \( |v_{n0}| \) obtained by low-pass filtering the measured voltage.
magnitude signal at bus $n$ on phase $\phi$, to reject some of the noise in the measurements $|v_{n_\phi,t}|$. In particular, $|\tilde{v}_{n_\phi,t}|$ and the limit on the choice of $\bar{q}$ are:

$$
|\tilde{v}_{n_\phi,t}| = |v_{n_\phi,t}| - 1 + \tau_n^C (|v_{n_\phi,t}| - |\tilde{v}_{n_\phi,t}|),
$$

$$
\bar{q}^2 + (f^n(|\tilde{v}_{n_\phi,t}|))^2 \leq \bar{s}^2,
$$

where $\tau_n^C$ is the time constant of the low pass filter, $|v_{n_\phi,t}|$ is the measured voltage magnitude, and $\bar{s}$ is the capacity of the inverter. The power injected also shall not change suddenly. Hence, voltage control is completed by the following dynamics on the injected active and reactive powers:

$$
p_{n_\phi,t} = p_{n_\phi,t-1} + \tau^n (f^n(|\tilde{v}_{n_\phi,t}|) - p_{n_\phi,t-1}),
$$

$$
q_{n_\phi,t} = q_{n_\phi,t-1} + \tau^n (f^n(|\tilde{v}_{n_\phi,t}|) - q_{n_\phi,t-1}),
$$

where $\tau^n$ is a time constant, and the complex power injected into the distribution system is $s_{n,t} = p_{n,t} - jq_{n,t}$. The goal of this paper is to determine an optimum control policy to select the set-points of the inverter. In the next section we describe the setup for DRL whose task is to learn a policy to control the VV and VW curves set-points. In particular, as shown in Fig. 1(c) and Fig. 1(d), to simplify the action space of the DRL algorithm, we will only shift the VV curve.

B. DRL Design for VVC

In this paper the neural network (NN) trained by the DRL algorithm is the agent selecting specific inverter setpoints, and the environment, where the agent takes multiple control actions, is the electric distribution network. The agent NN must be trained to respond to a variety of conditions and take control actions with respect to the given operating condition to achieve VVC. The interaction between the agent and the environment at time $t$ is described by: the state, comprising a set of past samples $(x_t, \ldots, x_{t-K})$, the action $a_t$, and the reward $r_t$. We describe these three elements next.

1) State and Action: The tuple of actions for the VVCs of the inverters in the bus set $N_s$ is denoted by:

$$
a = [a_1, a_2, \ldots, a_i, \ldots, a_i N_s]^T, a_i \in A_i,
$$

$$
\eta_i' = \eta_i + a_i * 1,
$$

where $|N_s|$ represents the number of smart inverters, $a_i$ denotes the control action on the $i$th smart inverter, $1 \in \mathbb{R}^5$ represents a vector of all one elements, $A_i$ represents the action space of the $i$th action, and $\eta_i'$ defines the default and shifted values of VV-VW curves. As shown in Fig. 1 in this paper, instead of controlling arbitrarily the parameters of the VV/VW curves, we only shift the value from the default value defining the curves [6] [24]. The vector of actions $a$, output of the NN that approximates the optimum policy, is a function of the three-phase voltages at all, or part, of buses in the distribution system, which is the observation/input of the STGCN-DRL. The state vector/observation is $x = [\theta, |v|^T]$, where $\theta$ is the vector of voltage phases and $|v|$ is the vector of voltage magnitudes.

2) Reward: In this paper, the definition of reward comprises multiple components in its expression, each targeting a measure of power quality. These components are defined below:

a) Voltage Magnitude Regulation: We define the component of regret caused by voltage deviation (VD) in the distribution network at bus $n_\phi$ at time $t$ as follows [13] [18]:

$$
rd_{n_\phi,t} = ||v_{n_\phi,t} - \bar{v}||, n_\phi \in N_s,
$$

where $\bar{v}$ denotes the desired voltage magnitude at bus $n$ (i.e., 1 p.u.), and $|v_{n_\phi,t}|$ is the measured voltage magnitude on phase $\phi$, respectively. The second component of the total regret that penalizes the active power curtailment is:

$$
rp_{n_\phi,t} = \left(1 - \frac{p_{n_\phi,t}}{p_{n_\phi,t}^\text{max}} \right)^2, n_\phi \in N_s,
$$

(7)

where $N_s$ denotes the set of smart inverters.

b) Oscillation Mitigation: To measure undesired variations of the voltage amplitudes, we define a filter whose output measures the “energy” associated with voltage variations in the distribution grid. The filter is the cascade of a high-pass filter, a square law non-linearity, and a low-pass filter. A discrete time block diagram of this part of the architecture is shown in Fig. 1(b), at the top left corner of Fig. 1. Specifically, $H_{HP}$ and $H_{LP}$ represent high-pass and low-pass filters, respectively, and $c$ is a positive gain. The high-pass filter removes DC contents from $v_{n_\phi,t}$, yielding $\Delta v_{n_\phi,t}$. After that, this signal is squared to produce a DC term that is then averaged by a low-pass filter. The filter parameters should be chosen such that the filter does not attenuate oscillations due to inverter instabilities.

Therefore, the reward function for each smart inverter is:

$$
r_{n_\phi,t}^{os} = - (\zeta_v \times rd_{n_\phi,t} + \zeta_p \times rp_{n_\phi,t} + \zeta_o \times ro_{n_\phi,t}), n_\phi \in N_s,
$$

(8)

where $\zeta_v$, $\zeta_p$ and $\zeta_o$ are positive weights. In particular, the third term $\zeta_o$ penalizes oscillations.

C. Voltage Control in Distribution Networks

While the policy learnt by the DRL is beneficial in general, the primary motivation in this paper is to respond to the nefarious effects of smart inverters VV/VM with inappropriate set-points due to a cyber-attack. The set of inverters in the system $|N_s|$ is composed of two subsets, denoted by $C$ and $U$, representing the “compromised” and “uncompromised” inverter, respectively. We assume that $U \neq \emptyset$ and use DRL to determine the optimum stochastic policy $\pi_{\theta}$ which represents the probability distribution of action $a \in A$ given a sequence of observations; $\theta$ are the neural network parameters that the DRL selects to maximize the reward defined in (8).

The overview of the problem statement and methodology is shown in Fig. 1. During the training stage, as illustrated in Fig. 1(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. 1(c)) during the training of the policy, computed with the neural network architecture in Fig. 1(f). Moreover, both policy function and value function are trained by the PPO algorithm, and share the same spatio-temporal GCN feature extraction layers, which will be introduced
explicitly in Section III. As a feedback, the policy gives an action to shift the VV-VW curves so as to change the active and reactive power injections, as shown in Fig. 1(c) and Fig. 1(d). Note the the input voltages of the VV-VW curves are low-pass filtered (see Fig. 1(c) and Fig. 1(h)) to smoothen noisy measurements. The active and reactive power injections from the inverters are low-pass filtered as well, to prevent sudden jumps. After the policy is well trained, the parameters of the neural network in Fig. 1(f) are fixed and the policy can be deployed in the real system. The details of the policy function design are presented in the following section.

### III. Spatio-Temporal Graph ConvNet-based Deep Reinforcement Learning

In this section, we first review the proximal policy optimization (PPO) method, and both graph signal processing (GSP) and GCN. We then present the novel STGCN-DRL framework, including Conv1D enabled spatio-temporal GCN and RNN enabled spatio-temporal GCN.

**A. A Brief Review of Proximal Policy Optimization**

Policy gradient methods employ a policy modeled by a neural network which is trained directly by gradient ascent on the expected return. Among these methods, Actor-Critic is one of the most important RL frameworks to learn both policy and value functions.

1) **Objective:** Let a stochastic policy \( \pi_\theta \), parameterized by \( \theta \), model the probability distribution of \( \mathbf{a}_t \in \mathcal{A} \) given a sequence of observations \( \mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t} \), where \( K_t \) represents the length of time windows. The goal of each agent is to find a policy, which maximizes its expected discounted return:

\[
\pi^* \in \arg \max \pi J(\pi) = \mathbb{E}_{\mu \sim \pi_\theta} \left[ \sum_{t=0}^{T} \gamma^t r_t(\mathbf{x}_t, \mathbf{a}_t) \right],
\]

where \( \mu \) is the trajectory generated by policy \( \pi_\theta \), i.e., the action \( \mathbf{a}_t \) is taken according to policy \( \pi_\theta(\cdot | \mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}) \), \( r_t \) represents rewards at time \( t \) (e.g. \( r_t = r_{\text{os},t}^{\text{os}} \) in (8)), and \( x_t \) denotes the state observation at time \( t \). The parameter \( \gamma \in (0, 1) \) is the discounting factor, discounting future rewards.

2) **Policy Gradient:** Let \( V_\pi^\theta(\mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}) \) be the value function parametrized by \( \vartheta \), estimating the cumulative discounted reward from the current state to the terminal state. The gradient ascent method is applied to solve the optimization problem in (9). For PPO, the gradient of \( J(\theta) \) is:

\[
\nabla J(\theta) = \mathbb{E}_{\mu \sim \pi_\theta} \left[ \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(\mathbf{a}_t | \mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}) \right],
\]

\[
A_\vartheta^\gamma(\mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}, \mathbf{a}_t) = r_t + \gamma V_\pi^\vartheta(\mathbf{x}_{t+1}, \cdots, \mathbf{x}_{t+1-K_t}) - V_\pi^\vartheta(\mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}),
\]

where \( A_\vartheta^\gamma(\mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}, \mathbf{a}_t) \) is the advantage function estimate, \( \gamma \) is the aforementioned discounting factor and \( T_b \) is a batch size. The policy and value functions are updated by gradient ascent/descent:

\[
\theta_{k+1} = \theta_k - \alpha \nabla J(\theta),
\]

\[
\vartheta_{k+1} = \vartheta_k - \beta \nabla \vartheta \left[ r_t + V_\pi^\vartheta(\mathbf{x}_{t+1}, \cdots, \mathbf{x}_{t+1-K_t}) - V_\pi^\vartheta(\mathbf{x}_t, \cdots, \mathbf{x}_{t-K_t}) \right],
\]

where \( \alpha \) and \( \beta \) is the constant step sizes. A PPO version using a clipped surrogate objective simplifies the aforementioned method and yields similar performance [25]:

\[
L_{\text{CLIP}}(\theta) = \mathbb{E} \left[ \min \left( \rho_t(\hat{\pi}_t), \rho_t(\hat{\pi}_t), 1 - \epsilon, 1 + \epsilon \right) \hat{A}_t \right],
\]

where clip operation encourages a more gradual update to the policy rather than large charges. The minimal operator between the unclipped and the clipped objective is used to
bound the unclipped objective. As such, the primary element of DRL is to design the policy function. In the following, we will present the STGCN-based policy function, which has the benefits of better capturing spatio-temporal features of the voltage phasors.

B. Graph Signal Processing and Grid Graph Convolutional Neural Network

To make the paper self-contained, we first review the basic theory of Graph Signal Processing (GSP) (more details can be found e.g. in [26]) and of graph neural networks.

1) Graph Signal Processing: In our description we consider a general graph \( G = (V, E) \), with vertex set \( V \) and edge set \( E \). The concepts defined here will be applied to the graph corresponding to the three-phase distribution network \( D = (\mathcal{N}, \mathcal{E}) \), where the node \( n_0 \) of bus \( n \) on phase \( \phi \) in \( D \) corresponds to the \( i^{th} \) node of \( G \) and the edges are the transmission lines connecting the buses.

A graph signal \( x \in \mathbb{R}^{|V|} \) (which in the grid is the state observation) is a vector, where \( x_i, \forall i \in V \) is the \( i \)-th entry of the state vector. The set \( \mathcal{N}_i \) denotes the subset of nodes connected to node \( i \). A graph shift operator (GSO) is a matrix \( S \in \mathbb{R}^{|V| \times |V|} \) that linearly combines graph signal neighbors’ values. Almost all operations including filtering, transformation and prediction are directly related to the GSO [26]. The GSO, denoted by \( S \in \mathbb{R}^{|V| \times |V|} \), is usually chosen as a graph weighted Laplacian:

\[
[S]_{i,j} = \begin{cases} \sum_{k \in \mathcal{N}_i} S_{i,k}, & i = j, \\ -S_{i,j}, & i \neq j. \end{cases}
\] (15)

In this work, we focus on real symmetric GSOs, i.e., \( S = S^\top \) that is appropriate for our power grid application. The GSO generalizes the \( s \) variable, that represents the derivative in the Laplace domain for signals in time. A graph filter is a linear matrix operator \( \mathcal{H}(S) \) that is a function of the GSO and operates on graph signals as follows

\[ w = \mathcal{H}(S)x. \] (16)

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

\[ \mathcal{H}(S) = \sum_{k=0}^K h_k S^k. \] (17)

Let the eigenvalue decomposition be \( S = U \Lambda U^\top \), where \( \Lambda \) is a diagonal matrix with eigenvalues \( \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|V|} \) and \( U \) be the eigenvector matrix \( U \) is unitary since the GSO \( S \) is symmetric. The Graph Fourier Transform (GFT) basis is \( U \), the GFT of a graph signal is \( \hat{x} = U^\top x \) and the eigenvalues \( \lambda_\ell, \ell = 1, \ldots, |V| \) are the graph frequencies. From (17) it follows that:

\[ \mathcal{H}(S) = U \left( \sum_{k=0}^K h_k \Lambda^k \right) U^{-1}. \] (18)

Note that the graph filter order \( K \) can be infinite.

The matrix \( \sum_{k=0}^K h_k \Lambda^k \) is a diagonal, with \( i^{th} \) entry \( \hat{h}(\lambda_i) \triangleq \sum_{k=0}^K h_k \lambda_i^k \). Hence, \( \hat{h} = [\hat{h}(\lambda_1), \ldots, \hat{h}(\lambda_{|V|})] \) is the transfer function for graph filters, and in the GFT domain the graph filter output corresponds to an element by element multiplication of the graph filter input, i.e.:

\[ w = \mathcal{H}(S)x \iff \hat{w} = \hat{h} \odot \hat{x}, \] (19)

where and \( \odot \) represents the element by element (Hadamard) vector product. To process time series of graph signals \( \{x_t\}_{t \geq 0} \) one uses graph temporal filters models:

\[ w_t = \sum_{\tau=0}^t \mathcal{H}_{t-\tau}(S)x_\tau \quad \mathcal{H}_t(S) = \sum_{k=0}^K h_{k,t} S^k, \] (20)

and for their analysis we can harness DSP tools, defining a combined GFT and \( z \)-transform:

\[ X(z) = \sum_{t=0}^{K_t} x_t z^{-t}, \quad \hat{X}(z) = U^\top X(z), \] (21)

where \( K_t \) is the length of the graph signal time series. In particular, considering a filter of order \( K_t \) and using \( S \odot z \) as the graph temporal GSO:

\[ \mathcal{H}(S \odot z) = \sum_{k=0}^K H_k(z) S^k, \quad H_k(z) = \sum_{t=0}^{K_t} h_{k,t} z^{-t} \] (22)

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:

\[ \hat{W}(z) = \mathcal{H}(S \odot z) \hat{X}(z), \] (23)

The graph-temporal joint transfer function is:

\[ \mathcal{H}(\Lambda, z) = \sum_{t=0}^{K_t} \sum_{k=0}^K h_{k,t} \Lambda^k z^{-t}. \] (24)

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

\[ \hat{W}(z) = \mathcal{H}(\Lambda, z) \hat{X}(z) \] (25)

which is again an element by element multiplication since \( \mathcal{H}(\Lambda, z) \) is a diagonal matrix. In a graph-convolutional neural networks (GCN), the \( h_{k,t} \) are the parameter learnt during training in the feature extraction layers [11]. In the left of Fig. 2 we illustrate differences of GCN, CNN and FNN with the same graph signal with specific neuron structures for each time instant. In this example, we observe that GCNs are generalizations of CNNs where time-series filter are replaced by the application-dependent graph temporal GSOs. FNN, instead, simply multiplies parameter matrices with graph signals but does not account for the graph-neighborhood structure directly.

In the following, we will introduce the application-specific GSO we will use in our GCNs distribution networks.
2) Grid Graph Convolutional Neural Network: Grid-GSP was proposed in [26] to provide an interpretation for the spatio-temporal properties of voltage phasor measurements by utilizing the admittance matrix as graph filters. In this work, due to the fact that the state of the art in GCNs currently takes only real-valued inputs [27], we derive a suitable graph shift operator defined as a function of the system susceptance matrix for the amplitude and phase of the voltage phasor.

The real graph signal components are the pairs of voltage magnitude $|v_{n_a}|$ and voltage phase $\varphi_{n_a}$. Let $s = p + iq$ be the vector of net apparent power at buses $(s = [s_1^T, \cdots, s_N^T]^T)$, with the $n$th entry $s_n = p_n + iq_n, s_n \in \mathbb{C}$. Further, let $v$ and $|v|$ be the vectors of bus voltage phasors and magnitudes, respectively, with $v \in \mathbb{C}^{\sum_{n \in N}|P_n| \times 1}$ and $|v| \in \mathbb{R}^{\sum_{n \in N}|P_n| \times 1}$, and let $i \in \mathbb{C}^{\sum_{n \in N}|P_n| \times 1}$ and $|i| \in \mathbb{R}^{\sum_{n \in N}|P_n| \times 1}$ be the vectors of net bus current phasors and magnitudes, respectively:

$$v_{n_o} = |v_{n_o}| e^{j\varphi_{n_o}}, \quad i_{n_o} = |i_{n_o}| e^{j\varphi_{n_o}}, \forall n \in N, \phi \in \mathcal{P}_n, i = Yv,$$  \hspace{1cm} (26)

where $Y$ in (26) is a block matrix of dimensions $\sum_{n \in N}|P_n| \times \sum_{n \in N}|P_n|$. More specifically, the blocks in $Y$ are:

1) the matrices $Y_{m,n}$, occupying the $|P_m| \times |P_n|$ off-diagonal block corresponding to line $(m, n) \in \mathcal{E}$; and,

2) the $|P_m| \times |P_m|$ diagonal block corresponding to node $n \in N$ with $N_n = \{m|(m, n) \in \mathcal{E}\}$:

$$[Y]_{P_m, P_n} = \sum_{m \in N_n} \left( \frac{1}{2} Y^s_{mm} + Y^m_{mn} \right)$$ \hspace{1cm} (27)

$Y^s_{mm}$ is the shunt element. The GSO is associated with $Y$, which is a complex Laplacian. As discussed in [28], Ohm’s law allows us to view voltage as the output low-pass filter by $v = Y^{-1}i$ (an integrator).

We used the approximation $\hat{p} = \hat{B} \hat{\varphi}$ and $\hat{q} = \hat{B} |v|$, which suggests using $\hat{\varphi}$ and $|v|$ as two graph signals, where $\hat{p}$, $\hat{q}$, $\hat{\varphi}$ and $|v|$ are defined in the appendix. Note that $B$ has the same structure with $B$, which is also defined in Appendix A.

Therefore, we have the following

$$\begin{bmatrix} \hat{p} \\ \hat{q} \end{bmatrix} = \begin{bmatrix} \hat{B} & 0 \\ 0 & \hat{B} \end{bmatrix} \begin{bmatrix} \hat{\varphi} \\ |v| \end{bmatrix}$$ \hspace{1cm} (28)

where $x = \begin{bmatrix} \hat{\varphi} \\ |v| \end{bmatrix}$ is the output of graph filter $S^{-1}$ and $S = \begin{bmatrix} \hat{B} & 0 \\ 0 & \hat{B} \end{bmatrix}$ are graph filters in our problem.

C. STGCN-based Deep Reinforcement Learning

Power systems are dynamic systems with time-varying voltage phasors (see the left side of Fig. 2). In order to fuse features from both spatial and temporal domains, we will consider the following two graph neural network architectures, namely Conv1D Graph Convolutional neural networks (CGCN) and Recurrent Graph Convolutional neural networks (RGCN).

1) STGCN I – CGCN: Based on (24), we can design the following transfer functions:

$$\hat{h}(\lambda, z) = \sum_{k=0}^{K} \theta_k \lambda^k \left( \sum_{t=0}^{K_t} h_{k,t} z^{-t} \right),$$  \hspace{1cm} (29)

where $\left( \sum_{t=0}^{K_t} h_{k,t} z^{-t} \right)$ are the parameters of temporal convolution, and then the GCN blocks are utilized. Accordingly, the graph signal $w_c^v$ from the first feature extraction layer (layer1) is defined as follows:

$$w_c^v = \sum_{k=0}^{K} \Theta_{k,1} s_k \left( \sum_{\tau=0}^{K_t} h_{k,\tau} x_{t-\tau} \right).$$  \hspace{1cm} (30)

As mentioned before the goal is to approximate the policy function and value function using graph neural networks. They are:

$$\pi_\theta = \sigma(\Theta_\pi^V \cdot \sigma(\Theta_2 \cdot w_1^c)),$$

$$V_{\vartheta} = \sigma(\Theta_\pi^V \cdot \sigma(\Theta_2 \cdot w_1^c)),$$  \hspace{1cm} (31)

where $\theta \triangleq \{(\Theta_3^V, \Theta_2, \Theta_{k,1}, h_{k,t})\}|k,t\}$ represent the parameters of $\pi_\theta$ that need to be learnt, and $\sigma(\cdot)$ represents the activation function. Here, we have omitted the biased term to unburden the notation. Recall that the value function $V_{\vartheta}$ share the STGCN layer and the second layer with the policy function. Therefore, the parameters of $V_{\vartheta}$ is defined as: $\vartheta \triangleq \{(\Theta_3^V, \Theta_2, \Theta_{k,1}, h_{k,t})\}|k,t\}$.

Eq. (30) and Eq. (31) are capable of capturing the spatial and temporal properties of the voltage phasors. In Fig. 2, we illustrate the proposed CGCN in the right block. Specifically, we have the Conv1D to convolve graph signals over time corresponding to $\left( \sum_{\tau=0}^{K_t} h_{k,\tau} x_{t-\tau} \right)$. Then we utilize the
spatial GCN to apply graph filters to graph signals. With the STGCN feature extraction layer, we construct the functions of the actor and critic in Eq. (31), which maps observations onto the probability of the actions and their advantage function values.

2) STGCN 2 – RGCN: RNNs are systems that exploit recurrence to learn dependencies in sequences of variable length. In the case of graph processes, we will thus adapt the operations performed by RNNs to take the graph structure into account. Based on [30], we have

$$ w_t^r = \sigma \left( \Theta_x \left( \sum_{k=0}^{K} h_k S^k x_t \right) + \Theta_h w_{t-1}^r \right), \quad (32) $$

where we have omitted the biased term to unburden the notation. Likewise, the policy function and value function are:

$$ \pi_\theta = \sigma (\Theta_3^Y \cdot \sigma (\Theta_2 \cdot w_t^r)), \quad V_\theta = \sigma (\Theta_3^V \cdot \sigma (\Theta_2 \cdot w_t^r)), \quad (33) $$

where $\theta \triangleq \{(\Theta_3^Y, \Theta_2, \Theta_x, \Theta_h, h_{k,t})|k,t\}$ represent the parameters of $\pi_\theta$ that need to be estimated by training. The parameters $\theta$ of $V_\theta$ are defined as $\vartheta \triangleq \{(\Theta_3^V, \Theta_2, \Theta_x, \Theta_h, h_{k,t})|k,t\}$. Here, we omit the bias term for brevity, but it is added in the implementation. Note that the value function $V_\theta$ shares the STGCN layer and the second layer with the policy function. Similar to our description of CGNN, we illustrate the proposed RGCN with $k = 4$ on the right side of Fig. 2. It shows that the graph signals are processed through GCN to capture spatial features of the signal, and then processed by RNN to capture the temporal correlations.

The use of the STGCN in Fig. 2 described above in the proposed STGCN-DRL framework is shown in Fig. 1 where the agent acquires global observations from measurements deployed to track the dynamics of the distribution system state, to use them as inputs of our DRL scheme, to compute rewards and the probability of the actions’ tuple $a = \{a_1, a_2, \cdots, a_{|N|}\}$ to control smart the inverters. Noticeably, the STGCN has multiple outputs with each one corresponding to each inverter, respectively. In particular, the actor and critic share the same feature extraction layers, i.e., the STGCN layer, to produce the policy actions and advantages. This kind of structure reduces the parameters to be learnt, accelerating and stabilizing the training process.

D. Partial Observation and PMU Selection

In this subsection, we provide the correct GSO for a down-sampled graph signal. The specific goal is to design a DRL algorithm that can use a reduced number of bus voltage phasors data as opposed to the complete state. Let $x_M$ (time index $t$ is ignored for simplicity) be the down-sampled voltage graph signal where $M \in \mathcal{N}$ in the set of node indices of the corresponding buses. In [26], it was shown that, not only the entire system state, but any down-sampled voltage phasor vector obtained from it has still low-pass property (its GFT is concentrated in the low graph frequencies) with respect to a GSO that is the Kron reduced admittance matrix. The result is summarized in the following lemma:

**Lemma 1** (Lemma 1) Let the GSO $S$ defined with respect to the full graph be invertible. Then, the GSO with respect to the reduced-graph of $M$ is denoted by $S_{red,M}$. We can model $x_M$ as follows:

$$ x_M = H(S_{red,M}) \kappa, \quad (34) $$

where the GSO for the reduced graph is given by Kron-reduction of $S$, $S_{red,M} = SC(S, S_{M^e,M^e})$. The $S_{M^e,M^e}$ is a sub-matrix of $S$ as:

$$ S = \begin{bmatrix} S_{M,M} & S_{M,M} \\ S_{M,M} & S_{M,M} \end{bmatrix}. \quad (35) $$

More specifically,

$$ x_M = \frac{H(S_{red,M})}{\|S_{M,M}\| - S_{M,M}S_{M,M}^{-1}} \kappa (S(S^{-1})w), \quad (36) $$

where $I$ is an identity matrix, and $S_{red,M}$ is the Schur complement of the block $S_{M^e,M^e}$ in the GSO, i.e.,

$$ S_{red,M} = SC(S, S_{M^e,M^e}) = S_{M,M} - S_{M,M}S_{M,M}^{-1}S_{M,M}^\top. \quad (37) $$

Therefore, $S_{red,M}$ represents the GSO for the reduced graph.

Recall that the right GSO is central in defining the structure of GCNs. Prior work (e.g. [23]) so far has assumed the full state as the input to the GCN. This may be undesirable for two reasons. If the state is accrued by measurements, in practice, in distribution systems PMUs are likely to be sparsely deployed and Distribution State Estimation is not necessarily common practice. Even if the state is available, being able to operate with a reduced input can benefit the scalability of the algorithm. Therefore, it is very useful for GCN networks to be able to accept an input that does not include the complete information about the state. However, the choice of $M$ affects the performance. This is why next we provide an optimized criterion to select $M$ leveraging GSP sampling theory.

1) Sampling method: Let the GFT basis corresponding to the first dominant $k$ graph frequencies be $U_K$. As shown in [26], the best $M$ is one-to-one with the subset of rows of $U_K$ with minimum correlation. Let $F_M$ be what is called the vertex limiting operator i.e. the matrix such that $F_M = Q_M Q_M^\top$, where $Q_M$ has columns that are the coordinate vectors pointing to each vertex/node in $M$. Mathematically, the optimal placement can be sought by maximizing the smallest singular value, $\max_{M} \lambda_{min}(F_M U_K)$, of the matrix $F_M U_K$. Such choice amounts to the selection of rows of $U_K$ that are as uncorrelated as possible, because the resulting matrix $F_M U_K$ has the highest conditional number [29].

E. Algorithm

In summary, we train the STGCN policies for oscillation mitigation while maintaining nodal voltage profiles within a desirable range. In the implementation stage, we utilize the trained policy for oscillation mitigation. The algorithm is summarized in Algorithm 1.
Algorithm 1: The STGCN-DRL Algorithm for Oscillation Mitigation

Input: in_training - True (stochastic) or False (static)

Initialization: The learning rates $\alpha$ and $\beta$, the discount rate $\gamma$, the size of replay buffer $RB$, default actions $a_0$;

Function RunPPO (in_training):

1. Initialize state $x_0$, $r_0 = \text{ChooseRewardState}(a_0)$, where $r_0$ is a vector with each element $r^{o_0}_{n_0, t=0}$ corresponding to each agent;
2. for $t \leftarrow 0$ to $T_{max}$ do
   
   for $j \leftarrow 0$ to $RB$ do
      
   Get actions and values $a_t, V^e_t = \text{stgcn}_ppo(in\_training)$, where $V^e_t$ is a vector $[V^e_t]_1 = V^e_{n_0, t}$ with respect to each agent;
   
   Get new state and rewards $x_{t+1}, r_{t+1} = \text{ChooseRewardState}(a_t)$;
   
   Store them as a transition $(x_t, \cdots, x_{t-K}, a_t, V^e_t, r_{t+1}, x_{t+1})$ in replay buffer;
   
   Sample from the replay buffer to obtain tuple $(x_t, \cdots, x_{t-K}, a_t, V^e_t, r_{t+1}, x_{t+1})$ to compute Eq. (10) and (11);
   
   Update the STGCN neural network by Eq. (12) and (13);

Function stgcn_ppo (in_training):

1. $V^e_t = \text{stgcn}_\text{critic}(x_t, \cdots, x_{t-K})$;
2. if in_training = True then
   
   $a^e_t = \text{stgcn}_\text{actor}(x_t, \cdots, x_{t-K})$, where $a^e_t$ is the probability of actions;
   
   Set agent policies to sample stochastic actions $a_t$;
3. else
   
   $a^e_t = \text{stgcn}_\text{actor}(x_t, \cdots, x_{t-K})$;
   
   Set agent policies to sample static actions $a_t$;
4. return actions $a_t$, values $V^e_t$;

Function ChooseRewardState ($a_t$):

1. Get rewards $[r_t]_1 = r^{o_0}_{n_0, t}$ by Eq. (9) and states $x_t$ from the environment envs ($a_t$);
2. return $x_t, r_t$;

IV. CASE STUDIES

In this section, we adopt and the 123-bus feeder distribution systems to validate the proposed STGCN-DRL algorithm, mitigating instabilities and maintaining nodal voltage profiles within a desirable range in the face of unbalanced load. The PV smart inverters bus locations, including all uncompromised and compromised ones on the 123-bus feeder, are shown in Fig. 3. We train these agents using the OpenDSS environment to estimate the grid response. The formulation and training for the proposed STGCN-DRL are performed by PyTorch.

The numerical experiments are divided in two parts. In the first part, we apply the proposed STGCN-DRL algorithm to control the smart inverters for oscillation mitigation, in combination with voltage regulation. We also validate the can of reduced-size input, and correspondingly reduced GSO in the STGCN scheme. Finally, we consider the impact of having a larger number of smart inverters in the system.

In the second part, we compare the proposed STGCN-DRL architecture with existing DRL methods focusing purely on voltage magnitude regulation. This is a problem that has been widely studied since the distribution networks have unbalanced demands on each phase, leading to voltage magnitudes that potentially go beyond the desired ranges. The target is to regulate the voltage magnitudes of the buses, leveraging the PV installed. We compare the learning curves of the proposed CGCN and RGCN with fully connected neural networks (FNN) and convolutions neural networks (CNN) in terms of voltage magnitude regulation performance, as measured by the same reward function. Here we set the weight of the oscillation component of the reward $\zeta_1 = 0$ for a fair comparison with the aforementioned schemes that have been proposed by other authors.

Fig. 4: This system has 3 smart inverters. (a) and (b) illustrate the learning (training) and testing curves of the STGCN-DRL. (c) and (d) illustrate the $r^{o_0}_{n_0, t}$ and voltage magnitude curves without the STGCN defense, respectively. (e) and (f) illustrate the $r^{o_0}_{n_0, t}$ and voltage magnitude curves with the STGCN defense, respectively.
A. Experiment Setup

We utilize historical PV and load data with a 1-minute resolution from the PecanStreet dataset for training and testing. The dataset includes 25 household loads and 14 PV panel powers from May 01, 2019 to July 31, 2019. In this dataset, we choose 4800 samples for training, and 2400 samples for testing. We show the average training curves in the oscillation mitigation application, i.e., Figs. 4(a), 5(a) and 6(a), where the bands represent the standard deviation over 16 runs. In Fig. 3, we have three PV smart inverters installed in the load buses (Bus 51a, Bus 53a, Bus 60a) with one compromised smart inverter (Bus 53a). In particular, the action range is set from $-0.05$ p.u to $0.05$ p.u with action range discretization $\Delta \eta = 0.01$ p.u. In order to mitigate voltage unbalance, we set the desired voltage magnitude $\bar{v} = 1$ p.u. We set $c = 5000$ for the high and low pass filters, and $\bar{v} = 1.05$ p.u. and $\bar{v} = 0.95$ p.u. The default configuration is $\eta = [0.94, 0.96, 1.04, 1.06, 1.1]$. Each epoch has 128 training samples. We set $\zeta_r = 1$, $\zeta_v = 0.2$, $\zeta_p = 0.005$ in Figs. 4(a) and 4(c), and $\zeta_r = 0.8$, $\zeta_v = 0.1$, $\zeta_p = 0.005$ in Fig. 5(a).

We introduce the STGCN-DRL specification 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 1.0. For the GCN part, we set $K = 4$ and $T = 10$ for the oscillation mitigation control. For the STGCN layer, the spatial channel is 10, and the temporal channel is 10 for both CGCN and RGCN. After the STGCN layer that performs the feature extraction, we have 512 neurons of a fully-connected neural networks to approximate the policy, and then the output layer.

| Systems                  | Bus Name |
|--------------------------|----------|
| 123-bus feeder with PMUs | 1.1, 1.2, 1.3, 10.1, 100.1, 100.2, 100.3, 101.1, 101.2, 101.3, 102.3, 103.3, 104.3, 108.1, 108.2, 105.4, 105.2, 105.3, 106.2, 107.2 |

B. Full Observations

The cyber attack is launched by translating the VV/VW curves and shifting $\eta$ of the compromised smart inverter to
induce an oscillation (the setup is borrowed from [6]). In regulations with full and partial observations, respectively.

Fig. 7: (a) and (b) have 3 smart inverters and (c) has 6 smart inverters. (a) and (b) illustrate the learning (training) curves of the STGCN-DRL algorithm for voltage magnitude results are shown in TABLE I. In particular, we select 20

selected according to the algorithm in Section III.C.2. The reduction algorithm to design GSO. The measurements are observations of voltage phasors. We apply the proposed Kron

C. Partial Observations

In this subsection, we test the idea of using partial observations of voltage phasors. We apply the proposed Kron-reduction algorithm to design GSO. The measurements are selected according to the algorithm in Section III.C.2. The results are shown in TABLE I. In particular, we select 20

buses from 278 buses (∼14% of the buses) in the three-phase 123-bus feeder system for CGCN and RGCN, respectively. Considering that distribution feeders include thousands of buses, this would brings the cost for PMU measurements systems to reasonable levels.

We illustrate the performance of STGCN with the following numerical tests, which are similar to the previous ones that used the full state observation. The first example shows the [ro_n,t] and the voltage curves without the policy defense in Fig 4(c) and Fig 4(e). In Fig. 4(e) a voltage oscillation event happens from t = 55s to t = 205s. Its [ro_n,t] surges when the attack is launched in Fig. 4(c). After the attack ends, the voltage magnitudes are about 0.96 per unit. With the policy beginning at t = 10s, [ro_n,t] is always very small even though the attack is launched from t = 55s to t = 205s. The corresponding voltage magnitude curves are shown in Fig 4(f) and demonstrate that the policy not only regulates voltage magnitudes within desired ranges, but also defends against the oscillation attack, that is lasting from t = 55s to t = 205s.

D. Comparison With Other Neural Networks

In this section, we compare CGCN and RGCN with benchmark algorithms in terms of voltage deviation regulation as well as learning stability in the training phase. The benchmark algorithm FNN-DRL has 3 layers with 512 neurons each layer. Another benchmark algorithm CNN-DRL has three hidden layers with the 32, 64 and 32 output channels, respectively. We only consider voltage deviation regulation for three smart inverters by setting ζ = 0 in Eq. (8). The learning curves of RGCN-DRL, CGCN-DRL, FNN-DRL, and CNN-DRL with the full and partial observations are shown in Fig. 7(a) and Fig. 7(b) respectively. The two figures show the average training reward, where the bands represents the standard deviation over 5 runs. In particular, the results in Fig. 7(a) show that the VDs of RGCN and CGCN, i.e., \(\sum_{n \in N_a} rd_{n,a}\), are 0.0425 p.u. and 0.0390 p.u., which outperform FNN and CNN that have 0.0610 p.u. and 0.0799 p.u., respectively. Another observation is that RGCN and CGCN are competitive in convergence time and performance. With the partial observations (20 out of 278), the results in Fig 7(b) show that \(\sum_{n \in N_a} rd_{n,a}\) of CGCN and RGCN converge into 0.0538 p.u. and 0.0535 p.u., respectively. However, \(\sum_{n \in N_a} rd_{n,a}\) of FNN and CNN
converge into 0.0668 p.u. and 0.0684 p.u., respectively. When we scale these policies with more smart inverters, Fig. 7(d) shows the learning curves of FNN and CNN decreases after 50 episodes, which indicates that they trigger the deadly triad of DRL. In contrast, CGCN and RGCN validate the excellent performance of enhancing the stability of DRL.

V. CONCLUSIONS

This paper proposed a novel STGCN-DRL algorithm to control the smart inverters in unbalanced distribution systems. The proposed STGCN-DRL algorithm uses graph filters to extract more efficiently the features of the voltage phasors that are relevant to the control policy. The general GCN structure is specialized to two architecture, the CGCN and RGCN models, to consider both spatial and temporal correlations. Moreover, we utilize the low-pass property of down-sampled voltage graph signal to prove that even having roughly 15% of the state values allows to define a reduced GSO that makes the STGCN-DRL very effective as compared to the schemes that use the full state. Our STGCN-DRL algorithm is capable of controlling the smart inverters in unbalanced distribution systems.

APPENDIX

To obtain the GSO in the real domain, the power flow equations are expressed as follows.

\[ i_n = \sum_{m \in \mathcal{N}_n} \left[ \frac{1}{2} Y_{mn}^s + Y_{mn}^{(n)} \right] v_n + Y_{mn}^{(m)} v_m \]

Note that, in general, for distribution lines \( Y_{mn}^{(m)} = -Y_{mn}^{(n)} \) with the exception of transformer or regulators. In the following analysis, we omit the influence of transformer or regulators so that we assume \( Y_{mn}^{(m)} = Y_{mn}^{(n)} \).

The power flowing from bus \( n \in \mathcal{N} \) to bus \( m \in \mathcal{N} \) is:

\[ s_n = D \left( v_n (i_n^H) \right), \forall \phi \in \mathcal{P}_n, n \in \mathcal{N}, \]

where \( D(\cdot) \) is the operator that returns the diagonal values of a matrix, and the superscript \( H \) stands for Hermitian.

Taking \( |\mathcal{P}_n| = 3 \) for example, we assume \( B = \mathbb{R}(Y) \) dominates over the conductance (i.e. \( \mathbb{R}(Y) \)) and have:

\[ s_n \approx D \left( \sum_{m \in \mathcal{N}_n} -j [v_n v_n^H \left( \frac{1}{2} B_{mn}^s + B_{mn}^{(n)} \right)^T + v_n v_m^H (B_{mn}^{(m)})^T] \right), \]

where \( B_{mn}^s, B_{mn}^{(n)} \) and \( B_{mn}^{(m)} \) are symmetric.

\[ v_n = \begin{bmatrix} v_n | e^{j \varphi_{na}}, \varphi_{na}, v_n | e^{j \varphi_{nb}} | \varphi_{na}, | e^{j \varphi_{nc}} \end{bmatrix}^T. \]

The goal is to obtain two decoupled equations that are real and that describe the dependence between the active and reactive power on the magnitude and phases of \( v_n v_n^H \) and \( v_n v_m^H \), using the expansion \( e^{j x} = 1 + j x \) for the phase terms of \( v_n^* \) and \( v_m^* \), re-centered respectively around their respective phases in the balanced case \( \psi = [1, e^{-12\pi/3}, e^{12\pi/3}]^T \). More specifically, let \( \bar{v}_n = [\bar{\varphi}_{na}, \bar{\varphi}_{nb}, \bar{\varphi}_{nc}] \triangleq \varphi_{na} \), \( \bar{v}_n = \frac{2\pi}{3}, \varphi_{nc} = -\frac{2\pi}{3} \). Also, let \( \Psi_n^{(3)} \triangleq \text{diag}(\psi) \), where \( \text{diag}(\cdot) \) which is a diagonal matrix with diagonal elements equal to the entries of the vector in the argument of \( \text{diag}(\cdot) \). We denote \( \circ \) as the Hadamard product. Let \( c_a \) and \( c_b \) represent \( \cos(2\pi) \) and \( \sin(2\pi) \), respectively. \( \text{I} \) is the all-ones vector and \( \text{J} = [1, \ldots, 1]^T \) is the all-ones matrix. In the following, we assume \( \Psi_n^{(3)} = \Psi_m^{(3)} \) without other shifts by some specific electrical elements, such as transformers.

We also present the following propositions that will be useful in our transformation. Note that \( A, B, C \) and \( E \) are real square matrices, and \( a \) and \( b \) are real vectors.

**Proposition 1 (P1)** If \( C \) and \( E \) are diagonal matrices, then \( C(A \circ B)E = A \circ (CBE) \).

**Corollary 1 (C1)** If \( C \) and \( E \) are diagonal matrices, then \( CAE = A \circ (C(11^T)E) = A \circ (\text{diag}(C)(\text{diag}(E))^T) \).

**Proposition 2 (P2)** \( D(AB) = \sum_j (A \circ B_j^T)_{ij} \).

**Proposition 3 (P3)** If \( B \) and \( C \) are symmetric, \( D((A \circ B)C) = D(A(B \circ C)) = D(A(B \circ C)^T) \).

**proof:**

\[ D((A \circ B)C) = \sum_i ((A \circ B)C_{ij}) = \sum_i (A \circ (C^T \circ B_{ij})) = \sum_i (A \circ (C \circ B_j^T))_{ij} = D(A(C \circ B)) = D(A(B \circ C)). \]

**Proposition 4 (P4)** \( D(ab^T) = \text{diag}(b)a \) and \( D(A) = D(A^T) \).

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 \( P1 \) or \( C1 \), with blue color. By adding and subtracting from the phase angle in \( v_n \), we obtain:

\[ v_n = \Psi_n^{(3)} \text{diag}(v_n) \begin{bmatrix} e^{j \bar{\varphi}_{na}} \\ e^{j \bar{\varphi}_{nb}} \\ e^{j \bar{\varphi}_{nc}} \end{bmatrix} = \Psi_n^{(3)} \text{diag}(v_n) e^{j \bar{\varphi}_n}. \]

Therefore, the outer product \( v_n v_m^H \) are:

\[ v_n v_m^H = \Psi_n^{(3)} \text{diag}(v_n) e^{j(\bar{\varphi}_n + \bar{\varphi}_m^T - \bar{\varphi}_m^T)} \text{diag}(v_m) (\Psi_m^{(3)})^H. \]

\[ \Psi_n^{(3)} \Psi_m^{(3)^H} \text{diag}(v_m) \]

\[ \Xi_{\Psi_n}^{(3)} \Xi_{\Psi_m}^{(3)^H} \text{diag}(v_m) \]

\[ \Xi_{\Psi_n}^{(3)} \text{diag}(v_m) \]

\[ \Xi_{\Psi_m}^{(3)^H} \text{diag}(v_m) \]

\[ \Xi_{\Psi_n}^{(3)} \text{diag}(v_m) \]

\[ \Xi_{\Psi_m}^{(3)^H} \text{diag}(v_m) \]
where $\Gamma$ can be expressed as:

$$\Gamma = \begin{bmatrix} 1 & e^{\frac{i\pi}{2}} & e^{-\frac{i\pi}{2}} \\ e^{-\frac{i\pi}{2}} & 1 & e^{\frac{i\pi}{2}} \\ e^{\frac{i\pi}{2}} & e^{-\frac{i\pi}{2}} & 1 \end{bmatrix} = \begin{bmatrix} 1 & e^{\frac{i\pi}{2}} & e^{-\frac{i\pi}{2}} \\ e^{-\frac{i\pi}{2}} & 1 & e^{\frac{i\pi}{2}} \\ e^{\frac{i\pi}{2}} & e^{-\frac{i\pi}{2}} & 1 \end{bmatrix}$$

Next we use the approximation in developing the component relative to the phase term we use the approximation\(^3\) that $|v_n| \approx 1$ and $|v_m| \approx 1$ in \([42]\). With this approximation, we substitute \([42]\) in \([38]\). Therefore, the first real part of \([38]\) is

$$\Re\left\{-D\left[v_n v_n^H \left(\frac{1}{2} B_{mn}^s + j B_{mn}^{(n)}\right)\right]\right\} =$$

$$-D\left\{\Re\left[\left(\left[J - (\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T) \odot \Gamma\right) \left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right)\right]\right]\right\} = D\left\{J \odot \Gamma_s + (\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T) \odot \Gamma_c\right\} \left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right)$$

(43)

We separate the biased part that does not involve in $(\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T)$ from \([43]\), and define it as:

$$p_n^{inc} \triangleq D \left(\Gamma_s \left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right)\right).$$

(44)

where $J \odot \Gamma_s = \Gamma_s$. The remaining part of \([43]\) that involves in $(\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T)$, denoted by $p_n^{out}$, can be expressed as:

$$p_n^{out} \triangleq D \left(\frac{A_{inc}}{B_{inc}} \right) \left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right)$$

(45)

$$= D \left(\hat{\varphi}_n \left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right) \odot \Gamma_c\right)$$

(46)

$$= D \left\{\hat{\varphi}_n \left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right) \odot \Gamma_c\right\}$$

(47)

Note that $\hat{B}_{mn}^s$ and $\hat{B}_{mn}^{(n)}$ are symmetric due to the fact that the Hadamard product is commutative and $\Gamma_c$, $B_{mn}^s$ and $B_{mn}^{(n)}$ are symmetric.

Replacing $m$ with $n$ and $\left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right)$ with $\hat{B}_{mn}^{(n)}$, this form applies also to $v_n v_n^H$. Therefore, the second real part of \([38]\) is

$$\Re\left\{-D\left(v_n v_n^H \left(j B_{mn}^{(n)}\right)\right)\right\} =$$

$$= D \left\{\Gamma_s + (\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T) \odot \Gamma_c\right\} \left(\hat{B}_{mn}^{(n)}\right)$$

(48)

Likewise, we separate the biased part that does not involve in $(\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T)$ from \([48]\), and define it as:

$$p_n^{out} \triangleq D \left(\Gamma_s \left(\hat{B}_{mn}^{(n)}\right)\right).$$

(49)

The remaining part of \([48]\) that involves in $(\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T)$, denoted by $p_n^{inc}$, can be expressed as:

$$D \left[\left((\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T) \odot \Gamma_c\right) \left(\hat{B}_{mn}^{(n)}\right)\right] =$$

$$= \text{diag} \left[\left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right) \odot \left(\hat{B}_{mn}^{(n)}\right)\right]$$

(50)

Finally, by excluding $p_n^{inc}$ and $p_n^{out}$ from GSO, we have

$$\hat{p}_n \triangleq p_n - p_n^{inc} - p_n^{out} =$$

$$= \sum_{m \in N_n} \left\{\text{diag} \left[\left(\frac{1}{2} B_{mn}^s + B_{mn}^{(n)}\right) \odot \left(\hat{B}_{mn}^{(n)}\right)\right]\right\} \hat{\varphi}_n$$

(51)

To write it in a compact way, we have

$$\hat{p} = \hat{B} \hat{\varphi},$$

(52)

where $\hat{p}$ an $\hat{\varphi}$ are denoted by

$$\hat{p} = \left[\begin{array}{c} \hat{p}_1 \\ \vdots \\ \hat{p}_{|N|} \end{array}\right], \quad \hat{\varphi} = \left[\begin{array}{c} \varphi_1 \\ \vdots \\ \varphi_{|N|} \end{array}\right],$$

(53)

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

B. Reactive Power Injection

The reactive power analysis is similar to the active power analysis. In particular, we use the approximation that $\hat{\varphi}_n 1^T - \hat{\varphi}_n 1^T = 0$ in \([42]\), where $0$ is the all-zeros matrix. Therefore, the first part of \([38]\) is

$$-D \left(v_n v_n^H \left(\frac{1}{2} B_{mn}^s + j B_{mn}^{(n)}\right)\right) \approx -D \left\{\begin{bmatrix} C_{inc} \odot \text{diag}(v_n) & A_{inc} \odot \Gamma_c \odot \text{diag}(v_n) & E_{inc} \odot \text{diag}(v_n) \end{bmatrix} \left(\frac{1}{2} B_{mn}^s + j B_{mn}^{(n)}\right)\right\}$$

(54)

Then we take the imaginary part of \([54]\) as:

$$-D \left\{\left(\text{diag}(v_n)\otimes \Gamma_c\right) \left(\frac{1}{2} B_{mn}^s + j B_{mn}^{(n)}\right)\right\}$$

(55)
By summing (57) and (58) together, we have

\[
q_n \approx -D \left\{ \left( \|v_n\|^T + |v_n|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) + \hat{B}_{mn}^{(n)} \right\} - D \left\{ \left( \|v_m\|^T \right) \circ \left( \hat{B}_{mn}^{(m)} \right) \right\}.
\]

(59)

where we add and minus this item, i.e., \(v_n\|v_n\|^T\), in order to split (59) into three parts, i.e., \(\hat{q}_n^{inc}\), \(\hat{q}_n^{inc}\), and \(\hat{q}_n^{ext}\). Specifically, we have

\[
\hat{q}_n^{inc} \triangleq -D \left\{ \left( \|v_n\|^T - |v_n|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) \right\}
\]

(60)

\[
= D \left( \left( \|v_n\|^T - |v_n|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) \right) \quad \text{P1} \succeq \text{P4}
\]

(61)

\[
= \text{diag} \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) |v_n| - \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) |v_n|.
\]

(62)

where the transformation from (61) to (62) is the same as the transformation from (46) to (47). Likewise, we could replace \(m\) with \(n\) and \(\hat{B}_{mn}^{(m)}\) with \(\hat{B}_{mn}^{(m)}\), and have the second part:

\[
\hat{q}_n^{inc} \triangleq -D \left\{ \left( \|v_n\|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) \right\}
\]

(63)

\[
\hat{q}_n^{ext} \triangleq -D \left\{ \left( \|v_n\|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s \right) \right\}.
\]

(65)

With \(|v_n|\) \(\approx 1\), \(\hat{q}_n^{ext}\) could be relaxed as a biased part that does not involve in \(v_n\|v_n\|^T - \|v_n\|^T\):

\[
\hat{q}_n^{ext} \approx \hat{q}_n^{inc} \triangleq -D \left\{ \left( \|v_n\|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s \right) \right\}
\]

(66)

Finally, by excluding \(\hat{q}_n^{ext}\) from GSO, we have

\[
q_n = -D \left\{ \left( \|v_n\|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) \right\} - \left\{ \text{diag} \left[ \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right] \right\} |v_n|.
\]

(67)

In the same way with active power injects, we have

\[
\hat{q} = \hat{B} |v|,
\]

(68)

where \(\hat{p}\) an \(\hat{\varphi}\) are denoted by

\[
q \left[ \begin{array}{c} 1 \\ \vdots \end{array} \right], \quad \varphi \left[ \begin{array}{c} \varphi_1 \\ \vdots \end{array} \right].
\]

(69)

Therefore, we have the following

\[
\begin{pmatrix}
\text{GSO} \\
\text{GS}
\end{pmatrix} =
\begin{pmatrix}
\hat{B} & 0 \\
0 & \hat{B}
\end{pmatrix}
\]

\[
\begin{pmatrix}
\hat{q} \\
\hat{\varphi}
\end{pmatrix} =
\begin{pmatrix}
\hat{B} & 0 \\
0 & \hat{B}
\end{pmatrix}
\]

where \(x = \left[ \begin{array}{c} \varphi \\varphi \end{array} \right] \) is the output of graph filters \(S^{-1}\) and \(S = \left[ \begin{array}{cc} \hat{B} & 0 \\
0 & \hat{B}\end{array} \right] \) are graph filters in our problem.

\[
q_n \approx -D \left\{ \left( \|v_n\|^T \right) \circ \left( \frac{1}{2} \hat{B}_{mn}^s + \hat{B}_{mn}^{(n)} \right) \right\}
\]

References

[1] H. Liu and W. Wu, “Two-stage deep reinforcement learning for inverter-based volt-var control in active distribution networks,” IEEE Transactions on Smart Grid, vol. 12, no. 3, pp. 2037–2047, 2020.

[2] Y. Gao, W. Wang, and N. Yu, “Consensus multi-agent reinforcement learning for volt-var control in power distribution networks,” IEEE Transactions on Smart Grid, 2021.

[3] X. Sun and J. Qiu, “Two-stage volt/var control in active distribution networks with multi-agent deep reinforcement learning method,” IEEE Transactions on Smart Grid, 2021.

[4] Y. Zhang, X. Wang, J. Wang, and Y. Zhang, “Deep reinforcement learning based volt-var optimization in smart distribution systems,” IEEE Transactions on Smart Grid, vol. 12, no. 1, pp. 361–371, 2021.

[5] D. Cao, J. Zhao, W. Hu, N. Yu, F. Ding, Q. Huang, and Z. Chen, “Deep reinforcement learning enabled physical-model-free two-timescale voltage control method for active distribution systems,” IEEE Transactions on Smart Grid, 2021.

[6] C. Roberts, S.-T. Ngo, A. Milesi, S. Peisert, D. Arnold, S. Saha, A. Scaglione, N. Johnson, A. Kocheturov, and D. Fridkin, “Deep reinforcement learning for DER cyber-attack mitigation,” in 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 2020, pp. 1–7.

[7] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, “Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision,” arXiv preprint arXiv:2102.01780, 2021.

[8] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” IEEE Signal Processing Magazine, vol. 34, no. 6, pp. 26–38, 2017.

[9] R. Jiang, T. Zahavy, Z. Xu, A. White, M. Hessel, C. Blundell, and H. Van Hasselt, “Emphatic algorithms for deep reinforcement learning,” in Proceedings of the 35th International Conference on Machine Learning, vol. 139. PMLR, 18–24 Jul 2021, pp. 5023–5033.

[10] H. Van Hasselt, Y. Doron, F. Strub, M. Hessel, N. Sonnerat, and J. Modayil, “Deep reinforcement learning and the deadly triad,” arXiv preprint arXiv:1812.02648, 2018.

[11] D. Detering, X. Bresson, and P. Vanderheyst, “Convolutional neural networks on graphs with fast localized spectral filtering,” Advances in neural information processing systems, vol. 29, pp. 3844–3852, 2016.

[12] J. G. Vlachogiannis and N. D. Hatziargyriou, “Reinforcement learning for reactive power control,” IEEE Transactions on Power systems, vol. 19, no. 3, pp. 1317–1325, 2004.

[13] H. Xu, A. D. Domínguez-García, and P. W. Sauer, “Optimal tap setting of voltage regulation transformers using batch reinforcement learning,” IEEE Transactions on Power Systems, vol. 35, no. 3, pp. 1990–2001, 2019.

[14] Q. Yang, G. Wang, A. Sadeghi, G. B. Giannakis, and J. Sun, “Two-timescale voltage control in distribution grids using deep reinforcement learning.”
learning.” IEEE Transactions on Smart Grid, vol. 11, no. 3, pp. 2313–2323, 2019.

[15] J. Duan, D. Shi, R. Diao, H. Li, Z. Wang, B. Zhang, D. Bian, and Z. Yi, “Deep-reinforcement-learning-based autonomous voltage control for power grid operations,” IEEE Transactions on Power Systems, vol. 35, no. 1, pp. 814–817, 2019.

[16] S. Wang, J. Duan, D. Shi, C. Xu, H. Li, R. Diao, and Z. Wang, “A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning,” IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4644–4654, 2020.

[17] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in International conference on machine learning. PMLR, 2018, pp. 1861–1870.

[18] D. Cao, W. Hu, J. Zhao, Q. Huang, Z. Chen, and F. Blaabjerg, “A multi-agent deep reinforcement learning based voltage regulation using coordinated pv inverters,” IEEE Transactions on Power Systems, vol. 35, no. 5, pp. 4120–4123, 2020.

[19] T. Zhao and J. Wang, “Learning sequential distribution system restoration via graph-reinforcement learning,” IEEE Transactions on Power Systems, 2021.

[20] X. Y. Lee, S. Sarkar, and Y. Wang, “A graph policy network approach for volt-var control in power distribution systems,” arXiv preprint arXiv:2109.12077, 2021.

[21] 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.

[22] A. S. Zamzam and N. D. Sidiroopoulos, “Physics-aware neural networks for distribution system state estimation,” IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4347–4356, 2020.

[23] M. Liang, Y. Meng, J. Wang, D. L. Lubkeman, and N. Lu, “Feedergan: Synthetic feeder generation via deep graph adversarial nets,” IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1163–1173, 2020.

[24] C. Roberts, S.-T. Ngo, A. Milesi, A. Scaglione, S. Peisert, and D. Arnold, “Deep reinforcement learning for mitigating cyber-physical der voltage unbalance attacks,” in 2021 American Control Conference (ACC). IEEE, 2021, pp. 2861–2867.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, 2017.

[26] R. Ramakrishna and A. Scaglione, “Grid-Graph Signal Processing (Grid-GSP): A Graph Signal Processing Framework for the Power Grid,” IEEE Transactions on Signal Processing, vol. 69, pp. 2725–2739, 2021.

[27] L. Ruiz, F. Gama, and A. Ribeiro, “Graph neural networks: Architectures, stability, and transferability,” Proceedings of the IEEE, vol. 109, no. 5, pp. 660–682, 2021.

[28] R. Ramakrishna and A. Scaglione, “On modeling voltage phasor measurements as graph signals,” in 2019 IEEE Data Science Workshop (DSW). IEEE, 2019, pp. 275–279.

[29] A. Anis, A. Gadde, and A. Ortega, “Efficient sampling set selection for bandlimited graph signals using graph spectral proxies,” IEEE Transactions on Signal Processing, vol. 64, no. 14, pp. 3775–3789, 2016.