Reinforcement Learning using Physics Inspired Graph Convolutional Neural Networks

Tong Wu, *Member*, Anna Scaglione, *Fellow, IEEE*, Daniel Arnold, *Member, IEEE*

2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton) | 979-8-3503-9998-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ALLERTON49937.2022.9929321

***Abstract*—In this work, we propose a physics inspired Graph Convolutional Neural Network (GCN)-Reinforcement Learning (RL) architecture to train online controllers policies for the optimal selection of Distributed Energy Resources (DER) set-points. While the use of GCN is compatible with any DRL scheme, we test it in combination with the popular proximal policy optimization (PPO) algorithm and, as application, we consider the selection of set-points for Volt/Var and Volt/Watt control logic of smart inverters as the case study for DER control. We are able to show numerically that the GCN scheme is more effective than various benchmarks in regulating voltage and mitigating undesirable voltage dynamics generated by cyber-attacks. In addition to exploring the performance of GCN for a given network, we investigate the case of grids that are dynamically changing due to topology or line parameters variations. We test the robustness of GCN-RL policies against small perturbations and evaluate the scheme so called "transfer learning" capabilities.**

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 (DERs) 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 representations for signals whose supports are large-scale networks. When appropriately applied it is shown that they are empirically capable of 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. 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 Work

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 but the approach requires the full state and presents convergence issues [1]. Very recently, [19, 20] leveraged GNN in their DRL design. However, the authors ignored the temporal correlation of their time series. and require the full system state.

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. 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 the STGCN architecture for the extraction of spatio-temporal features from the voltage phasors of the system.

Tong Wu 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, as337}@cornell.edu). Daniel Arnold is with Lawrence Berkeley National Laboratory (e-mail: dbarnold@lbl.gov).

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.

- 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.

D. Notation

The symbols and notations are summarized as follows. The grid has an associated graph with N nodes, whose set is denoted by $\mathcal{N} = \{1, \cdots, N\}$ and a set of lines that are the graph edges $\mathcal{E} \subsetneq \mathcal{N} \times \mathcal{N}$ 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 $v_{n_\phi} \in \mathbb{C}$ be the complex line-to-ground voltage at node $n \in \mathcal{N}$ of phase $\phi \in \mathcal{P}_n$. Let $\boldsymbol{v}$ represent the vector of voltage phasors over all buses, i.e., $\boldsymbol{v} = [\boldsymbol{v}_1^\top, \cdots, \boldsymbol{v}_N^\top]^\top$ with the n^{th} entry $\boldsymbol{v}_n = [v_{n_\phi} | \phi \in \mathcal{P}_n] \in \mathbb{C}^{|\mathcal{P}_n| \times 1}$ with voltage phase θ_{n_ϕ} and voltage magnitude $|v_{n_\phi}|$. The vector sizes of the current injection phasors $\boldsymbol{i}$, apparent power injections $\boldsymbol{s}$, active and reactive power injections, $\boldsymbol{p}$ and $\boldsymbol{q}$, respectively, are consistent with the size of $\boldsymbol{v}$. $\boldsymbol{s} = \boldsymbol{p} + \mathrm{j}\boldsymbol{q}$ be the vector of net apparent power, where $\mathrm{j} = \sqrt{-1}$ represents the imaginary unit.

II. Spatio-Temporal Graph ConvNet-based Deep Reinforcement Learning

Neural networks are known for their function approximation capabilities. In the context of DRL neural networks provide a parametric form for approximating the policy and value function that is general enough to come close to the theoretical optimum. The parametric form of the STGCN-based policy functions sacrifice generality by internalizing the inherent local correlation structures of data that originate from network interactions. The underlying abstraction is the notion of graph filter, generalizing of the notion of time-filters in digital signal processing (DSP) that inspired convolutional neural networks. In this section, motivated by the power flow equations which govern the structure of power systems state data, we first introduce graph filters and unveil the key ingredient to apply their abstraction to such data based on the grid physical constraints. Then, we review the proximal policy optimization (PPO) method, and introduce the STGCN-DRL framework.

A. Physics-Aware Grid Graph Filters in the real domain

The goal of this work is to present a DRL design based on STGCN whose GCN layers are designed based on the physics of the power flow equations, extending our results in [21]. In a nutshell, by using graph filters one can enhance the representation capabilities of the feature extraction layers compared to conventional neural networks.

To define graph filters we need to introduce some notation and the definition of Graph Shift Operator (GSO). Let $\mathcal{G} = (\mathcal{V}, \mathcal{L})$ be a graph, with vertex set $\mathcal{V}$ and edge set $\mathcal{L}$. A graph signal $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{V}|}$ is a vector indexed by the network nodes (e.g. for the grid the state vector of the voltage phasors at each bus). 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 is an neighborhood operator that is, its entries can only mix neighbors' values. Almost all operations including filtering, transformation and prediction are directly related to the GSO 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:

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

This is the case in the proposed DRL architecture, where we will use a real symmetric GSOs, i.e., $\mathbf{S} = \mathbf{S}^\top$. 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 [1]:

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

It is important to remark that there is a lot of freedom in choosing the GSO and, in many applications, there are different types of data that are indexed by the nodes that one can choose from, and different way to interpret the results of filtering. This is where the application for graph signal processing requires a further layer of care compared to conventional multi-dimensional filtering and the modeling effort may require to integrate domain expertise. This is the case of grid applications, where there are three nodal (bus) signals to choose from, apparent power injections, currents, voltages, and the AC power flow, as well as Ohm's law which are at the basis of the data structure spatially.

The first physics aware notion of GSO that interpreted these equations as a generative model for the voltage phasor data appeared in [22], where the authors proposed to use a GSO equal to the system admittance matrix $\boldsymbol{Y}$. The latter is a complex symmetric weighted graph Laplacian, and [22] extended several basic notions of real valued graph signal processing to cope with the differences. The extensive tools that exist for real-valued Graph NN are not, however, directly accessible using the insights in [22]. In fact, the naïve idea of writing the algebra in terms of real and imaginary parts works poorly. Let $\boldsymbol{v}$ represent the vector of the voltage phasors at each bus on the grid, and $\angle\boldsymbol{v}$ and $|\boldsymbol{v}|$ the vectors of its phase angles and amplitudes respectively. Intuitively, the grid topology constrains the $\angle\boldsymbol{v}$ and $|\boldsymbol{v}|$ to be similar at neighboring nodes, which is the graph equivalent of smoothness characterizing *low-pass graph signals* [22]. No such property can be

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

claimed for the real and imaginary parts of $\boldsymbol{v}$. This is why, in [21, Proposition 1], we have derived a physics-aware GSO is in the real domain for three-phase unbalanced distribution network and for a graph signal $\boldsymbol{x}$ concatenating $\angle\boldsymbol{v}$ and $|\boldsymbol{v}|$. More specifically, starting from the AC powerflow equations, [21] we showed that they yield the following approximate relationship:

$$\begin{bmatrix}\boldsymbol{p}\\ \boldsymbol{q}\end{bmatrix} - \begin{bmatrix}\boldsymbol{p}^{cst}\\ \boldsymbol{q}^{cst}\end{bmatrix} = \overbrace{\begin{bmatrix}\hat{\mathbf{B}} & \mathbf{0}\\ \mathbf{0} & \hat{\mathbf{B}}\end{bmatrix}}^{\mathbf{S}:\ \text{GSO}} \overbrace{\begin{bmatrix}\boldsymbol{\varphi}\\ |\boldsymbol{v}|\end{bmatrix}}^{\boldsymbol{x}:\ \text{GS}} = \mathbf{S}\boldsymbol{x} \tag{4}$$

where $\boldsymbol{x}$ and $\mathbf{S}$ are graph filters in our problem, and $\boldsymbol{\varphi}$ is obtained re-centering voltage phase angles $\angle\boldsymbol{v}$, by subtracting their balanced three-phase component, and $|\boldsymbol{v}|$ is the vector of voltage magnitudes and the matrix $\hat{\mathbf{B}}$, which defines the GSO, is a function of the system susceptance matrix $\mathbf{B}$, which is defined as

$$\begin{aligned}&\hat{\mathbf{B}} \triangleq ((\mathbb{1}\mathbb{1}^\top)_N \otimes \Gamma_c) \circ \mathbf{B},\\ &[\Gamma_c]_{kn} = \Re[e^{\mathrm{j}\frac{2(k-n)\pi}{3}}],\ k,n \in \{0,1,2\}.\end{aligned} \tag{5}$$

where $\circ$ is the Hadamard product and $\otimes$ is the Kronecker product. The two vectors $\boldsymbol{p}^{cst}$ and $\boldsymbol{q}^{cst}$ are:

$$\begin{aligned}\boldsymbol{p}_n^{cst} &\triangleq \tfrac{1}{2}D\left(\Gamma_s\mathbf{B}_{mn}^s\right), & \boldsymbol{q}_n^{cst} &\triangleq -\frac{1}{2}D(\hat{\mathbf{B}}_{mn}^s),\\ [\Gamma_c]_{kn\in\{0,1,2\}} &= \Im[e^{\mathrm{j}\frac{2(k-n)\pi}{3}}], & \hat{\mathbf{B}}_{mn}^s &\triangleq \Gamma_c \circ \mathbf{B}_{mn}^s,\end{aligned}$$

and $D(\boldsymbol{A})$ is a vector whose entries are the diagonal entries of the matrix $\boldsymbol{A}$ in the argument.

The notable facts are: 1) the relationship between phase angles and active power resembles that of the DC power-flow equations, with subtle differences; 2) a similar relationship is found for the reactive power and the voltage amplitudes; 3) the two are decoupled. Naturally, the lack of coupling is an artifact of our approximations, but one needs to keep in mind that the subsequent layers of the NN architecture mix the outputs of all entries that emerge from the graph filter layer and, thus, can learn the patterns that relate the features of the phase angles and those of the voltage magnitudes.

B. 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 π_θ, parameterized by θ, model the probability distribution of $\boldsymbol{a}_t \in \mathcal{A}$ given a sequence of observations $\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}$, 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:

$$\pi^* \in \arg\max J(\pi) = \mathbb{E}_{\mu\sim\pi_\theta}\left[\sum_{t=0}^{T}\gamma^\top r_t(\boldsymbol{x}_t, \boldsymbol{a}_t)\right], \tag{6}$$

where μ is the trajectory generated by policy π_θ, i.e., the action $\boldsymbol{a}_t$ is taken according to policy $\pi_\theta(\cdot|\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t})$, r_t represents rewards at time t (e.g. $r_t = r_{n_\phi,t}^{os}$ in (22)), and $\boldsymbol{x}_t$ denotes the state observation at time t. The parameter $\gamma \in (0,1)$ is the discounting factor, discounting future rewards.

2) Policy Gradient: Let $V_\vartheta^\pi(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t})$ be the value function parametrized by ϑ, 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 (18). For PPO, the gradient of $J(\theta)$ is:

$$\begin{aligned}\nabla J(\theta) = \mathbb{E}_{\mu\sim\pi_\theta}\Bigg[\sum_{t=0}^{T_b}\nabla_\theta \log\pi_\theta(\boldsymbol{a}_t|\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t})\\ A_\vartheta^\pi(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}, \boldsymbol{a}_t)\Bigg],\end{aligned} \tag{7}$$

$$\begin{aligned}A_\vartheta^\pi(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}, \boldsymbol{a}_t) = r_t + \gamma V_\vartheta^\pi(\boldsymbol{x}_{t+1}, \cdots, \boldsymbol{x}_{t+1-K_t})\\ -V_\vartheta^\pi(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}),\end{aligned} \tag{8}$$

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

$$\theta_{k+1} = \theta_k + \alpha\nabla_\theta J(\theta), \tag{9}$$

$$\begin{aligned}\vartheta_{k+1} = \vartheta_k - \beta\nabla_\vartheta(r_t + V_\vartheta^\pi(\boldsymbol{x}_{t+1}, \cdots, \boldsymbol{x}_{t+1-K_t})\\ -V_\vartheta^\pi(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}))^2,\end{aligned} \tag{10}$$

where α and β are the constant step sizes. A PPO version using a clipped surrogate objective simplifies the aforementioned method and yields similar performance [23]:

$$\begin{aligned}&L^{\text{CLIP}}(\theta) = \hat{\mathbb{E}}\left[\min\left(\rho_t(\theta)\hat{A}_t, clip(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right],\\ &\rho_t(\theta) \triangleq \frac{\pi_\theta(\boldsymbol{a}_t|\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t})}{\pi_{\theta_{old}}(\boldsymbol{a}_t|\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t})},\\ &\hat{A}_t \triangleq A_\vartheta^\pi(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}, \boldsymbol{a}_t),\end{aligned} \tag{11}$$

where clip operation encourages a more gradual update to the policy rather than large charges. The minimum between the unclipped and the clipped objective values is used to bound the unclipped objective.

C. STGCN-based Deep Reinforcement Learning

Power systems are dynamic systems with time-varying voltage phasors. In order to fuse features from both spatial and temporal domains in the DRL policy and value function we propose to use the following feature extraction layer (layer1) [21]:

$$\boldsymbol{w}_t^c = \sum_{k=0}^{K}\hbar_k\mathbf{S}^k\left(\sum_{\tau=0}^{K_t}h_{k,t}\boldsymbol{x}_{t-\tau}\right). \tag{12}$$

where $\hbar_k$ and $h_{k,t}$ are trainable parameters. As mentioned before the goal is to approximate the policy function and value function using graph neural networks. They are:

$$\boldsymbol{\pi}_\theta = \sigma\big(\Theta_3^\pi \cdot \overbrace{\sigma\left(\Theta_2 \cdot \boldsymbol{w}_t^c\right)}^{\text{2nd layer}}\big), \boldsymbol{V}_\vartheta = \sigma\big(\Theta_3^V \cdot \overbrace{\sigma\left(\Theta_2 \cdot \boldsymbol{w}_t^c\right)}^{\text{2nd layer}}\big), \tag{13}$$

where $\theta \triangleq \{(\Theta_3^\pi, \Theta_2, \hbar_k, h_{k,t})|\forall k, t\}$ represent the parameters of $\boldsymbol{\pi}_\theta$ 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 $\boldsymbol{V}_\vartheta$

share the STGCN layer and the second layer with the policy function. Therefore, the parameters of $\boldsymbol{V}_{\vartheta}$ is defined as: $\vartheta \triangleq \{(\Theta_3^V, \Theta_2, \hbar_k, h_{k,t}) | \forall k, t\}$. The resulting DRL algorithm is summarized in Algorithm 1.

Algorithm 1: The STGCN-DRL Algorithm

```
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 $\boldsymbol{a}_0$;
Function RunPPO($in\_training$):
    Initialize state $\boldsymbol{x}_0, \boldsymbol{r}_0 = ChooseRewardState(\boldsymbol{a}_0)$,
      where $\boldsymbol{r}_0$ is a vector with each element $r^{os}_{n_\phi,t=0}$
      corresponding to each agent;
    for $t \leftarrow 0$ to $T_{max}$ do
        for $j \leftarrow 0$ to $RB$ do
            Get actions and values
              $\boldsymbol{a}_t, \boldsymbol{V}^\pi_t = stgcn\_ppo(in\_training)$, where
              $\boldsymbol{V}^\pi_t$ is a vector $[\boldsymbol{V}^\pi_t]_i = V^\pi_{n_\phi,t}$ with respect to
              each agent;
            Get new state and rewards
              $\boldsymbol{x}_{t+1}, \boldsymbol{r}_{t+1} = ChooseRewardState(\boldsymbol{a}_t)$;
            Store them as a transition $(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}, \boldsymbol{a}_t,$
              $\boldsymbol{V}^\pi_t, \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1})$ in replay buffer;
        Sample from the replay buffer to obtain tuple
          $(\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}, \boldsymbol{a}_t, \boldsymbol{V}^\pi_t, \boldsymbol{r}_{t+1}, \boldsymbol{x}_{t+1})$ to compute
          (7) and (8);
        Update the STGCN neural network by (9) and (10);
Function stgcn_ppo($in\_training$):
    $\boldsymbol{V}^\pi_t$ = stgcn_critic($\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}$);
    if $in\_training$ = $True$ then
        $\boldsymbol{a}^p_t$ = stgcn_actor($\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}$), where $\boldsymbol{a}^p_t$ is the
          probability of actions;
        Set agent policies to sample stochastic actions $\boldsymbol{a}_t$;
    else
        $\boldsymbol{a}^p_t$ = stgcn_actor($\boldsymbol{x}_t, \cdots, \boldsymbol{x}_{t-K_t}$);
        Set agent policies to sample static actions $\boldsymbol{a}_t$;
    return actions $\boldsymbol{a}_t$, values $\boldsymbol{V}^\pi_t$;
Function ChooseRewardState($\boldsymbol{a}_t$):
    Get rewards $[\boldsymbol{r}_t]_i = r^{os}_{n_\phi,t}$ by (22) and states $\boldsymbol{x}_t$ from the
      environment $envs(\boldsymbol{a}_t)$;
    return $\boldsymbol{x}_t$, $\boldsymbol{r}_t$;
```

Next, we introduce an application of the proposed scheme to smart-total inverters.

III. Case study: Policies for Smart Inverters

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 proposed 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. 1(c) and Fig. 1(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 $\boldsymbol{\eta} = [\eta_1, \cdots, \eta_5]^\top \in \mathbb{R}^5$. Mathematically, they are:

$$f^q_n(|\tilde{v}_{n_\phi}|) \triangleq \begin{cases} \bar{q} & |\tilde{v}_{n_\phi}| \in [0, \eta_1] \\ \left(\frac{\eta_2 - |\tilde{v}_{n_\phi}|}{\eta_2 - \eta_1}\right)\bar{q} & |\tilde{v}_{n_\phi}| \in (\eta_1, \eta_2] \\ 0 & |\tilde{v}_{n_\phi}| \in (\eta_2, \eta_3] \\ -\left(\frac{\eta_3 - |\tilde{v}_{n_\phi}|}{\eta_4 - \eta_3}\right)\bar{q} & |\tilde{v}_{n_\phi}| \in (\eta_3, \eta_4] \\ -\bar{q} & |\tilde{v}_{n_\phi}| \in (\eta_4, \infty) \end{cases} \tag{14}$$

$$f^p_i(|\tilde{v}_{n_\phi}|) \triangleq \begin{cases} \tilde{p} & |\tilde{v}_{n_\phi}| \in [0, \eta_4] \\ \left(\frac{\eta_5 - |\tilde{v}_{n_\phi}|}{\eta_5 - \eta_4}\right)\bar{p} & |\tilde{v}_{n_\phi}| \in (\eta_4, \eta_5] \\ 0 & |\tilde{v}_{n_\phi}| \in (\eta_5, \infty] \end{cases} \tag{15}$$

where $\bar{p}$ and $\bar{q}$ are the active and reactive powers injected into the system, in response to the estimated voltage amplitude $|\tilde{v}_{n_\phi}|$ obtained by low-pass filtering the measured voltage magnitude signal at bus n on phase ϕ, 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:

$$\begin{aligned} &|\tilde{v}_{n_\phi,t}| = |\tilde{v}_{n_\phi,t-1}| + \tau^c_n(|v_{n_\phi,t}| - |\tilde{v}_{n_\phi,t-1}|), \\ &\bar{q}^2 + (f^p(|\tilde{v}_{n_\phi,t}|))^2 \leq \bar{s}^2, \end{aligned} \tag{16}$$

where τ^c_n 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:

$$\begin{aligned} p_{n_\phi,t} &= p_{n_\phi,t-1} + \tau^o(f^p_n(|\tilde{v}_{n_\phi,t}|) - p_{n_\phi,t-1}), \\ q_{n_\phi,t} &= q_{n_\phi,t-1} + \tau^o(f^q_n(|\tilde{v}_{n_\phi,t}|) - q_{n_\phi,t-1}), \end{aligned} \tag{17}$$

where τ^o 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 as action a specific inverter setpoints, and the environment the agent acts upon, 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 $(\boldsymbol{x}_t, \ldots, \boldsymbol{x}_{t-K_t})$, the action $\boldsymbol{a}_t$, and the reward $\boldsymbol{r}_t$. We describe these three elements next.

1) State and Action: The tuple of actions for the VVCs of the inverters in the bus set $\mathcal{N}_s$ is denoted by:

$$\begin{aligned} &\boldsymbol{a} = [a_1, a_2, \cdots, a_i, \cdots, a_{|\mathcal{N}_s|}]^\top, a_i \in \mathcal{A}_i, \\ &\boldsymbol{\eta}'_i = \boldsymbol{\eta}^o_i + a_i * \mathbb{1}, \end{aligned} \tag{18}$$

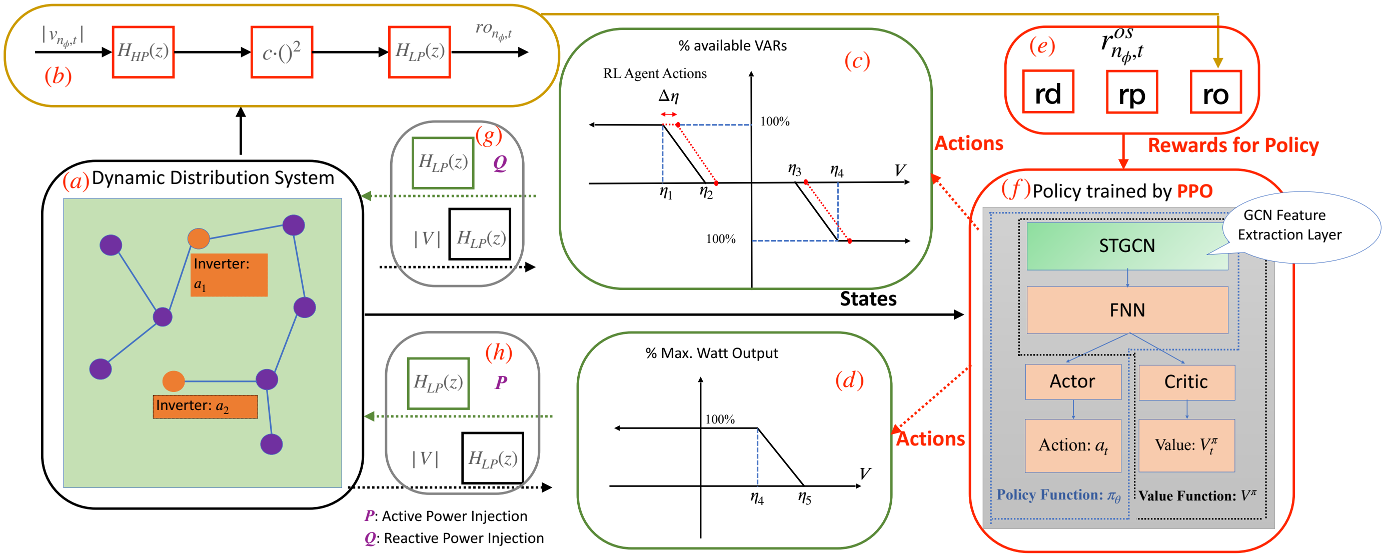


Fig. 1: Overview of STGCN for VVC.

where $|\mathcal{N}_s|$ represents the number of smart inverters, a_i denotes the control action on the ith smart inverter, $\mathbb{1} \in \mathbb{R}^5$ represents a vector of all one elements, $\mathcal{A}_i$ represents the action space of the ith action, and $\boldsymbol{\eta}_i^o$ and $\boldsymbol{\eta}_i'$ define the default and shifted values of VV-VW curves that define the intervals in (14) and (16). As shown in Fig. 1, in this paper, instead of controlling arbitrarily the parameters of the VV/VW curves, we shift only by a common constant the intervals [6, 24]. The vector of actions $\boldsymbol{a}$, 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 $\boldsymbol{x} = [\boldsymbol{\varphi}; |\boldsymbol{v}|]^\top$, where $\boldsymbol{\varphi}$ is the vector of voltage phase angels and $|\boldsymbol{v}|$ 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_ϕ at time t as follows [13, 18]:

$$\mathrm{rd}_{n_\phi,t} = \left||v_{n_\phi,t}| - \bar{v}\right|, n_\phi \in \mathcal{N}_s, \tag{19}$$

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 ϕ, respectively. The second component of the total regret that penalizes the active power curtailment is:

$$\mathrm{rp}_{n_\phi,t} = \left(1 - \frac{p_{n_\phi,t}}{p^{\max}_{n_\phi,t}}\right)^2, n_\phi \in \mathcal{N}_s, \tag{20}$$

where $\mathcal{N}_s$ 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, in Fig. 1(b), we define the oscillation regret $\mathrm{ro}_{n_\phi,t}$:

$$\mathrm{ro}_{n_\phi,t} = h_t^{LP} \star \left(h_t^{HP} \star |v_{n_\phi,t}|\right)^2, \tag{21}$$

where h_t^{HP} is the impulse response of a high-pass filter (here we just used the finite difference), and h^{LP} is a low-pass filter (e.g. a moving average), and $\star$ defines the convolution operator. Finally, the reward function for each smart inverter is:

$$r^{os}_{n_\phi,t} = -(\zeta_v \times \mathrm{rd}_{n_\phi,t} + \zeta_p \times \mathrm{rp}_{n_\phi,t} + \zeta_o \times \mathrm{ro}_{n_\phi,t}), n_\phi \in \mathcal{N}_s, \tag{22}$$

where ζ_v, ζ_p and ζ_o are positive weights. In particular, the third term ζ_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 $|\mathcal{N}_s|$ is composed of two subsets, denoted by $\mathcal{C}$ and $\mathcal{U}$, representing the "compromised" and "uncompromised" inverter, respectively. We assume that $\mathcal{U} \neq \emptyset$ and use DRL to determine the optimum stochastic policy π_θ which represents the probability distribution of action $\boldsymbol{a} \in \mathcal{A}$ given a sequence of observations; θ are the neural network parameters that the DRL selects to maximize the reward defined in (22).

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(e)) 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(g) 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.

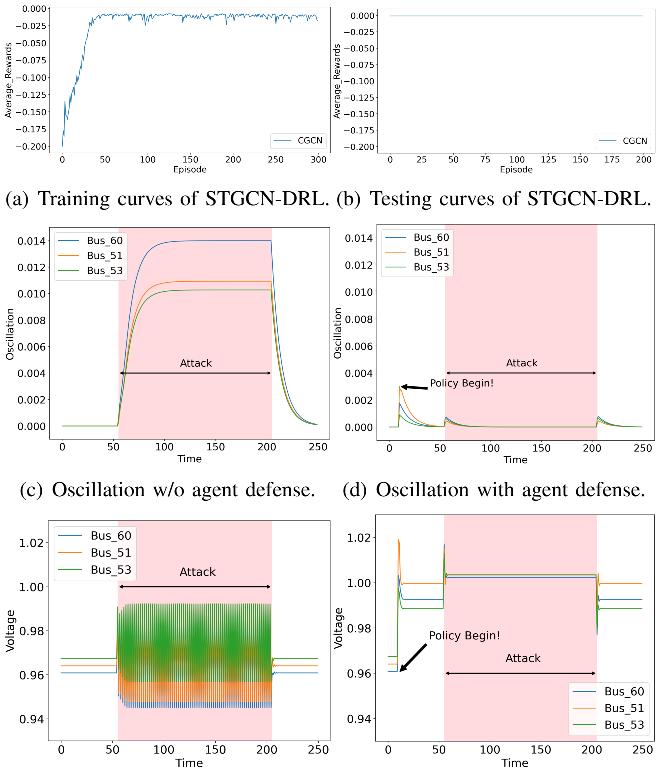
IV. Numerical Analysis

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. 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.

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[2]. In this dataset, we choose 4800 samples for training, and 2400 samples for testing. In Figs. 2(a) and 4(a) we show the training curves averaged over 16 runs in the oscillation mitigation application (the bands represents the standard deviation). 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 discrete step $\Delta\eta = 0.01$ 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 $\overline{V} = 1.05$ p.u. and $\underline{V} = 0.95$ p.u. The default configuration is $\boldsymbol{\eta} = [0.94, 0.96, 1.04, 1.06, 1.1]$. Each epoch has 128 training samples. We set $\zeta_o = 1$, $\zeta_v = 0.2$ $\zeta_p = 0.005$ in Figs. 2(a) and 2(c) and $\zeta_o = 0.8$, $\zeta_v = 0.1$ $\zeta_p = 0.005$ in Fig. 4(a).

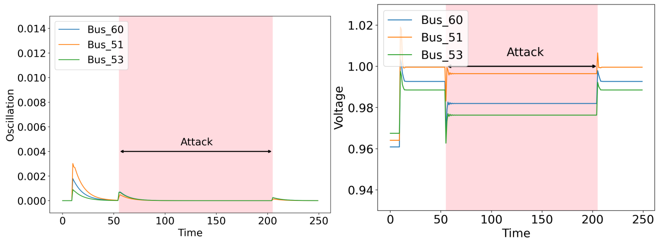


(a) Training curves of STGCN-DRL. (b) Testing curves of STGCN-DRL.

(c) Oscillation w/o agent defense. (d) Oscillation with agent defense.

(e) VM curves w/o STGCN-DRL. (f) VM curves with STGCN-DRL.

Fig. 2: This system has 3 smart inverters. (a) and (b) illustrate the learning (training) and testing curves of the STGCN-DRL. In (c) and (d), "Oscillation" represents the oscillation regret $\mathrm{ro}_{n_\phi,t}$ with the STGCN defense and without the STGCN defense, respectively. (e) and (f) illustrate the voltage magnitude curves with the STGCN defense and without the STGCN defense, respectively.

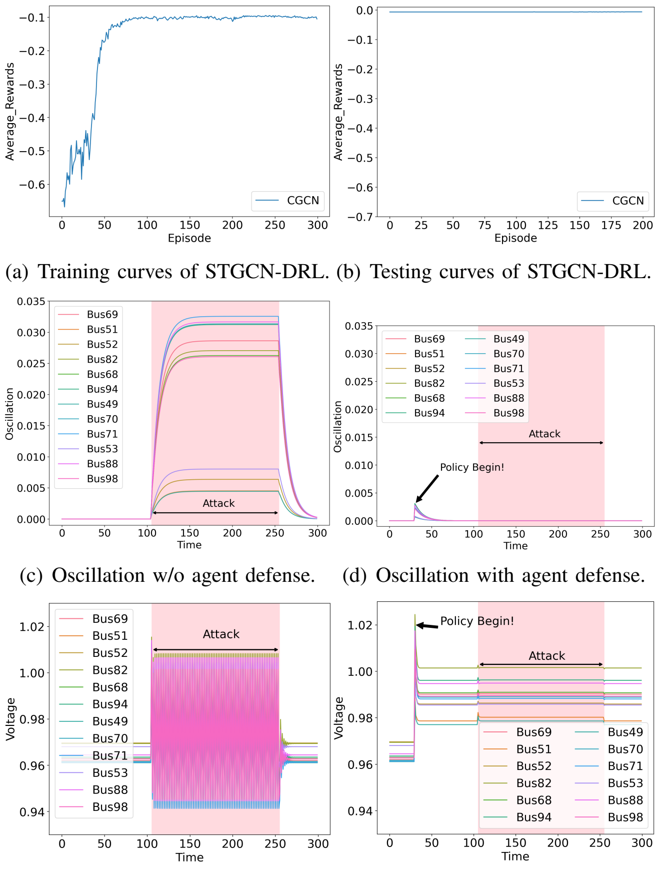


(a) Oscillation with agent defense. (b) VM curves with STGCN-DRL.

Fig. 3: Transfer Learning with GSO changes. This system has 3 smart inverters. (a) and (b) illustrate the $\mathrm{ro}_{n_\phi,t}$ and voltage magnitude curves by the STGCN defense with GSO changes, respectively.

The STGCN-DRL parameters are as follows. The learning rate is 0.0007. The discounted factor γ is 0.99. The PPO clip parameter ϵ 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 the graph filter order is 10, and the temporal filter order is also 10. 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. The network is quite effective, even though it is relatively shallow.

[2] https://www.pecanstreet.org/dataport/

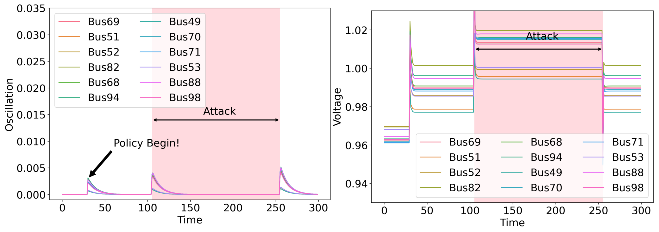


(a) Training curves of STGCN-DRL. (b) Testing curves of STGCN-DRL.

(c) Oscillation w/o agent defense. (d) Oscillation with agent defense.

(e) VM curves w/o STGCN-DRL. (f) VM curves with STGCN-DRL.

Fig. 4: This system has 12 smart inverters. (a) and (b) illustrate the learning (training) and testing curves of the STGCN-DRL. In (c) and (d), "Oscillation" represents the oscillation regret $\mathrm{ro}_{n_\phi,t}$ with the STGCN defense and without the STGCN defense, respectively. (e) and (f) illustrate the voltage magnitude curves with the STGCN defense and without the STGCN defense, respectively.



(a) Oscillation with agent defense. (b) VM curves with STGCN-DRL.

Fig. 5: Transfer Learning with GSO changes. This system has 12 smart inverters. (a) and (b) illustrate the $\mathrm{ro}_{n_\phi,t}$ and voltage magnitude curves by the STGCN defense with GSO changes, respectively.

B. Results

The cyber attack is launched by shifting the VV/VW curves of the compromised smart inverters till they induce oscillations (the setup is borrowed from [6]). In Fig. 2(a), after 300 epochs, the total reward $\sum_{n_\phi} r^{os}_{n_\phi,t}$ increases greatly from -0.2 to -0.01. Fig. 2(b) shows that in the tests of the STGCN-DRL policy. The values of $|\sum_{n_\phi} r^{os}_{n_\phi,t}|$ are very small, indicating that the agents STGCN-DRL policy successfully activated in the uncompromised agents is successful in damping the voltage profile oscillations. Recall also that small values of $|\sum_{n_\phi} r^{os}_{n_\phi,t}|$ indicate that the voltage magnitudes are very closed to 1 p.u, which means that the system is also dealing well with unbalanced injections.

The first example without the policy defense is shown in Fig.2(c) and Fig.2(e) on the left. In Fig.2(c) and Fig.2(d), the y-axis "Oscillation" represents the oscillation regret $\mathrm{ro}_{n_\phi,t}$ in Eq. (21). In Fig. 2(e), the attack starts at time $t = 55$s and ends at time $t = 205$s. Correspondingly, the oscillation regret $\mathrm{ro}_{n_\phi,t}$ surges when the attack is launched in Fig. 2(c). After the attack ends, the voltage magnitudes are very far from 1 p.u. In contrast, in Fig. 2(f), before the policy begins, the voltage magitudes are around 0.96 p.u because of the unbalanced demands and inadequate reactive power support. When the STGCN policy begins at $t = 10$s, the STGCN policy regulates the voltage magnitudes closed to 1 p.u. by providing the reactive power support. The staging of the attack is attained by shifting VV/VW curves of compromised inverters so that excessive amounts of reactive power VV are injected or drawn from the distribution network; this behavior, sustained from $t = 55$s to $t = 205$s, create oscillation events. The STGCN policy controlling the uncompromised smart inverters responds to the jump in oscillation regret $\mathrm{ro}_{n_\phi,t}$ and is able to tame its value to be very small in the attack time window. After the attack ends at $t = 205$s, the STGCN policy continues to regulate voltage magnitudes closed to 1 p.u. by taking optimal actions that shift VV/VW curves.

In the grid, the system matrix changes as well, due to switching or control equipment and retraining a new model based on the new environment is time-consuming. In this experiment, we test the transferability of the proposed STGCN-DRL parameters to a case where the grid has changed topology, by simply changing the GSO but retaining all other STGCN parameters the same. In the simulation, the change corresponds to a line tripping in the original system. The results shown in Fig. 3, are very similar to those in the previous case, where the training and testing environment had the same GSO (i.e. no line tripped). This clearly indicates that STGCN-DRL perform well in the new environment and retraining is not necessary for small changes.

C. Large penetration of Smart Inverters

We further increase the ratio of compromised inverters to 50% in the 123-bus feeder, including 6 out of 12 inverters in the subset. In particular, Fig. 4(a) and Fig. 4(b) show the training and testing curves of STGCN-DRL, respectively. Also in this case, without the policy defense, the values of the oscillation regret $\mathrm{ro}_{n_\phi,t}$ shown in Fig. 4(c) are high during the attack, lasting from $t = 105$s to $t = 255$s with significant oscillations shown in Fig. 4(e). However, with the policy defense in Fig. 4(d), the oscillations are swiftly mitigated, and the voltage magnitudes rapidly converge close to the desired values of 1 per unit. The values of the oscillation regret $\mathrm{ro}_{n_\phi,t}$, shown in Fig. 2(f), are also very small. Also in the presence of more smart inverters, the STGCN parameters are robust to changes in the GSO, as illustrated in Fig. 5 which shows that the STGCN-DRL still performs well in the new environment determined by having a line tripped in the system.

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 the STGCN model, to consider both spatial and temporal correlations. Our STGCN-DRL algorithm is capable of both mitigating oscillation due to unwanted dynamics caused by a set of compromised inverters, while maintaining nodal voltage profiles within a desirable range.

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. Fradkin, "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:2102.01168*, 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 38th 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] M. Defferrard, X. Bresson, and P. Vandergheynst, "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," *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:2109.12073*, 2021.

[21] T. Wu, I. L. Carreno, A. Scaglione, and D. Arnold, "Graph convolutional neural networks for physics-aware grid learning algorithms," 2022. [Online]. Available: https://arxiv.org/abs/2203.16732

[22] 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.

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

[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.