



Delft University of Technology

Document Version

Final published version

Licence

CC BY

Citation (APA)

Rajaei, A., Palensky, P., & Cremer, J. L. (2027). Transferable graph learning for transmission congestion management via busbar splitting. *Electric Power Systems Research*, 263, Article 113760. <https://doi.org/10.1016/j.epsr.2026.113760>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

In case the licence states "Dutch Copyright Act (Article 25fa)", this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

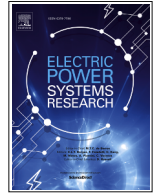
Sharing and reuse

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.



Transferable graph learning for transmission congestion management via busbar splitting

Ali Rajaei ^{*}, Peter Palensky , Jochen L. Cremer

Department of Electrical Sustainable Energy, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

Keywords:

Network topology optimization
Congestion management
Graph neural networks
Transferability

ABSTRACT

Network topology optimization (NTO) via busbar splitting can mitigate transmission grid congestion and reduce redispatch costs. However, solving this mixed-integer nonlinear problem for large-scale systems in near-real-time is currently intractable with existing solvers. Machine learning (ML) approaches have emerged as a promising alternative, but they have limited generalization to unseen topologies, varying operating conditions, and different systems, which limits their practical applicability. This paper formulates NTO for congestion management considering linearized AC power flow, and proposes a graph neural network (GNN)-accelerated approach. We develop a heterogeneous edge-aware message passing GNN to predict effective nodes for busbar splitting actions as candidate NTO solutions. The proposed GNN captures local flow patterns, improves generalization to unseen topology changes, and enhances transferability across systems. Case studies show up to $10^4\times$ speed-up, delivering AC-feasible solutions within one minute and a 2.3% optimality gap on the GOC 2000-bus system. These results demonstrate a significant step toward near-real-time NTO for large-scale systems with topology and cross-system generalization.

1. Introduction

Congestion management (CM) in transmission networks is becoming increasingly challenging due to the growing integration of distributed energy resources and electrification across sectors. NTO, particularly substation reconfiguration via busbar splitting, is a promising approach to relieve congestion and reduce re-dispatch costs. However, solving this mixed-integer non-linear optimization problem for large-scale systems in near-real-time remains computationally challenging. As a result, operators often rely on expert knowledge or predefined manuals, which can lead to sub-optimal system performance [1]. Recent approaches have explored ML to identify effective topological actions in near-real-time, but most models lack scalability, remain system-specific and fail to generalize to new topologies, operating conditions, or unseen systems. This lack of generalization and transferability limits their practical use due to: 1) the need for retraining and costly data collection for each new system, 2) difficulty for operators to detect when models become outdated, and 3) performance drops under rapidly changing grid conditions. In this context, this paper proposes a GNN-based approach that exploits the locality of CM to scale to large-scale systems, generalize to topology changes, and transfer to unseen grids [2].

Optimal substation reconfiguration (OSR) extends the optimal power flow (OPF) problem by introducing binary variables for breaker statuses,

allowing changes to the network topology. Several works [3–6] apply sensitivity analysis as a heuristic to identify candidate busbar splitting actions. These heuristic approaches assume a fixed substation configuration and focus solely on selecting splitting actions, but they do not provide an optimal substation configuration. Heidarifar and Ghasemi [7], Goldis et al. [8] formulate DC-OSR as a MILP using node-breaker modeling and shift factors, respectively. Morsy et al. [9] develops a heuristic approach to solve DC-OSR by focusing on local congested regions. Building on [7,10] propose a sparse tableau formulation of AC PF, while [11] develops a MISOCP formulation. However, these approaches either rely on heuristics, are limited to DC power flow, or become computationally intractable for large-scale systems.

Recent research has investigated supervised and reinforcement learning (RL) approaches for NTO [12]. For instance, Chauhan et al. [13] develops an RL agent based on proximal policy optimization, while [14,15] propose curriculum-based RL to improve learning performance. Other works such as [16,17] mitigate unforeseen action consequences by simulating future outcomes, but at the cost of a computationally intensive Monte-Carlo tree search. Hassouna et al. [18] proposes a GNN-based imitation learning approach to predict the probability of each action's effectiveness (as a soft label) in reducing maximum overloading, and [19] trains a heterogeneous GNN to imitate a rule-based expert agent for day-ahead operation and improve generalization to N-1

^{*} Corresponding author.

E-mail addresses: a.rajaei@tudelft.nl (A. Rajaei), p.palensky@tudelft.nl (P. Palensky), j.l.cremer@tudelft.nl (J.L. Cremer).

<https://doi.org/10.1016/j.epsr.2026.113760>

Received 15 October 2025; Received in revised form 13 March 2026; Accepted 3 July 2026

Available online 11 July 2026

0378-7796/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Nomenclature

Indices and Sets

$i \in \mathcal{V}$	Index for substations (nodes).
$b \in \mathcal{B}$	Index for busbars $\mathcal{B} = \{b1, b2\}$.
$l_{ij} \in \mathcal{L}$	Index for transmission lines (from i to j).
$g \in \mathcal{G}$	Index for generators.
$d \in \mathcal{D}$	Index for load demands.
$\mathcal{L}_i, \mathcal{G}_i, \mathcal{D}_i$	Elements $e \in E_i$ connected to substation i .
$\mathcal{L}^{cong}$	Set of congested transmission lines.
$\mathcal{V}^p$	Set of substations selected by the proximity filter.
$\mathcal{V}^{cnd}$	Set of candidate substations selected by the GNN for potential busbar splitting.

Parameters

$\hat{P}_d, \hat{Q}_d$	Active and reactive power demand.
$\hat{P}_g, \hat{Q}_g$	Active and reactive power generation.
$\underline{V}_i, \bar{V}_i$	Minimum and maximum squared voltage magnitude.
$\bar{Z}^s$	Maximum number of busbar splitting.
b_l, g_l	Susceptance and conductance of line l .
S_l	Maximum apparent power flow of line l .
ϕ_d	Tangent of the power factor angle of loads.
γ	Congestion cost parameter.
s^{th}	Line loading penalization threshold.
M^θ, M^v	Big-M parameters.

Variables

$P_{g,b}, Q_{g,b}$	Auxiliary variables for power generation.
$P_{d,b}, Q_{d,b}$	Auxiliary variables for load demand.
$P_{l,i}, Q_{l,i}$	Active and reactive power flow of line l from i side.
$P_{l,i,b}, Q_{l,i,b}$	Active and reactive power flow of line l through busbar b at substation i .
$P_{l,i}^L, Q_{l,i}^L$	Active and reactive power loss of line l .
$\theta_{l,b}, V_{l,b}$	Voltage angle and squared voltage magnitude of substation i busbar b .
$\theta_{l,i}, V_{l,i}$	Voltage angle and squared voltage magnitude at the end of line l on the side of i
z_i	Connection status of coupler in substation i (0: open, 1: close).
z_g, z_d	Connection status of generator and demand to busbars (0 : b1, 1 : b2).
$z_{l,i}$	Connection status of line l to substation i busbars (0 : b1, 1 : b2).
$\mathcal{O}^{CM}$	Congestion management objective.

conditions. Meng et al. [20] investigates a different application of NTO, aiming to maximize cross-zonal capability rather than to relieve congestion. In [20], two separate GNNs predict breaker statuses and DC flows for a given topology, followed by a repair optimization to enforce DC PF equations. Nevertheless, these approaches rely on reduced action spaces, cannot guarantee operational feasibility, and struggle with generalization and transferability. Here, we refer to **generalization** as the ability to perform well on unseen topologies or operating conditions within the same grid, and **transferability** as the ability of a trained model to be efficiently applied or adapted to different grids.

This paper formulates NTO for CM with linearized AC PF constraints (AC-NTO-CM) as a MIP. We propose an edge-aware GNN that quickly predicts effective splitting actions, which are used as candidate solutions for AC-NTO-CM and also prioritized as branching variables in the branch-and-bound MIP solver. The proposed GNN is fully localized to learn local PF patterns, while a proximity filter improves learning performance by focusing on the most relevant regions. Thus, the proposed approach addresses fundamental challenges in ML-based NTO: scalability, generalization to unseen topologies, and transferability to different systems.

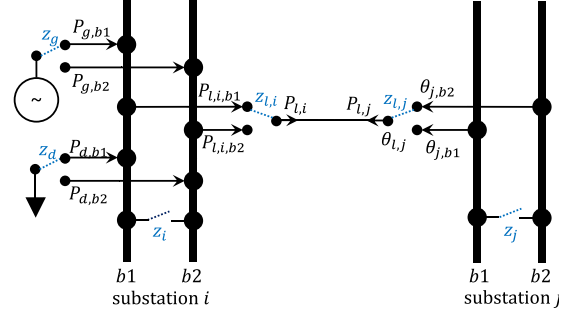


Fig. 1. Schematic of the substation model for AC power flow.

The main contributions are:

- The MIP formulation of NTO for CM considering linearized AC equations.
- The proposed edge-aware GNN to predict busbar splitting candidates. The GNN's expressiveness, the learning task design and the proximity filter, enable generalization to unseen N-k topologies and transferability across different power systems.
- Significant computational speed-ups while maintaining constraint feasibility and minor optimality gaps.
- A novel regression index that estimates the effectiveness of splitting actions, improving interpretability.

The paper is organized as follows. Section 2 introduces the proposed NTO MILP formulation. Section 3 presents the GNN-based solution approach. Section 4 presents the case studies, and Section 5 concludes the paper.

2. Congestion management via busbar splitting

Substation reconfiguration via busbar splitting can effectively reduce transmission congestion by reconfiguring the grid to reroute flows, providing economic advantages over re-dispatch or load shedding measures. This paper focuses on two commonly used substation arrangements: the double-bus double-breaker and the breaker-and-a-half arrangements [7,11,21]. We extend the model in [7] to consider linear AC PF constraints. Fig. 1 shows a substation i connected to substation j through a transmission line l_{ij} . Each connected element $e \in E_i$ (lines, demands and generators) can be switched to either of the busbars $b1$ or $b2$ with the binary variable z_e , while z_i denotes the busbar coupler status.

In this work, we consider the generation dispatch and load demand as input parameters. The constraints are:

$$P_{g,b1} = (1 - z_g)\hat{P}_g, \quad P_{g,b2} = z_g\hat{P}_g \quad g \in \mathcal{G} \quad (1a)$$

$$Q_{g,b1} = (1 - z_g)\hat{Q}_g, \quad Q_{g,b2} = z_g\hat{Q}_g \quad g \in \mathcal{G} \quad (1b)$$

$$P_{d,b1} = (1 - z_d)\hat{P}_d, \quad P_{d,b2} = z_d\hat{P}_d \quad d \in \mathcal{D} \quad (1c)$$

$$Q_{d,b} = \phi_d P_{d,b} \quad d \in \mathcal{D} \quad (1d)$$

Each transmission line $l_{ij} \in \mathcal{L}_i$ can be connected to either busbars of substation i through the binary variable $z_{l,i}$ as:

$$-z_{l,i}M^\theta \leq \theta_{l,i} - \theta_{i,b1} \leq z_{l,i}M^\theta \quad (2a)$$

$$-z_{l,i}M^v \leq V_{l,i} - V_{i,b1} \leq z_{l,i}M^v \quad (2b)$$

$$-(1 - z_{l,i})M^\theta \leq \theta_{l,i} - \theta_{i,b2} \leq (1 - z_{l,i})M^\theta \quad (2c)$$

$$-(1 - z_{l,i})M^v \leq V_{l,i} - V_{i,b2} \leq (1 - z_{l,i})M^v \quad (2d)$$

$$-(1 - z_{l,i})\bar{S}_l \leq P_{l,i,b1} \leq (1 - z_{l,i})\bar{S}_l \quad (2e)$$

$$-(1 - z_{l,i})\bar{S}_l \leq Q_{l,i,b1} \leq (1 - z_{l,i})\bar{S}_l \quad (2f)$$

$$-z_{l,i}\bar{S}_l \leq P_{l,i,b2} \leq z_{l,i}\bar{S}_l \quad (2g)$$

$$-z_{l,i}\bar{S}_l \leq Q_{l,i,b2} \leq z_{l,i}\bar{S}_l \quad (2h)$$

$$P_{l,i} = P_{l,i,b1} + P_{l,i,b2} \quad (2i)$$

$$Q_{l,i} = Q_{l,i,b1} + Q_{l,i,b2} \quad (2j)$$

where $\theta_{i,b}$ is the voltage angle of the substation i on each busbar b , and $\theta_{l,i}$ is the voltage angle of line ij on the i side, as Fig. 1 shows.

Constraints (3a)–(3d) consider operational assumptions of substations and improve the computational complexity:

$$\sum_{i \in \mathcal{V}} (1 - z_i) \leq \bar{Z}^s \quad (3a)$$

$$2(1 - z_i) \leq \sum_{l \in \mathcal{L}_i} z_{l,i} \quad i \in \mathcal{V} \quad (3b)$$

$$2(1 - z_i) \leq \sum_{l \in \mathcal{L}_i} (1 - z_{l,i}) \quad i \in \mathcal{V} \quad (3c)$$

$$z_i + z_e \leq 1 \quad e \in E_i \quad (3d)$$

where (3a) limits the total number of busbar splitting actions, as set by the system operator. (3b)–(3c) ensure system security by enforcing that at least two lines are connected to each busbar if that substation splits, which prevents busbar isolation due to a line contingency [7,11]. When the busbars are not split, the substation operates as a single node, and (3d) removes the symmetry in the formulation by enforcing all elements to connect to $b1$ in that case.

AC PF equations are crucial for accurately representing modern power systems, whereas DC formulations neglect reactive power and voltage constraints, which can lead to infeasible or sub-optimal solutions. However, using full AC equations results in a computationally expensive MINLP problem. In this paper, we use the linearized AC approximation [22] to achieve a balance of accuracy and efficiency in a MILP formulation:

$$\sum_{g \in \mathcal{G}_i} P_{g,b} - \sum_{d \in \mathcal{D}_i} P_{d,b} = \sum_{l \in \mathcal{L}_i} P_{l,i,b} \quad i \in \mathcal{V}, b \in \mathcal{B} \quad (4a)$$

$$\sum_{g \in \mathcal{G}_i} Q_{g,b} - \sum_{d \in \mathcal{D}_i} Q_{d,b} = \sum_{l \in \mathcal{L}_i} Q_{l,i,b} \quad i \in \mathcal{V}, b \in \mathcal{B} \quad (4b)$$

$$P_{l,i} = g_i V_{l,i} + g_l \frac{V_{l,i} - V_{l,j}}{2} - b_l (\theta_{l,i} - \theta_{l,j}) + P_{l,i}^L \quad l_{ij} \in \mathcal{L} \quad (4c)$$

$$Q_{l,i} = -b_l V_{l,i} - b_l \frac{V_{l,i} - V_{l,j}}{2} - g_l (\theta_{l,i} - \theta_{l,j}) + Q_{l,i}^L \quad l_{ij} \in \mathcal{L} \quad (4d)$$

$$P_{l,i}^2 + Q_{l,i}^2 \leq \bar{S}_l^2 \quad l_{ij} \in \mathcal{L} \quad (4e)$$

$$\underline{V}_i \leq V_{i,b} \leq \bar{V}_i \quad i \in \mathcal{V}, b \in \mathcal{B} \quad (4f)$$

where (4a)–(4b) present power balance for each substation's busbars. Active and reactive power flows are defined in (4c)–(4d), with $P_{l,i,c}^L, Q_{l,i,c}^L$ representing active and reactive power losses. (4e) models the thermal limit of lines. (4f) represents voltage limits. We use a piecewise linearization of quadratic constraint (4e), and a linear approximation of loss terms $P_{l,i}^L, Q_{l,i}^L$ around an initial operating point as in [22].

The NTO for CM (denoted as AC-CM-NTO) is written as (5), with the objective aiming to minimize grid congestion:

$$\min_z \quad \mathcal{O}^{CM} = \sum_{l \in \mathcal{L}} f \left(\left| \frac{S_l}{\bar{S}_l} \right|^\gamma \right) \quad (5)$$

s.t. (1)–(4).

where $\gamma \geq 1$ is a parameter, and $f(\cdot)$ is a convex function. The objective penalizes normalized line apparent flows (i.e., current magnitudes) [6]. In this work, we use $f(\cdot) = \text{clip}(\cdot, s^{th}, \infty)$, to restrict penalization to lines whose loading exceeds a specified threshold s^{th} (e.g., 80%), whereas the parameter $\gamma \geq 2$ shapes the penalty function by emphasizing highly loaded lines and down-weighting lightly loaded ones. We apply a piecewise linearization of the objective to obtain a MILP formulation.

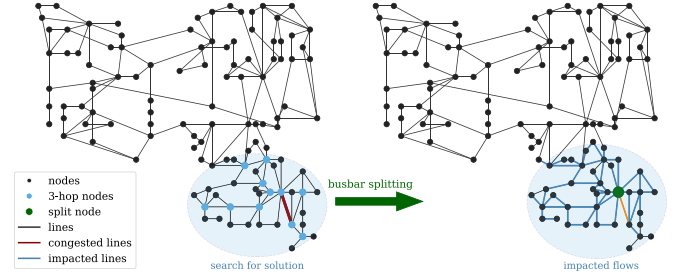


Fig. 2. An example of congestion in the IEEE 118-bus system. Left: potential busbar splitting actions within 3 hops. Right: line flows impacted by the splitting action.

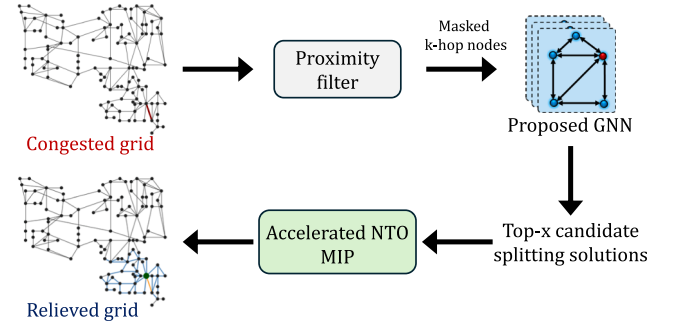


Fig. 3. Workflow of the proposed GNN-accelerated NTO approach.

3. Transferable graph learning approach

3.1. GNN-accelerated NTO approach

This section presents the proposed GNN-accelerated NTO (GNN-NTO) approach which significantly reduces the computational complexity of (5), enabling near-real-time solutions while ensuring linear AC PF constraints. As shown in Fig. 2 and previous studies [9,17], a busbar splitting action mostly impacts its local region, so effective actions are typically in close proximity to the congested line. This locality property of the NTO-CM problem makes GNNs a natural fit, as they use local information exchange and can improve scalability, generalization and transferability.

Fig. 3 and Algorithm 1 present the workflow of the proposed approach, consisting of three stages: proximity filter, GNN-based candidate prediction, and accelerated NTO.

3.1.1. Proximity filter

Given a congested grid state, the first step identifies substations likely to influence congestion. Congested lines $\mathcal{L}^{cong}$ are detected based on a threshold s_l^{th} (Line 1). A proximity filter then selects substations within k hops (e.g., 5 hops) of the congested lines and with at least four connected lines, as defined in (3c)–(3d), forming the proximity node set $\mathcal{V}^p$ (see Fig. 2). The filter does not remove input features; instead, it restricts the GNN output space to substations more likely to affect congestion, improving learning efficiency. During training, backpropagation is also applied only to these nodes.

3.1.2. GNN-based candidate prediction

The grid state is encoded as a graph and processed by the proposed GNN, which predicts a splitting score $\hat{y}_i$ for each substation $i \in \mathcal{V}^p$ (Line 1). These scores indicate how promising splitting a substation is for congestion mitigation. Substations are then ranked based on $\hat{y}_i$, and the top- x (e.g., top-3) are selected as candidate nodes $\mathcal{V}^{cnd}$ (Line 8).

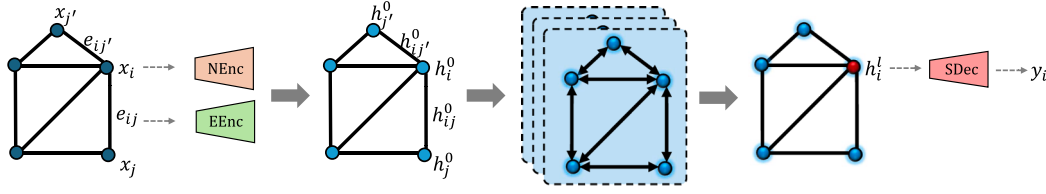


Fig. 4. Schematic of the proposed encoder-GNN-decoder architecture.

Algorithm 1: GNN-NTO Approach

Input: $\hat{P}_d, \hat{Q}_d, d \in \mathcal{D}, \hat{P}_g, \hat{Q}_g, g \in \mathcal{G}, P_l, Q_l, l \in \mathcal{L}$

- 1 Detect congested lines $\mathcal{L}^{cong} = \{l \in \mathcal{L} \mid S_l \geq s^{th}\}$;
- Proximity Filter**
- 2 **for** $l \in \mathcal{L}^{cong}$ **do**
- 3 **for** $i \in \mathcal{V}$ **do**
- 4 **if** $dist(l, i) \leq k$ **and** $Deg(v) \geq 4$ **then**
- 5 add i to $\mathcal{V}^p$;
- GNN-Based Candidate Prediction**
- 6 Forward pass of GNN $\hat{y}_i = \phi^{GNN}(x, e) \quad i \in \mathcal{V}^p$;
- 7 Select the top- x substations i from $\mathcal{V}^p$ ranked by $\hat{y}_i$, and add them to $\mathcal{V}^{cnd}$;
- Accelerated NTO;**
- 8 Fix non-candidate substations $z_i = 1 \quad i \notin \mathcal{V}^{cnd}$;
- 9 Prioritize candidate variables $z_i \quad i \in \mathcal{V}^{cnd}$ in B&B;
- 10 Solve reduced NTO* problem (5);
- 11 **return** optimal topology;

3.1.3. Accelerated NTO

The predicted candidate substations are incorporated into the NTO optimization. Substations outside the candidate set are fixed as unsplit nodes ($z_i = 1 \quad i \notin \mathcal{V}^p$), significantly reducing the number of binary variables (Line 10). Moreover, candidate variables $z_i \quad i \in \mathcal{V}^p$ are prioritized as *branching* variables in the branch-and-bound (B&B) process of the MIP solver [23] (Line 11). In other words, these variables are assigned higher branching priority compared to other binary variables (e.g., $z_{l,i}, z_g, z_d$), guiding the solver to explore promising topology decisions earlier in the search process. The reduced problem, denoted NTO*, is then solved using a commercial solver (Line 12).

By focusing the search on high-quality candidates predicted by the GNN, the proposed approach significantly accelerates the NTO-CM solution while maintaining high solution quality.

3.2. Learning task

This paper investigates two learning objectives: (1) *whether* to split a node, modeled as classification (Clf), and (2) *how much* congestion is reduced by splitting a node, modeled as regression (Reg). In both settings, the effect of splitting substation i is evaluated independently, assuming that all other substations remain unsplit.

Congestion can be reduced by reconfiguring substation i as:

$$\min_{z_i, \{z_e\}_{e \in E_i}} \mathcal{O}_i^{CM} \quad (6)$$

s.t. (1) – (4).

where (6) solely solves for the binary variables of elements of substation i . Consider $\mathcal{O}_0^{CM}$ as the initial objective without any topological actions. Then, the congestion reduction by splitting substation i is given by $\Delta \mathcal{O}_i^{CM} := \mathcal{O}_0^{CM} - \mathcal{O}_i^{CM}$.

Let y_i denote the GNN output label for substation i . In the Clf task, the goal is to predict *whether* splitting substation i reduces congestion. Therefore, $y_i^{Clf} = 1$ if $\Delta \mathcal{O}_i^{CM} > \mathcal{O}^{th}$, and $y_i^{Clf} = 0$ otherwise, where $\mathcal{O}^{th} \geq 0$ is a pre-defined threshold.

In the Reg task, the goal is to predict *how much* congestion reduction is achieved by splitting substation i , expressed as $y_i^{Reg} = \Phi(\Delta \mathcal{O}_i^{CM})$, where Φ is a convex monotonic function. The Reg task provides a richer estimate of action effectiveness, allowing better interpretability and prioritization of candidate actions by operators/MIP solvers. However, it requires hyperparameter tuning of γ in (5) and selecting Φ . In this work, we use a clipping function $\Phi(\cdot) = \text{clip}(\cdot, \Delta \mathcal{O}_i^{low}, \Delta \mathcal{O}_i^{high})$ to limit the influence of extreme values and stabilize the learning.

3.3. Transferable heterogeneous GNN architecture

We propose a transferable fully-local GNN approach to predict busbar splitting decisions on the substation level. Fig. 4 shows the proposed encoder-GNN-decoder architecture. Node and edge features are first processed through their respective encoders (each shared among all nodes or all edges) to obtain node and edge embeddings. The GNN layer then performs local message passing, followed by a splitting decoder that predicts the busbar splitting decision. Notably, the architecture relies solely on localized computations and is independent of the specific system or topology, i.e., the number of model parameters does not change with system size.

3.3.1. Edge-aware message passing

This section presents the homogeneous edge-aware message passing neural network (MPNN). The transmission grid is modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{L})$, where each node $i \in \mathcal{V}$ corresponds to a substation, and each edge $(i, j) \in \mathcal{L}$ represents a transmission line l_{ij} . Each line is represented as a pair of bi-directional edges to enhance expressiveness and capture flow symmetry.

Consider the nodal feature as $x_i = [p_i, q_i, V_i, \text{Deg}_i, \text{Dis}_i]$, where p_i, q_i are the net power injections (generation minus demand), V_i is voltage magnitude, Deg_i is the number of connected edges, and Dis_i is the number of hops to the closest congestion. The edge feature is defined as $e_{i,j} = [P_{l,i}, Q_{l,i}, S_{l,i}, \bar{S}_l, r_l, x_l]$. Let h_i^l and h_{ij}^l denote the node and edge embeddings at layer l . The proposed MPNN is:

$$h_i^0 = \phi^{NEnc}(x_i) \quad i \in \mathcal{V} \quad (7a)$$

$$h_{i,j}^0 = \phi^{EEnc}(e_{i,j}) \quad ij \in \mathcal{L} \quad (7b)$$

$$\tilde{h}_{i,j}^l = \phi_{\text{msg}}^l(|h_i^{l-1}| |h_j^{l-1}| |h_{i,j}^{l-1}|) \quad ij \in \mathcal{L} \quad (7c)$$

$$h_{i,j}^l = h_{i,j}^{l-1} + \tilde{h}_{i,j}^l \quad ij \in \mathcal{L} \quad (7d)$$

$$m_i^l = \sum_{j \in N_i} (h_{i,j}^l) \quad i \in \mathcal{V} \quad (7e)$$

$$h_i^l = h_i^{l-1} + \phi_{\text{upd}}^l(|h_i^{l-1}| |m_i^l|) \quad i \in \mathcal{V} \quad (7f)$$

$$\hat{y}_i = \phi^{SDec}(h_i^l) \quad i \in \mathcal{V}^p \quad (7g)$$

Here, $\phi(\cdot)$ shows a two-layer MLP with ReLU activation, and $||\cdot||$ denotes concatenation. (7a)-(7b) show the node and edge encoders, respectively. Edge messages are constructed in (7c), and the edge embeddings are updated with a residual connection in (7d). Messages are aggregated in

Algorithm 2: Training data generation.

```

1 for data sample  $s \in \Omega^s$  do
2   Sample inputs:  $\hat{P}_{d,s}, \hat{Q}_{d,s} \ d \in \mathcal{D}$ , outages  $l \in \Omega_s^o$ ,
   generation costs  $c_g \ g \in \mathcal{G}_s$ ;
3   Solve LP-AC-OPF to obtain  $\hat{P}_g, \hat{Q}_g \ g \in \mathcal{G}$ ;
4   Compute base congestion cost  $O_0^{CM}$ ;
5   if no congestion then
6     continue to next sample;
7   Apply proximity filter to obtain  $\mathcal{V}^p$ ;
8   for  $i \in \mathcal{V}^p$  do
9     Solve (6) for  $z_i, z_{e \in E_i}, O_i^{CM}$ ;
10    Compute congestion reduction  $\Delta O_i^{CM}$ ;
11    Assign label  $y_i$  according to Section III-B;

```

(7e), and node embeddings are updated in (7f), using a residual connection. After L layers of message passing, the split decoder (SDec) in (7g) outputs the final splitting decision. As mentioned earlier, we apply a proximity filter to improve learning performance: nodes $i \in \mathcal{V}^p$ are masked for prediction and used for backpropagation.

3.3.2. Heterogeneous GNN

We extend the homogeneous MPNN in (7) to a heterogeneous design that differentiates among node and edge types, providing a richer representation of the underlying power grid. To this end, we consider four node types: generators $g \in \mathcal{G}$, demands $d \in \mathcal{D}$, splitting nodes $i \in \mathcal{V}^p$, and non-splitting nodes $i \notin \mathcal{V}^p$. The splitting nodes are determined by the proximity filter for each grid sample. Moreover, we consider two edge types: transmission line $l_{ij} \in \mathcal{L}$, and pseudo connectors that link generators and demands to nodes. The pseudo connectors are featureless. We use separate $\phi_i(\cdot)$ functions to encode and process the embeddings of each node type i .

3.4. Training data generation

Algorithm 2 summarizes the training data generation process. First, random demands, planned $N-k$ line outages, and generation costs are sampled to emulate diverse operating conditions. The LP-AC-OPF is then solved to obtain the generation dispatch and initial power flows. To improve learning performance, only congested grid states are retained for training [17].

Next, the proximity filter is applied to determine the proximity node set $\mathcal{V}^p$. For each $i \in \mathcal{V}^p$, (6) is solved to evaluate the effectiveness of splitting substation i for congestion management. Since (6) only involves elements of substation i and assumes other substations as single electric nodes, these problems can be solved very efficiently and in parallel for all $i \in \mathcal{V}^p$. The label is then assigned based on the prediction task.

4. Case studies

4.1. Setting and test networks

The case studies are performed on small to large-scale IEEE and GOC test networks from Power Grid Library [24], assuming the substation model in Fig. 1. The thermal limits of the IEEE 118-bus network is reduced to 80% to induce congestion. Demand points are sampled within $\pm 20\%$ of the nominal load, following a Kumaraswamy(1.6, 2.8) distribution with a Pearson correlation coefficient of 0.75 to capture load correlations [25]. For topology variations, up to N-2 planned line outages are sampled uniformly from non-radial lines. Generation costs are also sampled uniformly within $\pm 20\%$ from nominal values to reflect

flow redistribution and potential generator outages. Initial dispatches and flows are determined by solving a cost minimizing OPF. The proximity filter assumes $k = 5$ hops. $\gamma = 2$ and $s_{th} = 0.8$ are used in (5). For each N-k case, 10K samples are generated under DC PF, and 10K samples are generated for the N-0 case under LP-AC PF. Non-congested and infeasible samples are excluded from the dataset. All optimizations are modeled in Pyomo and solved using Gurobi 10.0 with Python 3.12.

The dataset is randomly divided into 70% training, 10% validation, and 20% testing sets. ML models are implemented using PyTorch 2.5.0 and PyG 2.6.1. For classifier training, binary cross-entropy loss with class weighting is used. The output threshold is set to 0.5, and $\mathcal{O}^{th} = 0.05$ is used for labeling, though these thresholds can be adjusted. For regression training, the mean squared error (MSE) loss and $\Phi(\cdot) = \text{clip}(\Delta \mathcal{O}_i^{CM}, -0.2, \infty)$ are considered. All models are trained with the Adam optimizer up to 100 epochs at a learning rate of 10^{-3} with early stopping. Training and testing are performed on the DelftBlue supercomputer using 8 CPU cores (Intel Xeon E5-6248R 3.0 GHz), 32 GB RAM and NVIDIA V100S GPUs. For case studies, all models are run over 3 random seeds for data shuffling and weight initialization. The reported results are the average over these runs.

The ML models considered include an MLP, which uses concatenated node and edge features $x_i, e_{i,j}$ and one-hot encoding for line outages, and several GNN-based models with 5 layers and 64-dimensional hidden size: **Hom-GCN** (homogeneous graph convolutions), **Hom-MPNN** (homogeneous message passing Section 3.3.1), and **Het-MPNN** (heterogeneous message passing Section 3.3.2). All GNNs include the encoders and the splitting decoder in Fig. 4.

The following NTO-CM approaches are compared:

- **Org-NTO**: the original NTO-CM optimization in (5) which serves as the ground-truth.
- **5-hop-NTO**: busbar splitting limited to substations within $k = 5$ hops of congested lines.
- **No-SW**: no busbar splitting (switching) is applied.
- **Clf-GNN-NTO**: the proposed GNN-NTO using the top-5 actions predicted by the classification Het-MPNN.
- **Reg-GNN-NTO**: the proposed GNN-NTO using the top-5 actions predicted by the regression Het-MPNN.

4.2. Prediction accuracy

Table 1 compares the performance of different ML models in predicting the busbar splitting status of substations. Among the Clf evaluation metrics, accuracy can be misleading due to class imbalance (a model predicting mostly 'no-split' may still achieve high accuracy). F1-score balances precision and recall, reflecting how effectively the model identifies true splitting actions. The GCN shows the weakest performance due to its neglect of edge features. The MPNNs outperform the MLP, particularly in the larger GOC 2000-bus system by achieving an F1-score of 0.95 and an accuracy of 0.98. In other words, the fully-local MPNN with only 5 layers of message passing outperforms the *global* MLP with more parameters, which confirms the locality of the NTO-CM problem and shows that MPNNs scale more efficiently while improving performance by learning local PF patterns. Finally, the Hom-MPNN and Het-MPNN perform similarly, as effective splitting actions (on substation level) depend primarily on local flow patterns rather than inter-type relations.

4.3. Generalization to topology changes

Table 2 reports F1 scores and compares model generalization under topology changes. We consider two scenarios: (I) zero-shot generalization to unseen topologies, by training and testing on different N-k cases (e.g., training on N-0 samples and testing on N-1 samples), and (II) training on a combined dataset of N-0, N-1, and N-2 samples. For the IEEE-14 system, all models experience a performance drop from N-0 to N-1, while the drop from N-1 to N-2 is smaller for MPNN models. This

Table 1
Performance comparison for N-0 cases.

		F1 Score	Accuracy	Precision	Recall
ieee-118	MLP	0.87	0.93	0.81	0.94
	Hom-GCN	0.84	0.91	0.77	0.92
	Hom-MPNN	0.90	0.95	0.85	0.97
	Het-MPNN	0.90	0.95	0.85	0.95
ieee-300	MLP	0.87	0.91	0.83	0.91
	Hom-GCN	0.78	0.85	0.72	0.86
	Hom-MPNN	0.91	0.94	0.88	0.94
	Het-MPNN	0.91	0.94	0.86	0.95
goc-500	MLP	0.85	0.94	0.77	0.96
	Hom-GCN	0.79	0.91	0.68	0.95
	Hom-MPNN	0.89	0.96	0.83	0.97
	Het-MPNN	0.90	0.96	0.82	0.98
goc-2000	MLP	0.88	0.94	0.82	0.95
	Hom-GCN	0.86	0.94	0.77	0.96
	Hom-MPNN	0.95	0.98	0.91	0.98
	Het-MPNN	0.95	0.95	0.92	0.98

Table 2
Generalization to N-k topology changes.

	Train set	N-0	N-0	N-1	N-1	N-0, N-1, N-2
	Test set	N-0	N-1	N-1	N-2	N-0, N-1, N-2
ieee-14	MLP	0.97	0.79	0.93	0.79	0.92
	Hom-GCN	0.97	0.70	0.92	0.77	0.93
	Hom-MPNN	0.98	0.80	0.95	0.85	0.96
	Het-MPNN	0.98	0.81	0.96	0.86	0.96
ieee-118	MLP	0.87	0.79	0.82	0.79	0.83
	Hom-GCN	0.84	0.80	0.80	0.78	0.82
	Hom-MPNN	0.90	0.86	0.87	0.86	0.89
	Het-MPNN	0.91	0.88	0.88	0.87	0.90
ieee-300	MLP	0.87	0.85	0.85	0.84	0.87
	Hom-GCN	0.78	0.78	0.78	0.77	0.80
	Hom-MPNN	0.91	0.90	0.89	0.88	0.90
	Het-MPNN	0.91	0.90	0.89	0.88	0.91

is because MPNNs can capture topological features from N-1 samples, improving generalization to N-2 cases. As system size increases, the zero-shot performance drop diminishes, as line outages in larger grids have a smaller relative impact, and models experience more diverse local topologies during training. As expected, performance on the combined dataset is higher than in the zero-shot cases. The proposed MPNN enhances model expressiveness over baseline models, which improves generalization to topology changes and superior performance once trained on different topologies. This benefit is particularly pronounced in large-scale systems where topology changes frequently occur in practice.

4.4. Transferability across systems

Table 3 investigates the generalization and transferability of models across different power systems. We consider three cases: (I) **zero-shot (ZS)**: the GNN is trained on one system and tested on a different system without any retraining; (II) **transfer learning (TL)**: the GNN is first trained on the source system, and then only the encoders and decoder are retrained using 10% of the samples from the target system (this setup reflects a realistic scenario where generating training data for large systems is time-consuming); (III) **combined dataset (CD)**, the GNN is trained on a merged dataset from both source and target systems and evaluated on the target system. Training and testing samples include all N-0, N-1, and N-2 samples for each system. The MLP is excluded here due to different input and output dimensionality across systems.

Each power system has a distinct input and output distribution, posing a fundamental challenge for ZS transferability, as models trained on one system fail to generalize directly to another. However, in the TL case, retraining only the encoder and decoder with a limited number of

samples effectively mitigates the distribution shift, while preserving the learned GNN embeddings. As system size increases, models encounter more diverse topologies and feature patterns, which improves TL performance, achieving an F1-score of up to 0.92 for the GOC-500 and GOC-2000 systems. Finally, the proposed MPNN achieves the best results in the CD case, highlighting its capacity to capture local PF patterns that improve cross-system generalization. This result shows that, instead of generating extensive training data for large-scale grids (which is computationally expensive), one can efficiently train a universal GNN using data from decomposed, congestion-prone regions of the grid.

4.5. Computational efficiency

Table 4 compares the computational efficiency of different NTO-CM approaches over three cases: DC PF with maximum 1 splitting, DC PF with maximum 3 splittings, and AC PF with maximum 1 splitting. The MIPGap in Gurobi is set to 1%, and the time limit is 10 h. Top 5 predictions are used as candidates in GNN-NTO. The high optimality gap of the No-SW indicates the effectiveness of busbar splitting in reducing grid congestion (up to 10%). The negligible optimality gap of under 1% for the 5-hop-NTO confirms that considering the proximity filter and exploiting the locality property of the NTO-CM problem is effective, especially in larger systems where solving the full optimization becomes intractable. The GNN-NTO approaches achieve substantial computational speed-ups, up to 4 orders of magnitude for the GOC-2000 system. In this context, the Reg model achieves slightly lower optimality gaps than the Clf model, as it can better prioritize impactful splitting actions using the proposed congestion reduction index. However, this improved ranking performance comes with increased sensitivity to hyperparameters and training design. The practical differences between the Clf and Reg tasks are discussed in the next section. In the multiple switching case, the GNN-NTO optimality gap increases up to maximum 6% as the GNN is trained to predict the effectiveness of a single splitting action and may miss multiple cooperative actions. Finally, the computational advantage of GNN-NTO becomes even more pronounced for large-scale systems under AC PF, where even the proximity-filtered optimization is intractable; in such cases, GNN-NTO can still provide feasible solutions within a minute and 2.3% optimality gap. With more powerful hardware resources and efficient implementation (e.g., in lower-level languages or on GPUs), MIP process can be further reduced to a few seconds, achieving near-real-time performance.

4.6. Discussion

The presented case studies demonstrate promising results for the proposed GNN-NTO approach for congestion management. Table 1 shows that the MPNNs improve prediction accuracy compared to other architectures, while the heterogeneous GNN model does not provide additional benefit for the node candidate prediction task in this work. Table 2 shows that generalization to topology changes improves in larger systems, although zero-shot generalization to unseen topologies remains limited. Table 3 shows that the local formulation of the learning task together with the MPNN design improves transferability across systems with only minimal additional training or by training a universal model, while zero-shot generalization to entirely unseen systems remains a challenge. Finally, Table 4 demonstrates the substantial computational speed-up achieved by the GNN-NTO approach while maintaining relatively small optimality gaps. However, some limitations of our approach can be noted.

Regarding the learning tasks, the Reg provides a more informative signal for ranking candidate substations. However, it is also more sensitive to the choice of hyperparameters γ and Φ that define the congestion reduction index. Large values of γ emphasize highly congested lines but may increase sensitivity to small variations in the grid state. In addition, the regression index can take negative values when a split worsens

Table 3
Transferability across systems.

Source System	Target System	Zero-Shot	Transfer Learning	Combined Dataset	
ieeee-14	ieeee-118	MLP	–	–	–
		Hom-GCN	0.39	0.78	0.85
		Hom-MPNN	0.28	0.86	0.90
		Het-MPNN	0.30	0.86	0.90
ieeee-118	ieeee-300	MLP	–	–	–
		Hom-GCN	0.28	0.77	0.80
		Hom-MPNN	0.39	0.87	0.90
		Het-MPNN	0.31	0.87	0.91
goc-500	goc-2000	MLP	–	–	–
		Hom-GCN	0.32	0.83	0.85
		Hom-MPNN	0.30	0.91	0.93
		Het-MPNN	0.28	0.92	0.95

Table 4
Computational efficiency of different NTO-CM approaches.

		DC 1-split			DC 3-split			AC 1-split		
		Time	Speed up (×)	Gap (%)	Time	Speed up (×)	Gap (%)	Time	speed up (×)	gap (%)
ieeee-118	Org-NTO	18.3 s	–	–	2.2 min	1 ×	0.0	1.5 min	–	–
	5-hop-NTO	18.3 s	1 ×	0.4	45.7 s	3 ×	0.8	9.5 s	10 ×	0.5
	No-SW	0.4 s	46 ×	8.9	0.5 s	283 ×	23.1	0.6 s	149 ×	6.7
	Clf-GNN-NTO	0.9 s	21 ×	2.2	5.2 s	26 ×	7.3	2.7 s	34 ×	3.3
	Reg-GNN-NTO	1.3 s	14 ×	1.7	10.0 s	13 ×	2.2	3.2 s	29 ×	1.2
ieeee-300	Org-NTO	49.8 s	–	–	12.1 min	1 ×	0.0	3.4 min	–	–
	5-hop-NTO	24.0 s	2 ×	0.4	8.1 min	1 ×	0.7	2.0 min	2 ×	0.2
	No-SW	0.6 s	84 ×	10.2	0.8 s	949 ×	41.1	1.5 s	140 ×	1.4
	Clf-GNN-NTO	1.5 s	33 ×	5.5	6.6 s	110 ×	12.1	4.0 s	51 ×	0.4
	Reg-GNN-NTO	2.1 s	24 ×	1.1	22.9 s	32 ×	6.0	9.3 s	22 ×	0.2
goc-500	Org-NTO	4.1 min	–	–	35.6 min	1 ×	0.0	46.7 min	–	–
	5-hop-NTO	2.1 min	2 ×	0.8	5.9 min	6 ×	3.2	17.2 min	3 ×	0.0
	No-SW	1.1 s	230 ×	2.5	1.2 s	1838 ×	10.6	3.9 s	712 ×	1.7
	Clf-GNN-NTO	2.3 s	108 ×	1.9	11.0 s	194 ×	5.1	8.9 s	314 ×	2.6
	Reg-GNN-NTO	4.1 s	61 ×	0.7	17.8 s	120 ×	2.5	19.1 s	147 ×	2.1
goc-2000	Org-NTO	>10.0 hr	–	–	>10.0 hr	–	–	>10.0 hr	–	–
	5-hop-NTO	3.4 hr	>3 ×	–9.2	>10.0 hr	>1 ×	–3.0	>10.0 hr	>1 ×	–6.7
	No-SW	5.5 s	>6510 ×	0.0	6.3 s	>5735 ×	0.0	14.0 s	>2573 ×	0.0
	Clf-GNN-NTO	20.0 s	>1799 ×	–4.3	38.3 s	>940 ×	–0.5	54.0 s	>666 ×	–2.8
	Reg-GNN-NTO	11.8 s	>3049 ×	–6.3	2.5 min	>244 ×	–1.1	1.3 min	>462 ×	–4.4

congestion, which can bias the learning objective if not properly controlled. Therefore, clipping in Φ is used for stable training. In this work, the regression formulation is explored as a proof-of-concept for predicting congestion reduction magnitudes to improve candidate selection. Further work is needed to refine this approach.

The GNN predicts the effectiveness of a busbar splitting action while assuming that no other substations are split. Consequently, when multiple splitting actions are required to mitigate congestion, the current approach identifies them sequentially, which does not guarantee finding the optimal combination. In addition, although the GNN-NTO approach provides significant computational speed-ups during online use, generating labeled training data remains computationally expensive for large-scale systems. Improving zero-shot transferability can address this limitation by enabling models trained on smaller systems to be applied directly to larger systems.

5. Conclusion and future work

This paper presents a GNN-accelerated approach (GNN-NTO) for congestion management via busbar splitting. The proposed approach significantly reduces computations, enabling near-real-time solutions with operational feasibility. To this end, we develop a linearized AC NTO formulation to reduce grid congestion. A novel edge-aware MPNN is proposed to predict effective nodes for busbar splitting actions, which serve as candidate solutions for the MIP problem. A proximity filter

further improves learning performance by focusing the prediction task on the most relevant local regions. Two learning tasks are explored: a classification task that predicts whether splitting a substation alleviates congestion, and a regression task that estimates the congestion reduction associated with each splitting action. While the regression formulation can improve the prioritization of candidate nodes and provide interpretability, it is also more sensitive to hyperparameter choices and requires further investigation. Case studies demonstrate that the proposed MPNN learns generalizable features from local PF patterns. In particular, the results show that generalization to topology changes improves in larger systems, and that transfer learning across systems is effective with only minimal additional training, allowing a pre-trained GNN to be efficiently adapted to new grids or enabling a universal model to be trained across multiple systems. These properties are particularly valuable for large-scale systems with frequent topology changes, varying grid conditions, and expensive data collection. However, zero-shot generalization to entirely unseen systems remains challenging due to distributional shifts in the input and output spaces. Moreover, the GNN-NTO achieves up to 4 orders of magnitude speed-ups for the large-scale GOC 2000-bus system under AC PF, providing a feasible solution in one minute with a minor optimality gap.

Future research will focus on zero-shot generalization to unseen grids, addressing distributional shifts in ML-based optimization proxies, and extending the approach for multiple busbar splitting actions.

CRediT authorship contribution statement

Ali Rajaei: Writing – original draft, Visualization, Software, Methodology, Conceptualization; **Peter Palensky:** Supervision; **Jochen L. Cremer:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work is supported by the TU Delft AI Labs & Talent Programme. The authors are thankful to Olayiwola Arowolo from Delft University of Technology who provided valuable discussions.

Data availability

Data will be made available on request.

References

- [1] A. Marot, B. Donnot, S. Tazi, P. Panciatici, Expert system for topological remedial action discovery in smart grids, in: Mediterranean Conference on Power Generation, Transmission, Distribution and Energy Conversion (MEDPOWER 2018), IET, 2018, pp. 1–6.
- [2] T. Wu, A. Scaglione, S. Miguel, D. Arnold, Universal graph learning for power system reconfigurations: transfer across topology variations, *arXiv preprint arXiv:2509.08672* (2025).
- [3] Y. Zhou, H. Zhu, Bus split sensitivity analysis for enhanced security in power system operations, in: 2019 North American Power Symposium (NAPS), IEEE, 2019, pp. 1–6.
- [4] L. Wang, H.-D. Chiang, Bus-bar splitting for enhancing voltage stability under contingencies, *Sustain. Energy Grids Netw.* 34 (2023) 101010.
- [5] J. van Dijk, J. Viebahn, B. Cijssouw, J. van Casteren, Bus split distribution factors, *IEEE Trans. Power Syst.* 39 (3) (2023) 5115–5125.
- [6] S. Babaeinejadsarookolaee, B. Park, B. Lesieutre, C.L. DeMarco, Transmission congestion management via node-breaker topology control, *IEEE Syst. J.* 17 (3) (2023) 3413–3424.
- [7] M. Heidarifar, H. Ghasemi, A network topology optimization model based on substation and node-breaker modeling, *IEEE Trans. Power Syst.* 31 (1) (2015) 247–255.
- [8] E.A. Goldis, P.A. Ruiz, M.C. Caramanis, X. Li, C.R. Philbrick, A.M. Rudkevich, Shift factor-based SCOPF topology control MIP formulations with substation configurations, *IEEE Trans. Power Syst.* 32 (2) (2016) 1179–1190.
- [9] B. Morsy, J. Stiasny, A. Anta, J. Cremer, Configure-and-bound: a fast heuristic for network topology reconfiguration, *Int. J. Electr. Power Energy Syst.* 174 (2026) 111517.
- [10] B. Park, C.L. Demarco, Optimal network topology for node-breaker representations with AC power flow constraints, *IEEE Access* 8 (2020) 64347–64355.
- [11] M. Heidarifar, P. Andrianesis, P. Ruiz, M.C. Caramanis, I.C. Paschalidis, An optimal transmission line switching and bus splitting heuristic incorporating AC and N-1 contingency constraints, *Int. J. Electr. Power Energy Syst.* 133 (2021) 107278.
- [12] E. van der Sar, A. Zocca, S. Bhulai, Optimizing power grid topologies with reinforcement learning: a survey of methods and challenges, *arXiv preprint arXiv:2504.08210* (2025).
- [13] A. Chauhan, M. Baranwal, A. Basumatary, PowRL: a reinforcement learning framework for robust management of power networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, 37, 2023, pp. 14757–14764.
- [14] A.R.R. Matavalam, K.P. Guddanti, Y. Weng, V. Ajjarapu, Curriculum based reinforcement learning of grid topology controllers to prevent thermal cascading, *IEEE Trans. Power Syst.* 38 (5) (2022) 4206–4220.
- [15] M. Lehna, J. Viebahn, A. Marot, S. Tomforde, C. Scholz, Managing power grids through topology actions: a comparative study between advanced rule-based and reinforcement learning agents, *Energy AI* 14 (2023) 100276.
- [16] M. Dorfer, A.R. Fuxjäger, K. Kozak, P.M. Blies, M. Wasserer, Power grid congestion management via topology optimization with alphazero, *arXiv preprint arXiv:2211.05612* (2022).
- [17] G.J. Meppelink, A. Rajaei, J.L. Cremer, A hybrid curriculum learning and tree search approach for network topology control, *Electr. Power Syst. Res.* 242 (2025) 111455.
- [18] M. Hassouna, C. Holzhüter, M. Lehna, M. de Jong, J. Viebahn, B. Sick, C. Scholz, Learning topology actions for power grid control: a graph-based soft-label imitation learning approach, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2025, pp. 129–146.
- [19] M. de Jong, J. Viebahn, Y. Shapovalova, Generalizable graph neural networks for robust power grid topology control, *arXiv preprint arXiv:2501.07186* (2025).
- [20] D. Meng, R. Haider, P. van Hentenryck, Flow-aware GNN for transmission network reconfiguration via substation breaker optimization, *arXiv preprint arXiv:2508.01951* (2025).
- [21] C.R. Bayliss, B.J. Hardy, Substation layouts, in: Transmission and Distribution Electrical Engineering, Newnes, Oxford, third edition edition, 2007, pp. 92–114. <https://doi.org/10.1016/B978-075066673-2/50007-0>
- [22] M.I. Alizadeh, F. Capitanescu, A tractable linearization-based approximated solution methodology to stochastic multi-period AC security-constrained optimal power flow, *IEEE Trans. Power Syst.* 38 (6) (2022) 5896–5908.
- [23] L. Gurobi Optimization, Gurobi optimizer reference manual, 2023, (n. d). <https://www.gurobi.com>.
- [24] S. Babaeinejadsarookolaee, et al., The power grid library for benchmarking ac optimal power flow algorithms, *arXiv preprint arXiv:1908.02788* (2019).
- [25] B.N. Giraud, A. Rajaei, J.L. Cremer, Constraint-driven deep learning for N-k security constrained optimal power flow, *Electr. Power Syst. Res.* 235 (2024) 110692.