Contents lists available at ScienceDirect

Energy and AI

journal homepage: www.elsevier.com/locate/egyai

Graph reinforcement learning for power grids: A comprehensive survey

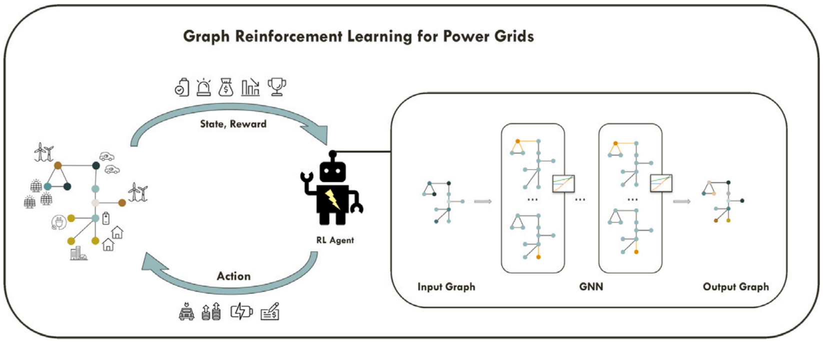
Mohamed Hassouna [a,b,*,1], Clara Holzhüter [a,b,*,1], Pawel Lytaev [c], Josephine Thomas [d], Bernhard Sick [b], Christoph Scholz [a,b]

[a] Fraunhofer Institute for Energy Economics and Energy System Technology (IEE), Joseph-Beuys-Straße 8, Kassel, 34117, Germany
[b] Intelligent Embedded Systems, University of Kassel, Germany
[c] Department of Sustainable Electrical Energy Systems, University of Kassel, Germany
[d] Machine Learning, University of Greifswald, Germany

HIGHLIGHTS

- Graph Reinforcement Learning provides scalable solutions where traditional solvers fail.
- Graph Neural Networks outperform standard neural networks in grid control.
- The simulation-to-reality gap is the main barrier to Graph Reinforcement Learning deployment.
- Trustworthy approaches demand safety, transparency, and physics integration.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Graph Reinforcement Learning
Graph Neural Networks
Reinforcement Learning
Power grid control
Voltage control

ABSTRACT

The increasing share of renewable energy and distributed electricity generation requires the development of deep learning approaches to address the lack of flexibility inherent in traditional power grid methods. In this context, Graph Neural Networks are a promising solution due to their ability to learn from graph-structured data. Combined with Reinforcement Learning, they can be used as control approaches to determine remedial actions. This review analyzes how Graph Reinforcement Learning can improve representation learning and decision-making in power grid applications, particularly transmission and distribution grids. We analyze the reviewed approaches in terms of the graph structure, the Graph Neural Network architecture, and the Reinforcement Learning approach. Although Graph Reinforcement Learning has demonstrated adaptability to unpredictable events and noisy data, its current stage is primarily proof-of-concept, and it is not yet deployable to real-world applications. We highlight the open challenges and limitations for real-world applications.

1. Introduction

The role of electrical grid operators, both for transmission and distribution grids, is to ensure cost-efficient system availability at all times. However, power systems worldwide are undergoing a paradigm shift driven by the need for CO2 neutrality. The integration of distributed renewable generation and additional load demand due to heating and traffic sector electrification introduce complexities that

[*] Corresponding authors at: Fraunhofer Institute for Energy Economics and Energy System Technology (IEE), Joseph-Beuys-Straße 8, Kassel, 34117, Germany.
E-mail addresses: mohamed.hassouna@iee.fraunhofer.de (M. Hassouna), clara.juliane.holzhueter@iee.fraunhofer.de (C. Holzhüter).

[1] Both authors contributed equally to this work.

https://doi.org/10.1016/j.egyai.2025.100671
Received 25 June 2025; Received in revised form 10 December 2025; Accepted 21 December 2025
Available online 2 January 2026

traditional power system operation struggles to cope with. These trends require advanced methods for optimal operation [1,2]. The ongoing energy transition also impacts other stakeholders, including energy market participants. They need to adapt to the decentralized structure and new market players such as Electric Vehicle (EV) charging operators. Additionally, the ongoing digitization and build-up of communication systems transform the classical power system into a cyber-physical energy system (CPES) [3]. These new challenges introduce a new layer of complexity to power grid operation.

Traditionally, grid operation has mostly relied on optimization approaches for Optimal Power Flow (OPF) problems [4,5]. However, due to the non-linear and non-convex nature of OPFs, these approaches struggle to scale to real power grids such that exact results cannot be obtained in a reasonable time [6,7]. Therefore, relaxation techniques are used to reduce the complexity [8–10]. However, they can produce imprecise results and cannot guarantee optimality, which casts doubt on their effectiveness in practical applications.

Furthermore, noisy or missing measurements cannot be reliably handled by classical approaches as they often assume ideal conditions and typically do not incorporate statistical models to distinguish between signal and noise [11]. As a result, even small perturbations in the input can strongly affect the output. Therefore, practitioners are exploring deep learning solutions [12–14] for power flow problems. Such solutions are promising alternatives to classical approaches, addressing the challenges of time criticality, scalability, and reliability of results.

Deep Reinforcement Learning (DRL) techniques can identify and exploit underutilized flexibility in power grids, which is often overlooked by traditional methods and human operators [2]. By learning from interactions with the grid environment, DRL agents can dynamically adapt to changing conditions and unforeseen events, which can potentially prevent cascading failures and blackouts [1,15]. Furthermore, their ability to consider long-term horizons aligns with the dynamic nature of power grids [16]. However, the development and training of DRL agents requires extensive simulations, as direct interaction with the physical grid is impractical. These simulations often need to abstract from reality and rarely use real data, leading to challenges in transferring the solutions back to real-world applications [17]. Furthermore, the large combinatorial action spaces in power grids hinder the application of DRL [18], highlighting the need for handcrafted action spaces and other reduction techniques [19–21]. Despite these challenges, the potential for Reinforcement Learning (RL) in power grid management is significant, particularly as power systems strive to meet decarbonization goals [22,23]. The aim is not to replace human operators but to provide them with RL-driven action recommendations [16,24]. Despite impressive proof-of-concept results, DRL research for grid control is still in its early stages, and significant gaps remain to be filled before deployment.

Power grids can be naturally modeled as a graph, in which nodes and edges correspond to grid elements and the connections between them [16]. Since Graph Neural Networks (GNNs) are designed explicitly for such graph-structured data, they are highly suitable for modeling interdependencies in power systems [25]. They can capture relationships between elements and enable an effective feature extraction from the grid. While standard, i.e., non-graph-based, neural networks struggle to produce accurate results when the grid's topology changes, GNNs are more robust to modifications of the graph structure. This is an advantage, as several grid actions, such as bus-bar splitting, can transform a single node into two distinct nodes (or combine two nodes into one). This is uncommon in other types of networks [26] and thus requires tailored methods. Similarly, long-range dependencies and the rapid propagation speed of electricity pose unique challenges. As GNN research is strongly driven by standard benchmark datasets such as citation networks, molecules, or social networks, most architectures are not entirely suited for the graph structure and properties of power grids. Therefore, the design of tailored GNN architectures, including the generalization across different topologies and grids, still requires extensive research [27]. With regard to the practical applicability of GNNs, the interpolation capabilities of GNNs are crucial in reconstructing missing information and smoothing out noisy measurements [28]. This takes into account the fact that sensors can experience connectivity problems, resulting in incomplete or unreliable data.

The combination of GNNs and RL represents a synergy that exploits the strengths of both paradigms. GNNs provide a powerful tool for feature extraction in graph-structured data, enhancing the RL agent's understanding of the complex relationships within power grids. As pointed out by [29], the performance of RL agents heavily depends on the state encoding, and GNNs are much better encoders for graph-structured environments. Incorporating them into RL has the potential for more informed decision-making, better adaptability to changing network conditions and noise, and improved generalization across different scenarios and topologies.

Existing works. This survey provides a comprehensive overview of existing GRL approaches for grid applications, filling a significant gap in the literature. While the Graph Reinforcement Learning (GRL) literature on power systems is growing, it remains fragmented across GNN, RL, GRL, and other deep learning approaches for power grids. This highlights the need for a comprehensive comparison of existing approaches to inform future work.

Table 1 lists the works discussed in more detail below, along with the methodologies and use cases they address. For clarity, the table only includes works that address at least one methodology and one use case. Broader GNN and RL surveys are not included.

From a methodological perspective, several broader surveys on GNNs cover various methodologies and applications. For example, [30, 31], and [32] provide overviews of common architectures. There are also various reviews with a specific focus, such as [33], which focuses on dynamic graphs that evolve over time, and [34] examines GNNs in recommender systems. For the power grids use case, two reviews on specialized GNNs are available. [25] highlight the superior performance of GNNs over standard neural networks, particularly in fault analysis, time series prediction, power flow calculation, and data generation. The second paper, [35], focuses on the power flow problem and the respective benefits and challenges. Neither paper covers the combination with RL or other sequential decision-making algorithms.

Similarly in RL, general reviews include [36,37], and [38]. [39] explores DRL for energy systems, focusing on problems such as demand response, electricity markets, and operational control. Specialized reviews, such as [40], focus on demand response in smart grids but do not cover GNNs, which is a more recent development compared to traditional RL approaches. Furthermore, a recent study analyzes RL approaches for power grid control, applied to the Grid2op framework [41].

Literature reviews combining GNN and RL are rare. [29] survey 80 relevant papers, categorizing them into DRL-enhancing GNNs and GNNs-enhancing DRL. The former includes DRL for architecture search and improving GNN explainability, while the latter covers the use of GNN in DRL, which is more closely related to our work. They explore areas like combinatorial optimization and transportation, but exclude energy applications.

[42] survey GRL approaches with a focus on the methodology of GNNs and RL, especially in multi-agent settings where GNNs facilitate agent communication. They primarily explore how graphs and RL interact, while we focus on using GNNs as feature extractors for graph-structured power grid data. Although they briefly mention an energy-related application, it is not analyzed in detail as their review does not emphasize application-specific approaches.

Similarly, the survey presented by [43] examines GRL methodologically, detailing how RL can enhance GNNs and address graph problems. They cover various transportation and medical research applications but do not address energy-related use cases.

Several surveys focus on specific aspects of GRL: [44] examines GNNs for representation learning in combinatorial optimization within

Table 1
Overview of existing surveys that address at least one power grid use case and one methodological aspect.

Reference	Methodology			Grid use case		
	GNNs	RL	GRL	Transmission	Distribution	Other
[25] (Liao et al. 2021)	✓			✓	✓	
[35] (Li et al. 2023)	✓			✓	✓	
[39] (Zhang et al. 2019)		✓		✓	✓	✓
[40] (Vázquez-Canteli et al. 2019)		✓		✓	✓	✓
[41] (van der Sar et al. 2025)		✓	(✓)	✓		
[42] (Fathinezhad et al. 2023)	✓	✓	(✓)			(✓)
Ours	✓	✓	✓	✓	✓	✓

RL. [45] detail how graph algorithms can enhance RL through action abstraction, while [46] explore hierarchical RL with graph-based approaches for discovering subtasks. None of these surveys addresses power grids.

Further related works analyze specific power grid problems such as voltage control (e.g. [6,47]). However, they mostly review traditional optimization and heuristics. To our knowledge, no comprehensive overview of GRL in power grids exists. Therefore, this work aims to fill this gap.

Contribution and structure. Addressing the fragmented existing literature on GRL approaches for power grids mentioned in the previous section, the core contribution of this survey is to provide the first comprehensive analysis of GRL approaches designed for power grid applications, covering transmission and distribution grids. We also investigate other power grid applications, such as the energy market, communication networks within power systems, and EV charging management, if they consider the underlying power grid in their approach. We focus on approaches that utilize GNNs to capture the graph-structured state space of power grids while applying DRL for sequential decision-making.

The main contributions are as follows:

- **Comprehensive review:** We are the first to provide a comprehensive analysis of existing GRL approaches for power grid use cases. We provide an overview of the applied GRL techniques, including states, actions, and rewards, and analyze the proposed GNN architecture in detail. We highlight the commonalities and differences between the analyzed methods and identify the most common approaches.
- **Categorization**: We categorize the approaches based on the specific scenarios they address. Our analysis focuses on applications in distribution and transmission grids. Within the distribution grid, we differentiate between regular voltage control and emergency situations. For the transmission grid, our focus is on topology control and the relevant frameworks.
- **Future Directions:** We highlight crucial aspects for the application of GRL in real-world scenarios and investigate limitations and open challenges of the proposed approaches.

The papers we analyze were published between 2020 and May 2025, as GRL is a relatively new field. Our selection includes those papers that explicitly address power grids, including the underlying grid structure. To maintain a focused and in-depth analysis of GRL applied to large-scale grid operation, we deliberately limit our primary scope to transmission and distribution systems. Although GRL approaches exist for microgrids, the distinct operational and modeling complexities of these systems (e.g., islanding and distributed control) fall outside the scope of the present review. Therefore, we only include approaches for zoned grids and microgrids if they focus on the use case of operational control.

This review is structured as follows: In Ch. 2, we present the basics of transmission and distribution grids, GNNs and RL. In the main part of this review, we discuss the presented methods in detail and categorize them by the use case they address. This part begins with approaches for common problems in transmission grids (Ch. 3) and continues with applications in distribution grids (Ch. 4). Then, in Ch. 5, we highlight several relevant papers catering to related use cases, such as EV charging. Finally, we give an overall conclusion and overview of key challenges and future directions.

2. Fundamentals: Power grids, graph neural networks and reinforcement learning

2.1. Power grids

Power grids are essential to modern society and a crucial part of today's infrastructure. In the face of the energy transition, power system engineers encounter challenges in all aspects of the grid. Their purpose is to transport electricity from generation units to consumers, who are typically not located in the same area. Traditionally, generation has been centralized, for example, at fossil-fueled or nuclear power plants, resulting in a unidirectional power flow from generation to consumption. With the worldwide expansion of renewable energy, generation units are spread across the power system, resulting in a decentralized structure and a bi-directional power flow. With the electrification of the transportation and heating sectors, the consumption side of power grids is also undergoing a significant shift. This sector coupling not only increases overall consumption but also introduces new patterns of simultaneous demand. For instance, during winter, multiple heat pumps may operate concurrently across the network, which can significantly impact its load.

Power grids are typically divided into two levels: transmission and distribution. These levels are split by substations (see Fig. 1) and differ in voltage levels, purpose, and characteristics. We elaborate on both levels in the following subsections.

2.1.1. Transmission grids

The purpose of transmission grids is to transport large amounts of electricity over long distances that vary from a few hundred kilometers to a few thousand [48]. Transmission grid operation aims to achieve at least N-1 secure operation. This means that, if one asset in the grid fails, the remaining grid is in a safe state. Therefore, transmission grids are typically built in meshed structures with redundancies installed, e.g., multiple transformers and busbars at substations [49]. The structure of transmission grids yields a highly complex system that necessitates solving large-scale optimization problems.

One measure to prevent high grid congestion, such as line overloading, is re-dispatch, which refers to changing generator injections [50]. Since generation and consumption in a power system must be balanced at all times, changing the set point of one generator set must result in the corresponding adjustment of another. This can lead to renewable generators being shut down and fossil fuel generators ramping up, which is undesirable in terms of both cost and CO2 neutrality. Therefore, other means of flexibility are being explored, such as topology control. By controlling the switching state in substations, the grid's topology can be modified, helping to reduce or even eliminate the need for re-dispatch. Optimizing the topology is a challenge in itself, as it results in a Mixed-Integer Non-Linear Problem. Here, deep learning solutions such as RL can help [12].

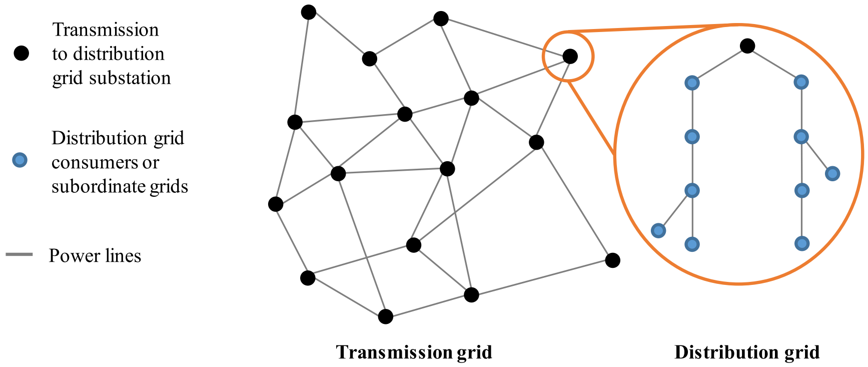

Transmission to distribution grid substation
Distribution grid consumers or subordinate grids
Power lines
Transmission grid
Distribution grid

Fig. 1. Visualization of the power grid structure with transmission and distribution level.

2.1.2. Distribution grids

Distribution grids cover smaller areas, such as villages, and often do not follow the N-1 safety criterion due to the lower impact of asset failures [51]. They are undergoing a major shift due to the transformation from unidirectional power flows to bidirectional power flows coming from distributed generation.

Voltage volatility is higher due to time-varying generation and consumption, particularly with the increased adoption of photovoltaic generation. This can lead to voltage fluctuations. Traditional voltage control uses regulated transformers, shunt capacitors, and voltage regulators [6]. With digitalization, voltage control options are expanding to include inverter-based technologies such as smart PV inverters [52], vehicle-to-grid systems [53], and stationary batteries [54]. These are flexibilities that grid operators must tap to maintain the system in a secure state.

Additionally, given the number of distribution grids, optimized control strategies need scalable and replicable solutions [53].

2.1.3. Grid models

In research, synthetic grid models are widely used as a standard benchmark for evaluating new approaches and comparing results. Examples in literature include [55–58]. In the transmission and distribution grid use cases mentioned above, the IEEE test cases are the most commonly used benchmarks, available in several power system calculation tools, such as MATPOWER [59], pandapower [60], and DIgSILENT PowerFactory [61]. Although they comprise grid topology data, line connectivity, generator and load placements, line impedances, and capacity limits, these benchmark cases are simplified representations of power systems that do not fully capture their complexities and challenges. They are mostly simple bus branch models with loads and generators connected to the buses. More sophisticated assets, such as inverter-based generator controls, transformer tap changers, and shunt elements, are not commonly found in such grid models but are prevalent in real power grid operations. Moreover, these test cases often lack critical operational conditions such as fluctuating demand, renewable generation variability, or network contingencies like line faults. Therefore, methods benchmarked on such synthetic grids must be treated with caution, as their application in practice requires further consideration. .

2.2. Graph neural networks

GNNs are designed to extract information from graph-structured data by applying multiple layers of graph convolutions. They can be interpreted as a generalization of Convolutional Neural Networkss (CNNs) to non-Euclidean structured data. The general idea is to combine information from local regions of the inputs in a learnable way and to grow these local regions from layer to layer. In this way, CNN layers learn increasingly abstract features from the input data. CNNs perform exceptionally well on grid-structured data, such as images. However, many real-world phenomena involve relationships or complex dependencies that cannot be represented as regular grid structures without losing information. For example, in an image, every node (pixel) has the same number of neighbors, but in power grids, not every component is connected to the same number of power lines.

Graphs consist of an unordered set of nodes and edges, where the edges define the neighborhood of a node. Graphs can, therefore, be used to model complex relationships, such as neighborhoods of arbitrary size or multiple types of edges or nodes. They can have attributes that describe properties of nodes and edges, such as node features or the strength of an edge. This makes graphs a flexible model for many real-world applications, such as power grids. Additionally, feed-forward neural networks operating on Euclidean data typically treat nodes as independent samples. This means that they neglect the relationships between nodes or stack them unsystematically into a vector. GNNs, on the contrary, make use of the information about node connectivity. They can solve all common learning tasks, i.e., classification, regression, and clustering, for entire graphs and at node- or edge-level.

2.2.1. Message passing

GNNs update the embeddings of the graph nodes by repeatedly aggregating information of their neighborhoods in a learnable way. This general scheme is known as message passing (see Fig. 2), where each node updates its embedding based on the messages it receives from its neighbors. The representation $\boldsymbol{h}'_u$ of a target node u generated by a general message passing layer is computed as:

$$\boldsymbol{h}'_u = \sigma\left(\boldsymbol{h}_u, \bigoplus_{v\in\mathcal{N}(u)} \psi(\boldsymbol{h}_u, \boldsymbol{h}_v)\right), \tag{1}$$

where ψ corresponds to a message function equipped with learnable weights that compute a message between node u and its neighbor v. $\boldsymbol{h}_u$ and $\boldsymbol{h}_v$ are the respective node embeddings. In the first layer, these embeddings are simply the initial node features. $\mathcal{N}(u)$ is the neighborhood of node u, $\bigoplus$ refers to an aggregation function that defines how messages are passed [63], and σ is an activation function. There exist various implementations of message passing layers; the scheme shown above is the most general one [63]. This section describes the message passing schemes most frequently used among the analyzed approaches.

Message passing in GNNs is linked to how electrical quantities are coupled in a power grid. The neighbor aggregation reflects the way buses in a power grid are electrically linked via the nodal admittance matrix. Stacking layers captures multi-hop electrical influence, similar to a few iterations of simple iterative power-flow methods (e.g., Gauss–Seidel method for meshed grids), where voltage and flow updates propagate along the network until nodal balances are satisfied. This provides a natural, physics-aligned inductive bias for learning on power grids.

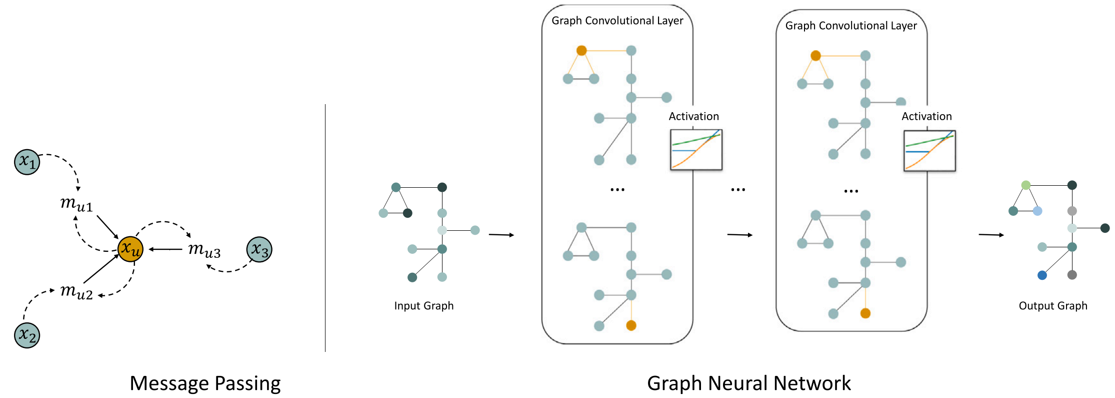


Fig. 2. **Left**: Visualization of the general message passing scheme in GNNs (modeled after [62]) — The target node (orange) receives messages $m_u i$ from its neighbors and aggregates them. The messages can be constructed from the information of both the target and neighboring node, depending on the message passing scheme. **Right**: Illustration of a GNN (modeled after [32]) - The graph is input to the GNN layers, which compute node embeddings based on the messages from neighboring nodes. As indicated in orange, this is done for each node in the graph. After all embeddings are computed, an activation function is applied. This is repeated for a given number of layers. In the end, the GNN outputs a graph with new node embeddings from which a prediction can be made.

Spatial graph convolution. A simple form of message passing is the spatial graph convolution. Here, messages are the embeddings of the neighboring nodes transformed using a learnable weight matrix. The aggregation corresponds to the summation operation. [64], for example, propose such an intuitive formulation:

$$\boldsymbol{h}'_u = \sigma(\boldsymbol{W}\boldsymbol{h}_u + \sum_{v\in\mathcal{N}(u)} \boldsymbol{W}\boldsymbol{h}_v) \tag{2}$$

with $\boldsymbol{W}$ being learnable weight matrices, also referred to as filters. Typically, the weights are shared across all nodes and neighbors. This concept follows the parameter sharing approach in CNNs.

GraphSage. The architecture proposed by [65] is a special case of spatial graph convolution based on sampling. Instead of aggregating the entire neighborhood of a node in each layer, a fixed number of neighbors of the target node are randomly sampled. The neighbors are aggregated using a permutation-invariant function, such as the mean or maximum. GraphSage is trained in an unsupervised manner using a special loss function. It consists of two terms: one that enforces nodes close in the input graph have similar embeddings, and the other that pushes apart the embeddings of two nodes that are far apart in the graph.

Graph attention network. A method commonly used to improve the performance of a GNN model is to equip the graph convolution with an attention mechanism. Such layers are a special case of general message passing [63] where attention coefficients are learned for each connected pair of nodes. They are computed from the features of the neighboring nodes and the target node and determine the influence a neighbor has on the target node. [66] defines such an attention convolution that would extend Eq. (2) to:

$$\boldsymbol{h}'_u = \sigma(\boldsymbol{W}\boldsymbol{h}_u + \sum_{v\in\mathcal{N}(u)} \alpha_{u,v}\boldsymbol{W}\boldsymbol{h}_v) \tag{3}$$

where W refers to a learnable weight matrix, $\alpha_{u,v}$ refers to the attention coefficient for node v in the neighborhood aggregation of node u indicating the importance of node u to node v. It is computed as:

$$\boldsymbol{\alpha}_{u,v} := \mathrm{softmax}_v\left(\sigma(\boldsymbol{a}^T[\boldsymbol{W}\boldsymbol{h}_u \parallel \boldsymbol{W}\boldsymbol{h}_v])\right) \tag{4}$$

with $\boldsymbol{a}$ and $\boldsymbol{W}$ being weight vector and matrix respectively. σ again refers to an activation function, [66] for example use ReLu. The $\parallel$ corresponds to the concatenation operation, so the transformed features of both nodes are concatenated before the attention mechanism $\boldsymbol{a}$ is applied. The coefficients are normalized using softmax to make them comparable across the neighbors of the target node.

Spectral graph convolution. Besides the aforementioned spatial GNN layers, GNNs can be formulated using spectral theory, which refers to the study of the properties of linear operators. Similar to signal processing, where a signal can be decomposed into sine and cosine functions by Fourier decomposition, a graph signal $\boldsymbol{x}$ (a scalar for each node) can be transformed into the spectral domain by the graph Fourier transform F and back with its inverse. Convolution in the spectral domain results in an element-wise multiplication, after which, the convolved signal is transformed back into the graph domain:

$$\boldsymbol{g} * \boldsymbol{x} = F^{-1}(F(\boldsymbol{g})F(\boldsymbol{x})) = \boldsymbol{U}(\boldsymbol{U}^T\boldsymbol{g}\boldsymbol{U}^T\boldsymbol{x}) \tag{5}$$

where $\boldsymbol{U}$ is the matrix of eigenvectors of the normalized graph Laplacian $\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}$ and is determined by the eigendecomposition $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. $\boldsymbol{U}^T\boldsymbol{g}$ is the filter in the spectral domain. Since the normalized graph Laplacian $\boldsymbol{L}$ is composed of the degree matrix $\boldsymbol{D}$ and adjacency matrix $\boldsymbol{A}$, intuitively the eigenvectors and eigenvalues indicate the main directions of information diffusion through the graph. A first-order approximation of the spectral graph convolution has been proposed by [67]:

$$\boldsymbol{H}' = \sigma(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}}HW) \tag{6}$$

where H corresponds to the node feature matrix and W is a learnable weight matrix.

While spatial and spectral formulations of GNNs are equivalent, spatial GNNs are more commonly used in practice due to the high computational cost of spectral GNNs from eigendecomposition. However, they are more common in physical systems.

Graph capsule networks. The idea of Graph Capsule Networks, proposed by [68], is to capture more informative local and global features through spectral convolutions. This is achieved using a capsule vector that contains sufficient discriminative features to enable proper reconstruction. These vectors are constructed using a capsule function that maps the node features to higher-order statistics depending on the given dimensionality of the capsule vector. In a simple form, this could mean that the resulting vector contains the mean and standard deviation of the node's neighborhood. The second key component of graph capsule networks is the aggregation function, which is based on the covariance of a graph and provides information such as norms or angles between node features.

Common architectures for power grid problems. The literature on GNN approaches to power grid problems, such as power flow, optimal power flow (OPF), and stability assessment, reveals a set of preferred architectures and design patterns that prioritize accuracy, robustness, and physical fidelity.

The most frequently used architectures include the above-mentioned Graph Attention Network (GAT), e.g. [69,70], and [71], which often yield the best performance across various grid sizes due to their ability to assign different attention weights to critical nodes. GraphSAGE, as described above, is valued for its scalability and robustness, making it a promising approach for inductive learning on large systems. This is done by e.g. [70–72]. The above-described spatial graph convolution serves as the fundamental baseline model. For dynamic stability assessment, specialized convolutions, such as ARMA filters, show superior performance. These are based on the auto-regressive moving average and designed to provide a more flexible frequency response [73].

Successful GNN design for power systems is defined by several key choices. In terms of graph representation, most models are bus-centric but often treat the graph as undirected to aid information diffusion. The input features are comprehensive, encompassing the adjacency matrix, nodal features such as injected power and voltage, and edge features including resistance and reactance. For model structure, an Encode-Process-Decode pattern is common. For large grids, models require deep message-passing (up to 48–60 layers) because the physical solution depends on the entire grid topology [74], although this must be balanced against the oversmoothing challenge, which is described in more detail in Sec. 2.2.3. Finally, there is a strong trend toward Physics-Informed Learning such as [26,70,75] described below in Section 2.2.2.

2.2.2. Graph neural network training

Since GNNs are differentiable functions, they can be trained just like ordinary neural networks using gradient descent, backpropagation, batches, or mini-batches. Commonly used loss functions include the negative log-likelihood of softmax functions for the node or graph-level classification. For link prediction, pairwise node embedding losses, such as cross-entropy or Bayesian personalized ranking loss, are common. There are several specific considerations to be taken into account when applying GNNs to power grid problems. The core distinction in training GNNs for power grids is the use of Physics-Informed Loss Functions to incorporate domain knowledge. These losses go beyond standard supervised error by actively minimizing the violation of physical constraints like Kirchhoff's Laws and AC-OPF equality/inequality conditions as done in most power grid-related GNN studies, e.g. [70,74,76]. This constraint-augmented approach forces the model to learn physically feasible and robust solutions rather than just fitting the training data. Typically, the inequalities or constraints are added to the supervised loss, similar to a regularization term; however, there are also approaches that train solely on the physics loss, as seen in [26]. The proposed Graph Neural Solver model is trained unsupervised by aiming to directly minimize the violation of Kirchhoff's Laws at every bus in the power grid. The architecture consists of a fixed number of GNN layers, each acting as an iterative correction step. These steps push the predicted bus voltage magnitude and angles closer to achieving power equilibrium, thereby satisfying the physical laws.

2.2.3. Challenges

Due to their specific functionality, GNNs suffer from oversmoothing and oversquashing. Oversmoothing refers to the phenomenon that node features become increasingly similar as the number of layers increases. This problem can be addressed by regularization or normalization. Oversquashing refers to the distortion of information from distant nodes and is difficult to handle [77]. In power grid problems, these challenges are primarily addressed through specific architectural choices and computational efficiency. Oversmoothing is addressed by utilizing GNN architectures that incorporate skip-connections or ARMA filters [78]. These connections ensure that node features do not become overly homogenized [79], allowing for the deployment of deep models, which are essential because solutions to physical problems depend on the entire grid topology [74]. While this addresses the depth needed for physical fidelity, it is worth noting that literature on certain tasks, such as Neural State Estimation, sometimes finds that shallower models perform better in specific zero-shot scenarios [71]. Regarding oversquashing, the current literature does not explicitly detail specific mechanisms designed to counteract the distortion of information from distant nodes in power grids.

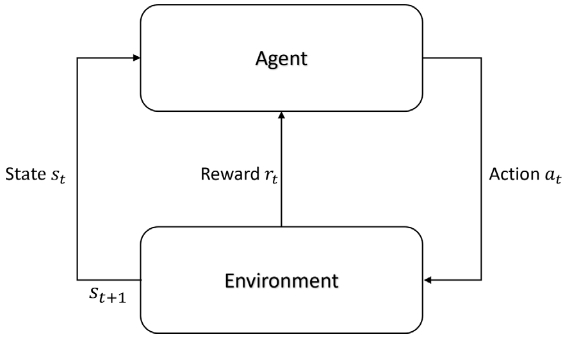


Fig. 3. The agent–environment interaction is a cyclical process where the agent selects actions based on the current state, leading to state transitions and rewards, guided by a policy π, hence generating a sequence of states, actions, and rewards.

2.2.4. Scalability

While general GNNs face significant challenges when processing very large graphs—often requiring complex sampling-based methods or mini-batches of subgraphs, which can lead to exponentially growing computational complexity [80] — this issue is less severe in power grid applications. Physical power systems have a limited and manageable size compared to massive virtual networks, such as social graphs. Even large-scale grids, typically with around 10,000 nodes, can be trained effectively without relying on complicated graph splitting or extensive sampling [74]. Compared to traditional power flow methods, such as Newton–Raphson, which have quadratic or cubic complexity, the scalability of GNNs stands out as a major strength [81]. Due to their linear computational complexity, GNNs achieves a speed-up that can be up to three to four orders of magnitude faster than conventional power flow solvers. For example, the assessment of dynamic stability using GNNs requires only 1 s, whereas dynamic simulations are up to seven orders of magnitude slower [73]. Furthermore, the use of techniques like GraphSAGE for neighborhood sampling enables efficient training on large grids [72].

2.3. Reinforcement learning

RL is a key branch of machine learning that focuses on training agents to make sequential decisions in dynamic environments to maximize cumulative rewards. It primarily uses the framework of Markov Decision Processs (MDPs) to model decision-making problems. An MDP is defined by the tuple $M = (S, A, R, T, \gamma, H)$, capturing essential elements of an agent's interaction with its environment. Fig. 3 shows the schematic procedure of a RL framework.

In this formalism, S is the state space of all possible states, and A is the action space of feasible actions. The reward function R maps states and actions to real-valued rewards, providing immediate feedback. The transition function T describes state transitions in response to actions. The discount factor γ balances the importance of future rewards against immediate ones. Finally, the horizon H defines the length of an episode, consisting of a sequence of states, actions, and rewards. In contrast, partially observed MDPs incorporates situations where the agent has incomplete knowledge about the current state. They provide a more realistic framework for many real-world problems where the agent must make decisions based on partial and uncertain information about the environment.

The primary objective of an agent is to learn a policy function $\pi(s)$ that prescribes actions to states, aiming to maximize the expected cumulative discounted sum of rewards over the time horizon H that defines the length of the episode. Here, π^* denotes the optimal policy.

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{H} \gamma^t r(s_t, a_t)\right] \tag{7}$$

RL algorithms are commonly classified into two main types: model-free and model-based methods. Model-free methods operate without requiring knowledge of the environment's transition functions; instead, they utilize the experiences gathered by the agent. These methods can be subdivided into two primary categories: policy-based and value-based methods, depending on their approach to solving an MDP. In contrast, model-based approaches focus on scenarios where the transition function is either known or can be learned. Examples of model-based methods include Monte Carlo Tree Search (MCTS) algorithms like AlphaZero [82], MuZero [83] and EfficientZero [84]. In the following, we introduce two important concepts for model-free RL, namely value-based and policy-based learning, as well as actor–critic approaches. Then, we will briefly present the widely used model-based technique MCTS.

2.3.1. Model-free reinforcement learning

Value-based learning. Value-based learning estimates the quality of state–action pairs to select optimal actions, i.e., actions with maximum value. For this purpose, the action-value function $Q^{\pi}(s,a)$ represents the expected sum of future discounted rewards, beginning from state s, executing action a, and subsequently adhering to a given policy π.

$$Q^{\pi}(s,a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid \pi, s_0 = s, a_0 = a\right] \tag{8}$$

The value function has a key recursive property linking the value of state s to the values of subsequent states s', which is fundamental to many value-based RL techniques. This is expressed by the Bellman equation

$$Q^{\pi}(s,a) = r(s,a) + \gamma \sum_{s'} p(s,a,s') \max_{a'} Q^{\pi}(s',a') \tag{9}$$

where $p(s,a,s')$ models the state transition dynamics. In value-based approaches, finding the optimal policy involves identifying the optimal value function $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$. Explicit solutions to the Bellman equation are possible only when the dynamics function is known [85]. Therefore, approximation methods are typically used. Here, we present two such methods.

Q-Learning aims to derive an optimal policy by directly updating values in a Q-table, a lookup table where each entry $Q(s,a)$ estimates the expected cumulative reward for taking action a in state s [86]. Q-Learning approximates the optimal action-value function Q^* through the following iterative updates.

$$\begin{aligned} &Q(s,a) \leftarrow Q(s,a) + \\ &\alpha\left[r(s,a) + \gamma \max_{a} Q(s',a) - Q(s,a)\right] \end{aligned} \tag{10}$$

Here, the agent explores the environment with a behavior policy, updating the Q-table based on the discrepancy between the actually observed and the previously expected reward. **Deep Q-Networks (DQNs)** uses neural networks to approximate action value functions in high-dimensional input spaces, minimizing the error between current and target Q-values [87]. It uses two networks, one to select actions and another to compute target Q-values. The target network is periodically updated with the weights of the primary network to stabilize the training. The agent stores its experience in a replay buffer from which it draws samples to train the neural network. Variants such as Double Deep Q-Network (DDQN) [88], dueling DQN [89], and Rainbow [90] further improve performance by addressing overestimation and efficiency issues.

Policy-based learning. Policy-based learning directly estimates policies without intermediate value functions. It optimizes parameterized policies $\pi_\theta(a \mid s, \theta)$ that specify the probability of action a given state s and parameters θ to maximize the expected cumulative rewards. A policy can be any mapping from state to action, for example, a neural network. Unlike value-based approaches, policy-based methods update parameters using gradient-based optimization and are suitable for continuous action spaces and stochastic policies [91].

The objective function $J(\theta)$ aims to maximize the true value function $v_{\pi_\theta}(s_0)$ from the initial state s_0. According to the policy gradient theorem [85], $J(\theta)$ is proportional to the sum of the action-value function multiplied by the gradient of the policy:

$$J(\theta) \propto \sum_{s} \mu(s) \sum_{a} q_{\pi}(s,a) \nabla \pi_\theta(a \mid s, \theta) \tag{11}$$

Here, $\mu(s)$ is the distribution under π, $q_\pi(s,a)$ is the action-value function, and $\nabla \pi_\theta(a \mid s, \theta)$ is the gradient of π with respect to θ. The update of the policy parameters proceeds in the direction of the gradient of the objective function to be maximized: $\Delta\theta = \alpha \nabla_\theta J(\theta)$, where α is the learning rate.

Actor–critic methods. These algorithms combine the strengths of value-based and policy gradient-based learning. The actor learns policies to maximize rewards, while the critic evaluates these policies by estimating the value function. This framework addresses the limitations of both approaches and is fundamental to various RL algorithms, including Asynchronous Advantage Actor-Critic (A3C) [92], Deep Deterministic Policy Gradient (DDPG) [93], Proximal Policy Optimization (PPO) [94], and Soft Actor-Critic (SAC) [95].

A3C updates policy and value networks asynchronously through multiple agents, using an advantage function to reinforce better-than-average actions. Further, it applies entropy regularization to enhance exploration, i.e., try new actions rather than exploiting knowledge already gained. Similarly, DDPG, tailored for continuous action spaces, simultaneously learns a state–action value function (critic) and a policy (actor), employing experience replay and target networks. PPO uses a clipped surrogate objective for smooth policy updates that balance exploration and exploitation, making it a popular choice in RL research. The clipping mechanism limits the policy update to a constrained range, preventing large, potentially destabilizing updates and improving training stability. Finally, SAC combines actor–critic methods with entropy regularization, training a policy network and two Q-value networks concurrently to encourage diverse action exploration and reduce overestimation.

While established algorithms such as PPO have been used extensively, recent innovations are often overlooked, particularly in power grid control. For instance, Bigger, Better, Faster (BBF) [96] is an advanced method that trains large neural networks in a sample-efficient manner. It employs a ResNet architecture with widened layers, a high replay ratio [97] with periodic network resets, and adaptive strategies like dynamic update horizon and discount factor schedules. BBF discards NoisyNets [98] in favor of weight decay for regularization, and outperforms state-of-the-art agents in both computational efficiency and performance, enhancing DRL for constrained environments.

2.3.2. Model-based reinforcement learning

Monte Carlo tree search. Both value-based and policy-based approaches in RL operate as model-free methods, meaning they do not utilize the environment model to plan ahead by simulating future steps. This is where MCTS [99] comes into play; it is a heuristic search that combines the accuracy of tree search with the power of random sampling to explore large state spaces efficiently. The algorithm builds a search tree incrementally, where nodes represent states and edges represent actions.

The process begins with selection, where the algorithm chooses the most promising child nodes from the root until it reaches a leaf node. If the leaf node is not terminal, the expansion phase adds one or more child nodes. Next, the algorithm runs a simulation from these new nodes to a terminal state, typically using random actions. Finally, in backpropagation, the simulation results are used to update the values of the nodes in the path from the leaf to the root, propagating the success or failure of the simulation.

MCTS effectively balances exploring new actions and exploiting known high-reward actions using the Upper Confidence Bound for Trees (UCT) formula to select nodes. This balance has made MCTS a powerful tool, with notable implementations such as AlphaZero [82], MuZero [83], and EfficientZero [84] achieving superhuman performance in complex games.

Designed to master complex games such as chess, shogi, and go, AlphaZero uses deep neural networks combined with MCTS and relies on predefined game rules. It learns by playing and using RL [82]. MuZero extends this approach by generalizing to environments with unknown rules, integrating RL, MCTS, and learned models to predict the environment dynamics [83]. EfficientZero builds on MuZero and achieves superhuman performance on the 100k Atari benchmark, significantly outperforming previous state-of-the-art results [84]. It introduces innovations such as self-supervised consistency losses for accurate next-state prediction and end-to-end value prefix prediction to deal with state aliasing issues. These enhancements improve exploration and action search capabilities, making EfficientZero highly effective in data-limited scenarios.

3. Graph reinforcement learning for transmission grids

To ensure safe and reliable transmission, human experts manually manage power grids. However, the rise in renewable energy and demand necessitates automated, data-driven optimization [1], shifting grid operation to adapt to generated power rather than predicted demand [100].

Many transmission grid control methods focus on generation or loads, like re-dispatch [101–103] or load shedding [104]. Topology actions, such as substation reconfiguration through busbar splits [105], offer a cost-effective alternative and enable an efficient power flow rerouting. In this way, curtailment of renewable energy can be avoided to some extent, and more power can be transmitted using the same infrastructure.

Solving this topology control problem is a significant challenge. It is an inherently non-linear, non-convex, and large-scale combinatorial problem. Classical methods, such as linear programming, quadratic programming, and non-linear programming, often rely on convex relaxation, linearization, or assumptions of continuity in the objective function. Yet they face critical limitations that render them unsuitable for modern transmission grid operations. First, classical optimization methods struggle with the non-convexity and non-linearity of AC power flow equations, which are essential for accurate grid modeling in close-to-real-time operations [106]. Secondly, mathematical solvers like IPOPT or CPLEX may fail to converge within the tight time windows required for real-time operations, such as during cascading failures or sudden renewable generation drops [107,108]. For many critical grid operations, such as topology control, these traditional solvers are computationally intractable [109].

The core advantage of GRL over these high-performance mathematical solvers (e.g., MISOCP or MIP) is not necessarily in finding a provably optimal solution, but in its ability to find high-quality, feasible solutions within a near-real-time decision window. For practitioners, this level of performance is often more than sufficient, as timely, robust decisions typically outweigh the marginal gains of exact optimality in transmission grid operation. A GNN-accelerated approach can deliver feasible solutions orders of magnitude faster than mathematical solvers [109]. It further allows for adapting to uncertainty in real-time in a way that is impossible for solvers that must recompute from scratch for every new scenario [1,109]. This massive gain in speed, scalability, and adaptability is the primary driver for GRL.

DRL-based grid operation approaches are validated through simulations, often using the Grid2op [110] environment from the Learning to Run a Power Network (L2RPN) challenges [12]. Grid2Op is crucial for developing methods that tackle grid congestion and enhance reliability; however, it is an abstraction of reality and may potentially limit its real-world applicability. Within this context, GRL has emerged as a promising paradigm that leverages the structural properties of power grids. The following section reviews existing GRL approaches for transmission grid control, highlighting their algorithmic design, graph representations, and evaluation setups. Furthermore, we discuss potential pitfalls that arise from overreliance on Grid2op and outline mitigation strategies at the end of the chapter.

We identified eleven GRL approaches for managing transmission grids in real-time. Due to a robust pre-dispatch schedule also given in real-world grid control, actions are necessary only in critical states. In stable conditions, agents usually do not act and only intervene when the line loading exceeds a threshold. This procedure is common across many machine learning approaches [19,21,111–117]. Table 2 lists the RL method, action type, GNN architecture, grid size, and overall focus of the analyzed approaches, and Fig. 5 illustrates the logical flow in GRL for grid control. A clear research trend is the gradual shift from monolithic to hierarchical and multi-agent designs, as well as the use of attention-based GNNs to improve interpretability and generalization.

3.1. RL framework

All GRL approaches reviewed in this chapter formulate the transmission grid control task as a Markov Decision Process (MDP), as introduced in Section 2.3. The following sections detail how these approaches specifically define the core components of this framework: the states (S), actions (A), and rewards (R).

Rewards. The overall goal of all approaches analyzed here is to manage line flows while mitigating congestion and minimizing overall costs. The reward functions, which guide the GRL agents, can be categorized into three objectives, as detailed in Table 3.

Firstly, Grid Stability and Survival: Several approaches prioritize keeping the grid operational under high congestion. For instance, [112,118,123] explicitly penalize line overflows or heavy loading ($\rho > 0.9$) to prevent cascading failures. Similarly, [120,122] design custom survival rewards to encourage agents to maintain grid stability for as long as possible during extreme events. Specifically, [122] comprise three components: a logarithmically scaled survival reward, an overload penalty proportional to the number of over-threshold lines, and an action term that favors conservative do-nothing policies unless line loadings exceed a critical threshold. This formulation encourages stability through minimal interventions. [120] augment their survival reward with potential-based reward shaping [126] to improve credit assignment across distributed agents. **Secondly, Grid Efficiency:** A second cluster of works focuses on maximizing the composite L2RPN Score. Since this score heavily penalizes blackouts, these approaches inherently prioritize survival as a prerequisite for optimizing efficiency. Approaches such as [113,114,119] reward the ratio of generated to served electricity, incentivizing the agent to reduce transmission losses while ensuring the grid remains operational. [121] also targets efficiency by minimizing power loss through topological changes. **Thirdly, Operational Constraints:** Finally, approaches dealing with continuous control variables, such as [124,125], utilize complex reward functions that combine operational costs (e.g., generator dispatch) with soft penalties for physical constraint violations, such as voltage limits or power flow equations.

Overall, most GRL agents use composite reward formulations balancing congestion penalties with efficiency and action costs. However, no standardized reward definition exists, making cross-comparison difficult.

Table 2
Overview of GRL approaches proposed for transmission grids. We categorize approaches by the used RL algorithm and GNNs, specifying action types and emphasizing topological actions (Topo) alongside others. Grid sizes indicate the number of buses evaluated. The focus column describes distinctive aspects of each approach.

env	Approach	Control algorithm	Action	GNN	Grid size	Focus
grid2op	Xu et al. 2020 [118]	Double-Q-Network	Topo	GCN	14	Simulation-constrained RL
	Taha et al. 2022 [112]	MCTS for action selection	Topo	GCN with residual connections	118 (2020)	GCNs for power flow estimation
	Sar et al. 2023 [114]	Multi-Agent SACD& PPO	Topo	GCN	5	Hierarchical RL
	Yoon et al. 2021 [113]	SMAAC	Topo	Transformer GNN	5, 14, 118 (2020)	Afterstate representation, goal topology actions
	Qui et al. 2022 [119]	SMAAC	Topo	GAT	5,14, 118 (2020 subset of 36 substations)	Afterstate representation, attention mechanism
	Fabrizio et al. 2025 [120]	Distributed Dueling DQN + DQfD	Topo	Shared GAT (line graph)	14	Distributed RL, pre-training, potential-based reward shaping
	Anguiano-Batanero et al. 2025 [121]	PPO	Topo	Transformer GNN	5	Action Masking, multiple graph representations
	Peter et al. 2025[122]	PPO	Topo	GCN	14	Dual-policy, N-k contingency
	Xu et. al 2022 [123]	Double-Dueling DQN	Topo & redispatch	GAT	118 (2020)	MCTS for action space reduction, sub action spaces for multiple agents
	Zhao et al . 2022 [124]	PPO	Redispatch, curtailment, battery storage	GraphSAGE	118	Representation of Power Grids, simulation of bus additions
other	Wu et al. 2023 [125]	Primal-Dual Constrained TD3	Active & reactive power control, battery operation	Cplx-STGCN	14, 30	Feasible control for SDOPF optimization

Actions. In critical states, agents can re-dispatch or alter the grid topology by changing bus configurations or line connectivity, often reconnecting disconnected lines. The actions considered by each approach are listed in Table 2. Since topology actions can often be done by the grid operators themselves and, thus, are cheaper than redispatch actions, they are favored if possible. However, the inherent combinatorial complexity of real-world busbar configurations makes it impractical to simulate every configuration in larger grids. Grid2Op curtails this by excluding electrically symmetrical actions, as well as any actions that would lead to a disconnected or islanded grid. While this reduces the action space, it remains large nonetheless. All works limit the enormous topological action space via masking, reduction, or hierarchical control. None has yet attempted to learn action embeddings or continuous latent actions.

States. Most approaches use the information provided by Grid2op, although typically not all features are used. For most approaches, the state includes grid topology, connected elements, and features such as bus-bar data, generation and loads, voltages, and line flows. Furthermore, the ratio (ρ) between the current flow and the thermal limit of each line is a critical feature. Typically, the states are modeled as graphs embedded using a GNN (see Section 3.3). The state information is consistent primarily across the GRL approaches, only [125] operate in a different environment and use only voltages as states.

3.2. Overall approach and RL algorithms

While the analyzed GRL frameworks share similar state, action, and reward structures, they diverge significantly in their core RL algorithm and architectural design. These approaches can be broadly categorized by how they structure the decision-making process and how they contend with the massive combinatorial action space and safety constraints inherent in grid control. We can identify several key trends: a move from monolithic agents to hierarchical or multi-agent architectures, the use of planning algorithms to guide exploration, and the integration of imitation learning to accelerate training.

Monolithic agents and action space management. A first group of approaches uses a single (monolithic) agent but employs specific techniques to make the decision-making process tractable and safe.

Early work by [118] employ value-based methods (see Section 2.3.1) with safety layers. Recognizing that naive exploration violates constraints, they introduced a “soft constraint” to replace invalid actions with a “do-nothing” action. This allows exploration without triggering constraint violations. During exploitation, they verify the top N actions with the highest Q-values of their Double-Q-learning agent through power flow simulation, creating a safety layer that prevents catastrophic failures.

Table 3
Overview of Reward Variables and Experimental Metrics (Transmission Grid).

Reference	Reward Variables (Optimization Objective)	Experimental Performance Metrics
Grid Stability & Survival (*Focus: Maximizing survival time*)		
Xu et al. 2020 [118]	Line overflow proximity, Survival bonus (+1/−1), Redispatch cost	Avg. Survival Steps, Successful episodes rate
Xu et al. 2022 [123]	Penalty for Overload ($\rho > 1$) and Heavy load ($\rho > 0.9$)	Avg. Survival Steps, Decision time
Fabrizio et al. 2025 [120]	Survival reward (+1), Potential-based reward shaping	Survival Time, Inference speed
Peter et al. 2025 [122]	Logarithmic survival ($\alpha \log t$), Overload penalty	Survival rate under **N-k contingencies**
Taha et al. 2022 [112]	Binary safety reward ($\rho < 1$) cumulative over MCTS horizon	**Failure rate reduction**
Grid Efficiency (L2RPN) (*Focus: Minimizing energy losses*)		
Yoon et al. 2021 [113]	Grid Efficiency ($\sum Load / \sum Gen$)	L2RPN Score, Survival rate
Sar et al. 2023 [114]	Grid Efficiency ($\sum Load / \sum Gen$)	Training score **convergence** (Multi-agent vs. Single-agent)
Qiu et al. 2022 [119]	Grid Efficiency ($\sum Load / \sum Gen$)	**L2RPN Score**, Management steps
Anguiano et al. 2025 [121]	Power Loss Reduction	**Steps to Complete (S2C)** (Convergence metric), Cost savings
Operational Constraints (*Focus: Voltage limits and dispatch costs*)		
Zhao et al. 2022 [124]	Operational costs + Penalties for voltage/flow violations	Reward convergence, **Topology generalization** (t-SNE)
Wu et al. 2023 [125]	Fuel cost (f_{cost}) + Battery loss (f_{ess}), Lagrangian constraints	**Optimality Gap**, **Feasibility Rate**

A more direct method for ensuring safety is dynamic action masking. [121] implement a purely model-free PPO agent (cf. Section 2.3.1) equipped with a "Topological Action Converter" (TAC). This module dynamically masks all infeasible or redundant topological actions at the policy output. This action-masking mechanism, implemented within PPO, ensures that only valid busbar configurations are selected during training and inference. This effectively reduces the exploration space without sacrificing flexibility.

Tackling the problem from a formal optimization perspective, [125] address the stochastic dynamic OPF problem with Renewable Energy Sources (RES) and decentralized energy systems. They use an actor–critic method where separate neural networks predict voltages. The critic networks are refined with temporal difference learning. They integrate constraints with Lagrangian multipliers, leveraging the duality principle to optimize both primal and dual variables through gradient-based updates. This approach is also used in similar constrained RL problems, such as in [127].

Planning-based and model-based strategies. Instead of learning a policy directly from rewards, a second group of methods adopts the model-based RL paradigm from Section 2.3.2. They use planning algorithms like Monte Carlo Tree Search (MCTS), a model-based heuristic search technique to explore the action space more intelligently.

Instead of Grid2op simulations, [112] took a model-based RL direction. They first train a GNN to predict the resulting grid state, i.e., line loading ρ for different topologies. They estimate the state evolution under different actions and select the trajectory maximizing cumulative rewards using MCTS. They iteratively build a tree, starting with the initial state as the root and actions as edges leading to subsequent nodes. This directly applies the MCTS framework described in Section 2.3.2, where the tree search effectively prunes the vast combinatorial action space by simulating future states rather than relying solely on a learned policy. Only nodes with sufficiently low loads are retained, and a given number of steps is simulated using a 'do-nothing' agent to determine the node's value. The GNN predicts line loading for each action. After simulating steps with a do-nothing agent, they select optimal actions by maximizing node values and the number of possible future actions.

[123] present another MCTS-based approach with a similar tree structure. Rather than a learned model, they use the simulator to build the tree. The leaf nodes represent overload states from feasible actions; the highest-value path determines the best action. Using double-dueling Q-networks, they train multiple sub-agents to select actions from different sub-action spaces obtained by dividing the reduced action space from MCTS into a fixed number of subspaces. The MCTS search effectively reduces the vast action space, creating a hybrid between a monolithic planner and a multi-agent execution system. A long-short-term strategy balances immediate and long-term benefits and manages sub-agents effectively. Each agent simulates n actions at overload states, selecting the best through efficient comparison. They also constrain topological actions and re-dispatch to stay within physically plausible limits.

Hierarchical and multi-agent decomposition. To manage the sheer combinatorial complexity of the grid's action space, a prominent trend is to decompose the monolithic MDP (Section 2.3) into smaller, manageable sub-problems. This is achieved through hierarchical and multi-agent architectures, which are purpose-built to divide the control task. This decomposition can be based on the **type of decision** or the **grid's physical components**.

Basing the decomposition on the type of topological action, [113, 119] propose a hierarchical policy. A high-level agent generates a desired goal topology rather than a specific action. A low-level policy is then responsible for executing the changes to reach that goal. This strategy avoids learning individual actions by focusing on suitable topologies for critical situations. Both utilize an afterstate representation to capture the grid topology after a topological action, which is advantageous when sequences of actions lead to identical topology changes. This directs their RL algorithm to understand the stochastic dynamics following each action. They use rule-based approaches like CAPA to prioritize substations with high-capacity usage and ensure timely responses. Their actor–critic algorithm enhances exploration and value function determination using the afterstate representation and goal topology predictions. [119] utilizes the whole architecture and present a similar approach with a different attention mechanism in their Graph Convolutional Network (GCN).

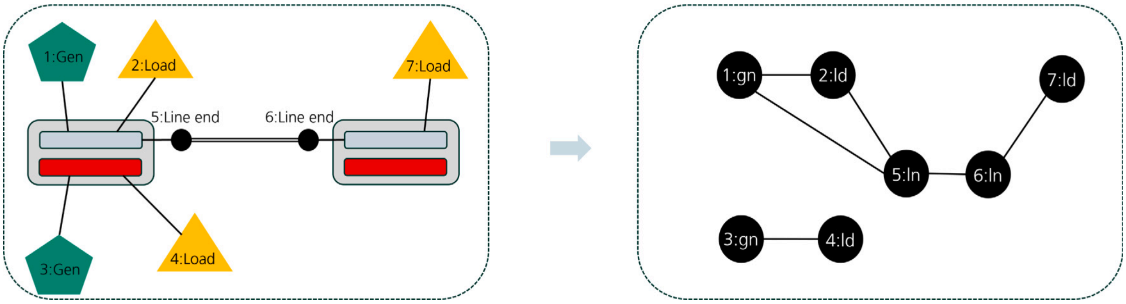


Fig. 4. Transformation from the physical power grid to the graph input for the GNN. Each grid component—loads, generators, and both ends of transmission lines—is represented as a node. Edges are defined by the grid's physical connectivity, linking nodes within substations and across substations via transmission lines.

On the other hand, [122] decompose the problem based on the operational scenario. They train two PPO policies: a "general" policy for stable conditions and a "critical" policy for emergency states, which is activated when line loading exceeds a threshold. This separation allows each policy to specialize in different operational regimes, improving robustness against extreme contingencies. Both policies share a GCN state encoder but have separate actor–critic networks, allowing each to specialize. To further stress-test the agent, an opponent model randomly disconnects multiple lines to emulate cyberattacks or cascading failures, effectively simulating $N-k$ events up to $k = 5$. The agent learns to mitigate these disruptions through topological switching actions while minimizing unnecessary interventions during normal operation. This dual-policy setup conceptually resembles hierarchical RL structures but differs in that it maintains fully independent actor–critic networks for each regime rather than nested decision levels. This architecture does not necessarily address the action space exploration problems if applied to a larger grid.

Decomposing the problem based on grid components, [114] present a three-level hierarchy. A top-level agent decides on the need for action and identifies critically loaded lines. It activates the mid-level agent in unsafe scenarios that prioritizes high-load substations using CAPA [113]. At the lowest level, substation-specific agents select bus assignments from a predefined action space. This approach is noteworthy for comparing parameter-shared versus independent critics in PPO and SAC.

Similarly, [120] introduce a fully distributed framework. A high-level manager coordinates a team of low-level agents, each responsible for controlling a single power line. This deep decomposition relies on a shared GNN to provide local context to each agent. Each low-level agent controls a single power line. Unlike prior substation-based approaches [113,114,128], their model decomposes both the action and observation spaces. Each agent receives a local view processed through a shared GNN, which encodes neighborhood information and mitigates partial observability. The high-level controller—also a Dueling DQN—decides which line-level agent acts at each step.

Hybrid approaches with imitation learning. Finally, several approaches accelerate the difficult learning process by leveraging expert data through imitation learning.

[120] pre-train their distributed agents using Deep Q-learning from Demonstrations (DQfD) on expert trajectories and further refine the policies through bootstrapped reward shaping. This allows the agents to learn a competent policy from expert trajectories before exploring on their own. This setup promises a scalable, modular coordination and enables GNN-based transfer learning across grid sizes.

To address the variability in grid topologies caused by, e.g., extreme weather or maintenance, [124] focus on re-dispatching, curtailment, and battery storage to ensure stability. Similar to [19,21], they use imitation learning to pre-train a PPO agent with a GNN based on GraphSAGE [129].

In summary, the field is clearly evolving from applying standard, monolithic RL algorithms toward designing sophisticated, structured architectures. These hierarchical, multi-agent, and planning-based systems are purpose-built to decompose the high-dimensional, combinatorial, and safety-critical nature of power grid control. Furthermore, pre-training agents using imitation learning has provided very promising results in pure imitation learning studies [116,117], refining policy initialization and improving sample efficiency in subsequent RL phases. Consequently, the combination of imitation learning and RL is increasingly explored to leverage the strengths of both paradigms—using expert demonstrations to guide early learning while allowing agents to further optimize their performance through exploration and interaction with the environment.

3.3. Graph embeddings

In all analyzed approaches, the power grid state is modeled as a graph, and a GNN is used to learn a latent representation that captures both the grid's features and its current topology. While this general procedure is universal, the approaches differ significantly in three key areas: the graph representation, which defines what is modeled as nodes and edges; the GNN architecture, which determines how information is processed within the graph; and the role of the GNN, which specifies the purpose of the learned embedding, such as for policy learning, value estimation, or world modeling.

Defining the graph representation. The choice of graph structure is critical, as it defines the information available to the GNN. Most approaches use the graph representation illustrated in Fig. 4. It is the default graph structure provided by Grid2Op [114,118,122], where nodes represent loads, generators, and the two ends of transmission lines. These nodes are connected according to their busbar assignments within substations, and the endpoints of each transmission line are also connected. This design naturally captures possible connections across busbars and facilitates the representation of substations that may be split. In some variants, transmission lines are instead modeled as edges equipped with line-specific features, while nodes represent only the electrical buses. Node features in both cases typically include all available state information. Fig. 4 illustrates how a topological configuration of two substations is transformed into a graph representation.

Moreover, [121] provide a systematic comparison of representations, evaluating three levels of abstraction: a flat representation, which uses a raw observation vector without any graph structure; a *SubstationGraph*, which models substations as nodes and transmission lines as edges; and an *ElementGraph*, a detailed representation similar to the Grid2Op default, where each component—load, generator, and line—is represented as a node. Their findings show that the most detailed representation, the *ElementGraph*, achieves the best performance and stability, underscoring the importance of incorporating fine-grained structural information into the GNN.

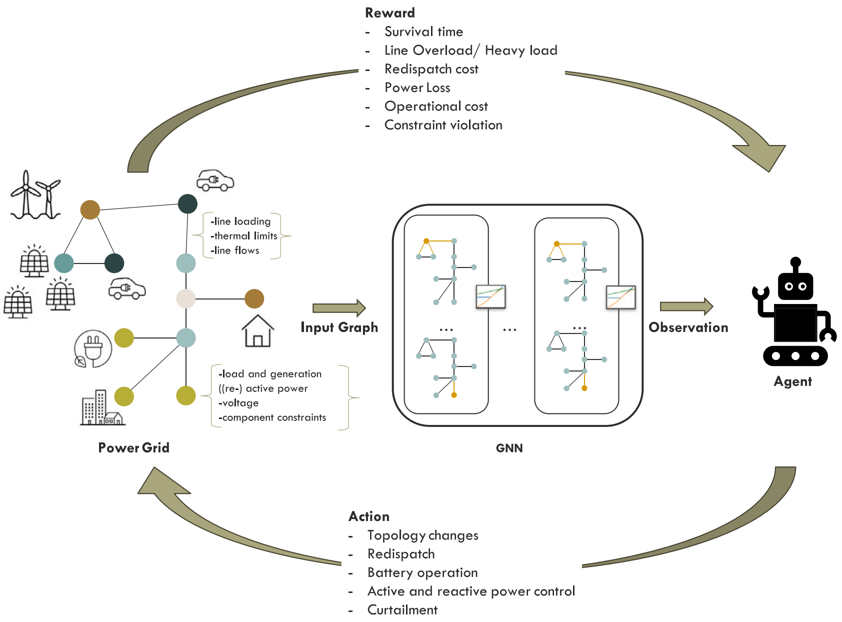


Fig. 5. Illustration of the Logical Flow of GRL for Transmission Grids: First, the power grid, including relevant information about lines and grid nodes, is modeled as a graph with node and edge features. This graph is then input into a GNN model, which learns a representation of the grid. This representation serves as an observation for the agent. Based on this observation, the agent selects an action from the action space. This may include simulations or other verification strategies to validate the action. The final action is executed in the environment (i.e., the simulated power grid). The agent receives a reward corresponding to the quality of the action and updates its weights. Depending on the RL algorithms employed, multiple (Graph) Neural Networks must be updated; for example, in actor–critic approaches.

A key challenge in grid modeling is the busbar information asymmetry [116], where the default graph structure (cf. Fig. 4) only passes messages between elements connected to the same busbar, while messages from elements on the other busbar within the same substation are not passed. This limits the model's expressiveness, as it makes elements on a different busbar (but at the same substation) indistinguishable from objects at entirely different substations. To address this, [120] propose a homogeneous line-graph. In this elegant representation, each power line is represented as a node, and edges connect lines that share a common substation. This design inherently addresses the asymmetry problem and facilitates a distributed agent framework where each agent controls one line.

Graph neural network architectures. Given a graph representation, the next choice is the GNN architecture used to extract features. A common baseline is the GCN, a spectral GNN defined in (6). Basic 2 or 3-layer GNNs are used by [114,118,122] as a robust encoder. [112] also use a GCN, but augment it with residual connections to create a deeper 3-layer model for predicting system dynamics.

Furthermore, a clear trend is the adoption of attention mechanisms to allow the model to weigh the importance of different neighbors. [123] directly compare GAT models against GCN and find that the GAT architecture leads to longer stable operation. Leveraging the attention mechanism defined in Eq. (3) (Section 2.2.1), this architecture allows the agent to dynamically weigh the importance of connected grid elements—crucial for identifying specific lines that are nearing thermal limits—rather than weighting all neighbors equally as in standard spatial convolutions. The model in [120] is also GAT-based, using attention to enrich each line agent's local observation with contextual information from neighboring lines. [113,121] employ transformer attention, which is an improved variant of attention convolutions (cf. Eq. (3)). It introduces nonlinearity to the computation of the attention coefficients, allowing for more diverse coefficients. Similar to the original transformer [130], it includes additional weight matrices that project the features from the node itself and neighboring features separately. This enhances the handling of edge features and enables deeper GNN architectures due to improved convergence properties. However, it also increases the computational complexity.

Some works employ architectures designed for specific challenges, such as GraphSAGE and spatio-temporal models.[124] use GraphSAGE [129], an inductive GNN that learns by sampling and aggregating features from a node's local neighborhood as described in . By utilizing the neighborhood sampling technique, the model avoids processing the entire graph simultaneously. This directly addresses the scalability and overfitting challenges discussed in Section 2.2.3, allowing the agent to handle larger grid variations without retraining. This is explicitly chosen to improve generalization to unseen grid topologies, as GraphSAGE has been applied in GNN-based imitation learning approaches for transmission grids [116]. [125] introduce the only spatio-temporal model, a Spatio-Temporal Graph Convolutional Network (STGCN) trained using a physics loss that represents the violation of the power flow equation as described in Section 2.2.2. It models the dynamics in power grid topologies by combining temporal convolutional layers with spatial convolutional layers to capture both types of dependencies. The temporal 1D-CNN layer extracts time-dependent features, and the spatial graph convolutions use the grid's admittance matrix as a graph shift operator to learn node embeddings. In both types of layers, the inputs are complex-valued, and the imaginary and real components are processed separately.

Overall, anywhere high robustness and interpretability are critical, attention-based GNNs are the prevailing choice. Benchmarks show GAT as superior across various scenarios, effectively handling disturbances. Conversely, GraphSAGE is the preferred architecture when the main goal is generalization to unseen topologies and scalability.

Role of the graph neural network in the RL framework. Finally, the learned embeddings are used in different ways within the RL loop, reflecting distinct algorithmic philosophies.

The most common approach is to use the GNN as a **shared policy/value encoder**. In this setup, the GNN functions as the primary state encoder, processing the raw graph-structured state into a latent embedding vector. This vector is then fed as input to both the actor (policy) and critic (value) networks. This design is implemented with various GNN architectures. For instance, [122] utilize the spectral GCN output embedding as the common input for their two separate PPO algorithm instances. Similarly, [123] use a two-layer GAT architecture for both the actor and critic in their PPO agent. [121] aggregates their GraphTransformer node embeddings into a global latent vector, which in turn serves as the input to their PPO actor and critic networks.

Sharing the GNN parameters across networks is a common strategy to promote efficient representation learning. [113] shares the learned embeddings across the actor and critic heads. The actor learns the parameters of a normal distribution for action sampling, while the critic transforms the embeddings into a scalar representing the value function of their afterstate representation. This sharing principle is also central in multi-agent settings. [114] apply shared GNN blocks in both their single and multi-agent frameworks. Likewise, the distributed framework by [120] relies on a shared GNN as a common feature extractor for all its line-level agents. This shared embedding enriches each agent's local observation with contextual information from its neighbors, enabling implicit coordination.

In contrast to these end-to-end trained encoders, some approaches decouple representation learning from the RL objective. [124] first train their GraphSAGE network in an **unsupervised manner** to learn structurally meaningful embeddings independent of the PPO loss. This pre-trained, "frozen" encoder is then used by the RL agent, a strategy chosen to improve generalization across different grid topologies.

A third, distinct role for the GNN is as a **world model for planning**. In a model-based RL approach, [112] use their GCN not as a policy encoder, but as a learned physics model. The GCN is trained to provide node level predictions of the line loads (ρ) for different actions, and this predictive model is then used by an MCTS planner to select the best action sequence. For training, they utilize grid topologies that are similar to a reference topology. If the GNN shows increased generalization error, they revert to the reference topology, which helps maintain grid stability, as supported by [21,111,117].

In summary, the field is moving beyond using standard GCN architectures on default grid graphs. A clear trend is emerging toward more sophisticated, problem-specific representations and more powerful attention-based architectures. Furthermore, the role of the GNN is diversifying, being used not just as a policy input, but also as a predictive world model or an unsupervised feature extractor to enhance generalization.

3.4. Experiments and evaluation

Most approaches train and evaluate their agents on grid2op because of its extensive power grid simulations with realistic data. [118, 120] focuses on an IEEE 14-bus system with 20 transmission lines, 6 generators, and 11 loads across 1004 scenarios over 4 weeks at 5-minute intervals. [112,123], and [124] use the larger IEEE 118 grid, while [114] uses the IEEE 5 grid for a hierarchical multi-agent proof of concept. [113] evaluate on all three of them and [125] applies their method to IEEE 14 and IEEE 30 bus systems with wind power data for power flow control using Battery Energy Storage Systems (BESS). In terms of evaluation, however, the presented approaches on grid control are not as comparable as one could hope, considering that most are based on the same framework and utilize similar data. Table 3 provides a comprehensive summary of the experimental variables used across these studies, clustering approaches by their primary reward objectives (Stability, Efficiency, or Operational Constraints) and detailing the key performance metrics reported.

[118] compare their simulation-constraint double dueling DQN agent with a basic double dueling DQN agent. The simulation-constraint agent outperforms the basic agent, maintaining grid stability for longer durations per episode. They also find that agents with GNN layers outperform those without. While results are promising, further testing on larger grids is needed to confirm the effectiveness of the approach.

The experiments in [121] are conducted exclusively on a very small 5-bus grid. Agents are compared only against the "Do Nothing" baseline. Results show faster convergence and higher stability when using the ElementGraph formalization with some agents achieving L2RPN scores of 100. However, the study does not include comparisons with existing GRL agents, nor does it test scalability on larger systems such as the IEEE 118 grid. Consequently, while the results demonstrate internal consistency, the practical scalability and competitiveness of the approach remain unverified, particularly given that the 5-bus can typically be solved by simple heuristics.

[112] trained their GNN using features representing power lines and a reduced injection horizon for speed. To gauge topology generalization, the GNN was tested on topologies differing by actions from the reference, showing that the RMSE increases logarithmically with action distance. Although no comparison with non-graph-based neural networks is provided, their MCTS agent significantly reduces failure rates from 15.1% to 1.5%, demonstrating the effectiveness of combining GNNs for prediction and MCTS for optimization, and paving the way for model-based RL with GNNs.

[113] validated their Semi-Markov Afterstate Actor-Critic (SMAAC) approach against baselines like DDQN and SAC, which underperformed on medium and large grids due to inefficient action exploration and potential failures. While they compared GNN-based and non-GNN methods, detailed validation of specific GNN architectures was lacking. The SMAAC/AS method, which incorporated goal topology without afterstate representation, performed poorly, highlighting the value of afterstate representation. Another baseline from the same L2RPN challenge struggled with the vast action space despite initial promise. SMAAC significantly outperformed other methods.

In [114], Soft Actor-Critic Discrete (SACD) and PPO are evaluated in both independent and dependent multi-agent settings. Independent agents, each with its own actor, critic, and replay buffer, face coordination and stability issues. Dependent versions utilize a centralized critic for improved coordination, thereby enhancing stability and performance. SACD performs well in a single-agent setting but is unstable in multi-agent scenarios, except for DSACD with tuned parameters. PPO converges effectively in both single- and multi-agent settings, with faster convergence in single-agent and less sensitivity to hyperparameters in multi-agent scenarios. The GNN used is not compared to feedforward networks or other GNN architectures.

[124] uses GraphSAGE networks on a modified IEEE 118-bus system, training them unsupervised and testing on various unseen grid topologies. They use 2D t-SNE to demonstrate the robust representation of the grid across different setups. Compared to a dense-based PPO algorithm, the GraphSAGE-based method performs well even with changing grid structures, while the dense-based approach struggles to adapt effectively. The evaluation focuses on training outcomes rather than power grid performance metrics such as agent survival time.

[123] evaluate their simulation-driven GRL method using the L2RPN Robustness Track challenge dataset. The method, which combines decisions from sub-agents, prevents overloads more effectively than a "do-nothing" approach and achieves fast decision-making with an average time of 35 ms per step. GAT models show better stability and economic benefits compared to GCN, although no comparison with feedforward networks is provided. Their Long-Short-Term action deployment strategy outperforms fully reward-guided and enumeration strategies by managing overloads with fewer actions, and the action threshold of 0.98 is validated as optimal. However, multiple action thresholds have been used across the literature, and no single value consistently dominates in terms of performance. The action threshold choice appears to be dependent on both the grid size and the utilized method [41].

The distributed GNN-based architecture proposed in [120] is evaluated on Grid2op's IEEE 14-bus grid. Their whole system—combining GNN-based local observations, DQfD pre-training, and potential-based reward shaping—achieves over 6000 average survival steps, far exceeding the “do-nothing” baseline. Ablation studies confirm the necessity of both the GNN and imitation-learning components. Furthermore, inference time is an order of magnitude faster than the simulation-based expert used for demonstration generation, indicating the model’s potential for scalability and real-time applicability. Nevertheless, despite its conceptual scalability, the evaluation remains confined to a small 14-bus grid. This limitation is particularly relevant given that the 14-bus grid can often be solved even by a simple greedy baseline. Consequently, the claimed transferability and scalability of the approach to larger grids remain untested, and further validation on larger IEEE systems or real-world networks would be required to substantiate these claims. Moreover, the study does not report the quantitative performance of the expert system used to generate demonstrations, nor does it include comparisons with other published agents. As a result, while the proposed framework demonstrates strong internal consistency, its relative performance against state-of-the-art baselines remains unclear.

Looking at the evaluation of [122], all simulations are conducted on a modified IEEE 14-bus grid with an opponent increasing system load by 25%. The study reports average survival times across $N-k$ contingency scenarios ($k = 1-5$) and compare it to a baseline without any agent. While the dual-policy agent significantly extends survival under these conditions, no comparisons with existing RL or heuristic agents are provided, and the scalability to larger grids (e.g., IEEE 118) remains untested. However, their dual policy agents achieve comparable performance for contingency scenarios that go beyond $k = 1$, making it a very promising approach. Noteworthy, however, is the limitation to just 100 simulation steps per episode, which constitutes only a fraction of the full operational horizon typically evaluated in Grid2Op. As a result, the reported survival times and robustness metrics capture only short-term resilience rather than sustained stability over realistic time spans.

[125] evaluated their GRL approach for managing BESS against baseline methods such as DQN and DDPG. They compare their spatio-temporal GCN (Cplx-STGCN) with feedforward, convolutional, and recurrent networks, highlighting their effectiveness. The study also tests hybrid OPF solvers, DeepOPF, and DC3, and compares them with RL methods like TD3, assessing metrics such as testing rewards and control over power generation and voltage magnitude. Their constrained GRL framework outperforms traditional optimization and existing RL techniques.

3.5. Discussion

The reviewed works demonstrate the significant potential of Graph Reinforcement Learning for transmission grid control. However, a critical analysis reveals several shared limitations and methodological gaps that must be addressed for these approaches to move from proof-of-concept to real-world applicability. These challenges can be grouped into limitations of the simulation environment, the need for standardized evaluation, and open research questions in model design.

The simulation-to-reality gap. Almost all approaches rely on the Grid2Op framework [110] for training and testing. While transmission system operators have designed it, it abstracts from real-world grid aspects. There is a need for larger grids with real injection data, realistic failure handling (e.g., N-1 security), and more accurate modeling of substation constraints and operational practices [131]. Consequently, rewards, actions, states, and graph representations are limited to the functionalities and data provided. As a result, the presented approaches are, to some extent, tailored to the specific problem setup used in Grid2op. This customization means the approaches inherit the same abstractions and limitations, questioning their applicability to real-world grids.

In particular, several **concrete pitfalls** arise from the reliance on simplified IEEE test cases and the abstractions within Grid2Op. Most studies evaluate their methods on comparatively small grids, which remain significantly below the size and complexity of real transmission networks. The commonly used double-busbar abstraction captures only part of the complexity of real-world substations. Furthermore, most benchmarks rely on synthetic injection data, as real-world measurements with realistic temporal dynamics are rarely available for research use. Another limitation is that most benchmark tasks neglect temporal N-1 security—evaluating safety only for single-step contingencies—whereas real operations require the grid to remain secure for extended horizons following any equipment failure. Grid2Op approximates N-1 assessments through its adversarial opponent mechanism, which induces random line disconnections to emulate unforeseen contingencies. However, this adversarial setup does not constitute a full N-1 evaluation, as it captures only isolated failures and lacks the temporal dimension and systematic coverage required for real-world reliability assessment. Additionally, the stochastic elements in Grid2Op—such as opponent events or random maintenance schedules can lead to non-reproducible comparisons when random seeds and horizons are not standardized. It is important to note, however, that these limitations do not only originate from the Grid2Op framework itself. Grid2Op is technically capable of handling large-scale and realistic grids through custom solvers such as LightSim2Grid or PowSyBl integrations—and even supports full N-1 security evaluations with minor configuration changes. The primary bottleneck is instead the availability of open, high-fidelity grid and injection data. Recent initiatives such as RTE’s open-source 7000 Nodes Dataset [132], which provides the complete French transmission network in node-breaker topology, mark a crucial step toward addressing this gap.

The need for standardized evaluation. To mitigate these limitations, **standardized evaluation protocols** are required. Future studies should use consistent evaluation horizons and report stochastic seeds. Moreover, benchmarks should specify the level of stochasticity, contingency modeling, and grid size to ensure comparability across publications. Some studies provide in-depth evaluations of RL algorithms and architectures and propose averaging results over multiple random seeds to increase reliability [19,117,133]. Based on their evaluation setups, using 20 to 30 seeds appears to establish a reliable basis for comparison.

While evaluation on real grid injection data is the ultimate goal, utilizing widely available synthetic data remains a viable option for benchmarking methods. Such evaluation environments should be of sufficient size; we follow the example set by [21] and recommend a year’s worth of data (i.e., 52 weeks/chronics). Regarding the evaluation, the corresponding metrics should align with the objectives that real-world grid operators use in their decision-making. These objectives are [134]:

- Minimizing the N-1 load flow
- Minimize the number of switching actions
- Minimize the number of open busbar couplers
- Minimize topology distance, i.e., minimize the amount of switching when stepping from one topology to another

Need for consolidation. Beyond evaluation, the primary methodological gap appears to be a need for consolidation. The research landscape is characterized by numerous, isolated innovations. For instance, sophisticated graph representations, such as heterogeneous graphs [116], have been proposed but not yet combined with advanced algorithmic structures, like afterstate representations [113]. Likewise, specialized GNN architectures (such as [125]) and planning techniques (such as MCTS [112]) have been developed in parallel but have not yet been integrated into unified frameworks. A significant opportunity exists in synthesizing these innovations—for example, combining a heterogeneous GNN with a hierarchical agent and an afterstate representation.

In addition to this need for consolidation, several other promising research avenues remain largely underexplored. These include the deep, systematic design of graph representations, the development of specialized GNN architectures, and the crucial work of mapping grid problems to the most appropriate RL algorithms—particularly hybrid approaches combining imitation learning with GRL.

Open frontiers in graph neural network design. Most studies use similar state features but differ in the information included, demonstrating the flexibility of GRL. However, despite the clear performance advantage of using graph representations, the studies lack a deep, systematic analysis of how grids are optimally represented. Recent work in imitation learning, for example, has demonstrated that heterogeneous graph models (which explicitly model different connection types, such as "same-busbar" vs. "cross-busbar"), can overcome information asymmetry and improve generalization [116]. More sophisticated graph modeling, such as hyper-heterogeneous multi-graphs [135], has not yet been incorporated into GRL approaches.

GNNs are pivotal for extracting features from power grids, enhancing convergence and generalizability across different configurations. While GATs, GraphSAGE, and transformers are used, their evaluation against alternatives is often shallow. Furthermore, most applied architectures are generic; specialized architectures, such as the Cplx-STGCN [125], are the exception. There is a clear opportunity to design GNN architectures specifically adapted to power grid physics and topologies, moving from graph-level predictions of single actions toward richer, node-level predictions of the entire grid state [116]. The novel and power grid-specific architecture described in would be a promising pathway.

Mapping problems to RL algorithms. The diversity in RL methods highlights a critical gap: the lack of a clear mapping between specific grid problems and the most suitable algorithmic family. This review suggests the following trade-offs.

Topological Control – This problem is defined by its vast, discrete, and combinatorial action space. Standard model-free methods, and in particular on-policy RL like PPO are prone to face a severe failures when directly applied, suffering from poor sample efficiency as they struggle to explore the long-term consequences of complex switching sequences.

Addressing Topological Complexity – The "afterstate" methods [113], hierarchical/multi-agent decompositions [114,120], and action masking [121] are all direct responses to this challenge, designed to prune the action space. MCTS-assisted selection [112,123] is another solution, trading a degree of sample efficiency (e.g., requiring a simulator in the loop) for more robust, safety-verified planning, though this can be computationally expensive at inference. However, model-based techniques, such as those in [103], are rarely used, and no full model-based GRL agent has yet been developed. Moreover, advanced model-free methods such as BBF [96] that could improve sample efficiency have not been applied.

A key underexplored strategy is the combination of imitation learning (IL) with GRL. IL is a proven method to accelerate training and improve sample efficiency by overcoming the "cold start" problem [19, 111,116]. The success of GNN-based IL methods using rich supervision signals, such as soft labels [117], suggests that a hybrid IL-GRL agent could offer a powerful balance of expert-guided safety and RL-driven optimization.

Redispatch and Storage Control – In contrast, problems like generator redispatch or battery storage [124,125], which often involve continuous or smaller discrete action spaces, are more amenable to constrained RL or standard model-free algorithms. Furthermore, integrating topological actions with generator redispatch is crucial for a more flexible and cost-effective control strategy; agents can leverage near-zero-cost substation reconfigurations to alleviate the majority of congestion, resorting to expensive generation adjustments only for remaining overloads.

To sum up, while the proposed approaches demonstrate significant potential, they remain largely at the proof-of-concept stage and are not yet scalable to real-world operations. Future deployment hinges not just on performance, but on ensuring trustworthiness and transparency [136]. This necessitates a paradigm shift toward human-AI decision support frameworks rather than full automation [137]. Nevertheless, these works successfully pave the technical foundation for next-generation grid operation.

4. Graph reinforcement learning for distribution grids

Power generation has increasingly shifted from the transmission system to the distribution side [138] due to the rise of distributed renewable energy sources such as photovoltaics. This shift causes voltage fluctuations [6] that can threaten grid stability, as system voltages must remain within operational limits. Voltage control addresses these issues by flattening voltage profiles and reducing network losses using devices such as voltage regulators, switchable capacitors, and controllable batteries [139], as well as topology control [140]. RL is particularly promising for handling multiple objectives in voltage control optimization problems. While DRL has shown promise in this area [141], the combination with GNNs is still emerging. We note that DRL methods serve as a proof of concept and are still far from practical deployment.

Classical methods for distribution grid tasks, like OPF-based optimization, mixed-integer formulations for discrete switching, and rule- or heuristic-based control, remain the standard toolkit in practice. However, they face critical limitations in modern distribution networks, which are increasingly dominated by distributed energy resources [142]. Similarly to transmission grid operation, accurate AC behavior combined with discrete actuators (tap steps, capacitor switching, feeder reconfiguration, shedding) leads to nonconvex mixed-integer programs that are difficult to solve reliably within real-time deadlines [143]. Linearizations and convex relaxations lose exactness under high R/X ratios, tight voltage bounds, or temporary meshing conditions common in distribution networks [144,145]. Combinatorial action spaces in reconfiguration and restoration scale poorly, which often necessitates heuristics that trade optimality for robustness [146]. Many formulations also assume full observability and stable parameters, making them sensitive to missing or noisy measurements, topology uncertainty, and forecast errors [145].

In contrast, GRL learns state-to-action policies over the grid graph and can naturally handle the given challenges, which is explored in the remainder of this chapter. An overview of the logical flow in GRL for distribution grids can be found in Fig. 6.

4.1. Voltage control

Voltage control is the key task in the distribution grid, managing reactive power set points to maintain grid stability. While active power is the power that runs devices, reactive power is required to provide the voltage levels that enable the delivery of real power. The cost function in these tasks typically includes system-wide indicators such as power losses and congestion [6]. It is essential that the voltages remain within the prescribed limits, as any violation would have detrimental effects on the system.

The AC power flow in a grid is modeled by highly non-linear equations, making the optimization problem non-convex. To simplify, it is often linearized using methods like DC-OPF formulations [147]. For exact models, numerical methods such as Newton–Raphson or Gauss–Seidel are used. Heuristics, such as particle swarm optimization, address the non-convexity of the problem formulation [6]. Deep learning approaches offer an effective alternative since they are suited for non-linear problems and overcome the limitations of traditional methods in dealing with the complexity and dynamics of smart grids.

In this work, we distinguish between two cases of voltage control: operation control and emergency mode. In the case of operation control, i.e. a stable grid state, voltage control is typically addressed through reactive power control. Emergency situations require more drastic measures such as load shedding, i.e., the cutting of loads to prevent the grid from blackout. Table 4 and 5 list the RL method, action type, GNN architecture, grid size, and overall focus of the analyzed approaches.

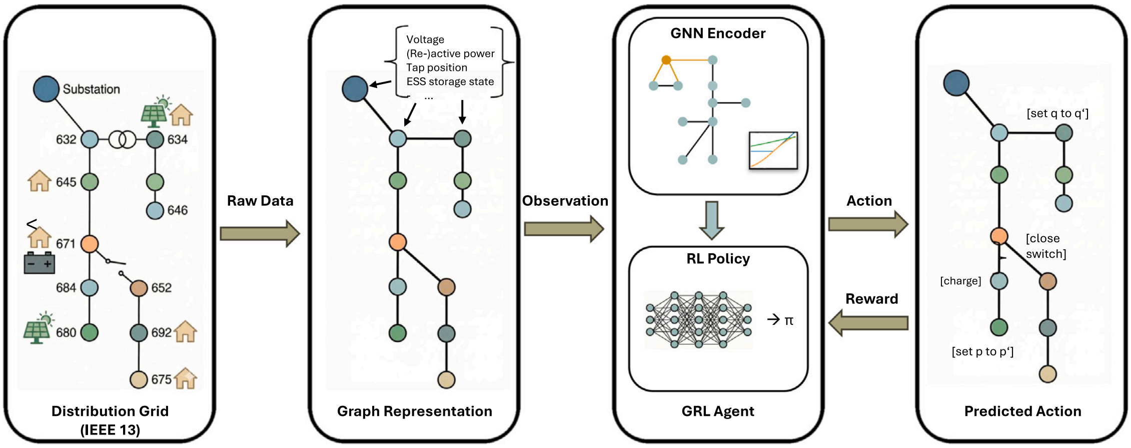


Fig. 6. Overview of the GRL approach for Distribution Grid Control. The environment represents the distribution grid, comprising various components such as distributed energy resources and energy storage systems connected via power lines, transformers, and switches. The left side shows a schematic of the IEEE 13-node test grid, which serves as the experimental use case in some of the presented approaches. This physical system is modeled as a graph, where nodes are characterized by specific features. This graph structure is passed to the GRL agent, which utilizes an GNN encoder to generate a comprehensive graph observation. An RL policy (typically Actor–Critic) then processes these embeddings to predict specific control actions, such as reactive power adjustments or tap position changes. These actions are executed within the grid environment, and a reward signal—calculated based on defined grid metrics, such as minimizing voltage deviation or power loss—is returned to the agent to update its weights.

4.1.1. Operational control

A commonly used approach to managing bus voltages is through the control of the reactive power which enables the generation of electromagnetic fields without delivering usable power to consumers. Modern inverter-based generation and digitized grids can maintain voltage within the desired range through reactive power control, aiming to minimize network loss, mitigate voltage oscillations, and reduce operational costs.

Rewards. Grid stability relies heavily on maintaining voltages within defined limits, so all approaches penalize voltage deviations from a reference value. Many methods combine penalties for voltage deviation with other terms such as power loss [127,148], equipment wear [151], and voltage barrier functions to constrain voltage ranges [148]. Table 6 provides a comprehensive summary of these reward formulations. Terms for PV curtailment, voltage oscillation [153,155], renewable integration, and generation costs [155] are also considered. [140,154] also base their reward on voltage deviation. But in addition, [154] introduce a surrogate model that estimates voltage and power loss based on the state and action and [140] penalize actions on already disconnected lines and define rewards based on the tree metric basic cut set to avoid loops or branch disconnections. In general, balancing multiple reward components is critical, as optimizing one can negatively impact others.

Actions. Voltage regulation mostly involves adjusting actuator set points. Approaches like [127,148,152–154,156] adjust reactive power outputs of PV inverters. While others also control the active power of Energy Storage Systems (ESS) [150,151,154,156], flexible loads [156] and static var compensation [154,156]. Adjusting the active power of generators of renewables [150,155] or diesel generators [150] alongside reactive power is another option. In contrast, [140] focuses on modifying grid topology by disconnecting and reconnecting lines.

States. The presented GRL methods predict actions based on states such as voltage measurements, load demand, and power generation. The status of actuators, e.g., ESS, tap changers [140,151,152], and electricity grid prices [150] are also considered. States are typically embedded using GNNs to encode the distribution grid's features and topology. These encoded representations are then used by the RL algorithm for decision-making.

Reinforcement learning algorithms. Common RL algorithms in these studies include Actor–Critic, PPO, and Q-Learning, each tailored to specific setups and objectives. Approaches like [151,153,155,156] use a GCN for grid embedding and train policies with DDPG or PPO. Multi-agent actor–critic setups [127,148] with one agent per zone manage zoned networks with centralized training including global observations and decentralized evaluation, i.e., based only on local observations. [127] integrate the GNN into the actor networks, while [148] use the GCN only in the critic to model agent interactions. To ensure that the physical equations of the physical system are satisfied, a primal–dual method similar to that of [125], as described in Section 3.2, is used.

A significant challenge in these multi-agent setups is the reliance on global information for centralized training. Addressing this, [149] propose a fully Decentralized Training and Decentralized Execution (DTDE) framework. Their algorithm, MASAC-HGRN, is based on Multi-Agent SAC and uses a stochastic policy with maximum entropy to improve exploration. Unlike centralized training methods, each agent in this paradigm learns using only its local observations and information communicated from its immediate neighbors, making it more robust to the partial observability inherent in real-world systems.

Similarly, [150] use multi-agent PPO with a GNN in both actor and critic for microgrid management, where each microgrid is controlled by an agent that manages its power schedule.

[152] propose a very different two-stage approach: day-ahead optimization of storage systems, tap changers and capacitor banks settings using Mixed Integer Second Order Cone Programming, followed by actor–critic learning for voltage regulation in a continuous action space with GCN-based grid embeddings. Similarly, [154] use a GCN to embed the grid, but they apply a subsequent fully connected deep autoencoder for feature dimension reduction in an actor–critic framework. Only [140] controls voltages by adjusting the grid topology, addressing the NP-hard complexity with a method to reduce the action space. First, they use deep Q-learning to predict a line to be disconnected. Second, they apply a branch exchange mechanism that considers the radial constraints (i.e., maintaining a tree structure without loops) when selecting a line to be reconnected.

Graph embeddings. The GNN architectures extract features from bus-centric graphs as described in where nodes represent buses including properties such as injections, voltage, active and reactive power and the status of controllable actuators). However, the GNNs vary

Table 4

Overview of GRL approaches for distribution grids under *operational control*. The *action* column lists devices or variables modified by predicted actions (*q* for reactive and *p* for active power, ESS for energy storage systems, SVC for static var compressors). The column *Grid size* lists the number of grid buses, and *Focus/Unique Feature* highlights key aspects or major differences to other approaches.

	Approach	RL	Action	GNN	Grid size	Focus/ Unique feature
Operational Control	Yan et al. 2023 [127]	MAAC	q (PV inverters)	Spectral GCN	141	Zoned grids, primal–dual approach
	Mu et al. 2023[148]	MAAC	q (PV inverters)	Spectral GCN	141, 39	Zoned grids
	Hu et al. 2024 [149]	MASAC	q (PV, SVC)	Hierarchical spatio-temporal GAT	33, 123	Fully decentralized training (DTDE), partial observability, robustness to communication failure
	Wang et al. 2023 [150]	Multi-agent PPO	ESS, p (generators)	Spatial GCN	69, 123	Real-time operation, multiple microgrids
	Lee et al. 2022 [151]	PPO	ESS, voltage regulators, capacitors	GAT	13, 34, 123, 8500	Graph augmentation, local readout
	Wu et al. 2023 [152]	AC	q (PV inverters)	Custom GSO (spectral)	33, 25	Two-stage hybrid (optimizer + RL), grid-specific filter
	Wu et al. 2022 [153]	PPO	q (PV inverters)	Custom GSO (spectral)	33, 119	Grid-specific filter in GNN
	Cao et al. 2023 [154]	AC	q (PV inverters), ESS, var compressors	GAT	33	Surrogate GNN (grid embedding + reward)
	Li et al. 2023 [155]	PPO	p (generators)	Spatio-temporal GAT	33, 69, 118	Consider temporal information
	Xing et al. 2023 [156]	DDPG	p (generators, PV, flexible loads), q (SVC)	GAT	33, 119	Computational efficiency, multiple objectives
	Xu et al. 2022 [140]	DQN	Topo	Spectral GCN	33, 69, 118	Action space reduction via GNN + branch exchange

across the presented approaches. Multiple approaches employ the popular GAT (cf. Eq. (3)) or some variant of it and train it using an RL algorithm. [151,155,156]. By calculating attention coefficients for each edge (as shown in Eq. (4)), these models can prioritize critical neighbors—such as a specific bus experiencing a voltage violation—while reducing the influence of less relevant distant nodes.

Some studies [150,151] extend their graph before applying a GNN. [151] introduce additional edges to allow faster information propagation through the graph. For decentralized microgrid voltage control, [150] construct a graph comprising only critical buses linked to generators, microgrids, or feeder endpoints that are connected via edges weighted by electrical distance.

Several approaches apply spectral graph convolutions (cf. Eq. (6)) for feature extraction [127,140,148,153]. [140] learn to predict the best line to be disconnected using three layers of graph convolutions, each followed by an MLP. The output is an edge embedding generated by aggregating the embeddings of the incident node. [127] employed a spectral GCN as actor-networks of each control agent. It acts like a low pass filter that suppresses noise in the input data and fills in missing values by aggregating the features of neighboring nodes. As defined in Section 2.2.1 (Eq. (6)), spectral convolutions operate by filtering graph signals in the frequency domain defined by the graph Laplacian. This property allows the GCN to efficiently attenuate high-frequency noise variations common in sensor data while preserving global structural signals. In contrast, [148] use the GNN only in the critic. It receives the action predicted by the actor, along with the observations, and combines them with the grid information from neighboring agents, i.e. neighbor grid zones.

To facilitate its decentralized, multi-agent framework, [149] design a novel architecture, the Hierarchical Graph Recurrent Network (HGRN), to explicitly handle partial observability and agent heterogeneity. In their model, each agent (representing a grid region) is a node in a meta-graph. The HGRN architecture, which is used in both the actor and critic, consists of three stages: (1) an encoder maps heterogeneous local observations to a common embedding space; (2) a Hierarchical GAT aggregates information from neighboring agents, allowing for communication; and (3) the resulting embedding is processed by a Gated Recurrent Unit, which maintains a memory to compensate for the partial observability of the environment.

[153] present a spatial temporal GNN including a graph shift operator, a spectral filter based on the relationship between voltage angle and magnitude of AC power flow equations similar to the architecture presented [125] (Section 3). Furthermore, the approach incorporates temporal information by aggregating the node embeddings of the last 10 time steps. [152] also introduce a graph shift operator but it is based on the grid topology and the correlation coefficient matrix obtained from the PV and historical load data. Both [152,153] use one convolutional layer and an MLP-based readout network comprising three layers. Another spatio-temporal approach is presented by [155]. Their attention mechanism learns the temporal dependency of a node's embedding at different time steps. The convolutions are applied sequentially to combine spatial and temporal information. To account for different periodic patterns, a fully connected layer combines three temporal resolutions (32 h, 16 days, 4 weeks) for the final prediction. The architecture is applied in both the actor and critic networks. In terms of training, [154] present an alternative to directly learning the GNN weights using an RL algorithm. They first train a surrogate GCN in a supervised manner on historical power flow data to predict node voltages like the approaches mentioned in . Then, the weights are copied to the representation networks of the actor–critic networks to perform feature extraction from the distribution network.

Table 5
Overview of GRL approaches for distribution grids under *emergency mode*. The *action* column lists the actions available to the agent. The column *Grid size* lists the number of grid buses and *Focus/Unique Feature* highlights key aspects or major differences to other approaches.

	Approach	RL	Action	GNN	Grid size	Focus/ Unique feature
Emergency Mode	Hossain et al. 2021 [157]	Double DQN	Binary load shedding	Spatial GCN	39	Consider temporal information
	Pei et al. 2023 [158]	Double-Dueling DQN	Two-level load shedding	GraphSAGE	39, 300	Adaptability to unseen topologies
	Zhao et al. 2022 [159]	Q-Learning	Reconnection of grid components	GCN	123, 8500	System restoration
	Jacob et al. 2024 [160]	PPO	load shedding, line switching	Graph capsule network	13, 34, 123	Achieves a close to optimal compliance with constraints

Experiments and evaluation. Most approaches evaluate their performance on IEEE grids ranging in size from 5 to 300 buses (see 4). The data for the injections can be randomly sampled or correspond to historical time series of power generation, mostly from photovoltaic. The authors evaluate the presented approaches using metrics such as voltage deviation, network energy loss, or voltage violation rates. Comparisons typically include traditional optimization methods, heuristics, and other deep RL approaches as benchmarks. Table 6 details the specific reward components used to guide the agents and the key performance metrics (e.g., robustness to noise, inference speed, violation rates) used to validate efficacy against baselines.

The advantage of GNNs over dense-based RL agents becomes evident in several studies. [151] tested their GNN-based PPO on PowerGym grids ranging from 13 to 8500 nodes, showing better performance and robustness, especially with noisy and missing data. In addition, the paper finds that voltage regulators affect the grid globally, while batteries and capacitors have local effects. To address this, the authors add edges between nodes with voltage regulators and use a local readout function for the controllable nodes, improving the robustness and performance of the GNN-based PPO approach. Similarly, [150] compared their graph PPO with a dense PPO method on IEEE grids from 33 to 123 buses, demonstrating near-optimal performance and better scalability than other multi-agent RL methods as well as optimization-based approaches.

Similar grid sizes are addressed by [155], whose GRL approach significantly outperforms optimization-based benchmarks with faster inference, higher rewards, lower voltage fluctuations, and greater renewable energy accommodation. Their spatio-temporal attention exhibits a faster convergence with the attention masks, indicating strong connections to high-power buses.

[127] found that their primal–dual GRL model minimized energy losses and voltage deviations and outperformed single-agent and multi-agent DDPG methods, especially with noisy data. Similarly, [148] reported robustness to line and bus deletions as well as stable voltages and fewer violations on 33-bus and 141-bus grids when compared to optimization methods and multi-agent DDPG. This is the same for [152], whose approach improved voltage profiles and reduced power losses on IEEE 33-node and 25-node systems compared to dense-based and CNN-based DDPG and conventional optimization.

[153] mitigated oscillations from cyber-attacks in 33 and 119-node systems, showing effective mitigation even with 50% of inverters compromised. They did not benchmark against other methods, hence, these results are difficult to interpret. Particularly, the utilization of GNNs remains to be validated. Similarly, [140] demonstrated that their method was faster than the heuristics and close to optimal. They confirmed the benefits of GCNs, branch exchange, and action separation in an ablation study.

[149] validate their MASAC-HGRN algorithm on the IEEE 33-bus and 123-bus systems, comparing it against a model-based SOCP benchmark as well as centralized (DQN) and centralized-training (MADDPG, MAAC) RL baselines. The decentralized (DTDE) approach outperformed all other RL methods in minimizing both power loss and voltage deviation. While all RL methods were orders of magnitude faster at inference than the SOCP solver, the key finding was in robustness. When communication links to agents were severed, the centralized DQN failed, and the CTDE-based MAAC degraded significantly. In contrast, the proposed DTDE approach showed the least performance drop, highlighting the resilience of a decentralized framework that does not depend on a central controller.

Lastly, [154] evaluated their physics-informed GAT-SAC on IEEE 33- and 119-node systems. Their approach outperforms other methods several control methods, including SAC variants and GCN-SAC in reducing voltage deviations and maintaining safe voltage levels, especially under noisy conditions. Ablation studies emphasized the importance of the GAT-based network and the added robustness from the deep autoencoder. Tests on the IEEE 119-node system confirmed the method's scalability and effectiveness in larger networks.

4.1.2. Distribution grid control in emergency mode

Another way to control grid voltages is load shedding, which involves deliberately disconnecting certain loads. This drastic measure is usually a last resort to avoid total blackouts. Conversely, in the event of a full or partial blackout occurs, system restoration is required to restart the grid. This involves coordinated steps to reconnect loads, restore generation capacity, and ensure the integrity of distribution networks.

Load shedding. The outlined approaches [157,158,160] present GRL agents for load shedding decision-making problems.

States and actions. The problem is framed as an MDP based on grid state observations such as power demand, generation, voltage measurements, and topology. [158] also use historical node voltages. [157,160] model decisions as binary (shed or not), while [158] consider shedding 5% or 10% of the load. All approaches consider only heavy load nodes which significantly reduces the action space. [160] additionally include line switching.

Reward. All approaches reward stable voltage levels and prevention of system collapse (cf. Table 6). For example, [157] give a large negative reward if the voltage has not returned to nominal within a specified time, and a positive reward if the voltages stay within predefined levels. The reward either minimizes load shedding [157,158] or maximizes power supply [160] and penalizes actions that violate system constraints. [160] introduce a high penalization when the load flow does not converge for the predicted grid configuration to avoid invalid grid states.

Reinforcement learning algorithm. [157,158] employ a DDQN with an ϵ-greedy strategy and experience replay with the latter also integrating a dueling architecture. In contrast, [160] adopt PPO using a hybrid policy network combining a fully connected network and a GCN.

Table 6
Overview of Reward Variables and Experimental Metrics (Distribution Grid).

Reference	Reward Variables (Optimization Objective)	Experimental Performance Metrics
Operational Control (*Focus: Voltage stability, loss minimization, and economic dispatch*)		
Yan et al. 2023 [127]	Voltage deviation penalty, Active power loss, Lagrangian safety constraints	Energy loss reduction, Voltage violation rate
Mu et al. 2023 [148]	Voltage deviation, Active power loss, Barrier function for limits	Robustness to **topology changes** (N-1 contingencies)
Hu et al. 2024 [149]	Power loss, Voltage deviation penalty	Performance maintenance under **communication failure**
Wang et al. 2023 [150]	Microgrid power schedule costs, Curtailment penalty	Scalability to larger grids, Near-optimal performance
Lee et al. 2022 [151]	Voltage deviation, Switching cost (Equipment wear)	Robustness to **noisy/missing sensor data**
Wu et al. 2023 [152]	Voltage deviation, Active power loss, Smoothness penalty	Voltage profile improvement, Loss reduction vs. DDPG
Wu et al. 2022 [153]	Voltage oscillation penalty (Frequency stability)	Mitigation rate of compromised inverters (Cyber-attack)
Cao et al. 2023 [154]	Voltage deviation, Surrogate model estimation error	Voltage deviation reduction, Training acceleration
Li et al. 2023 [155]	Voltage oscillation, Generation costs, Renewable usage	Inference speed, Reward convergence
Xing et al. 2023 [156]	Multi-objective: Voltage, Loss, and Comfort (Flexible load)	Computational efficiency, Pareto optimality
Xu et al. 2022 [140]	Voltage deviation, Tree metric (penalize loops/islands)	Inference speed vs Heuristics, Optimality gap
Emergency Mode (*Focus: Load shedding and system restoration*)		
Hossain et al. 2021 [157]	Binary Voltage Recovery (+10/−10), Stability penalty	**Convergence rate** (Episodes to stability), Recovery time
Pei et al. 2023 [158]	Minimize total load shedding magnitude (P_{shed})	Adaptability to **unseen fault locations**
Zhao et al. 2022 [159]	Restoration steps (minimize), Reconnection success	Inference speed vs CPLEX
Jacob et al. 2024 [160]	Maximize power supply, Penalize non-convergence	**Constraint compliance rate**, Optimality gap

Graph embedding. All studies model the grid as a graph with nodes representing buses, including substations, loads, and generators. Node features include grid state data such as voltage measurements and loads. The edges represent power lines and transformers. All approaches use a GCN for feature extraction but with different architectures. [157] use a simple GCN with three layers followed by a narrow fully connected layer processing the flattened embeddings. [160] employ a graph capsule network (cf. Section 2) to embed entire graphs. To reduce information loss in the convolution, higher-order statistics for each feature within the local neighborhood of a node are stored along with compressed node labels, similar to Weisfeiler-Lehman colorings [68]. Unlike standard message passing that aggregates features into a single scalar or vector (Eq. (1)), the capsule function described in Section 2.2.1 maps node features to vector representations that explicitly preserve feature statistics and orientation information, ensuring better reconstruction of local subgraphs. In the end, global grid information is integrated through a feedforward network. [158] apply GraphSAGE, which samples and aggregates the local neighborhood of nodes.

Experiments and evaluation. [157,158] use the IEEE 39-bus system, while [160] test on modified IEEE 13-bus and 34-bus systems. Each method trains on different topological configurations with random fault locations. All three approaches require less load shedding than using a fully connected network and outperform it in terms of convergence and average reward, especially on unseen topologies. It should be noted that none of the approaches is evaluated on completely new topologies as fault locations are randomly inserted in a known set of nodes. [157] test on 32 topologies in the IEEE 39-bus system, while [158] also apply their GNN method to a 300-bus system, effectively handling larger action spaces. [158,160] also outperform traditional optimization techniques in terms of speed and near-optimal performance. [158] show that GraphSAGE is more adaptable and efficient than classical GCN, but a comparison to a graph capsule network has not been made. The other two approaches do not compare to other GNN architectures.

System restoration. If earlier mitigation actions fail and a blackout occurs, rapid system restoration is crucial to reconnect loads and restart the grid promptly. [159] employ a multi-agent approach where a Q-network guides actions by using an encoder that takes observations such as generator capacities, and load conditions as input. A GCN extracts features from neighboring agents and passes the embeddings to the Q-networks. This method outperforms single-agent DQN and other multi-agent baselines using feedforward networks and the optimization method CPLEX in terms of accuracy and speed. Case studies validate this approach on IEEE 123 and 8500 node test systems [159].

4.2. Other use cases

Two other use cases for distribution grids include loss minimization and economic dispatch. Loss minimization focuses on optimizing the power grid state by adjusting the topology or the generator set points to reduce losses in branch elements. Economic dispatch minimizes operational generator costs.

Reinforcement learning approach. [161] minimize the Distribution Network Reconfiguration (DNR) loss and improve resilience by re-configuring the grid topology via sectionalizing and tie line switches. The states include the grid topology and other grid properties such as power demands and branch currents. The rewards penalize disconnections or radial structure changes and reward loss reduction and feasible network exploration.

Meanwhile, [162] focuses on economic dispatch in systems with high renewable energy, optimizing (renewable) generation and ESS power under dynamic conditions. The states include load demands, generator outputs and ESS state of charge. To optimize economic costs and improve system stability, the rewards are based on voltage violations and balancing economic operation costs and stability.

Graph embedding. The graph representations in the presented approaches denote substations and buses as nodes, while the edges represent power lines and transformers. Node features typically consist of properties such as load demand, generator outputs, ESS state, and time step information. Similar to [160,161] employ a Capsule-based Graph Convolutional Network (GCAPCN) (cf. Section 2) to capture local and global features that are used as input to the policy network. The value network of the PPO algorithm is a feedforward neural network. In contrast, [162] apply the SAC algorithm with the GCN layers in both the actor and critic to optimize dispatch policies off-policy.

Evaluation. [161] evaluate their GCAPS-RL method against a feedforward approach on modified IEEE 13 and 34-bus systems as well as two conventional baseline methods, Mixed-integer Second-order Conic Programming (MISOCP) and binary particle swarm optimization (BPSO). GCAPS-RL shows superior real-time decision-making and adherence to topological constraints and outperforms the feed-forward counterpart. [162] conduct case studies on a modified IEEE39 system with conventional generators, renewable sources, and an ESS. Their GRL approach outperforms the Optimal Solution with Perfect Information (OSPI), Model Predictive Control (MPC), and feed-forward SAC policies and shows strong convergence, effective policy performance, and superior scalability with cost reductions.

4.3. Discussion

Voltage and grid control in emergencies using DRL techniques involve several considerations. Reward functions typically address voltage deviations and may also integrate additional factors such as renewables or power loss. However, balancing these objectives poses a non-trivial challenge as mentioned in Section 3.5.

The control devices considered are mainly PV inverters, ESS, and generator power adjustments, with one notable approach that also includes topological actions. Multi-agent setups suit zonal or microgrid distribution networks, as agents can represent different zones and the conducted experiments show that GNNs improve the robustness in these scenarios. The strategies of handling global system knowledge varies between the approaches, using either global training with local evaluation or centralized critics. GNNs integrate information from different agents as shown in [148].

The graph representations used for distribution grid applications are consistent, however evidence from transmission grid literature shows that the choice of representation affects performance and stability. This underscores the need for more research on suitable representations for distribution grids. A start would be the evaluation of the provably meaningful representations used for transmission grid control. In contrast to the graph representations, GNN architectures vary significantly. Firstly, the role of the GNNs in the RL algorithms varies widely, with no consensus on their use in the actor, critic, or both. The node embeddings are mapped to action vectors through various readout methods, including neural networks, autoencoders, and 1D convolutions. Some approaches share weights between surrogate models and actors/critics. With respect to the architectures, Spectral GCNs, though less common in other domains, are more prevalent here probably due to their roots in signal processing, aligning closely with electrical engineering. Further they are crucial for noise mitigation and filtering, as they function as low-pass filters that suppress noise. They are a particularly strong choice when designing custom, problem-specific filters—such as a Custom Graph Shift Operator (GSO) used to map the relationship between the voltage angle and magnitude of the AC power flow equations—and often achieve high performance in specialized tasks.

As already mentioned in Section 3, GAT is frequently favored for when robustness is more important that scalability (general control and voltage regulation). For achieving scalability and generalization across diverse DG topologies (e.g., in emergency load shedding), GraphSAGE is the preferred choice. The fundamental Graph Convolutional Network (GCN) often acts as a reliable baseline encoder, particularly in multi-agent setups.

Temporal information is used in some approaches, but most approaches only consider static data. The benefit of using temporal data remains an open question, as it typically increases the complexity of the models. Although state-of-the-art architectures like ARMA Networks and TAGNet have demonstrated superior performance in solving power flow and stability tasks (as detailed in), they have not yet been integrated into GRL approaches for distribution grids, presenting a highly promising avenue for future research.

Similar to the conclusions in Section 3.5, GRL agents outperform DRL agents with fully connected neural networks in transferability and adaptability to topology changes, handling experiments with deleted grid elements or different structures without significant performance drops. The GNNs' ability to manage noisy or missing data is a considerable advantage, demonstrating robustness in experiments with generator failures, deleted lines, or nodes. This is particularly advantageous given the prevalence of faulty sensor data in real grid operations.

The experiments and evaluations of these approaches cover a wide range of considerations. Most studies conduct experiments on IEEE grids, with grid sizes ranging from small systems with as few as five buses to large networks with up to 8500 nodes. There is a considerable variation in grid size, affecting the approaches' scalability and generalizability. Notably, some methods show effective performance even without access to global information, highlighting the robustness of GRL strategies in this context—an essential advantage for real-world applications where data may be unavailable due to technical limitations or privacy concerns, and where local, faster decisions are critical for real-time use.

Evaluation metrics typically include voltage deviation, network energy loss, or voltage violation rates, reflecting the overarching goal of grid stability. Traditional optimization techniques and heuristics serve as benchmarks in many studies, as these are typically used in practice. The comparison highlights the comparative performance and efficiency gains of DRL-based approaches. Benchmarking against other DRL methods, including dense-based and CNN-based approaches, sheds light on the advantages of graph-based methods. The experiments highlight the stability and real-time performance of the proposed frameworks on comparatively small power grids. However, given the substantially larger scale of real-world grids, these methods are not yet practically applicable and can be regarded as preliminary proofs of concept.

It should be noted that none of the studies have been carried out on real data. Most approaches rely on simplifications, such as considering only binary actions for load shedding. This further limits the applicability and highlights the need for experimentation in more realistic scenarios. Nevertheless, the studies confirmed the potential of GRL for distribution system use cases.

5. Other applications

This chapter explores GRL approaches for related applications, including new energy markets, communication networks for power grids, and EV charging scheduling. We consider only approaches that take into account the underlying power grid structure and constraints. Table 7 provides an overview of the RL method, action type, GNN architecture, grid size, and overall focus of the approaches analyzed.

Table 7
Overview of other relevant GRL approaches.
In the left column, *Comm.* stands for *Communication Networks* and in the column *Action*, *CS* refers to charging station.

	Approach	RL	Action	GNN	Focus/ Unique feature
Energy Market	Rokhforoz et al. 2023 [163]	Multi-agent actor–critic	Pricing	GCN	Traditional electricity market
	Lee et al . 2022 [164]	DQN, DRQN, Bi-DRQG, PPO	Buy/sell/hold	GCN, Bi-LSTM	P2P power trading, maximizing integration of renewables
Comm.	Islam et al . 2023 [165]	Q-Learning	Routing, setting queue service rate	Spectral GCN	Reduce end-to-end latency and packet loss rate
EV charging	Xu et al. 2022 [166]	Double prioritized DQN(λ)	CS recommendation	GAT	Combine RL and Dijkstra for CS recomm. and rounting
	Xing et al. 2023 [167]	Rainbow DRL	CS recommendation, selection of road segments	GAT	Bi-Level RL for CS recomm. and routing

5.1. *Energy market*

GRL opens new possibilities in the energy market, especially in decentralized bidding or direct trading between entities. Traditionally, bidding strategies are centrally managed, requiring full information on all generation units. This is often computationally infeasible due to privacy concerns and results in large-scale problems. Distributed decision-making, using multi-agent RL and GNNs, has the potential to provide efficient and scalable solutions.

We limit our focus to the context of power grids and review two papers using GRL to optimize energy trading strategies considering grid topology. [163] focuses on the traditional market where generation units set their prices, and a market operator optimizes bids for the lowest overall cost. In contrast, [164] explores P2P trading, where individuals trade electricity directly, promoting renewable integration.

Approaches. [163] propose a two-level optimization as follows: first, each unit sets a bidding price; second, the market operator determines the market price such that the overall market cost is minimized and the load demand is satisfied. The aim of each generation unit is to maximize its profit. Accordingly, the rewards are calculated based on the determined market prices. The proposed approach is a multi-agent actor–critic algorithm, with one agent per generation unit. At each time step, the actor network of each generation unit selects an action, i.e. a bidding strategy and a GNN critic network which updates the bidding strategy. As input, it receives a graph consisting of buses as nodes including the respective demand as features. Experiments on the IEEE 30-bus and 39-bus systems show that the GNN approach outperforms the baseline using an MLP-based critic, particularly under varying generation capacities. When tested across different systems, the GNN also demonstrated better transfer capability.

In the approach by [164], energy is traded directly between prosumers without an intermediary market. The setup includes multiple nanogrids, an information network, and a business network for trading. The proposed RL algorithm learns trading strategies to minimize maximum load and maximize renewable integration, including power from discharging EVs. The agent's actor and critic are hybrid models combining a GCN with a Bi-LSTM to process time series data on prosumer consumption and production. The model inputs include cluster demand, renewable supply, system price, and demand response. The actions are either buy, sell, or hold. The reward is based on a rule-based baseline, and multi-objective optimization includes load shifting. The authors compare various RL methods, including DQN, Bi-LSTM, and PPO, using a nanogrid with real usage data. The PPO GCN-Bi-LSTM approach achieves the lowest electricity cost and performs better than other methods, significantly reducing average electricity costs with P2P trading.

Discussion. Both studies demonstrate that GNNs are promising for optimizing energy markets, particularly as decentralized approaches gain popularity for computational and privacy reasons. GNNs allow consideration of neighboring market participants' information without creating large-scale problems, unlike traditional deep learning methods that typically treat participants as independent samples. Experiments show that incorporating information from nearby nodes enhances overall market profit. GNNs improve both Actor–Critic and Q-Learning RL methods by capturing interdependencies missed by MLP-based methods, thereby learning more representative grid embeddings crucial for RL decision-making. These findings highlight the potential of GRL in energy markets, with more GRL-based approaches expected in the future.

5.2. *Power communication networks*

Apart from the power transmission itself, modern energy systems also transmit information for monitoring and control, requiring efficient routing in communication networks to avoid critical information loss. Unlike physical power transmission (cf. Section 3 and Section 4), these networks operate on the cyber layer.

The study in [165] addresses packet routing and presents a prioritization strategy including different qualities of service, which is not implemented in practice. They distinguish two types of packets: periodic packets with fixed schedules and emergency packets needing low latency to avoid loss of critical packets. The goal is to reduce overall end-to-end latency and packet loss using software-defined networks that adapt dynamically to grid conditions.

Two Q-Learning RL algorithms are trained: one for routing paths and another for queue service rates to minimize congestion. The first agent selects feasible paths, while the second predicts queue service rates at switches. Rewards are based on the difference between switch capacity and queue state in order to accelerate queue emptying.

A GNN predicts future grid states to inform the queue service rate agent, though the GNN is trained separately from the RL agent. The model, using spectral GCN with Chebyshev polynomials, is trained on IEEE grid traffic data represented as a graph consisting of switches (nodes) and communication links between them (edges). Experiments on the cyber layers of the IEEE-14 and 39-bus systems show the approach's effectiveness in managing grid communication through message exchanges between devices and control centers.

5.3. Electric vehicles applications

The rapid growth of electromobility is challenging the grid infrastructure, as it increases electricity demand and introduces variable loads. In this context, DRL has been studied for charging management [168–171], station recommendation [166,167], navigation [166,167,172], and pricing optimization [173]. These applications optimize the allocation of electricity, the pricing, and the routing of EVs. We concentrate on those GRL approaches that consider the underlying power grid and its constraints.

Reinforcement learning algorithms. The study in [166] tackles the increasing demands of fast charging stations using a multi-objective DRL method to dynamically allocate EVs to stations based on the interest of EV owners, charging stations (CS), traffic nodes (TN), and power grid nodes (PG). The agent recommends CSs and guides EVs using Dijkstra's algorithm, optimizing waiting times, service balance, traffic congestion, and grid voltage deviation. The recommendation of a CS is fast, but the full process is completed only once the EV finishes charging. The double-prioritized DQN(λ) method is introduced to address this delay and unpredictability. It integrates λ-return and experience replay with a small buffer to improve efficiency. During training, high-quality samples are prioritized using an attention mechanism, along with a strategy to regulate boundary actions.

Similarly, [167] present a Bi-Level GRL approach for charging and routing in Transportation Electrification Coupled Systems. Using a Rainbow-architecture DRL block, the high level agent recommends CSs, while the low level agent selects routes with DRL. This bi-level approach addresses credit assignment by selecting charging stations at the high level, focusing on the target CS. The low level agent handles the route selection to that station. The reward considers charging costs, battery loss, time allocation, energy consumption, travel time, and voltage limit penalties, with the high level interacting with charging stations and power grids and the low level with traffic nodes.

Graph embeddings. GNNs leverage the inherent graph structure of transportation systems and power grids. Therefore, [166] design a graph structure based on the physical properties. The CSs connect to TNs, and PGs are based on geographical and power supply relations. A unified expression method with type-specific transformation matrices projects features into a shared space, and GATs extract meaningful features. These learned representations are integrated with EV features for input to a DRL agent.

Similarly, [167] utilize GATs and introduce an instantaneous adjacency matrix for connections among EVs, CSs, TNs, and PGs, with smaller matrices representing different relationships. Node features store energy and information features.

Experiments and evaluation. The approach in [166] is validated using a power-transportation simulation platform with an IEEE 33-node distribution network and a 25-intersection traffic network. They optimize traffic, user experience, and grid stability, outperforming distance-based methods even with charging station queue limits. Training a user-oriented graph DQN(λ) agent shows long-term benefits and improved user experience. Combining GATs and DQN(λ) improves training and decision-making, though a comparison with MLP-DQN(λ) would be required to assess the impact of GATs.

On the other hand, [167] evaluate their Bi-Level GRL approach using real transportation-electrification data, achieving a 10.08% cost reduction and 16.45% time savings for owners. Compared to other DRL and traditional methods, their GRL approach lowers the average total cost by 8.96% (distance-based) and 4.73% (DRL), demonstrating its efficiency. They recommend learning GNNs weights directly with RL for better robustness and scalability.

Discussion. The authors of [166] demonstrate the effectiveness of GRL in dynamic resource allocation for charging stations, emphasizing real-time responsiveness and multi-stakeholder considerations. Their approach highlights the role of sequential decision-making in balancing objectives across transportation and power networks. In contrast, [167] focuses on efficient charging and routing coordination through GNNs. Their method aims to reduce charging costs and travel time, demonstrating the potential of GRL in real-world scenarios. While both employ GAT architectures, other GNN architectures and robustness to noisy data should be investigated more closely to confirm the applicability of GRL in real transportation and energy infrastructures. While current GRL approaches demonstrate strong performance in simulation, real-world deployment remains in an early stage, though pilot projects for EV charging management exist [174]. Key challenges include interfacing with heterogeneous transportation and power systems, processing noisy and incomplete data, and coordinating the joint optimization of mobility scheduling and grid constraints. This is logistically complex due to the need for integrated, cross-domain control systems.

6. Conclusion and outlook

Conclusion. This survey provides the first comprehensive analysis of Graph Reinforcement Learning for power grid control, revealing a rapidly growing field that, while promising, remains largely in a proof-of-concept phase. Our primary finding is that the core motivation for Graph Reinforcement Learning is not to achieve optimality, but rather to achieve speed, scalability, and adaptability to uncertainty. It excels at finding high-quality, feasible solutions for complex, non-linear, and combinatorial problems—like real-time topology control—where traditional mathematical solvers are computationally intractable. Our review confirms that Graph Neural Networks consistently outperform non-graph-based neural networks in decision-making for power grids, offering superior generalization to unseen topologies and robustness to noisy or missing data. Architecturally, we identified a clear trend away from monolithic agents toward hierarchical and multi-agent systems, with a preference for attention-based and sampling based Graph Neural Networks to enhance robustness and scalability. While we briefly touched upon zoned grids, the specific application of Graph Reinforcement Learning to microgrids represents a vital future direction. As power systems shift toward decentralized generation, Graph Reinforcement Learning is well-suited to optimize the combinatorial switching required for efficient microgrid islanding and self-healing.

Key challenges and future directions. Despite this promise, Graph Reinforcement Learning is not yet deployable. The most significant barrier is the gap between simulation and reality. The field's reliance on the Grid2Op framework, while crucial for development, means that most studies are validated on small IEEE grids that abstract real-world complexities, such as busbar configurations or full N-1 security. This also introduces reproducibility issues if stochastic seeds and horizons are not standardized. Future work should focus on large-scale, open datasets and address the challenges of handling noisy, real-world data. There is an urgent need for a benchmark framework for the distribution grid, specifically for voltage control, to ensure experiments are comparable across all scenarios.

To bridge this gap, a two-pronged progress is needed. First, the field requires standardized evaluation protocols—including fixed stochastic seeds, consistent evaluation horizons, and operator-aligned metrics (such as minimizing N-1 load flow)—to ensure reproducible and comparable results. Second, the field needs methodological consolidation. Our review identified numerous isolated innovations, including heterogeneous graph representations and specialized Graph Neural Network architectures, hierarchical Reinforcement Learning algorithm designs, and imitation learning for pre-training, which have been developed in parallel but not yet integrated. Future work should combine these and explore state-of-the-art Reinforcement Learning methods like *Bigger,* *Better, Faster* or full model-based Graph Reinforcement Learning, which remain largely unexplored.

Finally, agents must be safe, moving beyond simple reward penalties to formal Safe Reinforcement Learning techniques. They must be transparent and handle multi-objective optimization to present operators with a Pareto-front of solutions, not just one black-box answer. Incorporating domain knowledge, such as physics-informed losses and custom graph operators, will be crucial for building this trust.

Ultimately, even after successfully navigating the challenges of realism, standardization, and trustworthiness, deploying Graph Reinforcement Learning into the actual power grid requires strict adherence to operational and legal constraints. From a regulatory perspective, compliance with obligations such as the EU AI Act is essential and mandates extensive testing, documentation, rigorous auditability, and guaranteed human supervision. In addition, real grids experience continuous distribution shifts caused by fluctuating generation and consumption or grid expansion, demanding reliable shift detection and automated retraining. This requires tight integration with existing operator systems which typically suffer from data latency and synchronization issues. Consequently, control agents must be embedded in a robust MLOps pipeline that supports continuous monitoring, retraining, and dependable interaction with operational technologies.

Despite these challenges, Graph Reinforcement Learning stands out as a promising solution, providing the essential scalability and adaptability required to operate future renewable-dominated power grids.

CRediT authorship contribution statement

Mohamed Hassouna: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Clara Holzhüter:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Pawel Lytaev:** Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation. **Josephine Thomas:** Writing – review & editing, Supervision, Funding acquisition. **Bernhard Sick:** Supervision, Resources, Funding acquisition. **Christoph Scholz:** Writing – review & editing, Supervision, Resources, Funding acquisition.

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.

Acknowledgments

This work is supported by

(1) Graph Neural Networks for Grid Control (GNN4GC), founded by the Federal Ministry for Economic Affairs and Climate Action, Germany, under the funding code 03EI6117 A.
(2) AI4REALNET has received funding from European Union's Horizon Europe Research and Innovation program under the Grant Agreement No 101119527. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.
(3) Reinforcement Learning for Cognitive Energy Systems (RL4CES) from the Intelligent Embedded Systems of the University Kassel and Fraunhofer IEE funded by the Ministry of Education and Research Germany (BMBF) unter the funding code 01IS22063 A.
(4) GAIN project funded by the Ministry of Education and Research Germany (BMBF), under the funding code 01IS20047 A, according to the 'Policy for the funding of female junior researchers in Artificial Intelligence'.

Data availability

No data was used for the research described in the article.

References

[1] Marot A, Donnot B, Dulac-Arnold G, Kelly A, O'Sullivan A, Viebahn J, Awad M, Guyon I, Panciatici P, Romero C. Learning to run a power network challenge: a retrospective analysis. In: NeurIPS 2020 competition and demonstration track. PMLR; 2021, p. 112–32.

[2] Kelly A, O'Sullivan A, de Mars P, Marot A. Reinforcement learning for electricity network operation. 2020, arXiv:2003.07339, [Cs, Eess, Stat], URL http://arxiv.org/abs/2003.07339.

[3] Steinbrink C, Köhler C, Siemonsmeier M, van Ellen T. Lessons learned from CPES co-simulation with distributed, heterogeneous systems. Energy Informatics 2018;1(1):38. http://dx.doi.org/10.1186/s42162-018-0042-2.

[4] Bienstock D, Escobar M, Gentile C, Liberti L. Mathematical programming formulations for the alternating current optimal power flow problem. Ann Oper Res 2022;314(1):277–315. http://dx.doi.org/10.1007/s10479-021-04497-z.

[5] Capitanescu F, Martinez Ramos J, Panciatici P, Kirschen D, Marano Marcolini A, Platbrood L, Wehenkel L. State-of-the-art, challenges, and future trends in security constrained optimal power flow. Electr Power Syst Res 2011;81(8):1731–41. http://dx.doi.org/10.1016/j.epsr.2011.04.003, URL https://www.sciencedirect.com/science/article/pii/S0378779611000885.

[6] Srivastava P, Haider R, Nair VJ, Venkataramanan V, Annaswamy AM, Srivastava AK. Voltage regulation in distribution grids: A survey. Annu Rev Control 2023. Publisher: Elsevier.

[7] Capitanescu F. Critical review of recent advances and further developments needed in AC optimal power flow. Electr Power Syst Res 2016;136:57–68. http://dx.doi.org/10.1016/j.epsr.2016.02.008, URL https://www.sciencedirect.com/science/article/pii/S0378779616300141.

[8] Molzahn DK, Hiskens IA. A Survey of Relaxations and Approximations of the Power Flow Equations. 2019, http://dx.doi.org/10.1561/3100000012.

[9] Bienstock D, Munoz G. On linear relaxations of OPF problems. 2014, arXiv:1411.1120.

[10] Coffrin C, Hentenryck PV. A linear-programming approximation of AC power flows. 2013, arXiv:1206.3614.

[11] Liu Y, Wang Y, Zhang N, Lu D, Kang C. A data-driven approach to linearize power flow equations considering measurement noise. IEEE Trans Smart Grid 2020;11(3):2576–87. http://dx.doi.org/10.1109/TSG.2019.2957799.

[12] Marot A, Guyon IM, Donnot B, Dulac-Arnold G, Panciatici P, Awad M, O'Sullivan A, Kelly A, Hampel-Arias Z. L2RPN: Learning to run a power network in a sustainable world neurips2020 challenge design. 2020, URL https://api.semanticscholar.org/CorpusID:243830891.

[13] Marot A, Donnot B, Chaouache K, Kelly A, Huang Q, Hossain R-R, Cremer JL. Learning to run a power network with trust. Electr Power Syst Res 2022;212:108487. http://dx.doi.org/10.1016/j.epsr.2022.108487, URL https://www.sciencedirect.com/science/article/pii/S0378779622006137.

[14] Chen K, Bose S, Zhang Y. Physics-informed gradient estimation for accelerating deep learning-based AC-OPF. IEEE Trans Ind Inform. 2025.

[15] Donnot B, Guyon I, Schoenauer M, Panciatici P, Marot A. Introducing machine learning for power system operation support. 2017, arXiv:1709.09527.

[16] Viebahn J, Naglic M, Marot A, Donnot B, Tindemans S. Potential and challenges of AI-powered decision support for short-term system operations. 2022, CIGRE paris Session 2022 ; Conference date: 28-08-2022 Through 02-09-2022.

[17] Kaspar M, Muñoz Osorio JD, Bock J. Sim2real transfer for reinforcement learning without dynamics randomization. In: 2020 IEEE/RSJ international conference on intelligent robots and systems. 2020, p. 4383–8. http://dx.doi.org/10.1109/IROS45743.2020.9341260.

[18] Dulac-Arnold G, Levine N, Mankowitz DJ, Li J, Paduraru C, Gowal S, Hester T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Mach Learn 2021;110(9):2419–68. http://dx.doi.org/10.1007/s10994-021-05961-4.

[19] E.I. Innovation Lab HT. Neurips competition 2020: Learning to run a power network (L2RPN) - robustness track. 2020, (Accessed 22 January 2023) on Github, https://github.com/AsprinChina/L2RPN_NIPS_2020_a_PPO_Solution.

[20] Lehna M, Hoppmann B, Scholz C, Heinrich R. A reinforcement learning approach for the continuous electricity market of Germany: Trading from the perspective of a wind park operator. Energy AI 2022;8:100139.

[21] Lehna M, Holzhüter C, Tomforde S, Scholz C. HUGO – highlighting unseen grid options: Combining deep reinforcement learning with a heuristic target topology approach. Sustain Energy, Grids Netw 2024;39:101510. http://dx.doi.org/10.1016/j.segan.2024.101510, URL https://www.sciencedirect.com/science/article/pii/S235246772400239X.

[22] Prostejovsky AM, Brosinsky C, Heussen K, Westermann D, Kreusel J, Marinelli M. The future role of human operators in highly automated electric power systems. Electr Power Syst Res 2019;175:105883. http://dx.doi.org/10.1016/j.epsr.2019.105883, URL https://www.sciencedirect.com/science/article/pii/S0378779619302020.

[23] Marot A, Rozier A, Dussartre M, Crochepierre L, Donnot B. Towards an AI assistant for power grid operators. In: HHAI2022: augmenting human intellect. Frontiers in artificial intelligence and applications, IOS Press; 2022.

[24] Marot A, Kelly A, Naglic M, Barbesant V, Cremer J, Stefanov A, Viebahn J. Perspectives on future power system control centers for energy transition. J Mod Power Syst Clean Energy 2022;10(2):328–44. http://dx.doi.org/10.35833/MPCE.2021.000673.

[25] Liao W, Bak-Jensen B, Pillai JR, Wang Y, Wang Y. A review of graph neural networks and their applications in power systems. J Mod Power Syst Clean Energy 2021;1–16. http://dx.doi.org/10.35833/MPCE.2021.000058.

[26] Donon B, Clément R, Donnot B, Marot A, Guyon I, Schoenauer M. Neural networks for power flow: Graph neural solver. Electr Power Syst Res 2020;189:106547. http://dx.doi.org/10.1016/j.epsr.2020.106547, URL https://www.sciencedirect.com/science/article/pii/S0378779620303515.

[27] Ringsquandl M, Sellami H, Hildebrandt M, Beyer D, Henselmeyer S, Weber S, Joblin M. Power to the relational inductive bias: Graph neural networks in electrical power grids. In: Proceedings of the 30th ACM international conference on information & knowledge management. New York, NY, USA: Association for Computing Machinery; 2021, p. 1538–47. http://dx.doi.org/10.1145/3459637.3482464.

[28] Kuppannagari SR, Fu Y, Chueng CM, Prasanna VK. Spatio-temporal missing data imputation for smart power grids. In: Proceedings of the twelfth ACM international conference on future energy systems. E-energy ’21, New York, NY, USA: Association for Computing Machinery; 2021, p. 458–65. http://dx.doi.org/10.1145/3447555.3466586.

[29] Munikoti S, Agarwal D, Das L, Halappanavar M, Natarajan B. Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications. IEEE Trans Neural Netw Learn Syst 2023;1–21. http://dx.doi.org/10.1109/TNNLS.2023.3283523, Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[30] Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M. Graph neural networks: A review of methods and applications. AI Open 2020;1:57–81.

[31] Thomas J, Moallemy-Oureh A, Beddar-Wiesing S, Holzhüter C. Graph neural networks designed for different graph types: A survey. Trans Mach Learn Res 2022.

[32] Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY. A comprehensive survey on graph neural networks. IEEE Trans Neural Netw Learn Syst 2020;32(1):4–24.

[33] Skarding J, Gabrys B, Musial K. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. IEEE Access 2021;9:79143–68.

[34] Wu S, Sun F, Zhang W, Xie X, Cui B. Graph neural networks in recommender systems: a survey. ACM Comput Surv 2022;55(5):1–37.

[35] Li Y, Xue C, Zargari F, Li YR. From graph theory to graph neural networks (GNNs): The opportunities of GNNs in power electronics. IEEE Access 2023;11:145067–84.

[36] Arulkumaran K, Deisenroth MP, Brundage M, Bharath AA. Deep reinforcement learning: A brief survey. IEEE Signal Process Mag 2017;34(6):26–38.

[37] Garcıa J, Fernández F. A comprehensive survey on safe reinforcement learning. J Mach Learn Res 2015;16(1):1437–80.

[38] Zhu Z, Lin K, Jain AK, Zhou J. Transfer learning in deep reinforcement learning: A survey. IEEE Trans Pattern Anal Mach Intell 2023.

[39] Zhang Z, Zhang D, Qiu RC. Deep reinforcement learning for power system applications: An overview. CSEE J Power Energy Syst 2019;6(1):213–25.

[40] Vázquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. Appl Energy 2019;235:1072–89. http://dx.doi.org/10.1016/j.apenergy.2018.11.002, URL https://www.sciencedirect.com/science/article/pii/S0306261918317082.

[41] van der Sar E, Zocca A, Bhulai S. Optimizing power grid topologies with reinforcement learning: A survey of methods and challenges. 2025, arXiv:2504.08210.

[42] Fathinezhad F, Adibi P, Shoushtarian B, Chanussot J, Fathinezhad F, Adibi P, Shoushtarian B, Chanussot J. Graph neural networks and reinforcement learning: A survey. IntechOpen; 2023, http://dx.doi.org/10.5772/intechopen.111651, URL https://www.intechopen.com/online-first/87170.

[43] Nie M, Chen D, Wang D. Reinforcement learning on graphs: A survey. IEEE Trans Emerg Top Comput Intell 2023.

[44] Mazyavkina N, Sviridov S, Ivanov S, Burnaev E. Reinforcement learning for combinatorial optimization: A survey. Comput Oper Res 2021;134:105400. http://dx.doi.org/10.1016/j.cor.2021.105400, URL https://www.sciencedirect.com/science/article/pii/S0305054821001660.

[45] Mendonca MR, Ziviani A, Barreto AM. Graph-based skill acquisition for reinforcement learning. ACM Comput Surv 2019;52(1):1–26.

[46] Pateria S, Subagdja B, Tan A-h, Quek C. Hierarchical reinforcement learning: A comprehensive survey. ACM Comput Surv 2021;54(5):1–35.

[47] Murray W, Adonis M, Raji A. Voltage control in future electrical distribution networks. Renew Sustain Energy Rev 2021;146:111100, Publisher: Elsevier.

[48] Li F, Qiao W, Sun H, Wan H, Wang J, Xia Y, Xu Z, Zhang P. Smart transmission grid: Vision and framework. IEEE Trans Smart Grid 2010;1(2):168–77. http://dx.doi.org/10.1109/TSG.2010.2053726.

[49] Gui Y, Jiang S, Bai L, Xue Y, Wang H, Reidt J, Ojetola ST, Schoenwald DA. Review of challenges and research opportunities for control of transmission grids. IEEE Access 2024;12:94543–69. http://dx.doi.org/10.1109/ACCESS.2024.3425272.

[50] Linnemann C, Echternacht D, Breuer C, Moser A. Modeling optimal redispatch for the European transmission grid. In: 2011 IEEE trondheim powerTech. 2011, p. 1–8. http://dx.doi.org/10.1109/PTC.2011.6019442.

[51] Moreno Escobar JJ, Morales Matamoros O, Tejeida Padilla R, Lina Reyes I, Quintana Espinosa H. A comprehensive review on smart grids: Challenges and opportunities. Sensors 2021;21(21). http://dx.doi.org/10.3390/s21216978, URL https://www.mdpi.com/1424-8220/21/21/6978.

[52] Howlader AM, Sadoyama S, Roose LR, Chen Y. Active power control to mitigate voltage and frequency deviations for the smart grid using smart PV inverters. Appl Energy 2020;258:114000. http://dx.doi.org/10.1016/j.apenergy.2019.114000, URL https://www.sciencedirect.com/science/article/pii/S0306261919316873.

[53] Gonzalez Venegas F, Petit M, Perez Y. Active integration of electric vehicles into distribution grids: Barriers and frameworks for flexibility services. Renew Sustain Energy Rev 2021;145:111060. http://dx.doi.org/10.1016/j.rser.2021.111060, URL https://www.sciencedirect.com/science/article/pii/S1364032121003488.

[54] Stecca M, Elizondo LR, Soeiro TB, Bauer P, Palensky P. A comprehensive review of the integration of battery energy storage systems into distribution networks. IEEE Open J Ind Electron Soc 2020;1:46–65. http://dx.doi.org/10.1109/OJIES.2020.2981832.

[55] Babaeinejadsarookolaee S, Birchfield A, Christie RD, Coffrin C, DeMarco C, Diao R, Ferris M, Fliscounakis S, Greene S, Huang R, Josz C, Korab R, Lesieutre B, Maeght J, Mak TWK, Molzahn DK, Overbye TJ, Panciatici P, Park B, Snodgrass J, Tbaileh A, Hentenryck PV, Zimmerman R. The power grid library for benchmarking AC optimal power flow algorithms. 2021, arXiv:1908.02788.

[56] Meinecke S, Sarajlić D, Drauz SR, Klettke A, Lauven L-P, Rehtanz C, Moser A, Braun M. Simbench—A benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis. Energies 2020;13(12). http://dx.doi.org/10.3390/en13123290, URL https://www.mdpi.com/1996-1073/13/12/3290.

[57] Meinecke S, Thurner L, Braun M. Review of steady-state electric power distribution system datasets. Energies 2020;13(18). http://dx.doi.org/10.3390/en13184826, URL https://www.mdpi.com/1996-1073/13/18/4826.

[58] Strunz K, Abbasi E, Fletcher R, Hatziargyriou N, Iravani R, Joos G. TF C6.04.02 : TB 575 – benchmark systems for network integration of renewable and distributed energy resources. 2014.

[59] Zimmerman RD, Murillo-Sánchez CE, Thomas RJ. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans Power Syst 2011;26(1):12–9. http://dx.doi.org/10.1109/TPWRS.2010.2051168.

[60] Thurner L, Scheidler A, Schäfer F, Menke J-H, Dollichon J, Meier F, Meinecke S, Braun M. Pandapower—An open-source python tool for convenient modeling, analysis, and optimization of electric power systems. IEEE Trans Power Syst 2018;33(6):6510–21. http://dx.doi.org/10.1109/TPWRS.2018.2829021.

[61] DIgSILENT GmbH. Digsilent PowerFactory, version 2024. 2024.

[62] Bronstein MM, Bruna J, Cohen T, Veličković P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. 2021, arXiv preprint arXiv:2104.13478.

[63] Bronstein MM, Bruna J, Cohen T, Veličković P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. 2021, CoRR, arXiv:2104.13478.

[64] Morris C, Ritzert M, Fey M, Hamilton WL, Lenssen JE, Rattan G, Grohe M. Weisfeiler and leman go neural: Higher-order graph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, (01):2019, p. 4602–9.

[65] Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. In: Proceedings of the 31st international conference on neural information processing systems. Red Hook, NY, USA: Curran Associates Inc.; 2017, p. 1025–35.

[66] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: International conference on learning representations. 2018.

[67] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. 2016, arXiv preprint arXiv:1609.02907.

[68] Verma S, Zhang Z-L. Graph capsule convolutional neural networks. 2018, arXiv preprint arXiv:1805.08090.

[69] Varbella A, Amara K, Gjorgiev B, El-Assady M, Sansavini G. Powergraph: A power grid benchmark dataset for graph neural networks. Adv Neural Inf Process Syst 2024;37:110784–804.

[70] Ghamizi S, Bojchevski A, Ma A, Cao J. Safepowergraph: Safety-aware evaluation of graph neural networks for transmission power grids. 2024, CoRR, arXiv:2407.12421.

[71] Berezin A, Balduin S, Oberließen T, Peter S, Veith E. On zero-shot learning in neural state estimation of power distribution systems. 2024, arXiv preprint arXiv:2408.05787.

[72] Talebi S, Zhou K. Graph neural networks for efficient AC power flow prediction in power grids. 2025, arXiv preprint arXiv:2502.05702.

[73] Nauck C, Lindner M, Schürholt K, Hellmann F. Toward dynamic stability assessment of power grid topologies using graph neural networks. Chaos Interdiscip J Nonlinear Sci 2023;33(10).

[74] Piloto L, Liguori S, Madjiheurem S, Zgubic M, Lovett S, Tomlinson H, Elster S, Apps C, Witherspoon S. Canos: A fast and scalable neural ac-opf solver robust to n-1 perturbations. 2024, arXiv preprint arXiv:2403.17660.

[75] Jeddi AB, Shafieezadeh A. A physics-informed graph attention-based approach for power flow analysis. In: 2021 20th IEEE international conference on machine learning and applications. IEEE; 2021, p. 1634–40.

[76] Varbella A, Briens D, Gjorgiev B, D'Inverno GA, Sansavini G. Physics-informed GNN for non-linear constrained optimization: PINCO a solver for the AC-optimal power flow. 2024, arXiv preprint arXiv:2410.04818.

[77] Di Giovanni F, Rusch TK, Bronstein M, Deac A, Lackenby M, Mishra S, Veličković P. How does over-squashing affect the power of GNNs? Trans Mach Learn Res 2024. URL https://openreview.net/forum?id=KJRoQvRWNs.

[78] Bianchi FM, Grattarola D, Livi L, Alippi C. Graph neural networks with convolutional arma filters. IEEE Trans Pattern Anal Mach Intell 2021;44(7):3496–507.

[79] Ringsquandl M, Sellami H, Hildebrandt M, Beyer D, Henselmeyer S, Weber S, Joblin M. Power to the relational inductive bias: Graph neural networks in electrical power grids. In: Proceedings of the 30th ACM international conference on information & knowledge management. Virtual Event Queensland Australia: ACM; 2021, p. 1538–47. http://dx.doi.org/10.1145/3459637.3482464, URL https://dl.acm.org/doi/10.1145/3459637.3482464.

[80] Ding M, Rabbani T, An B, Wang EZ, Huang F. Sketch-GNN: Scalable graph neural networks with sublinear training complexity. In: Oh AH, Agarwal A, Belgrave D, Cho K, editors. Advances in neural information processing systems. 2022, URL https://openreview.net/forum?id=4PJbcrW_7wC.

[81] Hamann HF, Gjorgiev B, Brunschwiler T, Martins LS, Puech A, Varbella A, Weiss J, Bernabe-Moreno J, Massé AB, Choi SL, et al. Foundation models for the electric power grid. Joule 2024;8(12):3245–58.

[82] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, van den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S, Grewe D, Nham J, Kalchbrenner N, Sutskever I, Lillicrap T, Leach M, Kavukcuoglu K, Graepel T, Hassabis D. Mastering the game of go with deep neural networks and tree search. Nature 2016;529(7587):484–9. http://dx.doi.org/10.1038/nature16961.

[83] Schrittwieser J, Antonoglou I, Hubert T, Simonyan K, Sifre L, Schmitt S, Guez A, Lockhart E, Hassabis D, Graepel T, Lillicrap T, Silver D. Mastering atari, go, chess and shogi by planning with a learned model. Nature 2020;588(7839):604–9. http://dx.doi.org/10.1038/s41586-020-03051-4.

[84] Ye W, Liu S, Kurutach T, Abbeel P, Gao Y. Mastering atari games with limited data. In: Ranzato M, Beygelzimer A, Dauphin Y, Liang P, Vaughan JW, editors. Advances in neural information processing systems. 34, Curran Associates, Inc.; 2021, p. 25476–88, URL https://proceedings.neurips.cc/paper_files/paper/2021/file/d5eca8dc3820cad9fe56a3bafda65ca1-Paper.pdf.

[85] Sutton RS, McAllester D, Singh S, Mansour Y. Policy gradient methods for reinforcement learning with function approximation. In: Proceedings of the 12th international conference on neural information processing systems. Cambridge, MA, USA: MIT Press; 1999, p. 1057–63.

[86] Watkins CJCH, Dayan P. Q-learning. Mach Learn 1992;8(3):279–92. http://dx.doi.org/10.1007/BF00992698.

[87] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33. http://dx.doi.org/10.1038/nature14236.

[88] Hasselt Hv, Guez A, Silver D. Deep reinforcement learning with double Q-learning. In: Proceedings of the thirtieth AAAI conference on artificial intelligence. AAAI Press; 2016, p. 2094–100.

[89] Wang Z, Schaul T, Hessel M, Van Hasselt H, Lanctot M, De Freitas N. Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd international conference on international conference on machine learning - volume 48. JMLR.org; 2016, p. 1995–2003.

[90] Hessel M, Modayil J, van Hasselt H, Schaul T, Ostrovski G, Dabney W, Horgan D, Piot B, Azar MG, Silver D. Rainbow: Combining improvements in deep reinforcement learning. 2017, CoRR, arXiv:1710.02298.

[91] Sutton RS, Barto AG. Reinforcement learning: An introduction. 2nd ed.. The MIT Press; 2018, URL http://incompleteideas.net/book/the-book-2nd.html.

[92] Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K. Asynchronous methods for deep reinforcement learning. In: Balcan MF, Weinberger KQ, editors. Proceedings of the 33rd international conference on machine learning. Proceedings of machine learning research, vol. 48, New York, New York, USA: PMLR; 2016, p. 1928–37, URL https://proceedings.mlr.press/v48/mniha16.html.

[93] Lillicrap T, Hunt J, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. 2015, CoRR.

[94] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, CoRR, arXiv:1707.06347.

[95] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. Pmlr; 2018, p. 1861–70.

[96] Schwarzer M, Obando-Ceron J, Courville A, Bellemare MG, Agarwal R, Castro PS. Bigger, better, faster: human-level atari with human-level efficiency. In: Proceedings of the 40th international conference on machine learning. JMLR.org; 2023.

[97] Fedus W, Ramachandran P, Agarwal R, Bengio Y, Larochelle H, Rowland M, Dabney W. Revisiting fundamentals of experience replay. In: Proceedings of the 37th international conference on machine learning. JMLR.org; 2020.

[98] Fortunato M, Azar MG, Piot B, Menick J, Osband I, Graves A, Mnih V, Munos R, Hassabis D, Pietquin O, Blundell C, Legg S. Noisy networks for exploration. 2019, arXiv:1706.10295.

[99] Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, Tavener S, Perez D, Samothrakis S, Colton S. A survey of Monte Carlo tree search methods. IEEE Trans Comput Intell AI Games 2012;4(1):1–43. http://dx.doi.org/10.1109/TCIAIG.2012.2186810.

[100] Viebahn J, Naglic M, Marot A, Donnot B, Tindemans SH. Potential and challenges of AI-powered decision support for short-term system operations. 2022, CIGRE 2022 Paris Session.

[101] Kamel M, Wang Y, Yuan C, Li F, Liu G, Dai R, Cheng X. A reinforcement learning approach for branch overload relief in power systems. In: 2020 IEEE power & energy society general meeting. 2020, p. 1–5. http://dx.doi.org/10.1109/PESGM41954.2020.9281402.

[102] Bai Y, Chen S, Zhang J, Xu J, Gao T, Wang X, Wenzhong Gao D. An adaptive active power rolling dispatch strategy for high proportion of renewable energy based on distributed deep reinforcement learning. Appl Energy 2023;330:120294. http://dx.doi.org/10.1016/j.apenergy.2022.120294, URL https://www.sciencedirect.com/science/article/pii/S0306261922015513.

[103] Fuxjäger AR, Kozak K, Dorfer M, Blies PM, Wasserer M. Reinforcement learning based power grid day-ahead planning and AI-assisted control. 2023, arXiv:2302.07654.

[104] Larik RM, Mustafa MW, Aman MN. A critical review of the state-of-art schemes for under voltage load shedding. Int Trans Electr Energy Syst 2019;29(5):e2828. http://dx.doi.org/10.1002/2050-7038.2828, e2828 ITEES-18-0343.R1.

[105] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. 2017, arXiv preprint arXiv:1712.01815.

[106] Lee D, Nguyen HD, Dvijotham K, Turitsyn K. Convex restriction of power flow feasibility sets. IEEE Trans Control Netw Syst 2019;6(3):1235–45. http://dx.doi.org/10.1109/TCNS.2019.2930896.

[107] Capitanescu F, Wehenkel L. Experiments with the interior-point method for solving large scale optimal power flow problems. Electr Power Syst Res 2013;95:276–83. http://dx.doi.org/10.1016/j.epsr.2012.10.001, URL https://www.sciencedirect.com/science/article/pii/S0378779612002994.

[108] Mohagheghi E, Alramlawi M, Gabash A, Li P. A survey of real-time optimal power flow. Energies 2018;11(11). http://dx.doi.org/10.3390/en11113142, URL https://www.mdpi.com/1996-1073/11/11/3142.

[109] Rajaei A, Palensky P, Cremer JL. Transferable graph learning for transmission congestion management via busbar splitting. 2025, arXiv:2510.20591.

[110] Donnot B. Grid2op- a testbed platform to model sequential decision making in power systems. 2020, https://GitHub.com/rte-france/grid2op.

[111] Lehna M, Viebahn J, Marot A, Tomforde S, Scholz C. Managing power grids through topology actions: A comparative study between advanced rule-based and reinforcement learning agents. Energy AI 2023;14:100276. http://dx.doi.org/10.1016/j.egyai.2023.100276, URL https://www.sciencedirect.com/science/article/pii/S2666546823000484.

[112] Taha S, Poland J, Knezovic K, Shchetinin D. Learning to run a power network under varying grid topology. In: 2022 IEEE 7th international energy conference. 2022, p. 1–6. http://dx.doi.org/10.1109/ENERGYCON53164.2022.9830198.

[113] Yoon D, Hong S, Lee B-J, Kim K-E. winning the l2rpn challenge: Power grid management via semi-markov afterstate actor-critic. 2021, p. 18.

[114] Sar E, Zocca A, Bhulai S. Multi-agent reinforcement learning for power grid topology optimization. 2023.

[115] de Jong M, Viebahn J, Shapovalova Y. Imitation learning for intra-day power grid operation through topology actions. 2024, arXiv:2407.19865.

[116] de Jong M, Viebahn J, Shapovalova Y. Generalizable graph neural networks for robust power grid topology control. 2025, arXiv:2501.07186.

[117] Hassouna M, Holzhüter C, Lehna M, de Jong M, Viebahn J, Sick B, Scholz C. Learning topology actions for power grid control: A graph-based soft-label imitation learning approach. In: Machine learning and knowledge discovery in databases. applied data science track and demo track. Cham: Springer Nature Switzerland; 2026, p. 129–46.

[118] Xu P, Pei Y, Zheng X, Zhang J. A simulation-constraint graph reinforcement learning method for line flow control. In: 2020 IEEE 4th conference on energy internet and energy system integration (EI2). 2020, p. 319–24. http://dx.doi.org/10.1109/EI250167.2020.9347305.

[119] Qiu Z, Zhao Y, Shi W, Su F, Zhu Z. Distribution network topology control using attention mechanism-based deep reinforcement learning. In: 2022 4th international conference on electrical engineering and control technologies. 2022, p. 55–60. http://dx.doi.org/10.1109/CEECT55960.2022.10030642.

[120] Fabrizio C, Losapio G, Mussi M, Metelli AM, Restelli M. Power grid control with graph-based distributed reinforcement learning. 2025, arXiv:2509.02861.

[121] Batanero EA, Fernández Á, Barbero Á. Graph-enhanced model-free reinforcement learning agents for efficient power grid topological control. 2025, arXiv:2503.20688.

[122] Peter BM, Korkali M. Robust defense against extreme grid events using dual-policy reinforcement learning agents. In: 2025 IEEE texas power and energy conference. 2025, p. 1–6. http://dx.doi.org/10.1109/TPEC63981.2025.10907039.

[123] Xu P, Duan J, Zhang J, Pei Y, Shi D, Wang Z, Dong X, Sun Y. Active power correction strategies based on deep reinforcement learning—Part I: A simulation-driven solution for robustness. CSEE J Power Energy Syst 2022;8(4):1122–33. http://dx.doi.org/10.17775/CSEEJPES.2020.07090, Conference Name: CSEE Journal of Power and Energy Systems.

[124] Zhao Y, Liu J, Liu X, Yuan K, Ren K, Yang M. A graph-based deep reinforcement learning framework for autonomous power dispatch on power systems with changing topologies. In: 2022 IEEE sustainable power and energy conference. IEEE; 2022, p. 1–5.

[125] Wu T, Scaglione A, Arnold D. Constrained reinforcement learning for predictive control in real-time stochastic dynamic optimal power flow. IEEE Trans Power Syst 2024;39(3):5077–90. http://dx.doi.org/10.1109/TPWRS.2023.3326121.

[126] Adamczyk J, Makarenko V, Tiomkin S, Kulkarni RV. Bootstrapped reward shaping. In: Proceedings of the thirty-ninth AAAI conference on artificial intelligence and thirty-seventh conference on innovative applications of artificial intelligence and fifteenth symposium on educational advances in artificial intelligence. AAAI'25/IAAI'25/EAAI'25, AAAI Press; 2025, http://dx.doi.org/10.1609/aaai.v39i15.33679.

[127] Yan R, Xing Q, Xu Y. Multi agent safe graph reinforcement learning for PV inverter s based real-time de centralized Volt/Var control in zoned distribution networks. IEEE Trans Smart Grid 2023. http://dx.doi.org/10.1109/TSG.2023.3277087, 1–1, Conference Name: IEEE Transactions on Smart Grid.

[128] van der Sar E, Zocca A, Bhulai S. Multi-agent reinforcement learning for power grid topology optimization. 2023, arXiv preprint arXiv:2310.02605.

[129] Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. 2017, CoRR, arXiv:1706.02216.

[130] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Process Syst 2017;30.

[131] Marchesini E, Donnot B, Crozier C, Dytham I, Merz C, Schewe L, Westerbeck N, Wu C, Marot A, Donti PL. RL2Grid: Benchmarking reinforcement learning in power grid operations. 2025, arXiv:2503.23101.

[132] Turkia SB, Pigeon-Schneider O, Pache C, Dolley B, Saludjian L, Panciatici P. D-GITT RTE 7000 nodes dataset. 2024, URL https://huggingface.co/datasets/OpenSynth/D-GITT-RTE7000-2021.

[133] Lehna M, Hassouna M, Degtyar D, Tomforde S, Scholz C. Fault detection for agents on power grid topology optimization: A comprehensive analysis. 2024, arXiv:2406.16426.

[134] Viebahn J, Kop S, van DIJK J, Budaya H, Streefland M, Barbieri D, Champion P, Jothy M, Renault V, Tindemans S. C2 - GridOptions tool: Real-world day-ahead congestion management using topological remedial actions. CIGRE Sci Eng 2024;2024-December(35).

[135] Donon B, Cubelier F, Karangelos E, Wehenkel L, Crochepierre L, Pache C, Saludjian L, Panciatici P. Topology-aware reinforcement learning for tertiary voltage control. Electr Power Syst Res 2024;234:110658. http://dx.doi.org/10.1016/j.epsr.2024.110658, URL https://www.sciencedirect.com/science/article/pii/S0378779624005443.

[136] Mussi M, Metelli AM, Restelli M, Losapio G, Bessa RJ, Boos D, Borst C, Leto G, Castagna A, Chavarriaga R, Dias D, Egli A, Eisenegger A, El Manyari Y, Fuxjäger A, Geraldes J, Hamouche S, Hassouna M, Lemetayer B, Leyli-Abadi M, Liessner R, Lundberg J, Marot A, Meddeb M, Schiaffonati V, Schneider M, Stadelmann T, Usher J, Van Hoof H, Viebahn J, Waefler T, Zanotti G. Human-AI interaction in safety-critical network infrastructures. IScience 2025;28(9). http://dx.doi.org/10.1016/j.isci.2025.113400.

[137] Leyli-abadi M, Bessa RJ, Viebahn J, Boos D, Borst C, Castagna A, Chavarriaga R, Hassouna M, Lemetayer B, Leto G, Marot A, Meddeb M, Meyer M, Schiaffonati V, Schneider M, Waefler T. A conceptual framework for AI-based decision systems in critical infrastructures. 2025, arXiv:2504.16133.

[138] Beinert D, Holzhüter C, Thomas JM, Vogt S. Power flow forecasts at transmission grid nodes using graph neural networks. Energy AI 2023;14:100262. http://dx.doi.org/10.1016/j.egyai.2023.100262, URL https://www.sciencedirect.com/science/article/pii/S2666546823000344.

[139] Fan T-H, Lee XY, Wang Y. Powergym: A reinforcement learning environment for volt-var control in power distribution systems. In: Learning for dynamics and control conference. PMLR; 2022, p. 21–33.

[140] Xu P, Wang B, Zhang Y, Zhu H. Online topology-based voltage regulation: A computational performance enhanced algorithm based on deep reinforcement learning. IET Gener Transm Distrib 2022;16(24):4879–92. http://dx.doi.org/10.1049/gtd2.12433, URL https://onlinelibrary.wiley.com/doi/abs/10.1049/gtd2.12433, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/gtd2.12433.

[141] Duan J, Shi D, Diao R, Li H, Wang Z, Zhang B, Bian D, Yi Z. Deep-reinforcement-learning-based autonomous voltage control for power grid operations. IEEE Trans Power Syst 2020;35(1):814–7. http://dx.doi.org/10.1109/TPWRS.2019.2941134.

[142] Yang C, Sun Y, Zou Y, Zheng F, Liu S, Zhao B, Wu M, Cui H. Optimal power flow in distribution network: A review on problem formulation and optimization methods. Energies 2023;16(16). http://dx.doi.org/10.3390/en16165974, URL https://www.mdpi.com/1996-1073/16/16/5974.

[143] Heid J, Bornhorst N, Tönges E, Härtel P, Mende D, Braun M. A computationally efficient method for solving mixed-integer AC optimal power flow problems. In: 2025 IEEE kiel powerTech. IEEE; 2025, p. 1–7. http://dx.doi.org/10.1109/powertech59965.2025.11180420.

[144] Pham TN, Shah R, Dao MN, Sultanova N, Islam S. Low and medium voltage distribution network planning with distributed energy resources: a survey. Electr Eng 2024;107(2):1797–828. http://dx.doi.org/10.1007/s00202-024-02535-0.

[145] Liao H. Review on distribution network optimization under uncertainty. Energies 2019;12(17). http://dx.doi.org/10.3390/en12173369, URL https://www.mdpi.com/1996-1073/12/17/3369.

[146] Lotfi H, Hajiabadi ME, Parsadust H. Power distribution network reconfiguration techniques: A thorough review. Sustainability 2024;16(23). http://dx.doi.org/10.3390/su162310307, URL https://www.mdpi.com/2071-1050/16/23/10307.

[147] Tinney WF, Hart CE. Power flow solution by Newton's method. IEEE Trans Power Appar Syst 1967;PAS-86(11):1449–60. http://dx.doi.org/10.1109/TPAS.1967.291823.

[148] Mu C, Liu Z, Yan J, Jia H, Zhang X. Graph multi-agent reinforcement learning for inverter-based active voltage control. IEEE Trans Smart Grid 2023.

[149] Hu D, Li Z, Ye Z, Peng Y, Xi W, Cai T. Multi-agent graph reinforcement learning for decentralized volt-var control in power distribution systems. Int J Electr Power Energy Syst 2024;155:109531. http://dx.doi.org/10.1016/j.ijepes.2023.109531, URL https://www.sciencedirect.com/science/article/pii/S0142061523005884.

[150] Wang Y, Qiu D, Wang Y, Sun M, Strbac G. Graph learning-based voltage regulation in distribution networks with multi-microgrids. IEEE Trans Power Syst 2023.

[151] Lee XY, Sarkar S, Wang Y. A graph policy network approach for volt-var control in power distribution systems. Appl Energy 2022;323:119530.

[152] Wu H, Xu Z, Wang M, Zhao J, Xu X. Two-stage voltage regulation in power distribution system using graph convolutional network-based deep reinforcement learning in real time. Int J Electr Power Energy Syst 2023;151:109158.

[153] Wu T, Scaglione A, Arnold D. Reinforcement learning using physics inspired graph convolutional neural networks. In: 2022 58th annual allerton conference on communication, control, and computing (allerton). 2022, p. 1–8. http://dx.doi.org/10.1109/Allerton49937.2022.9929321.

[154] Cao D, Zhao J, Hu J, Pei Y, Huang Q, Chen Z, Hu W. Physics-informed graphical representation-enabled deep reinforcement learning for robust distribution system voltage control. IEEE Trans Smart Grid 2023. http://dx.doi.org/10.1109/TSG.2023.3267069, 1–1, Conference Name: IEEE Transactions on Smart Grid.

[155] Li J, Zhang R, Wang H, Liu Z, Lai H, Zhang Y. Deep reinforcement learning for voltage control and renewable accommodation using spatial-temporal graph information. IEEE Trans Sustain Energy 2023.

[156] Xing Q, Chen Z, Zhang T, Li X, Sun K. Real-time optimal scheduling for active distribution networks: A graph reinforcement learning method. Int J Electr Power Energy Syst 2023;145:108637. http://dx.doi.org/10.1016/j.ijepes.2022.108637, URL https://www.sciencedirect.com/science/article/pii/S0142061522006330.

[157] Hossain RR, Huang Q, Huang R. Graph convolutional network-based topology embedded deep reinforcement learning for voltage stability control. IEEE Trans Power Syst 2021;36(5):4848–51. http://dx.doi.org/10.1109/TPWRS.2021.3084469, Conference Name: IEEE Transactions on Power Systems.

[158] Pei Y, Yang J, Wang J, Xu P, Zhou T, Wu F. An emergency control strategy for undervoltage load shedding of power system: A graph deep reinforcement learning method. IET Gener Transm Distrib 2023;17(9):2130–41.

[159] Zhao T, Wang J. Learning sequential distribution system restoration via graph-reinforcement learning. IEEE Trans Power Syst 2022;37(2):1601–11. http://dx.doi.org/10.1109/TPWRS.2021.3102870, Conference Name: IEEE Transactions on Power Systems.

[160] Jacob RA, Paul S, Chowdhury S, Gel YR, Zhang J. Real-time outage management in active distribution networks using reinforcement learning over graphs. Nat Commun 2024;15(1):4766.

[161] Jacob RA, Paul S, Li W, Chowdhury S, Gel YR, Zhang J. Reconfiguring unbalanced distribution networks using reinforcement learning over graphs. In: 2022 IEEE texas power and energy conference. IEEE; 2022, p. 1–6.

[162] Chen J, Yu T, Pan Z, Zhang M, Deng B. A scalable graph reinforcement learning algorithm based stochastic dynamic dispatch of power system under high penetration of renewable energy. Int J Electr Power Energy Syst 2023;152:109212. http://dx.doi.org/10.1016/j.ijepes.2023.109212, URL https://www.sciencedirect.com/science/article/pii/S0142061523002697.

[163] Rokhforoz P, Montazeri M, Fink O. Multi-agent reinforcement learning with graph convolutional neural networks for optimal bidding strategies of generation units in electricity markets. Expert Syst Appl 2023;225:120010. http://dx.doi.org/10.1016/j.eswa.2023.120010, URL https://www.sciencedirect.com/science/article/pii/S0957417423005122.

[164] Lee S, Nengroo SH, Jin H, Heo T, Doh Y, Lee C, Har D. Reinforcement learning-based cooperative P2P power trading between DC nanogrid clusters with wind and PV energy resources. 2022, http://dx.doi.org/10.48550/arXiv.2209.07744, arXiv:2209.07744, [cs, eess].

[165] Islam MA, Ismail M, Atat R, Boyaci O, Shannigrahi S. Software-defined network-based proactive routing strategy in smart power grids using graph neural network and reinforcement learning. E-Prime - Adv Electr Eng Electron Energy 2023;5:100187. http://dx.doi.org/10.1016/j.prime.2023.100187, URL https://www.sciencedirect.com/science/article/pii/S2772671123000827.

[166] Xu P, Zhang J, Gao T, Chen S, Wang X, Jiang H, Gao W. Real-time fast charging station recommendation for electric vehicles in coupled power-transportation networks: A graph reinforcement learning method. Int J Electr Power Energy Syst 2022;141:108030. http://dx.doi.org/10.1016/j.ijepes.2022.108030, URL https://www.sciencedirect.com/science/article/pii/S0142061522000746.

[167] Xing Q, Xu Y, Chen Z. A bilevel graph reinforcement learning method for electric vehicle fleet charging guidance. IEEE Trans Smart Grid 2023. http://dx.doi.org/10.1109/TSG.2023.3240580, 1–1, Conference Name: IEEE Transactions on Smart Grid.

[168] Bayani R, Manshadi SD, Liu G, Wang Y, Dai R. Autonomous charging of electric vehicle fleets to enhance renewable generation dispatchability. CSEE J Power Energy Syst 2022;8(3):13. http://dx.doi.org/10.17775/CSEEJPES.2020.04000, URL https://www.sciopen.com/article/10.17775/CSEEJPES.2020.04000.

[169] Sadeghianpourhamami N, Deleu J, Develder C. Definition and evaluation of model-free coordination of electrical vehicle charging with reinforcement learning. IEEE Trans Smart Grid 2020;11(1):203–14. http://dx.doi.org/10.1109/TSG.2019.2920320.

[170] Li H, Wan Z, He H. Constrained EV charging scheduling based on safe deep reinforcement learning. IEEE Trans Smart Grid 2020;11(3):2427–39. http://dx.doi.org/10.1109/TSG.2019.2955437.

[171] Silva FLD, Nishida CEH, Roijers DM, Costa AHR. Coordination of electric vehicle charging through multiagent reinforcement learning. IEEE Trans Smart Grid 2020;11(3):2347–56. http://dx.doi.org/10.1109/TSG.2019.2952331.

[172] Xing Q, Xu Y, Chen Z, Zhang Z, Shi Z. A graph reinforcement learning-based decision-making platform for real-time charging navigation of urban electric vehicles. IEEE Trans Ind Inform 2023;19(3):3284–95. http://dx.doi.org/10.1109/TII.2022.3210264, Conference Name: IEEE Transactions on Industrial Informatics.

[173] Zhang W, Liu H, Han J, Ge Y, Xiong H. Multi-agent graph convolutional reinforcement learning for dynamic electric vehicle charging pricing. In: Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining. Washington DC USA: ACM; 2022, p. 2471–81. http://dx.doi.org/10.1145/3534678.3539416, URL https://dl.acm.org/doi/10.1145/3534678.3539416.

[174] Lee ZJ, Lee G, Lee T, Jin C, Lee R, Low Z, Chang D, Ortega C, Low SH. Adaptive charging networks: A framework for smart electric vehicle charging. IEEE Trans Smart Grid 2021;12(5):4339–50. http://dx.doi.org/10.1109/TSG.2021.3074437.