Physics-Informed Heterogeneous Graph Neural Networks for DC Blocker Placement

Hongwei Jin*, Prasanna Balaprakash†, Allen Zou‡, Pieter Ghysels‡, Aditi S. Krishnapriyan§,
Adam Mate¶, Arthur Barnes¶, and Russell Bent‖

*Mathematics and Computer Science Division
Argonne National Laboratory, Lemont, IL

†Computing and Computational Sciences Directorate
Oak Ridge National Laboratory, Oak Ridge, TN

‡Computational Research Division
Lawrence Berkeley National Laboratory, Berkeley, CA

§ University of California, Berkeley, CA

¶Information Systems and Modeling Group and ‖Applied Mathematics and Plasma Physics Group
Los Alamos National Laboratory, Los Alamos, NM

Abstract—The threat of geomagnetic disturbances (GMDs) to the reliable operation of the bulk energy system has spurred the development of effective strategies for mitigating their impacts. One such approach involves placing transformer neutral blocking devices, which interrupt the path of geomagnetically induced currents (GICs) to limit their impact. The high cost of these devices and the sparsity of transformers that experience high GICs during GMD events, however, calls for a sparse placement strategy that involves high computational cost. To address this challenge, we developed a physics-informed heterogeneous graph neural network (PIHGNN) for solving the graph-based dc-blocker placement problem. Our approach combines a heterogeneous graph neural network (HGNN) with a physics-informed neural network (PINN) to capture the diverse types of nodes and edges in ac/dc networks and incorporates the physical laws of the power grid. We train the PIHGNN model using a surrogate power flow model and validate it using case studies. Results demonstrate that PIHGNN can effectively and efficiently support the deployment of GIC dc-current blockers, ensuring the continued supply of electricity to meet societal demands. Our approach has the potential to contribute to the development of more reliable and resilient power grids capable of withstanding the growing threat that GMDs pose.

Index Terms—geomagnetic disturbance, geomagnetically induced current mitigation, blocking devices, physics-informed machine learning, graph neural networks.

I. INTRODUCTION

Geomagnetic disturbances (GMDs) pose a serious threat to the continuous and reliable operation of the bulk energy system [1]. During severe space weather, charged and magnetized particles are blown away from the Sun. These disrupt the Earth's magnetic field and drive geomagnetically induced currents (GICs) – quasi-dc currents produced by the induced low-frequency electric fields – in the conductive infrastructure, which flow into the high-voltage network through the neutrals of transformers [2], [3]. Depending on intensity, GICs can adversely impact transmission networks and equipment through overheating and thermal degradation in transformers, misoperation in protective devices, voltage collapses and the loss of reactive power support, and in the worst case, widespread infrastructure damage and tripping of transmission lines, leading to cascading power outages [3]–[6].

A handful of strategies, for both system operation (transmission line switching, generation re-dispatch, and load-shedding) and system planning (placement of dc-blockers or series capacitors, and equipment stockpiling), have been explored over the years to mitigate the impacts of GMDs. This paper focuses on the placement of transformer neutral blocking devices – the installation of GIC dc-current blockers – which consists of injecting a shunt capacitor in series with the grounding point of transformer neutrals to interrupt the path of GICs [7]. Unlike other strategies, this approach does not reduce the power-carrying capacity of the grid, but may interfere with ground-fault relaying, which in turn requires additional switching devices and increased capital costs. Given the high initial costs of these devices, and that the number of transformers that experience high GICs during GMD events is sparse [8], sparse dc-blocker placement is necessary. This motivates the need for optimization-based placement formulations to guide the choices of where to place blocking devices.

Determining the optimal placement of GIC blocking devices is a challenging optimization problem formulated as a difficult mixed-integer non-linear program (MINLP). Developing effective techniques to solve such complex MINLP models for large-scale power systems has been an active research area for decades [9]–[11].

Recently, machine learning (ML) approaches have shown great promise for tackling these challenging optimization problems by training ML models to predict high-quality solutions efficiently [12]–[14], avoiding the need to solve the optimization directly. These ML surrogate models have demonstrated impressive performance on traditional power

systems optimization tasks.

Motivated by these promising ML developments, this paper explores applying machine learning techniques to the critical problem of optimally placing GIC blocking devices to mitigate geomagnetically induced current impacts during solar storms. We develop a novel ML surrogate model approach that leverages recent advances in graph neural networks and physics-informed neural networks to effectively capture the complex network structure and physical laws governing power grid systems.

More specifically, the application of graph neural networks (GNNs) in power grid systems has been a rapidly developing area of research in recent years. For example, [15] applied GNNs to learn a local and scalable parameterization of the optimal power flow (OPF) solution, while [16] learned to perform ac OPF computations by directly minimizing the violation of physical laws during training. Additionally, [17] proposed a new approach to predicting short-term power generation from renewable energy sources, combining a graph convolutional network and a long short-term memory model to capture both spatial and temporal correlations. Furthermore, [18] presents a novel approach for solving reliability assessment commitments (RAC) optimization problems in power systems using GNNs and uncertainty quantification.

However, these models are designed for homogeneous graphs, which cannot fully capture the complex relationships and dependencies between different types of nodes and edges in ac/dc networks. The physics of GMD impacts on power system cause induced low-frequency voltages on electric transmission lines. A reasonable approximation is to model these as dc voltages. GICs can be calculated on transmission lines by generating a dc network from an ac transmission network model. In this dc network, the nodes correspond to both buses in the ac network and substation grounding grids, while edges correspond to transmission lines and transformer windings. This dc network is linked to the ac transmission network by reactive power loss resulting from half-cycle saturation on transformers. This is calculated as a function of the weighted sum of GICs flowing though each winding of a transformer and mapped to one of the transformer terminals [19]. We will further discuss the details in Section III-A.

In addition, physics-informed neural networks [20] have drawn attention in recent years to address the problem of learning from data while respecting the underlying domain. With such an approach, constraints can be added to the ML model to respect the underlying physics [20]–[22], or by combining more explicit information from numerical solvers with the ML model [23], [24]. In this work, we incorporate the physical information of the power grid into the ML training process. This gap in the power grid literature motivates our research to explore the potential of heterogeneous graph neural networks in solving the dc-current blocker placement problem while ensuring the satisfaction of physical laws.

This paper proposes a novel heterogeneous graph neural network (HGNN) approach to model the complex ac/dc power grid network and predict optimal locations for placing GIC dc-current blockers to mitigate geomagnetic disturbance impacts. The HGNN captures the diverse node and edge types present in integrated ac/dc networks through tailored node and edge embeddings. Building upon this base HGNN model, we develop an innovative physics-informed heterogeneous graph neural network (PIHGNN) that incorporates the fundamental physical laws governing power grid operation. The PIHGNN combines the HGNN's predictions with a surrogate power flow model solver to ensure predictions respect the nonlinear ac/dc power flow constraints.

To the best of our knowledge, this is the first work that combines the powerful representational capabilities of graph neural networks for capturing network structure with physics-informed neural network techniques for enforcing physical law constraints. Applying this novel PIHGNN approach to the critical problem of optimizing GIC dc-current blocker placement is an important contribution. By addressing this gap between data-driven graph representations and physical power system models, our research enables more reliable and resilient power grid operation in the face of growing geomagnetic disturbance threats. The proposed PIHGNN model provides a cost-effective and computationally efficient machine learning surrogate for supporting real-world GIC blocker deployment decisions.

The rest of the paper is structured as follows: In Section II., we introduce the mathematical formulation of blocker placement with objectives and constraints. In Section III., we describe our ML surrogate approach and the methodology used to train and validate the model. In Section IV., we present two case studies that demonstrate the effectiveness of our approach and discuss the results. Finally, in Section V., we conclude with a brief discussion of the implications of our research and future directions for improving power grid resilience against GMDs with the use of neural networks.

II. Problem Formulation

In this section, we discuss the GIC dc-current blocker placement formulation from an optimization problem perspective. This formulation consists of four distinct pieces: the formulation used to model ac power flows in a transmission network, the formulation used to model GIC flows in a transmission network, the formulation that links GIC and ac formulations, and the formulation for modeling blockers. It is important to note that while the ac and GIC flows are formulated for the same physical system, the formulations utilize unique representations of the system.

A. AC Power Flow

The ac power flow constraints consist primarily of the set of equations that represent the physics of power flow on an ac transmission network, and the set of inequalities that represent the operational limits of the network. This is formulated as a MINLP maximum-load-delivered (MLD) formulation, as described in [25].

1) Objective:

The problem objective is to maximize the amount of load delivered. This can be alternately formulated as minimizing the cost associated with load-shedding:

$$\min. \sum_{i\in\mathcal{D}} \left|p_i^d\right| \kappa_i^d z_i^d \tag{1}$$

where $\mathcal{D}$ is the set of loads in the ac network; p_i^d is the dispatched power of load i; κ_i^d is the cost of shedding load i; and $z_i^d \in \{0,1\}$ is the load-shedding variable corresponding to load i.

2) Nodal power balance constraint:

$$\sum_{j\in\mathcal{E}_i^{a+}} \left(\mathbf{s}_j^+ - \mathbf{i}q_j^{loss+}\right) - \sum_{j\in\mathcal{E}_i^{a-}} \left(\mathbf{s}_j^- + \mathbf{i}q_j^{loss-}\right)$$
$$= \sum_{j\in\mathcal{G}_i} \mathbf{s}_j^g - \sum_{j\in\mathcal{D}_i} z_j^d \mathbf{s}_j^d - \mathbf{y}_{\mathbf{i}}^{\mathbf{s}} \left|\mathbf{v}_i\right|^2, \quad \forall i \in \mathcal{N}^a$$

where $\mathcal{N}^a$ is the set of ac buses; $\mathcal{E}_i^{a+}$ is the set of branches flowing into bus i; $\mathcal{E}_i^{a-}$ is the set of branches flowing out of bus i; $\mathbf{s}_j$ is the complex power flow through branch j; q_j^{loss} is the reactive power loss of the branch resulting from GIC; $\mathcal{G}_i$ is the set of generators connected to bus i; $\mathbf{s}_k^g$ is the complex power produced by generator k; $\mathcal{D}_i$ is the set of loads connected to bus i; $\mathbf{s}_k^d$ is the scheduled power of load k; $z_k^d \in [0,1]$ is the load shed variable; $\mathbf{y}_i^s$ is the shunt admittance of bus i; and $\mathbf{v}_i$ is the voltage phasor at bus i.

3) Branch flow constraint:
$\forall\, i \in \mathcal{E}^a$

$$\mathbf{s}_i^+ = \left(\mathbf{y}_i^* - \mathbf{i}\frac{\mathbf{b}_i^{sh}}{2}\right) \left|\mathbf{v}_j\right|^2 - \mathbf{y}_i^* \mathbf{v}_j \mathbf{v}_k^* \tag{2a}$$

$$\mathbf{s}_i^- = \left(\mathbf{y}_i^* - \mathbf{i}\frac{\mathbf{b}_i^{sh}}{2}\right) \left|\mathbf{v}_k\right|^2 - \mathbf{y}_i^* \mathbf{v}_j^* \mathbf{v}_k \tag{2b}$$

where $\mathcal{E}^a$ is the set of ac edges; $\mathbf{y}_i$ is the series admittance of branch i; and $\mathbf{b}_i^{sh}$ is the line charging capacitance (equal to 0 for transformers).

4) AC Operational Limit Constraints:

The voltage limits for each node (bus) $i \in \mathcal{N}^a$ are defined by,

$$\underline{v}_i \leq \left|\mathbf{v}_i\right| \leq \overline{v}_i \tag{3}$$

where the lower and upper bounds of the voltage magnitude at node $i \in \mathcal{N}^a$ are set by the range $[\underline{v}_i, \overline{v}_i]$.
The thermal limits of an edge $i \in \mathcal{E}^a$ are defined by,

$$\left|\mathbf{s}_i^-\right| \leq \overline{s}_i \tag{4a}$$

$$\left|\mathbf{s}_i^+\right| \leq \overline{s}_i \tag{4b}$$

where $\overline{s}_i$ is the thermal limit of edge i.
Voltage stability limits on the phase angle difference on edge i, between nodes j and k, are defined by,

$$\underline{\theta}_i \leq \arg(\mathbf{v}_k) - \arg(\mathbf{v}_j) \leq \overline{\theta}_i \tag{5}$$

where $\underline{\theta}_i$ and $\overline{\theta}_i$ represent the upper and lower bounds on the phase angle difference.

Given the set of generators $\mathcal{G}$, the capacity of power generation of $i \in \mathcal{G}$ is defined by,

$$\underline{p}_i^g \leq \mathcal{R}(\mathbf{s}_i^g) \leq \overline{p}_i^g \tag{6a}$$

$$\underline{q}_i^g \leq \mathcal{I}(\mathbf{s}_i^g) \leq \overline{q}_i^g \tag{6b}$$

where $\mathcal{R}(\mathbf{s})$ and $\mathcal{I}(\mathbf{s})$ are the real and imaginary components of is the real part of $\mathbf{s}$, respectively, $\underline{p}_i^g$ and $\overline{p}_i^g$ represent the lower and upper bounds on real power for generator i; and $\underline{q}_i^g$ and $\overline{q}_i^g$ represent the lower and upper bounds on real power for generator i.

B. GIC Flow

The dc power flow constraints consist of the set of equations that represent the physics of GIC flow on a dc representation of the transmission network. For a node $i \in \mathcal{N}^d$, the nodal flow balance constraint is defined by,

$$\sum_{j\in\mathcal{E}_i^{d+}} I_j - \sum_{j\in\mathcal{E}_i^{d-}} I_j = (1 - z_i^b)a_i^s V_i \tag{7}$$

where $\mathcal{N}^d$ is the set of nodes in the dc network; $\mathcal{E}_i^{d+}$ is the set of directed edges connected to node i in the dc network that are oriented towards node i, while $\mathcal{E}_i^{d-}$ is the set of directed edges connected to node i that are oriented against node i; for the problem of maximizing load delivered, $z_i^b \in \{0,1\}$ for node i is a parameter indicating if there is a blocker at i, which forces GIC flow into a transformer neutral to 0; for the blocker placement problem, it is the binary placement variable for the blocker; and finally, V_i is the quasi-dc voltage at node i and a_i^s is the resistance to remote earth at i.

Next, the dc Ohm's law constraint for $i \in \mathcal{E}^d$ is defined by,

$$I_i = a_i(V_j - V_k + V_i^{br}) \tag{8}$$

where a_i is the conductance of dc edge i; and V_i^{br} is the voltage induced on dc edge i by the GMD.

C. GIC and AC coupling

The notion of "effective" GIC is to calculate a weighted sum of currents through transformer windings that is proportional to the transformer core flux, and therefore to the GIC-induced reactive power losses of the transformer. This varies for different transformer configurations. These are defined $\forall i \in \mathcal{E}^a$ by,

$$\tilde{I}_i = \begin{cases} I_{iH} & \text{if } i \in \mathcal{E}^\Delta \\ \frac{\alpha_i I_{iH} + I_{iL}}{\alpha_i} & \text{if } i \in \mathcal{E}^y \\ \frac{\alpha_i I_{iS} + I_{iC}}{\alpha_i + 1} & \text{if } i \in \mathcal{E}^\infty \\ I_{iH} + \frac{I_{iL}}{\alpha_i} + \frac{I_{iT}}{\beta_i} & \text{if } i \in \mathcal{E}^3 \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

where $\mathcal{E}^\Delta$ is the set of grounded-wye delta transformers, $\mathcal{E}^y$ is the set of grounded-wye-grounded-wye transformers, $\mathcal{E}^\infty$ is the set of autotransformers, $\mathcal{E}^3$ is the set of three-winding transformers, $\alpha_i = V_{iH}^b/V_{iL}^b$ is the transformer turns ratio for the primary and secondary windings, while $\beta_i = V_{iH}^b/V_{i3}^b$

is the transformer turns ratio for the primary and tertiary windings for the case of three-winding transformers; $\widetilde{I}_i$ is the "effective" GIC current on the dc edge; I_{iH} is the high-side winding current and I_{iL} is the low-side for two- or three-winding transformers; I_{iS} is the series winding current and I_{iC} is the common winding current for auto-transformers; and $i^H = H_i^d$, $i^L = L_i^d$, $i^S = S_i^d$, $i^C = C_i^d$, $i^3 = T_i^d$ are set index variables for transformer terminal nodes in the dc network.

The magnitude of "effective" GIC for $\forall i \in \mathcal{E}^\tau$ is then defined by,

$$\overline{I}_i = \left| \widetilde{I}_i \right| \tag{10}$$

where $\mathcal{E}^\tau = \mathcal{E}^\Delta \cup \mathcal{E}^y \cup \mathcal{E}^\infty \cup \mathcal{E}^3$ is the set of all transformers.

Last, the reactive power loss increase for a transformer resulting from GIC-induced half-cycle saturation $\forall i \in \mathcal{E}^\tau$ is defined by,

$$q_i^{loss+} = \begin{cases} \sqrt{\frac{2}{3}} \cdot \frac{S_i^b}{V_k^b} \cdot |\mathbf{v}_j| \, K_i \overline{I}_i^d & \text{if } k = i^H \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

$$q_i^{loss-} = \begin{cases} \sqrt{\frac{2}{3}} \cdot \frac{S_i^b}{V_j^b} \cdot |\mathbf{v}_j| \, K_i \overline{I}_i^d & \text{if } j = i^L \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $j = H_i$ is the high-side node associated with transformer i; q_j^{loss} is the GIC-induced reactive power loss at the high-side node j; S_i^b is the base power of transformer i; V_j^b is the rated voltage at the high-side node j; and K_i is the GIC reactive power loss of transformer i.

The MLD problem is formulated by minimizing Eq. (1) subject to constraints Eqs. (2) – (12) being satisfied.

D. GIC DC-Current Blocker Placement

The blocker placement problem takes on the same form as the MLD problem, however, the blocker placement parameters z_i^b become binary decision variables and an additional constraint is added to limit the cost of blocker placement to within a budget:

$$\sum_{i \in \mathcal{N}_b^d} \kappa_i^b z_i^b \leq \overline{c}_b \tag{13}$$

where κ_i^b is the cost of placing a GIC dc-current blocker at node (bus) i; and $\overline{c}_b$ is the total budget for blocker placement.

The blocker placement problem is formulated by minimizing Eq. (1) subject to constraints Eqs. (2) – (13) being satisfied.

III. LEARNING MODEL

Given the mathematical formulation of blocker placement, in this section, we present our ML approach using HGNN to solve the GIC dc-current blocker placement problem as a binary classification problem with a physics-informed neural network model.
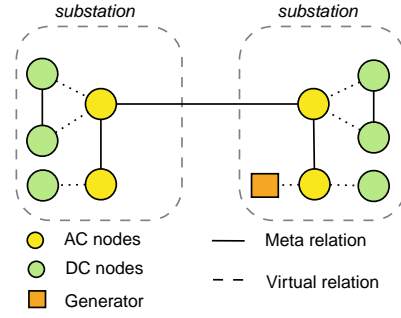

Fig. 1: An abstract view of a heterogeneous graph for power grid (B4GIC).

A. Heterogeneous Graph

The power grid can be modeled as a heterogeneous graph, where different types of nodes represent different components of the grid (such as ac bus and generator) and the edges between nodes represent the connections between these components (such as transmission lines and transformers). In addition to building a HGNN to predict blocker locations, we first consider the effects of GMD as quasi-dc current injection, and then we can use the ac-dc power flow model to solve the power flow problem. Therefore, we also have a set of nodes and edges from the dc network, and a set of new edges between ac and dc networks. We can use a combination of node and edge embeddings to represent the graph structure and node attributes (such as the type of power plant or the voltage level of a transmission line). Figure 1 provides an abstract view of a heterogeneous graph for power grid B4GIC[1], a typical small network studied for GIC problems.

Definition 1 (Heterogeneous Graph). *A heterogeneous graph is defined as an undirected graph $\mathbb{G} = (\mathcal{N}, \mathcal{E}, \mathcal{A}, \mathcal{R})$ where each node $v \in \mathcal{N}$ and each edge $e \in \mathcal{E}$ are associated with their type mapping function $\tau(v) : \mathcal{V} \to \mathcal{A}$ and $\phi(e) : \mathcal{E} \to \mathcal{R}$, respectively.*

Meta Relation: From an abstract view, a meta relation can be thought of as an edge $e = (s, t)$ that connects a source node s to a target node t. The tuple (s, e, t) represents the meta relation, where e is the edge between the nodes, and s and t are the nodes themselves. The features associated with the meta relation are the attributes of the edge and the nodes it connects. For example, the branch is an edge connecting two buses, denoted as (bus s, branch e, bus t), and has the attributes of the branch, the source bus, and the target bus. By leveraging the features associated with meta relations, we can build more accurate and effective models that can be used to make predictions and decisions about the power grid.

Virtual Relation: The power grid is hierarchical, and the meta relations alone are not sufficient to capture its hierarchical structure. For instance, generators are a subset of buses, and they have attributes that are not shared with buses. To address this limitation, we introduce virtual relations,

---

[1]PowerWorld simulator (B4GIC): https://www.powerworld.com/

a special type of meta relation that connects nodes without having edge features. The virtual relation (bus s, conn e, gen t) represents a bus that is also a generator, i.e., bus s and gen t are the same physical entity, and the virtual edge conn e is used to establish a connection between them. Thus, this relation does not have any edge features but can still capture information from node to node.

B. Heterogeneous Graph Neural Network

Our approach uses HGNNs to learn the node and edge embeddings of a power grid. Once the embeddings are learned, we use them to predict the likelihood of a blocker occurring in the dc buses, and identify the most effective locations to place blockers to mitigate the impact of GMDs. Thus, we can use a HGNN to predict GIC dc-current blockers as a binary classification problem on dc buses.

To build a HGNN to predict blocker locations, we use a combination of node and edge embeddings to represent the graph structure and node attributes, such as a set of features of nodes (e.g., type, real demand, reactive demand, etc.) and a set of features of transmission lines (e.g., length, capacity, voltage level, etc.).

In general, the GNNs learn the embedding of each node by aggregating the information from its neighbors as follows,

$$\boldsymbol{H}_v^t \leftarrow \mathbf{Aggr}_{\forall u \in N(v)} \left(f\left(\boldsymbol{H}_u^{t-1}; \boldsymbol{H}_v^{t-1} \right) \right) \quad (14)$$

where $N(v)$ denotes the neighbors of v; $\boldsymbol{H}_v^t$ is the embedding of node v at the t-th layer; and f is a learnable function given the node embeddings. HGNNs can learn the node and edge embeddings simultaneously.

HGNN is a type of neural network designed to work with heterogeneous graph-structured data, and it has been shown to be effective in a variety of tasks, such as node classification, graph classification, and link prediction. The key idea of HGNN is to iteratively update the node embeddings by aggregating information from neighboring nodes and learning a representation of the graph structure. The update rule for the node embeddings can be written as:

$$\boldsymbol{H}_i^{t+1} = \sigma \left(\sum_{j \in N(i)} \boldsymbol{H}_j^t \boldsymbol{W}_{ij}^{(\alpha)} + \sum_{j \in \mathcal{N}(i)} \boldsymbol{E}_{ij}^t \boldsymbol{W}_{ij}^{(\beta)} \right) \quad (15)$$

where $\boldsymbol{H}_i^t$ is the embedding of node i at iteration t; σ is an activation function; $N(i)$ is the set of neighboring nodes of i; $\boldsymbol{W}_{ij}^{(\alpha)}$ and $\boldsymbol{W}_{ij}^{(\beta)}$ are weight matrices that capture the importance of the edge between nodes i and j for the node and edge embeddings, respectively; and $\sum_{j \in N(i)} \boldsymbol{H}_j^t \boldsymbol{W}_{ij}^{(\alpha)} + \sum_{j \in N(i)} \boldsymbol{E}_{ij}^t \boldsymbol{W}_{ij}^{(\beta)}$ is the aggregated feature representation of the neighbors of i.

In addition to the node embeddings, the model also learns edge embeddings to represent the transmission lines and transformers between nodes. The edge embeddings are learned using a similar update rule:

$$\boldsymbol{E}_{ij}^{t+1} = \sigma \left(\sum_{k \in \mathcal{N}(j)} \boldsymbol{E}_{kj}^t \boldsymbol{W}_{ik}^{(\alpha)} + \sum_{k \in \mathcal{N}(i)} \boldsymbol{H}_k^t \boldsymbol{W}_{ik}^{(\beta)} \right) \quad (16)$$
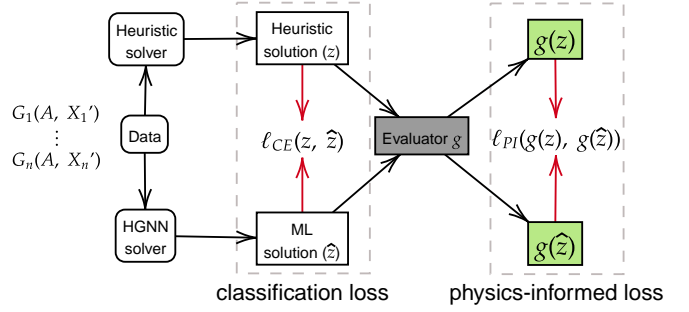

Fig. 2: PIHGNN Framework

where $\boldsymbol{E}_{ij}^t$ is the embedding of the edge between nodes i and j at iteration t. For $t = 0$, we take $\boldsymbol{H}^0$ and $\boldsymbol{E}^0$ as the input node and edge features, respectively.

Once we have learned the node and edge embeddings, we use them to predict the likelihood of a blocker occurring in the power grid. Specifically, we place blockers on the dc buses, where we define a probability distribution over the possible GIC blocker locations using a fully connected multi-layer perceptron (MLP) that takes the node and edge embeddings as input:

$$\mathrm{P}(\boldsymbol{z}|\boldsymbol{H},\boldsymbol{E}) = \mathrm{MLP}(\boldsymbol{H},\boldsymbol{E}) \quad (17)$$

where $\boldsymbol{H}$ is the set of node embeddings of dc nodes; $\boldsymbol{E}$ is the set of edge embeddings; and $\boldsymbol{z}$ is the set of possible binary GIC blocker locations the model tries to predict.

The HGNN is trained using a variety of loss functions, such as cross-entropy loss between the predicted probability and the true labels, to optimize the prediction performance. The learnable parameters of the HGNN include the weight matrices $\boldsymbol{W}_{ij}^{(\alpha)}$ and $\boldsymbol{W}_{ij}^{(\beta)}$ for the node and edge embeddings, and the parameters of the MLP for predicting the blocker locations, denoted as $\boldsymbol{\theta}$ in general.

C. Physics-Informed Heterogeneous GNN

Having location predictions simply provided by the HGNN is insufficient to solve the problem. Since the core blocker placement problem is MINLP, and a heuristic solver (ac-rect) [26] can only provide locally optimal solutions and not necessarily the global optimum, in this section, we introduce a novel approach that combines the strengths of HGNNs with the physics-based modeling of power grids.

The developed method, called physics-informed heterogeneous graph neural network (PIHGNN), leverages the predictive power of HGNNs with the physical laws governing power systems. This allows for more accurate and reliable predictions of GIC dc-current blocker locations, which can help prevent equipment damage and power outages caused by GMDs.

PIHGNN is based on a re-evaluation step after reading out the blocker location predictions from the HGNN. The predicted locations are sent into a surrogate problem, which measures the feasibility and optimality of the predictions in a real electric grid. We denote the surrogate problem as function
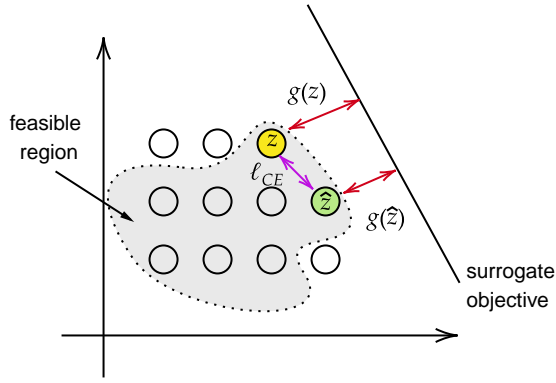
Fig. 3: Evaluation of predictions. The heuristic solver provides the label z (highlighted in yellow), and the GNN-based prediction $\hat{z}$ (highlighted in green) is evaluated by the evaluator g. The evaluator g is a surrogate MLD model that takes the predictions from the heuristic solver and the GNN model as inputs.

$g(\cdot)$, which is a physics-based model that solves the MLD problem of a power grid given both ac and dc formulations; it takes the predicted blocker locations as input, and outputs a set of feasible and optimal values of the surrogate problem.

The surrogate model, $g(\cdot)$, along with the HGNN, is trained using the backpropagation method. The training data are the same as the HGNN problem with the additional output from the heuristic solver into the surrogate problem. The total loss function was built not only from the HGNN problem but also from the surrogate problem as well, presented in Figure 2, and is given

$$\ell(\boldsymbol{A}, \boldsymbol{X}, \boldsymbol{z}) = \ell_{\mathrm{CE}}(\boldsymbol{z}, \hat{\boldsymbol{z}}) + \eta\ell_{\mathrm{PI}}\left(\mathrm{g}(\boldsymbol{z}), \mathrm{g}(\hat{\boldsymbol{z}})\right) \qquad (18)$$

where $\boldsymbol{A}$ are the heterogeneous structural adjacencies, $\boldsymbol{X}$ are the node and edge feature matrices, and η is the regularization parameter; $\ell_{\mathrm{CE}}(\cdot, \cdot)$ is the classification loss based on cross-entropy function from the HGNN problem; and $\ell_{\mathrm{PI}}(\cdot, \cdot)$ is the physics-informed loss from the surrogate formulation.

Figure 3 illustrates the rationale behind the loss function in PIHGNN. A classification loss, typically based on a cross-entropy function, only ensures that the solution obtained from the HGNN ($\hat{z}$) is close to the observed data (z). Due to the non-convex and non-smooth optimization of the blocker placement problem, the observed data is not the only solution to the original problem. Moreover, the prediction from the HGNN model is not guaranteed to be a feasible solution to the original blocker placement problem. To capture the underlying physical laws governing the behavior of power systems, we propose to use a surrogate model $g(\cdot)$ to evaluate the quality of predictions, both in feasibility and optimality. A physics-informed loss $\ell_{\mathrm{PI}}(\cdot, \cdot)$ offers a promising approach to simplify the complexity of the underlying physical laws but still serves as a surrogate measurement to the quality of solutions.

The advantages and contributions of PIHGNN are:

- **Improved accuracy**: PIHGNN combines the predictive power of HGNNs with the physical laws governing power

systems, leading to more accurate predictions of blocker locations.

- **Increased reliability**: The re-evaluation step using the surrogate model ensures that the predicted GIC dc-current blocker locations are feasible and optimal, reducing the risk of equipment damage and power outages caused by GMDs.
- **Flexibility**: PIHGNN can be adapted to different scenarios and can handle various types of uncertainties, such as uncertainty in the blocker locations, uncertainty in the grid topology, and uncertainty in the grid parameters.

Algorithm 1 Adaptive training of PIHGNN

1: **Data:** $\mathcal{G}_1(\boldsymbol{A}, \boldsymbol{X}_1, \boldsymbol{z}_1), \cdots, \mathcal{G}_n(\boldsymbol{A}, \boldsymbol{X}_n, \boldsymbol{z}_n)$
2: **HP:** η, p, lr, epoch, etc.
3: **procedure** TRAINING($\mathcal{G}_1, \cdots, \mathcal{G}_n$)
4:     **for** epoch **do**
5:         **for** sample $\mathcal{G}_i$ in batch **do**
6:             $\hat{\boldsymbol{z}}_i \leftarrow \mathrm{HGNN}(\mathcal{G}_i)$ ▷ feedforward
7:             $\ell_{\mathrm{CE},i} \leftarrow CE(\boldsymbol{z}_i, \hat{\boldsymbol{z}}_i)$ ▷ classification loss
8:         **end for**
9:     **if** epoch % p = 0 **then**
10:         **for** sample $\mathcal{G}_i$ in batch **do**
11:             $\ell_{\mathrm{PI},i} \leftarrow d(g(\hat{\boldsymbol{z}}_i), g(\boldsymbol{z}_i)))$ ▷ PI loss
12:         **end for**
13:     **end if**
14:     $\ell = \frac{1}{b}\sum_i(\ell_{CE,i} + \frac{\eta}{\sqrt{epoch}}\ell_{PI,i})$
15:     Update $\boldsymbol{\theta}$ ▷ backpropagation
16:     **end for**
17:     Return HGNN with parameters $\boldsymbol{\theta}$
18: **end procedure**

IV. CASE STUDIES

A. Dataset

Extensive experiments were conducted on two widely used synthetic test systems, EPRI21 and UIUC150, which are publicly available with PowerModelsGMD[2] (PMsGMD) [6]. The EPRI21 dataset is a 20-bus network with 7 generators and 31 branches, and the UIUC150 dataset is a 150-bus network with 27 generators and 218 branches. Details of the PMsGMD data format are summarized in Table I: tables "bus", "gen", and "gmd_bus" contain the nodes in the heterogeneous graph; tables "branch", "branch_gmd", and "gmd_branch" contain the meta relations with a set of specific features; and tables "bus_conn_gen" and "gmd_bus_attach_bus" contain the virtual relations without edge features.

Data preprocessing: To demonstrate the effectiveness of our proposed models, we first generate a set of synthetic data representing the GMD scenarios: 50 samples for each combination of E-field direction in $(45, 135)$ degree and magnitude $(5, 10, 15, 20)$, along with a uniformly random scalar perturbation in $[0.8, 1.2]$ for active and reactive demands on load buses. With a total 400 samples for each dataset, we split

---

[2]https://github.com/lanl-ansi/PowerModelsGMD.jl

the dataset into 80%, 10%, and 10% for training, validation, and testing, respectively. We also normalize the node and edge features to $[0, 1]$ in order to stabilize the training process[3].

B. Performance

Formulated as a binary classification problem, we evaluated the performance of the proposed models on the test systems using the accuracy and ROC-AUC metrics with 10-fold cross validation, where the accuracy is the ratio of the number of correct predictions to the total number of predictions, and the ROC-AUC is the area under the receiver operating characteristic curve, which is typically used to evaluate the performance of binary classification models. We use 4 layers of heterogeneous transformer layer with 4 heads, and 4 layers of MLP with 128 hidden dimensions. We apply the Adam optimizer [27] with a learning rate of 0.001 and weight decay of 0.0001 for the backpropagation with 200 epochs. Table II summarizes the prediction performance of the two models, HGNN and PIHGNN, on the two datasets. Results indicate that PIHGNN outperforms HGNN on both datasets, with average accuracy improvements of 0.052 and 0.013, and average ROC-AUC improvements of 0.046 and 0.007, respectively. Regarding the performance variations of the datasets, EPRI21 has generally higher accuracy and ROC-AUC scores than UIUC150, which is primarily due to the complexity of UIUI150 being a larger system. Results also suggest that PIHGNN is a more reliable and robust model than HGNN, especially when applied to a diverse and complex power system dataset.

One advantage of the proposed HGNN/PIHGNN model is that it can be trained efficiently in a supervised manner. To demonstrate this, Table III compares the efficiency of three methods – HGNN, PIHGNN, and a heuristic solver [26] – for solving the blocker placement problem. Training and testing times for each method, expressed in seconds per data sample, is displayed; training time is the time required to train the model on the training set, while testing time is the time required to evaluate the model with one sample on the test set. The heuristic solver [26], is a baseline method that uses a stochastic learning optimizer which is implemented in Julia, and its time is also shown per data sample. Table III shows that HGNN and PIHGNN have significantly lower training and testing times than the heuristic solver, with HGNN being the fastest method overall. In particular, HGNN has training of 4.75 and 8.12 seconds per data sample on average from EPRI21 and UIUC150, respectively. While on the testing set, HGNN has a negligible time of less than 1 second per data sample. In contrast, the heuristic solver has much longer times, ranging from 1 minute to 10 minutes per data sample in EPRI21 and UIUC150 respectively. These results indicate that both HGNN and PIHGNN are significantly more efficient than the heuristic solver, and are therefore more suitable for solving the blocker placement problem in large networks where the heuristic solver is time-consuming.

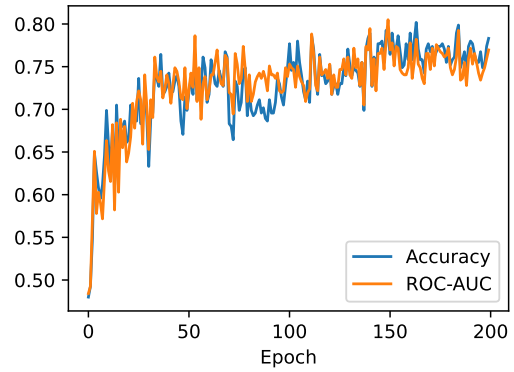[3]The source code is accessible: https://github.com/cshjin/swmp_ml/

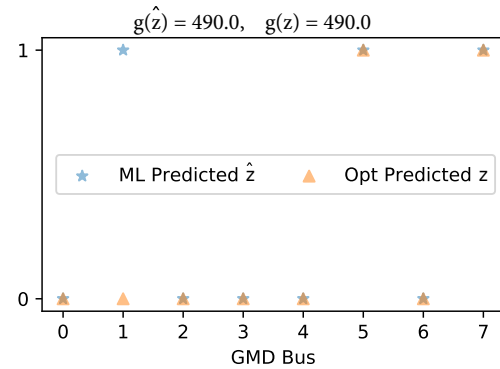Fig. 4: Training PIHGNN model on EPRI21

Fig. 5: A surrogate evaluator for EPRI21

Figure 4 shows the training process of the PIHGNN model on the EPRI21 dataset. Both accuracy and ROC-AUC scores are shown as a function of the number of training epochs. The figure shows that the model achieves a high accuracy and ROC-AUC score over 0.75, respectively, after 150 epochs. These results indicate that the PIHGNN model can be trained efficiently on the EPRI21 dataset, and can achieve a high prediction performance after a small number of training epochs. More interestingly, Figure 5 shows the surrogate evaluator on a single case in EPRI21. As shown, the PIHGNN prediction is slightly different from the heuristic solver, but the output from both solutions is exactly the same in the surrogate evaluator. This demonstrates the need for the surrogate evaluator to evaluate the PIHGNN prediction despite the binary labels from the HGNN model when compared with the MIP solver.

We further investigated the impact of the electric field (E-field) on the blocker placement problem by evaluating the PIHGNN model's predictions under different E-field magnitudes. Figure 6 depicts the objective function values for the PIHGNN model, applied to the EPRI21 dataset, as a function of increasing E-field magnitude (while fixing the direction at 45°). Notably, both the heuristic approach and the ML approach exhibit an increasing trend in objective function value with rising E-field magnitude. However, the ML approach achieves comparable objective function values to the heuristic approach,

TABLE I: Statistics of networks

(# entries, # features)	blocker locations	bus	gen	gmd_bus	branch	branch_gmd	gmd_branch	bus_conn_gen	gmd_bus_attach_bus
EPRI21	(8,)	(19, 15)	(7, 20)	(27, 3)	(31, 11)	(31, 18)	(37, 4)	(7, 0)	(27, 0)
UIUC150	(98,)	(150, 15)	(27, 20)	(248, 3)	(218, 11)	(218, 18)	(250, 4)	(27, 0)	(248, 0)

TABLE II: Prediction performance

Dataset	Model	Accuracy	ROC-AUC
EPRI21	HGNN	0.732 ± .08	0.755 ± .11
	PIHGNN	0.784 ± .12	0.801 ± .10
UIUC150	HGNN	0.702 ± .07	0.588 ± .07
	PIHGNN	0.715 ± .10	0.594 ± .12

TABLE III: Efficiency comparison. The training time is the time per data sample to train the model, and the testing time is the time per data sample to evaluate the model on the test set. The heuristic solver is the time per data sample to solve the problem in Julia.

Dataset	Model	Training (sec.)	Testing (sec.)
EPRI21	HGNN	4.75	$\ll 1$
	PIHGNN	7.50	2.31
	Heuristic Solver	54.056	54.12
UIUC150	HGNN	8.12	$\ll 1$
	PIHGNN	10.30	3.55
	Heuristic Solver	384.52	390.41

Fig. 6: Objective function values w.r.t. E-field magnitude on EPRI21

TABLE IV: Generalization of learned HGNN (accuracy scores reported)

Train \ Test	EPRI21	UIUC150
EPRI21	0.741	0.672
UIUC150	0.661	0.718

demonstrating similar average performance.

It is important to acknowledge that the non-convex and non-smooth nature of the optimization problem precludes exact equivalence between the objective function values of the PIHGNN model and the heuristic solver. Nevertheless, the observed closeness in their average values across various E-field magnitudes suggests the PIHGNN model's efficacy in predicting the objective function value of the blocker placement problem, which consequently allows for assessment of prediction quality. This aligns with the findings presented in Table III, emphasizing the suitability of the PIHGNN model for large-scale power systems, where the computational efficiency of training the model outweighs the substantial time required for the heuristic solver.

Last, we examined the generalization capability of our machine learning model by training it on one network and predicting on another. This approach leverages the knowledge acquired during training on known datasets to make predictions on similar datasets, even with different network sizes, thanks to the graph neural networks (GNNs). Such generalization ability is particularly useful in power systems where a model can learn patterns from one network and apply them to another without the need for explicit heuristics or simulated data.

Table IV shows the empirical results of training HGNN model (without true labels to be evaluated in the surrogate) on one network and testing it on another. The model trained on EPRI21 data and tested on UIUC150 achieved an accurac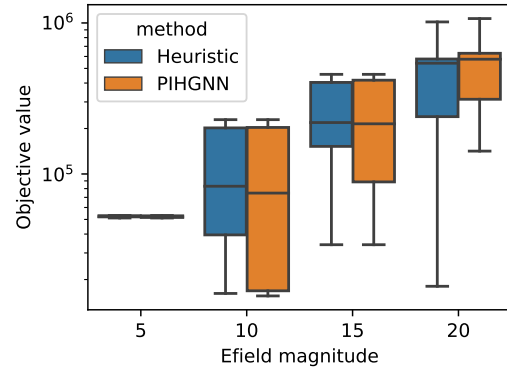y score of 0.672, which is lower than the model trained and tested on its own (0.718). Similarly, the model trained on UIUC150 and tested on EPRI21 had an accuracy score of 0.661, which is lower than the model trained from its own data. These results provide empirical evidence that a model can generalize to new data without requiring explicit labels or training data from the new network.

V. CONCLUSION

We introduced a physics-informed heterogeneous graph neural network (PIHGNN) for solving the graphically constrained blocker placement problem in power systems. The proposed method combines the strengths of graph neural networks (GNNs) and physics-informed neural networks (PINNs) to efficiently and accurately optimize the placement of geomagnetically induced currents (GICs) dc-current blockers. The GNN captures the complex relationships and dependencies between different types of nodes and edges in ac/dc networks, while the PINN ensures that the solution satisfies the physical laws of the power grid. Further work will extend the proposed method to solve other optimization problems on power systems, such as voltage stability in presence of GICs, investigate other machine learning techniques, such as reinforcement learning, and incorporate spatiotemporal data for explicit physics-informed neural networks in the future.

REFERENCES

[1] NERC and U.S. Department of Energy. High-Impact, Low-Frequency Event Risk to the North American Bulk Power System. Technical report, North American Electric Reliability Corporation, Jun. 2010.

[2] R. Pirjola. Geomagnetically Induced Currents During Magnetic Storms. *IEEE Plasma Sci.*, 28(6):1867–1873, Dec. 2000.

[3] NERC. Effects of Geomagnetic Disturbances on the Bulk Power System System. Technical report, North American Electric Reliability Corporation, 2012.

[4] A. K. Barnes et al. The Risk of Hidden Failures to the United States Electrical Grid and Potential for Mitigation. In *Proc. of the 2021 IEEE 53rd North American Power Symposium*, Nov. 2021. https://arxiv.org/abs/2107.03439.

[5] A. Mate et al. Relaxation Based Modeling of GMD Induced Cascading Failures in PowerModelsGMD.jl. In *Proceedings of the 2021 IEEE/PES 53rd North American Power Symposium*, 2021.

[6] A. Mate et al. Analyzing and Mitigating the Impacts of GMD and EMP Events on the Electrical Grid with PowerModelsGMD.jl, Jan. 2021. https://arxiv.org/abs/2101.05042.

[7] P.-H. Yang et al. Optimal Placement of Grounding Small Resistance in Neutral Point for Restraining Voltage Fluctuation in Power Grid Caused by Geomagnetic Storm. *IET Generation, Transmission & Distribution*, 13(8):1456–1465, Apr. 2019.

[8] R. Horton. Magnetohydrodynamic Electromagnetic Pulse Assessment of the Continental U.S. Electric Grid: Geomagnetically Induced Current and Transformer Thermal Analysis. Technical Report 3002009001, Electric Power Research Institute, Feb. 2017.

[9] D. K. Molzahn et al. A Survey of Distributed Optimization and Control Algorithms for Electric Power Systems. *IEEE Transactions on Smart Grid*, 8(6):2941–2962, 2017.

[10] L. A. Roald et al. Power Systems Optimization Under Uncertainty: A Review of Methods and Applications. *Electric Power Systems Research*, 214:108725, 2023.

[11] P. Panciatici et al. Advanced Optimization Methods for Power Systems. In *2014 Power Systems Computation Conference*, pages 1–18. IEEE, 2014.

[12] A. Kody et al. Modeling the AC Power Flow Equations with Optimally Compact Neural Networks: Application to Unit Commitment. *Electric Power Systems Research*, 213:108282, 2022.

[13] F. Fioretto et al. Predicting AC Optimal Power Flows: Combining Deep Learning and Lagrangian Dual Methods. In *Proceedings of the AAAI conference on artificial intelligence*, pages 630–637, 2020.

[14] W. Chen et al. Learning Optimization Proxies for Large-Scale Security-Constrained Economic Dispatch. *Electric Power Systems Research*, 213:108566, 2022.

[15] D. Owerko et al. Optimal Power Flow Using Graph Neural Networks. In *IEEE 2020 International Conference on Acoustics, Speech and Signal Processing*, pages 5930–5934. IEEE, 2020.

[16] B. Donon et al. Neural Networks for Power Flow: Graph Neural Solver. *Electric Power Systems Research*, 189:106547, 2020.

[17] W. Liao et al. Short-Term Power Prediction for Renewable Energy Using Hybrid Graph Convolutional Cetwork and Long Short-Term Memory Approach. *Electric Power Systems Research*, 211:108614, 2022.

[18] S. Park et al. Confidence-aware Graph Neural Networks for Learning Reliability Assessment Commitments. *IEEE Transactions on Power Systems*, 2023.

[19] Thomas J. Overbye, Trevor R. Hutchins, Komal Shetye, Jamie Weber, and Scott Dahman. Integration of geomagnetic disturbance modeling into the power flow: A methodology for large-scale system studies. In *North American Power Symposium (NAPS), 2012*, pages 1–7. IEEE, September 2012.

[20] M. Raissi et al. Physics-Informed Neural Networks: A Deep Learning Framework for Solving Forward and Inverse Problems Involving Nonlinear Partial Differential Equations. *Journal of Computational physics*, 378:686–707, 2019.

[21] Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

[22] Yiheng Du, Nithin Chalapathi, and Aditi Krishnapriyan. Neural spectral methods: Self-supervised learning in the spectral domain. *arXiv preprint arXiv:2312.05225*, 2023.

[23] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *International Conference on Machine Learning*, pages 3276–3285. PMLR, 2018.

[24] Nithin Chalapathi, Yiheng Du, and Aditi Krishnapriyan. Scaling physics-informed hard constraints with mixture-of-experts. *arXiv preprint arXiv:2402.13412*, 2024.

[25] C. Coffrin et al. Relaxations of AC Maximal Load Delivery for Severe Contingency Analysis. *IEEE Transactions on Power Systems*, 34(2):1450–1458, 2018.

[26] Minseok Ryu, Ahmed Attia, Arthur Barnes, Russell Bent, Sven Leyffer, and Adam Mate. Heuristic algorithms for placing geomagnetically induced current blocking devices. *arXiv preprint arXiv:2310.09409*, 2023.

[27] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.