IEEE ComSoc · IEEE Computer Society · IEEE Signal Processing Society · VTS
IEEE Transactions on Machine Learning in Communications and Networking

Received 1 December 2023; revised 19 March 2024 and 22 May 2024; accepted 27 May 2024.
Date of publication 3 June 2024; date of current version 10 June 2024.

The associate editor coordinating the review of this article and approving it for publication was A. Savard.

*Digital Object Identifier 10.1109/TMLCN.2024.3408723*

# Graph Neural Networks Approach for Joint Wireless Power Control and Spectrum Allocation

**MAHER MARWANI**[1] **(Student Member, IEEE), AND**
**GEORGES KADDOUM**[1,2] **(Senior Member, IEEE)**

[1]Electrical Engineering Department, École de Technologie Supérieure, Montreal, QC H3C 1K3, Canada
[2]Artificial Intelligence and Cyber Systems Research Center, Department of Computer Science and Mathematics,
Lebanese American University, Beirut 797751, Lebanon
CORRESPONDING AUTHOR: M. MARWANI (mahermarwani@gmail.com)

**ABSTRACT** The proliferation of wireless technologies and the escalating performance requirements of wireless applications have led to diverse and dynamic wireless environments, presenting formidable challenges to existing radio resource management (RRM) frameworks. Researchers have proposed utilizing deep learning (DL) models to address these challenges to learn patterns from wireless data and leverage the extracted information to resolve multiple RRM tasks, such as channel allocation and power control. However, it is noteworthy that the majority of existing DL architectures are designed to operate on Euclidean data, thereby disregarding a substantial amount of information about the topological structure of wireless networks. As a result, the performance of DL models may be suboptimal when applied to wireless environments due to the failure to capture the network's non-Euclidean geometry. This study presents a novel approach to address the challenge of power control and spectrum allocation in an N-link interference environment with shared channels, utilizing a graph neural network (GNN) based framework. In this type of wireless environment, the available bandwidth can be divided into blocks, offering greater flexibility in allocating bandwidth to communication links, but also requiring effective management of interference. One potential solution to mitigate the impact of interference is to control the transmission power of each link while ensuring the network's data rate performance. Therefore, the power control and spectrum allocation problems are inherently coupled and should be solved jointly. The proposed GNN-based framework presents a promising avenue for tackling this complex challenge. Our experimental results demonstrate that our proposed approach yields significant improvements compared to other existing methods in terms of convergence, generalization, performance, and robustness, particularly in the context of an imperfect channel.

**INDEX TERMS** Intelligent resource allocation, RRM, 6G, GNN, D2D, AI.

## I. INTRODUCTION

THE sixth generation (6G) of wireless communications is expected to feature heterogeneous networks capable of supporting a vast number of connected devices while delivering high data rates, low latencies, and energy efficiency. Several technologies have been developed to meet these requirements, including those referenced in [1], [2], [3], and [4]. However, the increasing complexity of radio resource management (RRM) has emerged as a significant challenge with the proliferation of new technologies and diverse demands [5], [6]. Previous RRM solutions are insufficient in adapting to the novel heterogeneous wireless environment in terms of convergence time, generalization to different wireless contexts, and maintaining satisfactory performance while scaling up the number of devices. Therefore, novel approaches are required to address these challenges and pave the way for the efficient management of wireless resources in the upcoming 6G era.

The focus of our research is on the N-link interference environment with shared channels, a wireless network architecture characterized by multiple communication links sharing the same available bandwidth. In this setting, the co-existence of multiple links causes significant interference and performance degradation, which necessitate effective management of transmission power control and spectrum allocation. This network structure can be observed in various wireless scenarios, including device-to-device (D2D) communication [7], [8], [9], [10], where multiple devices communicate directly without a network infrastructure, and uplink/downlink [11], [12], [13], [14], [15] scenarios, where a base station communicates with multiple users using the same spectrum, also known as non-orthogonal multiple access (NOMA). To maximize the network's sum rate, solving the mixed-integer, non-convex optimization problem involving power control and channel allocation is essential. However, obtaining a globally optimal solution within the required time is challenging. Therefore, researchers have proposed several near-optimal solutions for specific cases [16], [17], [18], [19], which tend to have high computational complexity and are impractical for real-time scenarios.

In recent years, researchers have explored the use of machine learning (ML) techniques to address wireless network optimization problems. Specifically, there has been interest in incorporating deep learning (DL) approaches, which have shown promise in a variety of applications. Two primary approaches have been pursued in this integration: (1) constructing end-to-end learnable architectures that can capture complex relationships between inputs and outputs [20], [21], [22], and (2) replacing computational blocks within existing solutions with DL architectures to reduce computational costs [23], [24]. Despite promising results, existing DL-based approaches have primarily focused on addressing isolated RRM tasks such as power control, user association, and link scheduling. Moreover, their scalability to large wireless networks is a concern as they scale linearly with respect to the size of the input data. Furthermore, techniques such as multi-layer perceptrons (MLPs) and convolutional neural networks (CNNs) can be subjected to overfitting and, thus, require large amounts of training data. Additionally, these methods rely on tabular data, such as channel state information (CSI), which ignores the network's underlying topology. Therefore, there is a need for further research to explore more effective ways to integrate DL approaches into wireless network optimization problems. Recent research has demonstrated the potential for improving the scalability and generalization of DL-based RRM solutions by integrating the target task's structure into the neural network architecture [25]. Given that wireless networks can be intuitively modeled as graph topologies, there is a growing interest in leveraging graph representation learning techniques to enhance the performance of RRM algorithms [26]. One such approach is Graph Neural Networks (GNNs), which possess several attractive properties, including permutation equivariance, scalability, generalization, high computational efficiency, and the ability to train efficiently on relatively small datasets [27]. The application of GNNs has yielded promising results in various domains, indicating its potential as an effective technique for enhancing the performance of RRM algorithms in wireless networks.

The primary aim of this research is to propose a solution that simultaneously addresses spectrum allocation and power control tasks. Initially, we formulate a network mean rate maximization problem, considering both RRM tasks and the minimum Quality of Service (QoS) required for each communication link. Subsequently, we create interference graphs from the network's CSI, enabling parallel processing without information loss. Additionally, we develop an end-to-end GNN-based framework that learns from these constructed graphs and embeds them into Euclidean space. This embedding is used to compute power and channel allocation solutions. In contrast to Deep Neural Network (DNN) models, our framework is both scalable and generalizable, requiring no retraining or architectural modification when changing the input size. It also excels in computational efficiency due to parallel execution. To enhance the model's generalization and training stability, we combine four loss functions: the supervised mean squared error for power control, the supervised cross-entropy for channel allocation, an unsupervised loss to avoid constraining the model with an upper-bound performance from supervised training, and a regulation loss to ensure QoS constraints are met. Lastly, we rigorously tested our approach, focusing on the training convergence, generalization across different wireless setups, network mean rate, QoS violation, scalability with input size, and robustness in the presence of imperfect channel estimation.

This paper is structured as follows. In Section II, a comprehensive literature review is presented to explore the previous work related to our research. In Section III, we introduce the N-link interference environment with shared channels, which is the problem setting that our proposed solution is designed to address. In Section IV, we present the optimization problem that we aim to solve. In Section V, we provide a detailed description of our proposed end-to-end solution architecture, which consists of various components, including CSI preprocessing, the GNN feature extractor, the MLP component, the loss function design, and the training process. In Section VI, we conduct extensive simulations to evaluate our proposed solution's performance in terms of stability, generalization, and robustness compared to the state-of-the-art methods. Finally, in Section 7, we present our conclusions and future research directions.

## II. LITERATURE REVIEW

Numerous studies have focused on solving the power control and spectrum allocation problems in different network topologies, either independently or concurrently. For example, in [16], the authors proposed a dual-based iterative algorithm that allocates resources to D2D pairs while maintaining the quality of service requirements. Another study [19] developed a two-stage algorithm to maximize the energy efficiency of D2D communication under cellular constraints, assuming that each D2D link could use one sub-channel at most to decrease the computational complexity. Conversely, this research [17] proposed a channel and power allocation scheme with channel reuse based on the Hungarian algorithm and a prioritizing method. Moreover, this work [18] employed a game-theory approach to manage the reuse of multiple channels by multiple D2D pairs. Despite the various proposed solutions, most of them were heuristic approaches or tended to convexify the RRM problems, resulting in a high computational complexity. Additionally, they did not provide complete flexibility in allocating multiple channels to multiple links, mainly because of convergence issues.

Given the limitations of model-based and heuristic solutions, researchers have turned to learnable approaches by integrating DL architecture to tackle RRM problems. For example, these works [21], [28], [29] integrated a DL component to learn the optimal pruning policy for the branch-and-bound (B&B) algorithm to solve mixed-integer nonlinear programming (MINLP) problems. While this approach simplifies the problem significantly and reduces the exponential computation of the traditional B&B algorithm, intense sampling is required to train the DL architecture since the training is supervised. To alleviate the need for training data, unsupervised learning approaches have been explored [30]. This work considers constructing a DNN framework to solve beamforming problems over an imperfect channel, which is trained in an unsupervised fashion using the negative of the sum rate. Similarly, [22] constructed a DNN that takes CSI and computes the power of each user. However, this work did not integrate the minimum rate constraints into the training process, which raises questions about the solution's feasibility. Another approach to training DL models is combining supervised and unsupervised losses. For instance, [31] proposed an end-to-end DL framework to solve resource allocation in multi-channel cellular systems with D2D links. Moreover, the approach can be implemented in a centralized manner, with full knowledge of the CSI, or distributed manner with partial CSI. However, the authors transform the continuous power variable into a set of discrete levels in order to use the cross-entropy loss. Following the same training approach in [24], a CNN model is employed to learn the patterns from CSI and output the power control that maximizes the energy or spectrum efficiency of the network. Despite the promising results achieved by the current DNN and CNN approaches, their lack of flexibility with input sizes is a significant limitation. Any alteration in input shape necessitates architectural modification. Furthermore, they prove inadequate in large wireless scenarios with a substantial number of connected devices. This deficiency stems from their heavy reliance on the quality of training data, which can be challenging to obtain in real-life situations. Moreover, the training process for these models is often time-consuming and typically conducted offline. Another drawback of DNN and CNN approaches is their disregard for the geometric information inherent in the input data.

To facilitate the incorporation of input data structures into DL models for RRM tasks, several solutions based on GNNs have been introduced [26]. These approaches have exhibited promise in tackling various RRM tasks, encompassing channel allocation, power control, and user association. For instance, in [32], a framework combining Deep Reinforcement Learning (DRL) and Graph Convolution Networks (GCNs) was proposed for channel allocation. This method enabled the agent to learn optimal channel assignments to access points using features extracted from the wireless environment as a state space. However, the model's testing was confined to a wireless setting with perfect channels and a relatively small number of devices. Similarly, [33] introduced a GNN-based framework for learning resource allocation strategies in wireless networks, offering reduced training times and improved scalability compared to conventional MLPs. Nonetheless, this framework was not well-suited for heterogeneous wireless devices or systems with single or multiple antennas. To address these limitations, [25] presented a more flexible GNN-based solution for constrained power allocation in a heterogeneous MIMO-interfering environment. Leveraging the permutation-invariant properties of RRM problems, this GNN architecture demonstrated excellent generalization across different problem scales with minimal training data. In addition, in [34], researchers aimed to find the optimal power control strategy in an uplink multi-cell network by combining DNNs with knowledge from the wireless network's topology, reducing training complexity and model parameters. However, this approach was tailored exclusively to the power control task. In contrast, [35] proposed employing GNNs to tackle power control and beamforming issues in heterogeneous D2D networks. Here, communication and interference links were represented as vertices in the wireless graph, and an unsupervised learning process was employed for the graph convolutional model. This method demonstrated favorable properties such as scalability and reduced execution time compared to alternative approaches. Similarly, [36] introduced an Access Point (AP) selection strategy for massive cell-free Multiple Input, Multiple Output (MIMO) systems based on GNNs. The authors constructed two graphs: a homogeneous one representing only AP nodes and a heterogeneous one containing both user equipments and AP nodes. However, these methods modeled the wireless network as a single graph, assuming that all communication links interfered with each other. GNNs can serve as end-to-end learnable solutions or feature extractors. For example, [37] proposed a joint optimization framework for user association and power control in a heterogeneous ultra-dense network. Similarly, [38] improved the Iteratively Weighted Minimum Mean Square Error (WMMSE) algorithm [39] by incorporating trainable components parametrized by GNNs. Simulations illustrated that the proposed method, unfolded WMMSE, delivered a comparable performance to WMMSE but with significantly lower time complexity.

The work in [40] introduced a trainable resilient RRM policy using an unsupervised primal-dual approach for power control and user association. Another paper [41] presented an edge-update empowered GNN architecture, enhancing GNNs' ability to handle node and edge variables and validating its Permutation Equivariance in power allocation scenarios. Additionally, [42] introduced Aggregation GNNs for decentralized resource allocation in wireless networks, utilizing a model-free primal-dual approach for asynchronous local information processing. The study in [43] proposed a distributed spectrum allocation scheme for vehicle-to-everything (V2X) networks using GNNs and multi-agent RL to optimize the network capacity. Furthermore, [44] discussed GNN-based frameworks for distributed power allocation in wireless networks, aimed at minimizing signaling overhead by incorporating Recurrent Neural Networks (RNNs) to capture temporal dynamics. The work in [45] offered a GNN framework to enhance power control and hybrid precoding in wireless systems, demonstrating scalability and efficiency. The work in [46] proposed a state-augmented algorithm for RRM in multi-user networks, ensuring feasible and nearly optimal decisions. In addition, [47] introduced a Heterogeneous GNN model for resource allocation in heterogeneous networks. The study in [48] presented a GNN-based scheme for RRM in wireless IoT networks, optimizing resources in D2D communications. Lastly, [49] explored the expressive power of GNNs in learning wireless policies, highlighting the limitations of Vertex-GNNs and the advantages of Edge-GNNs in resource allocation tasks.

Nevertheless, most of the mentioned GNN-based works are not adaptable to environments where multiple resource blocks have varying CSI. Additionally, many of these methods focus on addressing individual RRM tasks.

## III. SYSTEM MODEL

We denote $\mathcal{N} = \{1, 2, \dots, N\}$ a set of active (scheduled) links distributed randomly in a two-dimensional environment. The distance between transmitter-receiver pairs varies across links. We adopt a non-orthogonal scheme for all communication links, where $\mathcal{K} = \{1, 2, \dots, K\}$ is the set of resource blocks (RBs) with constant bandwidth $W$ that can be assigned to any link ($K \leq N$). In this environment, a centralized control unit (CCU) controls the transmission power of each link and allocates the required bandwidth to ensure effective communication, as illustrated in Fig. 1. We operate in a time slot scenario where the CCU obtains CSI from scheduled links, performs resource management, and communicates decisions to all transmitters. Although the CCU possesses full CSI knowledge, it may still be subject to noise and errors, leading us to evaluate our approach under noisy CSI conditions to assess its robustness in Section V.

We assume that the bandwidth $W$ of each RB is small enough to exhibit flat-fading channel characteristics. Additionally, due to block fading, CSI values change independently from one time slot to the next, requiring independent resource allocation for each frame. We introduce $g_{ii}^k \in \mathbb{R}$ to represent the direct channel gain between the transmitter and receiver of the $k$-th RB in the $i$-th link, and $g_{ij}^k \in \mathbb{R}$ to denote the channel gain between the transmitter of the $j$-th link and the receiver of the $i$-th link. To represent all channel gains, we define $G = [G^1, \dots, G^K] \in \mathbb{R}^{N\times N\times K}$ as the CSI tensor of all links in all resource blocks, where:

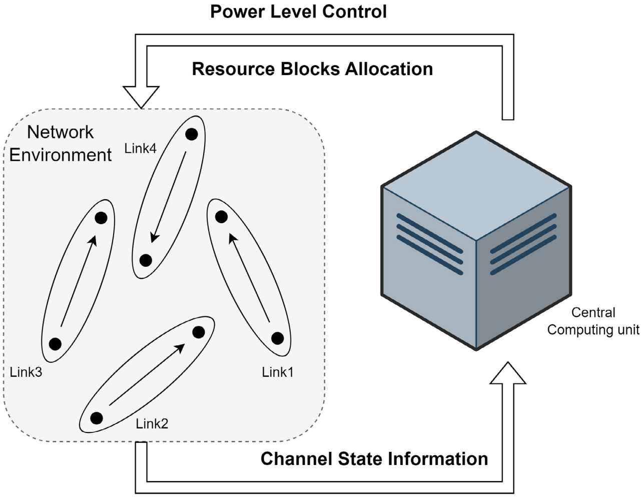


FIGURE 1. N-link interference environment with shared channels.

$$G^k = \begin{pmatrix} g_{11}^k & g_{12}^k & \cdots & g_{1N}^k \\ g_{21}^k & g_{22}^k & \cdots & g_{2N}^k \\ \vdots & \vdots & \ddots & \vdots \\ g_{N1}^k & g_{N2}^k & \cdots & g_{NN}^k \end{pmatrix}. \tag{1}$$

Taking into consideration the most common types of fading, the channel gain formula can be expressed as:

$$g_{ij}^k = \beta_{ij}^k \alpha_{ij}^k |h_{ij}^k|^2 \quad \forall k \in \mathcal{K}, \quad \forall (i,j) \in \mathcal{N} \times \mathcal{N}. \tag{2}$$

where $\beta_{ij}^k$ is the path loss proportional to the inverse of the distance, $\alpha_{ij}^k$ is the shadowing following the normal distribution, and $h_{ij}^k$ represents the small scale Rayleigh fading.

Each transmitter is equipped with a single antenna, and we represent the power allocation for all links as $P = [p_1, p_2, \cdots, p_N]$, where $p_i$ is the transmission power of the $i$-th link. We also consider a maximum transmission power limit, denoted as $P_{max}$, i.e., $p_i \leq p_{max}$. For RB assignment, we use binary variables $\psi_i^k \in \{0, 1\}$, where $\psi_i^k = 1$ indicates that the $i$-th link uses the $k$-th RB, and $\psi_i^k = 0$ otherwise. We denote the RB assignment for the $i$-th link as $\Psi_i = [\psi_i^1, \cdots, \psi_i^K]$. We assume that each link can use at most one RB per time slot, i.e., $\sum_{k=1}^K \psi_i^k \leq 1$.

In our analysis, we focus on the dedicated mode, where links experience no interference from other users. This means external interference is not considered in our calculations. We evaluate the signal-to-interference-plus-noise ratio (SINR) for the $k$-th RB in the $i$-th receiver, which is defined as:

$$\mathrm{SINR}_i^k = \frac{g_{ii}^k p_i}{\sum_{j\neq i}^{N} \psi_j^k g_{ij}^k p_j + N_0 W}, \forall (i,k) \in \mathcal{N} \times \mathcal{K}. \quad (3)$$

where $N_0$ is the noise density per unit bandwidth. Consequently, using the defined variables, we can calculate each link's achievable rate as:

$$\gamma_i = W \sum_{k=1}^{K} \psi_i^k \log_2 \left(1 + \mathrm{SINR}_i^k\right), \forall i \in \mathcal{N}. \quad (4)$$

Our objective is to find values for the variables $P$ and $\Psi$ that maximize the average network rate while ensuring quality of service, power constraints, and bandwidth limitations. This leads us to formulate the optimization problem as follows:

$$\begin{aligned} \max_{P,\Psi} \quad & \frac{1}{N} \sum_{i=1}^{N} \gamma_i & & \\ \text{s.t} \quad & \gamma_i \geq \gamma_{min} & & \forall i \in \mathcal{N} \\ & \sum_{k=1}^{K} \psi_i^k \leq 1 & & \forall i \in \mathcal{N} \\ & 0 \leq p_i \leq p_{max} & & \forall i \in \mathcal{N} \\ & \psi_i^k \in \{0, 1\} & & \forall (k, i) \in \mathcal{K} \times \mathcal{N} \end{aligned} \quad (5)$$

In (5), the first constraint ensures a minimum required data rate is met, the second constraint limits each link to using only one channel, and the third and fourth constraints restrict transmission powers within the defined maximum power and enforce binary channel indicators.

The problem at hand presents significant challenges due to the complexity of the objective function and the inclusion of mixed variables in the optimization process. Furthermore, the presence of time constraints, specifically related to the channel state, necessitates the adoption of a solver with a convergence time that is shorter than the coherence time to ensure the validity of the obtained solution. As such, we seek a universal approach capable of producing efficient solutions within the necessary timeframe while adhering to the imposed constraints. In the subsequent section, we introduce our novel GNN-based model, which offers a generalizable solution and yields promising outcomes, thereby fulfilling the aforementioned objectives.

## IV. SOLUTION ARCHITECTURE

In this section, we present a detailed exposition of our proposed model, including a description of the training process. Fig. 2 provides an overview of the model architecture. Essentially, the model takes the processed CSI, which has been transformed into separate interference graphs, as input and produces power levels and channel matrices as output. The model consists of two primary components: a feature extractor based on the GNNs and a CNN block that learns from the embedding vectors while simultaneously preserving the constraints inherent in the problem. The training process comprises two phases: a supervised phase that enhances the learning process and an unsupervised phase that maximizes our objective function while also mitigating overfitting.

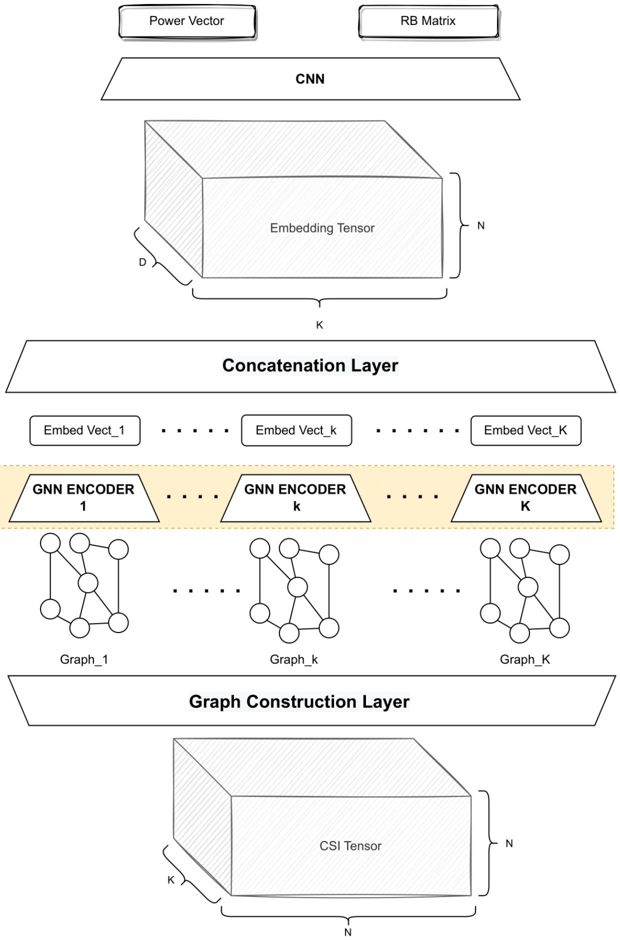


FIGURE 2. **Model architecture.**

### A. CHANNEL STATE INFORMATION PREPROCESSING

In this subsection, we describe the process followed to transform the CSI tensor $G$ into multiple graph structures to gain insights into their geometric properties. Our system model operates under specific constraints where each link is allocated a maximum of one RB, leading to potential interference only when multiple links share the same RB. Consequently, RBs are considered independent regarding interference, which simplifies the management of interactions between links. To effectively represent the CSI within this framework, we construct interference graphs for each RB, depicted as distinct subgraphs. This approach mirrors the uncorrelated flat fading characteristic of our system, where no correlation exists between different RBs. Since interference arises only when multiple links utilize the same bandwidth, we construct $K$ separate complete graphs ($N$ nodes and $N(N-1)/2$ edges) without any loss of information, where $K$ is the number of resource blocks. Specifically, we denote $\mathcal{G}_k(\mathcal{V}, \mathcal{E})$ as the interference graph of the $k$-th RB, where the nodes denote the communication links and the edges represent the interference links. Each node is labeled by $n_i = G^k_{ii} \quad \forall i \in \mathcal{N}$, indicating the signal strength or quality at each link, and each edge is labeled by $e_{ij} = G^k_{ij} \ \ \forall (i, j) \in \mathcal{N} \times \mathcal{N}$, representing the interference between links. This representation allows for a simplified yet effective understanding of the interactions and interference patterns within the network, leveraging the geometric properties of the graphs to facilitate analysis and optimization. Fig. 3 illustrates an example of a three-interference graph for three communication links.

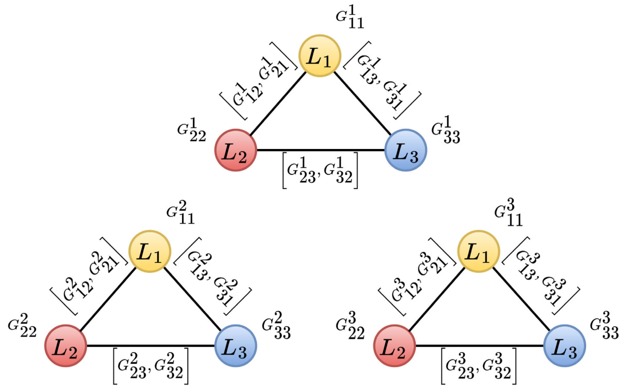


**FIGURE 3. Interference graphs of three communication links using three resource blocks.**

By modeling the CSI in this way, we enhance computational efficiency without sacrificing any essential information. This graph formulation enables us to parallelize the computation of our model, thereby reducing the model's training and execution time by approximately $1/K$. Additionally, handling relatively small graphs is more manageable in terms of complexity and memory control.

### B. GNN FEATURE EXTRACTOR

GNNs are a specialized type of neural network designed to operate on graph-structured data [50]. They share a multilayer structure akin to DNNs, where each node within the graph combines its individual features with an aggregation of the features of its neighboring nodes. Furthermore, GNNs update the embedding of each node through iterative aggregation and combination operations. This iterative process relies on a message-passing mechanism [51], where information is exchanged among nodes in the graph through their connecting edges to capture relationships between nodes. In essence, the $t$-th GNN layer for a node $v \in \mathcal{V}$ can be succinctly summarized through two key iterative equations (6):

$$\begin{aligned} m^{t+1}_{vu} &= \phi(z^t_v, w^t_{vu}, z^t_u) \\ z^{t+1}_v &= \sigma(z^t_v, \rho(\{m^{t+1}_{vu}, u \in \mathcal{N}(v)\})) \end{aligned} \tag{6}$$

Here, $z^t_v \in \mathbb{R}^{d_1}$ represents the hidden state of node $v \in \mathcal{V}$, and $w^t_{vu} \in \mathbb{R}^{d_2}$ denotes the feature vector associated with the edge $(v, u) \in \mathcal{E}$ at time $t$. In the $(t+1)$-th iteration, the edge features $w^t_{vu}$ are fused with the features of their incident nodes $\{z^t_u, z^t_v\}$ via the message function $\phi$. Subsequently, within each node $v$, messages from all neighboring nodes in $\mathcal{N}(v)$ are aggregated using the reduce function $\rho$. Finally, the node feature $z^{t+1}_v$ is updated using the update function $\sigma$. Different choices for combining and aggregation functions can lead to various types of GNNs [52], but the reduce function must always be a permutation-invariant operation, such as sum, mean, or max, to ensure that the input graph's global structure is preserved. Following a similar pattern, our computation equation (6) is defined as:

$$\begin{aligned} m^{t+1}_{vu} &= \mathrm{MLP}^t_1(z^t_u, g_{uv}, g_{vu}) \\ z^{t+1}_v &= \mathrm{MLP}^t_2(z^t_v, \max\{m^{t+1}_{vu}, u \in \mathcal{N}(v)\}) \end{aligned} \tag{7}$$

In this equation, the message function is represented as an MLP block, named $\mathrm{MLP}_1$, which consists of layers of neurons, non-linear activation functions, and batch normalization. It operates on the concatenation of node $u$'s hidden state and the features of edge $(v, u)$. The reduce function is essentially a max aggregation operation, which combines messages received from all neighbors of node $v$. The update function for node $v$ is similar to $\mathrm{MLP}_1$ but with a distinct number of neurons and is referred to as $\mathrm{MLP}_2$. It learns patterns from the aggregation of messages $m^{t+1}_{vu}$ from neighboring nodes and the node's previous hidden state. Notably, as the size of the hidden state $z^t_v$ of each node varies for each iteration $t$ due to the concatenation operations, each message-passing iteration has its specific message and update functions denoted as $\mathrm{MLP}^t_1$ and $\mathrm{MLP}^t_2$.

MLPs are preferred in our context for stability, efficiency, and performance. CNNs are less suitable due to the lack of spatial correlations in message function inputs in the message function input $(z^t_u, g_{uv}, g_{vu})$, and RNNs are limited by the absence of temporal patterns. Upon completing T iterations, the output of the GNN feature extractor comprises the embedding vectors for each node within each graph, represented as $z^T_{ik} \in \mathbb{R}^D; \ \ i \in \mathcal{V}^k$, where $\mathcal{V}^k$ denotes the node set of $\mathcal{G}_k$. These vectors are then stacked to form a global embedding tensor $Z \in \mathbb{R}^{N \times K \times D}$, where $D$ denotes the embedding dimension. Subsequently, the tensor $Z$ is used for computing the power vector and the channel allocation matrix in the next steps.

### C. CNN COMPONENT

In this section, we detail the CNN component of our approach, starting with the input of the embedding tensor $Z$ into a CNN block. This block executes a deconvolution operation along the embedding axis, outputting a matrix with dimensions $N \times K$. The CNN block consists of convolutional layers that extract features by progressively reducing the channel count, alongside ReLU activation functions, dropout for regularization, and batch normalization for stability. The result, denoted as $X = \mathrm{CNN}_d(Z)$, benefits from the CNN's ability to identify spatial patterns, augmenting the GNN's geometric insights.

We apply dimension manipulation techniques to process $X$ for our desired outputs—power vector and a Resource Block (RB) matrix. We treat $X$ as a matrix for channel allocation and use a softmax function across the relevant dimension to calculate channel probabilities. For power control, we condense $X$ along the $K$-axis into an $N \times 1$ vector and apply a sigmoid function to normalize the power values to a range between 0 and 1. We employ a softmax operation along the channel axis to compute the probabilities of selecting a channel. These probabilities are calculated as:

$$a_i^k = \frac{e^{x_{ik}}}{\sum_{s=1}^{K} e^{x_{ik}}} \tag{8}$$

Consequently, the channel allocation is calculated as follows:

$$\Psi_i^k = \begin{cases} 1 & \text{if } j = \text{argmax}_k(\{a_i^k, k \in \mathcal{K}\}) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The current formulation of $\Psi$ is not differentiable and would break the chain rule of backpropagation. Therefore, during deployment, we utilize this formulation to ensure that the output adheres to the constraints. However, during training, we backpropagate with the probabilities. This initially violates the constraint that each link can have at most one channel. However, through supervised learning, the model gradually learns to maximize the probability of selecting a single channel until it approaches 1 and consequently the other probabilities approach 0.

Regarding the power allocation, we first apply a sigmoid function to $X$. This transforms $X$ into values within the range of [0, 1]. We then compute the average of these values across the channel axes. Finally, we obtain the power allocation $p_i$ by multiplying the averaged value by $p_{max}$. The overall process is defined as follows:

$$p_i = \frac{1}{K}\sum_{k=1}^{K} \frac{p_{max}}{1 + e^{x_{ik}}} \tag{10}$$

It can be straighforwardly demonstrated that the values of $p_i$ falls always within the acceptable range, where $p_i \leq \frac{1}{K}(\sum_k 1) \cdot p_{max} \leq 1 \cdot p_{max}$.

Both the power vector and the channel allocation matrix originate from the same matrix $X$. Afterward, we apply different functions in the output layers to respect the constraints of each variable. Rather than employing distinct neural network blocks for each variable, we tackle the problem jointly. The majority of the model's parameters can be learned by optimizing a specific criterion. The following subsection will define the loss functions utilized to train our model.

### D. LOSS FUNCTION DESIGN

The loss function of our model is composed of three parts: a supervised segment, an unsupervised segment, and a regulation loss designed to maintain the required minimum data rate. Initially, the neural networks in the model leverage supervised learning to acquire a generalized strategy, drawing from the diverse array of solutions within our dataset generated by PYMOO. This dataset, rich in variety due to alterations in minimum data rates, the number of links, and link distance variations, equips the model with a wide-ranging understanding of potential wireless configurations. Such a broad perspective is crucial for the model's ability to effectively adapt and fine-tune to specific scenarios during the inference phase. Following the supervised learning stage, the networks further refine their ability to optimize the objective through unsupervised learning, all the while adhering to optimization constraints enforced by the regulation loss. This holistic approach ensures that the model not only learns generalizable strategies but also enhances its objective maximization capabilities and compliance with necessary constraints.

Specifically, power control is a supervised continuous prediction problem; thus, we employ the mean square error (MSE) to determine the prediction's cost function. Moreover, we consider the channel allocation as a multi-label supervised classification problem. Thus, we use the categorical cross-entropy (CCE) loss to calculate the cost of the sample's miss-classification.

$$\mathcal{L}_{sup}(\hat{P}, P, \hat{\Psi}, \Psi) = \sum_{i=0}^{N}\sum_{k=0}^{K} \left(\hat{p}_i - p_i\right)^2 + \psi_i^k \log(\hat{\psi}_i^k) \tag{11}$$

We adopt the negative network's mean rate as an unsupervised loss. The value of the loss function decreases when the data rate of each link, $\gamma_i$ increases.

$$\mathcal{L}_{rate}(\hat{\Psi}, \hat{P}) = -\sum_{i=0}^{N}\sum_{k=0}^{K} \frac{W\hat{\psi}_i^k}{N} \log_2\left(1 + \text{SINR}_i^k(\hat{p}_i^k, \hat{\psi}_i^k)\right) \tag{12}$$

In order to ensure that every link retains the necessary minimum capacity, we incorporate a regulation loss into our model, strategically managing the rate of each link. Contrary to the conventional approach that penalizes the model when rates drop below $\gamma_{\min}$ [31], our method imposes a penalty when certain rates are excessively elevated, as expressed in equation (13). This strategy pushes the model to generate rates as near to $\gamma_{\min}$ as possible. Consequently, while rates that are excessively high are brought down, those that are too low are also increased due to the shared radio resources. Mathematically,

$$\mathcal{L}_{\text{reg}}(\hat{\Psi}, \hat{P}) = \frac{1}{N}\sum_{i=0}^{N} \max(0, \gamma_i - \gamma_{\min}) \tag{13}$$

where $\mathcal{L}_{\text{reg}}$ computes the average extent to which each rate, $\gamma_i$, surpasses the minimum, $\gamma_{\min}$, considering only the excesses, due to the max function. This regulatory loss aims to steer the model to adhere closely to the minimum rate, preventing it from significantly exceeding it and thus ensuring a consistent rate output across all links. When paired with the maximization of the network mean rates, it lends appreciable stability to our model. We note that while supervised training is preferred, it is not mandatory. In the absence of a labeled dataset, the model can learn unsupervisedly.

### E. TRAINING AND DEPLOYMENT PROCESS

We employ supervised learning to guide the model toward acquiring an optimal initial and generalized strategy derived from the training dataset. During the deployment phase, we alter the training direction with the objective of enhancing the rates across all links and focusing on respecting the minimum data rate by minimizing a combination of the rate and regulation loss.

In both training and deployment, we first forward propagate to compute the prediction of $\hat{P}$ and $\hat{\Psi}$ to assess the loss value. Afterward, we back-propagate to calculate the gradients and update the model's weights accordingly using the Adadelta optimization technique [53]. For each epoch, we preprocess each sample's CSI and construct $K$ interference graphs. Then, we parallelly compute the node representations by evaluating (7) $T$ times for all the $K$ graphs. This parallel computation decreases the execution time by roughly $1/K$. Following that, we calculate $\hat{P}$ and $\hat{\Psi}$ accordingly. Lastly, we evaluate the loss and update the learnable weights of the model.

**Algorithm 1** Supervised Training Process Overview

```
INPUT: Dataset $\mathcal{D}$ contains tuples of $(G,P,\Psi)$
OUTPUT: Model's optimal parameters $\Theta$
Initialize $\hat{P}$, $\hat{\Psi}$
for Epoch in $[0$, MAX EPOCHS$]$ do:
    for all samples $(G_l, P_l, \Psi_l) \in \mathcal{D}$ $(l = 1, \dots, L)$ do:
        PreProcess $G_l$
        do in parallel: $(k = 0, \dots, K)$
            for all node $iin\mathcal{V}^k$ $(i = 1, \dots, N)$ do:
                $z_{ki}^0 \leftarrow g_{ii}$
                for t in $[0$,T$]$ do: Compute $z_{ki}^t$
        Compute $\hat{\Psi}$, $\hat{P}$
        Compute $\mathcal{L}_{sup}(\hat{\Psi}, \hat{P})$
        Update weights using $\nabla_\theta \mathcal{L}_{sup}(\hat{\Psi}, \hat{P})$
```

The supervised training procedure is thoroughly outlined in Algorithm 1. Meanwhile, Algorithm 2 provides a detailed description of the unsupervised deployment process, which, operates with a notably reduced number of iterations compared to the training phase.

While constructing the training dataset, we have the option of utilizing any MINLP optimization technique to obtain near-optimal solutions to our optimization problem. The exhaustive search is not feasible taking into account the number of possible solutions. Therefore in our case, we have chosen to utilize PYMOO [54] due to its widespread applicability and demonstrated efficacy in various fields. PYMOO provides a range of flexible genetic algorithm techniques that include evaluation features capable of assessing the solutions obtained, as well as parallel computation functionalities that serve to expedite the dataset construction process.

**Algorithm 2** Deployment Process Overview

```
INPUT: CSI $G$, pre-trained parameters $\Theta^*$
OUTPUT: $P^*$ and $\Psi^*$
for Epoch in $[0$, MAX EPOCHS$]$ do:
    PreProcess $G$
    Compute Z
    Compute $\hat{\Psi}$, $\hat{P}$ (8, 10)
    Compute $\mathcal{L}_{rate}(\hat{\Psi}, \hat{P}) + \mathcal{L}_{\text{reg}}(\hat{\Psi}, \hat{P})$
    Update weights using $\nabla_\theta(\mathcal{L}_{rate} + \mathcal{L}_{\text{reg}})$
Compute $\hat{\Psi}$ (9)
```

## V. PERFORMANCE EVALUATION

In this section, we evaluate the effectiveness of our proposed approach through a series of experiments and comparisons. Initially, we delve into a training convergence analysis, evaluating both supervised and unsupervised learning across various wireless network parameters. Subsequently, we showcase that our model outperforms in diverse wireless network setups by comparing it with the benchmarking schemes, focusing on network mean rate, QoS violation probability, and the level of QoS violation. Ultimately, we assess the robustness of our model in scenarios characterized by an imperfect channel.

### A. SIMULATION PARAMETERS

We construct a rectangular 2D layout with width $w_x = 200$ $m$ and height $w_y = 100$ $m$ that represents a wireless environment, and we randomly distributed the $N$ transmitters in the area. Consequently, we spread the receivers to be randomly distant from their corresponding transmitters in a range between $d_{\min}$ and $d_{\max}$. We adopted the channel model from the short-range outdoor model ITU-1411 with a distance-dependent path-loss [55], with 2.4 GHz carrier frequency, 1.5 m antenna height and 2.5 dBi antenna gain. The transmit power's maximum level is 4 dBm, and the background noise level is $-169$ dBm/Hz. We model the shadowing using the normal distribution, $\alpha_{ij}^k = 10^{\frac{S}{10}}, S \sim \mathcal{N}(0, \sigma)$ where $\sigma$ is the shadowing deviation in dB. We consider an urban outdoor environment, where $\sigma$ is between 4 dB and 12 dB. As for the fast fading channel, we use the Rayleigh fading model, $h_{ij}^k = R + jI; I, R \sim \frac{\mathcal{N}(0,1)}{\sqrt{2}}$. The number of channels is $K = 10$, where each has 500 Hz of bandwidth.

It is important to note that our model is **trained only once** with 10000 training samples over 30 epochs where $N = 50$, $d_{max} = 50$ m, $d_{min} = 5$ m, $\gamma_{min} = 200$ bps, and $\sigma = 4$ dB generated by PYMOO Single-Objective Optimization With Mixed Variables API [56]. We have also generated 1000 samples for testing the convergence of the supervised training following the same procedure. All the simulations and training have been conducted on the same hardware, processor Intel Intel(R) Xeon(R) W-1270 CPU, 16.0 GB memory, and 3.40 GHz. The code[1] is implemented using Python 3.9 with Deep Graph Learning library (DGL [57]) and PyTorch as a backend.

Regarding network parameters, we configure the number of GNN layers to $T = 4$, and the embedding dimension is established at $D = 10$. We design $\text{MLP}_1$ and $\text{MLP}_2$ with three hidden layers, between these layers, we apply a ReLU activation function and incorporated batch normalization. Our CNN block comprised 3 convolution filters, with ReLU activation functions interposed. The parameters for the convolution filters, including strides, padding, and dilation, were all set to 1. The kernel size is defined as $(3 \times 3)$, ensuring that the spatial dimensions of the input tensor remain unchanged. The channel input-output pairs are configured as $(10, 5, 2, 1)$. For the training and the deployment process, we opt for a fixed learning rate of 1.0 without any decay over epochs, as Adadelta would adapt it accordingly.

## B. COMPLEXITY ANALYSIS

Assuming sequential processing, the complexity of GNN encoders is $o(KTN[(N-1)\left((D+2)D+AD^2\right)+BD^2])$, where $A$ and $B$ denote the numbers of hidden layers in the message and update functions, respectively. Moreover, we process the sub-graphs in parallel, thus the complexity is divided by $K$. Given the parameters we set previously, the complexity is $o(240N(7N-2))$. Moreover, the CNN complexity is $o([\sum_{c\in\mathcal{C}} c_{i-1}c_i]9NK)$ where $\mathcal{C} = \{D, \ldots, 1\}$ is the set of consecutive number of channels, which in our case, is $o(588NK)$. Combining the two phases, the overall complexity of a single feed-forward is $o(240N(7N-2) + 558NK)$, which is roughly $o(N^2)$.

## C. BENCHMARKING SCHEMES

In our performance evaluation, we selected four distinct approaches for comparison: randomized, heuristic, convexification, and a learnable method. Since supervised training data can be scarce, we refer to our model in the following simulations as *GNN* when it is pre-trained in a supervised manner before deployment, and *DGNN* when it is not. We Benchmarking Schemes are explained as follows:

- **RANDOM:** We generate 40000 power and channel allocation solutions at random. Subsequently, we select the solution that minimizes QoS violations while maximizing the mean network rate.
- **PYMOO [56]:** The problem is addressed utilizing a genetic algorithm provided by PYMOO. Notably, this solution is identical to the one used to generate the training dataset.
- **SLSQP [58]:** Initially, greedy channel allocation [59] is assigned to all links. Following this, we resolve the power control task by employing the Sequential Least Squares Programming (SLSQP) technique.
- **DNN [31]:** The CSI is reshaped and fed into two DNN architectures. The first DNN architecture is responsible for power control, while the second DNN architecture handles channel allocation. However, due to the DNN architecture's inflexibility towards variations in the number of links, we adjust the number of neurons in the DNN to be suitable for the selected $N$ and retrain it with an adequate dataset.
- **REGNN [40]:** The problem is addressed using a Resilient GNN policy, trained using an unsupervised primal-dual approach. We average and normalize the CSI across resource blocks to construct a graph topology similar to the one used in the paper.

## D. TRAINING CONVERGENCE ANALYSIS

We present in this subsection convergence analyses for the supervised phase and the unsupervised phases in different wireless scenarios.

### 1) SUPERVISED CONVERGENCE

Fig. 4 and Fig. 5 illustrate the convergence patterns of the Mean Squared Error (MSE) and categorical Cross Entropy (CCE), respectively. Fig.6 shows the convergence pattern of the supervised (SUP) loss, which is the sum of both MSE and CCE losses. An exponential decrease is observed in the initial iterations, followed by the stabilization of the loss curves towards a minimum, which underscores the model's proficiency in emulating the RRM strategy inherent in the training dataset. After the 5000-th iteration, a plateau in the loss values is noticeable, suggesting that the model might have reached a state where further learning is limited and is potentially trapped in a local minimum. Meanwhile, Fig. 7 demonstrates the progression of the Average network rate with respect to the number of epochs (10000 iterations per epoch), emphasizing that the model continues its learning trajectory to align the performance between testing and training samples, hence toward generalization.

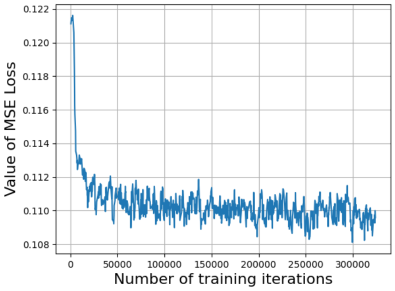


**FIGURE 4. Convergence of MSE loss.**

By the 30-th epoch, the test performance converges to the train performance, demonstrating that the model is adept at managing unseen samples. Such robust generalization capability can be linked to the permutation-invariant character of the GNN architecture. When trained on graphs derived from the CSI, the model naturally undergoes data augmentation, enhancing the generalization performance.

[1] https://github.com/mahermarwani/Graph-Neural-Networks-Approach-for-Joint-Wireless-Power-Control-and-Spectrum-Allocation

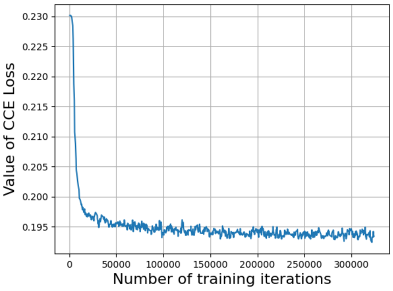


FIGURE 5. Convergence of CCE loss.

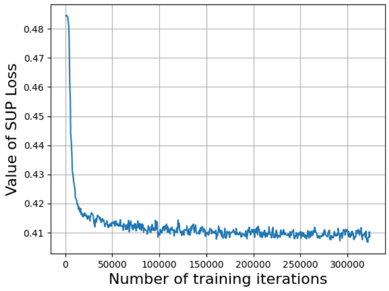


FIGURE 6. Convergence of SUP loss.

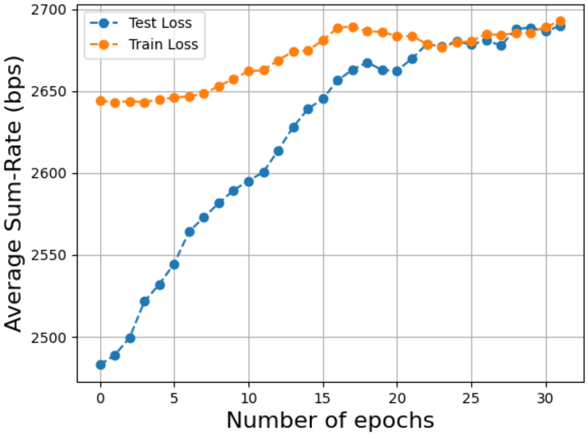


FIGURE 7. Training vs. test model's average sum rate evolution over supervised learning process.

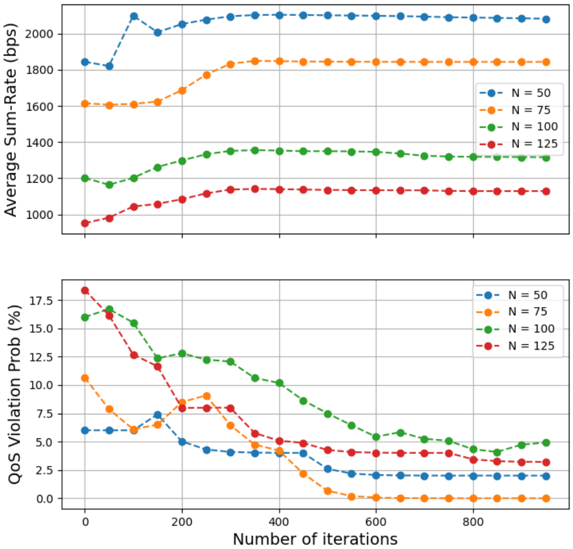


FIGURE 8. Model's average sum rate and QoS probability over unsupervised learning in different *N* values.

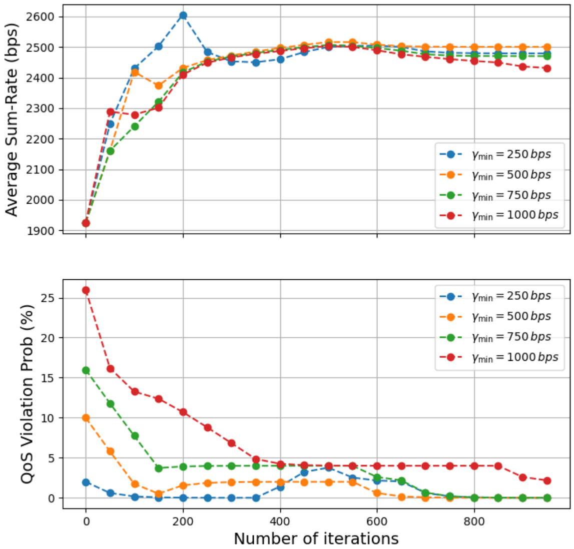


FIGURE 9. Model's average sum rate and QoS probability over unsupervised learning in different $\gamma_{min}$ values.

### 2) UNSUPERVISED CONVERGENCE

Supervised learning typically results in an upper-bound performance generated from the process used to create the training dataset, i.e., PYMOO. Thus, we demonstrate the impact of unsupervised learning to increase the model's performance by directly maximizing the mean network rate while minimizing the QoS violations. We conduct several tests across different wireless scenarios by adjusting the minimum required rate $\gamma_{min}$, the number of links $N$, the distribution of link locations $d_{max}$, and the shadowing deviation $\sigma$. For every parameter change, we generate 100 samples and analyze the average network mean rate evolution and QoS violation probability over the unsupervised learning iterations.

Fig. 8 illustrates the impact of varying the number of links on the average sum rate and QoS probability. We set $N = \{50, 75, 100, 125\}$, $d_{max} = 50$ m, $\gamma_{min} = 300$ bps, and $\sigma = 4$ dB. The top graph indicates a convergence trend in average sum rates for all the considered $N$ values, with a rise of around 20% after 1000 iterations. Higher $N$ values produce lower rates since the radio resources are finite and invariant. The bottom graph underscores the improvement in QoS adherence over the iterations. For instance, $N = 125$ decreases its initial violation probability from 17.5% to nearly 1%, while $N = 50$ reduces it from 5% to almost 0%. This demonstrates that our approach can achieve improved performances while maintaining strict QoS compliance for various numbers of links. Fig. 9, on the other hand, evaluates the model's performance against changing QoS values, specifically at $\gamma_{min} = \{250, 500, 750, 1000\}$ bps, $N = 50$, $d_{max} = 50$ m, and $\sigma = 4$ dB. We kept the same CSI in this analysis for a fair assessment. The Average Sum Rate displays an increased performance across the examined values, stabilizing near 2500 bps, a boost of approximately 31% from the initial supervised outcome. Meanwhile, the QoS Violation Probability segment reveals a sharp decrease, especially for the $\gamma_{min} = 1000$ bps curve, and by the 800-th iteration, all curves converge to under 5%, with most nearing zero. This observation underscores the model's aptitude to adjust to fluctuating QoS values, constantly optimizing rates and reducing QoS breaches.

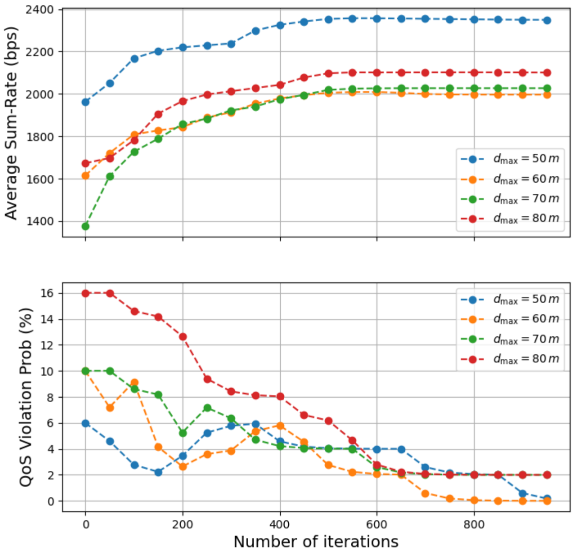


**FIGURE 10. Model's average sum rate and QoS probability over unsupervised learning in different $d_{max}$ values.**

Fig. 10 explores the effects of different link distances, where $d_{max} = \{50, 60, 70, 80\}$ m, $\gamma_{min} = 300$ bps, $N = 50$, and $\sigma = 4$ dB. Similarly, we observe a clear trend: the Average Sum Rate increases over iterations by 30 %, and higher $d_{max}$ values produce lower rates due to higher channel attenuations. However, although initially high, the QoS Violation Probability for all $d_{max}$ values declines rapidly, converging to 2% by the 800-th iteration. This emphasizes the model's increased performance versatility across varying link distances. Finally, Fig. 11 showcases the influence of shadowing, where $\gamma_{min} = 300$ bps, $N = 50$, $d_{max} = 50$ m, and $\sigma = \{4, 6, 9, 12\}$ dB. The top graph reveals that all curves consistently increase the Average Sum Rate as iterations continue. Higher values of $\sigma$ lead to slightly higher rates. This observation can be attributed to the fact that if shadowing results in a positive deviation (i.e., the signal strength is higher than expected), the SNR would increase, potentially leading to a higher Shannon capacity. However, it's important to note that this doesn't imply shadowing is inherently "beneficial." Rather, the random nature of shadowing can occasionally produce signal strengths that surpass deterministic predictions. The bottom graph, depicting QoS Violation Probability, suggests that the system is robust across diverse shadowing scenarios, with higher and lower $\sigma$ values witnessing significant violation reductions over iterations, reaching nearly 0 violations. This highlights the model's increased performance across varying shadowing deviations.

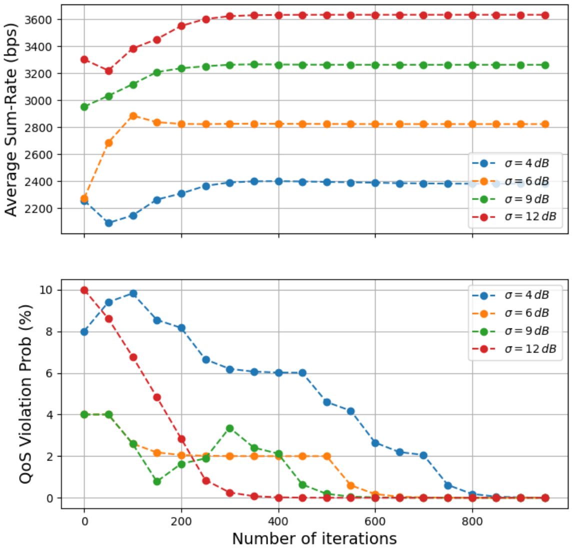
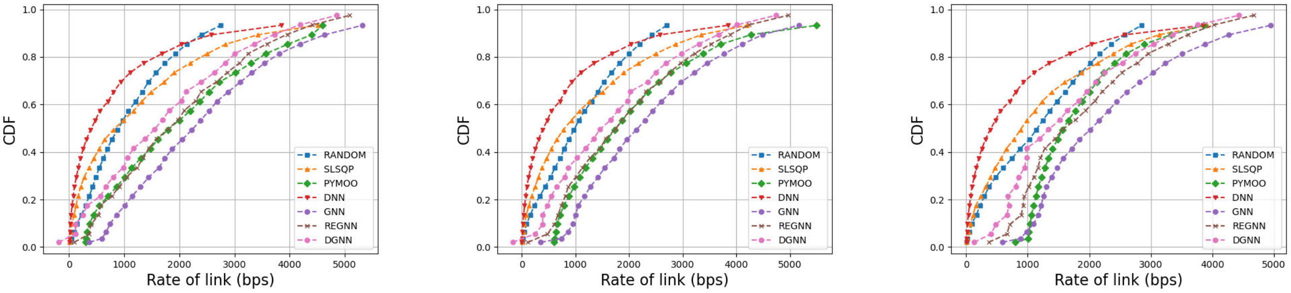


**FIGURE 11. Model's average sum rate and QoS probability over unsupervised learning in different $\sigma$ values.**

Overall, the supervised training provides a solid starting point, yet it doesn't achieve optimal results. Integrating unsupervised training during the deployment phase shows a notable improvement in the network's mean rate. Simultaneously, the QoS violation probability reduces across various wireless configurations. This underscores the effectiveness of our methodology in enhancing both performance and adaptability.

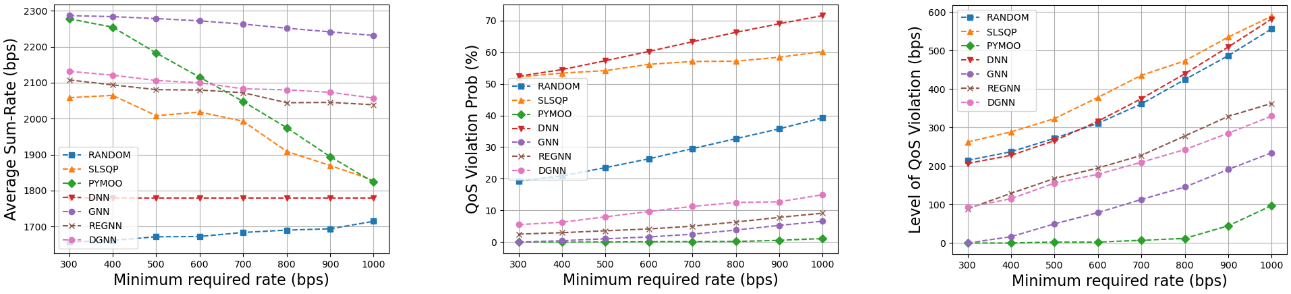
### *E. IMPACT OF QoS CONSTRAINTS ON PERFORMANCE*

This subsection examines the model's (GNN & DGNN) performance by changing the QoS values and comparing it with established benchmark schemes. For a fair comparison, the CSI remained the same while changing $\gamma_{min}$, as it is done during the convergence analysis. The Cumulative Distribution Function (CDF) plots presented in Fig. 12 provide a comparative analysis of the link rate performances of five benchmark schemes for $\gamma_{min} = \{300, 600, 1000\}$ bps. A noticeable rightward skew of the GNN curve in each plot highlights its capability to achieve higher link rates more often than other methods. Importantly, GNN consistently surpasses other schemes in every scenario, even DGNN which shows the importance of the supervised learning phase. This superiority is especially evident at extreme $\gamma_{min}$ values where most schemes find it challenging to uphold the QoS requirements. It's worth noting that consistently maintaining extreme $\gamma_{min}$ can be unfeasible for various



(a) Rates when $\gamma_{min} = 300\ bps$. (b) Rates when $\gamma_{min} = 600\ bps$. (c) Rates when $\gamma_{min} = 1000\ bps$.

**FIGURE 12.** **Rate's CDF of benchmarking schemes for different $\gamma_{min}$ values.**



(a) Average Sum-Rate of all links. (b) Probability of QoS violation. (c) Level of QoS violation.

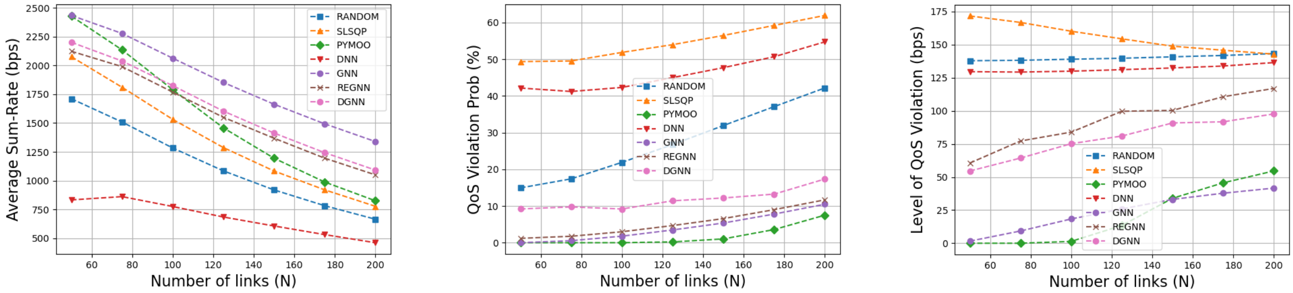
**FIGURE 13.** **Performance of benchmarking schemes with respect to $\gamma_{min}$.**

channel realizations. Our model, as shown in (c), registers approximately 5% QoS violations, whereas other schemes hover around 50%, except for the PYMOO and REGNN approach.

Next, we show the average sum rate of the links, the probability of QoS constraint violation, and the level of QoS violation, which is defined as the difference between the QoS requirement and the rate when the QoS constraint is violated, i.e., $\mathbb{E}[\gamma_{min} - \gamma \mid \gamma_{min} > \gamma]$, as a function of $\gamma_{min}$ in Figs. 13(a-c), respectively. In Fig. 13(a), the average sum rate is observed to decrease as $\gamma_{min}$ increases. This decline is attributable to the restrictions on all links' transmit power and channel usage, which are necessary to minimize interference and, thus, to meet the QoS constraints. Notably, the proposed scheme maintains superior performance even as its average sum rate marginally declines. Moreover, this behavior widens the performance disparity with the PYMOO scheme as $\gamma_{min}$ escalates. In contrast, other benchmark schemes consistently underperform relative to the proposed strategy. Regarding Fig. 13(b), the QoS violation probability for our model remains commendably low. Specifically, it hovers close to 0 when $\gamma_{min}$ is minimal and reaches approximately 5% at higher $\gamma_{min}$ values. This performance is comparable with the PYMOO, REGNN, and DGNN approaches. In contrast, the RANDOM, SLSQP, and DNN schemes exhibit substantially elevated violation probabilities, underscoring the efficacy of the proposed scheme. Lastly, Fig. 13(c) highlights the proposed scheme's performance in terms of QoS violation levels. Although the QoS constraint may occasionally be breached, the deviation of the link's rate from $\gamma_{min}$ is still minimal (around 10%). This ensures that even when violations occur, their impact remains largely inconsequential.

### F. IMPACT OF THE NUMBER OF LINKS

In Figs. 14(a) - 14(c), we illustrate the average sum rate of the network, the probability of QoS violation, and the level of QoS violation against the number of links, $N$. Fig. 14(a) highlights that as the value of $N$ ascends from 50 to 200, there is a marked decrease in the average sum rate across all benchmark schemes. This downward trend can be attributed to the intensified competition for available radio resources, resulting from the addition of links to the network. Other benchmark schemes consistently fall behind our proposed scheme by margins of $15 - 30\%$, highlighting the superiority of our proposed model. On the other hand, we observe in Fig. 14(b) and 14(c) a modest surge in both the probability of QoS violation and its level as $N$ proliferates. Despite this trend, our proposed scheme consistently outperforms other benchmark schemes, registering less than a 10% probability of violation and maintaining the level of violation below 50 bps at higher $N$ values. These metrics restate the scheme's adeptness at maintaining QoS requirements, even in denser networks. These tests highlight the scalability of our GNN-based model. It consistently delivers robust performance across expansive networks without the need for retraining. This characteristic emphasizes its capability to generalize to larger graphs, while trained only on smaller graphs.



(a) Average Sum-Rate of all links. (b) Probability of QoS violation. (c) Level of QoS violation.

**FIGURE 14.** **Performance of benchmarking schemes with respect to *N*.**

In contrast, the DNN approach, even with retraining at each distinct $N$ value, continually lags behind, revealing its inherent limited scalability.

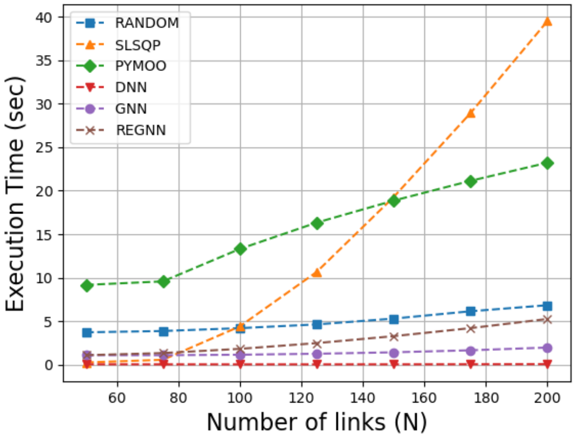


**FIGURE 15.** **Execution time for different *N* values.**

In Fig. 15, we assess the computation time required for resource allocation as a function of $N$. The simulation results focus solely on the computation time of the deployment phase, mainly because the GNN's training occurs just once and can be conducted offline prior to its actual implementation. It is essential to note that the execution time is intrinsically linked to our hardware specifications, leading to results being presented in seconds instead of milliseconds. Analyzing the data from Fig. 15, it is evident that the computation time for SLSQP and PYMOO increases almost linearly. This trend suggests that these methods may not be efficiently scalable for real-time operations. Contrarily, our method's computation time is marginally more than the DNN's, attributed to the unsupervised iterations involved, with the benefit of a significant increase in the overall performance. Moreover, the computation time remains unaffected by $N$. This is because the size of the GNN remains constant, irrespective of the number of links. For $N = 200$, the computation time stands at 2.3 seconds. However, this duration can be substantially reduced with superior hardware, indicating the viability of our proposed scheme for more extensive networks. Furthermore, incorporating parallel computing in graph embedding, stemming from our graph modeling, has decreased the execution time by a factor of $K$. As a result, considering a higher $K$ value renders our approach much more efficient than the other schemes.

### G. IMPACT OF THE LINKS LOCATIONS

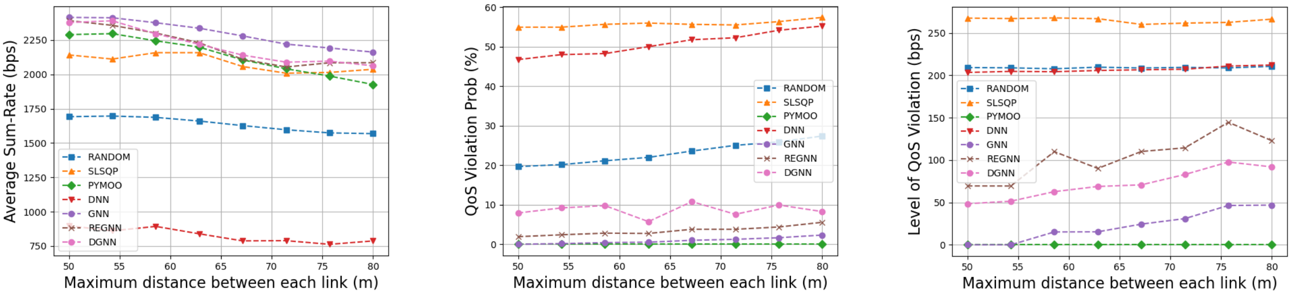
In Figs. 16(a) - 16(c), we illustrate the average sum rate of the network, the probability of QoS violation, and the level of QoS violation against the maximum distance between each link, $d_{max}$. In Fig. 16(a), the GNN curve starts at approximately 2200 bps at 50 $m$ and sees a steady decline, settling just above 2000 bps by 80 $m$. Notably, the GNN approach consistently surpasses the performance of the RANDOM, DNN, REGNN, and PYMOO algorithms across the entire distance range. Moving to Fig. 16(b), GNN demonstrates remarkable stability, ensuring a violation probability below 5% throughout all distances. This stability sets it apart from other methods, particularly DNN and SLSQP, the latter of which sees a hovering rate near 50%. Lastly, Fig. 16(c) underscores GNN's efficiency, with its curve initiating at around 50 bps at 50 $m$ and registering a slight surge to roughly 70 bps by 80 $m$. In this context, the GNN outperforms most other schemes, with the exception of PYMOO, which is precisely engineered to be resilient against optimization constraints at the expense of an extensive execution time.

### H. IMPACT OF THE FADING EFFECTS

In Figs. 17(a) - 17(c), we demonstrate the average sum rate of the network, the probability of QoS violation, and the level of QoS violation against the shadowing deviation, $\sigma$. Among the tested schemes, the GNN consistently outperforms its counterparts, achieving the highest average sum rate across all shadowing deviations, as seen in Fig. 17(a). Furthermore, when it comes to ensuring the quality of service, the GNN demonstrated resilience, exhibiting the lowest probability and level of QoS violations, as shown in Fig. 17(b-c). This consistent superiority of the GNN emphasizes its potential as a highly reliable solution in environments with varying shadowing deviations.

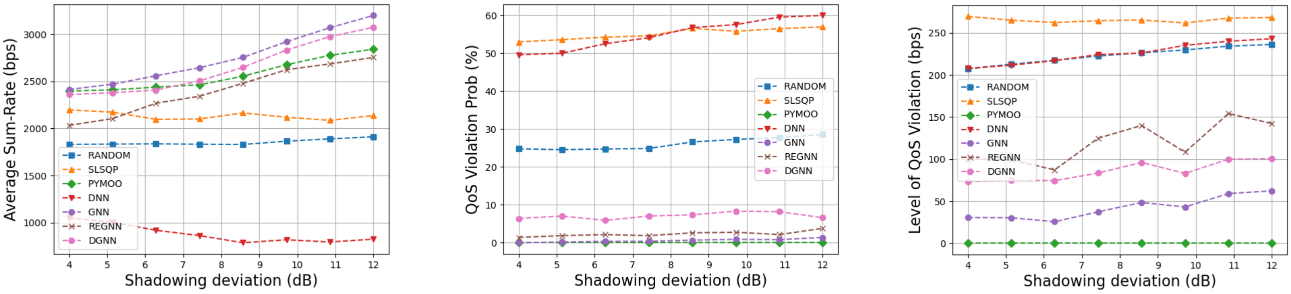
### I. IMPACT OF NOISY CSI

In this subsection, we evaluate the resilience of our model when subjected to channel imperfections. These imperfec-



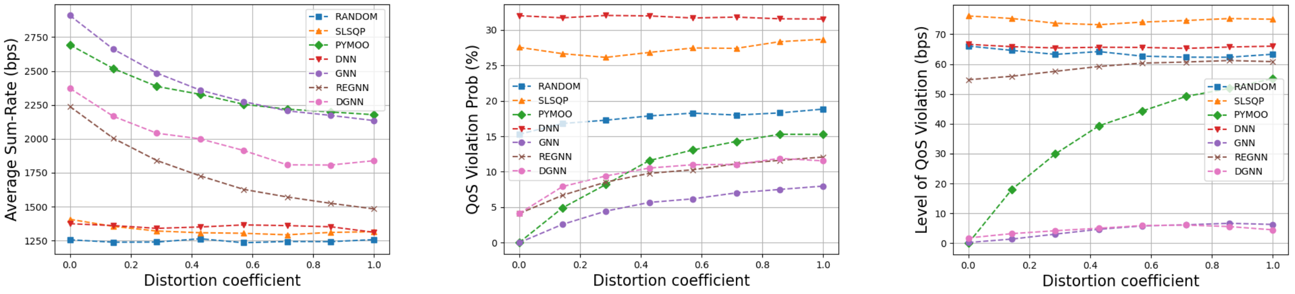
(a) Average Sum-Rate of all links. (b) Probability of QoS violation. (c) Level of QoS violation.

**FIGURE 16. Performance of benchmarking schemes with respect to $d_{max}$.**



(a) Average Sum-Rate of all links. (b) Probability of QoS violation. (c) Level of QoS violation.

**FIGURE 17. Performance of benchmarking schemes with respect to $\sigma$.**



(a) Average Sum-Rate of all links. (b) Probability of QoS violation. (c) Level of QoS violation.

**FIGURE 18. Performance of benchmarking schemes with respect to $\sigma_e$.**

tions are quantified using a specific expression that defines the discrepancy between the estimated and actual multipath fading effects. The relationship between the estimated multipath fading values, $h_{ij}^k$, and their true counterparts, $\tilde{h}_{ij}^k$, is described by the first-order Gauss-Markov process [60], expressed as follows:

$$\tilde{h}_{ij}^k = \sqrt{1-\sigma_e^2} h_{ij}^k + \sigma_e n_{ij}^k \tag{14}$$

Here, $n_{ij}^k$ represents the error associated with the estimated channel, $h_{ij}^k$, which adheres to a complex Gaussian distribution. The error coefficient $\sigma_e$ characterizes the precision of the CSI, where $\sigma_e$ ranges from 0 to 1. A smaller value of $\sigma_e$ indicates a higher CSI accuracy, approaching perfect accuracy as $\sigma_e$ tends to zero. Consequently, we can express the noisy channel with the following equation:

$$\tilde{g}_{ij}^k = \beta_{ij}^k \alpha_{ij}^k |\tilde{h}_{ij}^k|^2 \tag{15}$$

To test the robustness, we first generate a clean CSI. We then apply Equation (15) to introduce varying degrees of noise to the clean CSI by adjusting $\sigma_e$ within its defined range. Using the noise-inflicted CSI, we proceed to determine the power and channel allocation solutions for all benchmark schemes, and our approach. Upon finalizing these solutions, we calculate the actual link rates, employing the clean CSI. This method allows us to measure the effectiveness of our model in real-world conditions where channels are infrequently perfect.

Fig. 18(a) to Fig. 18(c) demonstrate that the GNN scheme excels in performance against a rising distortion coefficient, $\sigma_e$. In graph (a), all benchmark schemes experience a reduced average sum rate as $\sigma_e$ increases due to the distortion, yet the GNN maintains the highest rates, indicating a strong resistance to distortion. Graph (b) reveals that the GNN's probability of QoS violations stays under 10%, contrasting with the marked vulnerability of other schemes under the same conditions. Graph (c) shows the GNN's QoS violation level remains below 10 bps, surpassing by far other schemes, which worsen with higher $\sigma_e$. GNN's consistent robustness, attributed to its permutation invariant features, showcases its superior design in mitigating distortion and preserving service quality.

## VI. CONCLUSION

In conclusion, we presented a novel GNN-based framework for jointly solving power control and spectrum allocation in a non-orthogonal wireless environment. Our approach demonstrated superior performance in terms of average sum rate and Qos preservation in different wireless setups compared to other heuristic and learnable approaches and achieved robustness over an imperfect channel. Additionally, our approach exhibited scalability, stability, and generalization, making it suitable for various network structures with different setups, such as D2D networks and Downlink-Uplink cellular scenarios. This study establishes a foundation for advanced RRM in future wireless networks. Future research should delve into GNNs' capabilities for dynamic spectrum allocation, interference management, and network optimization in changing conditions. Additionally, in-depth theoretical analysis is needed to pinpoint the best graph representations of wireless networks and fine-tune the GNN embedding layer. Lastly, incorporating temporal dynamics into GNN training could further improve RRM outcomes.

## REFERENCES

[1] Y. Zhao, J. Zhao, W. Zhai, S. Sun, D. Niyato, and K. Lam, "A survey of 6G wireless communications: Emerging technologies," in *Advances in Information and Communication* (Advances in Intelligent Systems and Computing), vol. 1363, K. Arai, Ed. Cham, Switzerland: Springer, 2021.

[2] M. Alsabah et al., "6G wireless communications networks: A comprehensive survey," *IEEE Access*, vol. 9, pp. 148191–148243, 2021.

[3] C. D. Alwis et al., "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 836–886, 2021.

[4] M. Moussaoui, E. Bertin, and N. Crespi, "5G shortcomings and beyond-5G/6G requirements," in *Proc. 1st Int. Conf. 6G Netw. (6GNet)*, Jul. 2022, pp. 1–8.

[5] T. Akhtar, C. Tselios, and I. Politis, "Radio resource management: Approaches and implementations from 4G to 5G and beyond," *Wireless Netw.*, vol. 27, no. 1, pp. 693–734, Jan. 2021, doi: 10.1007/s11276-020-02479-w.

[6] F. Qamar, M. U. A. Siddiqui, M. N. Hindia, R. Hassan, and Q. N. Nguyen, "Issues, challenges, and research trends in spectrum management: A comprehensive overview and new vision for designing 6G networks," *Electronics*, vol. 9, no. 9, p. 1416, Sep. 2020.

[7] J. Huang, S. Huang, C.-C. Xing, and Y. Qian, "Game-theoretic power control mechanisms for device-to-device communications underlaying cellular system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 4890–4900, Jun. 2018.

[8] A. Ramezani-Kebrya, M. Dong, B. Liang, G. Boudreau, and S. H. Seyedmehdi, "Joint power optimization for device-to-device communication in cellular networks with interference control," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5131–5146, Aug. 2017.

[9] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2133–2168, 3rd Quart., 2018.

[10] S. Gupta, R. Patel, R. Gupta, S. Tanwar, and N. Patel, "A survey on resource allocation schemes in device-to-device communication," in *Proc. 12th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2022, pp. 140–145.

[11] Z. Lin and Y. Liu, "Joint uplink and downlink transmissions in user-centric OFDMA cloud-RAN," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7776–7788, Aug. 2019.

[12] M. K. Shehata, S. M. Gasser, H. M. El-Badawy, and M. E. Khedr, "Optimized dual uplink and downlink resource allocation for multiple class of service in OFDM network," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2015, pp. 597–601.

[13] D. H. N. Nguyen, L. B. Le, and Z. Han, "Optimal uplink and downlink channel assignment in a full-duplex multiuser system," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[14] R. Ruby, S. Zhong, H. Yang, and K. Wu, "Enhanced uplink resource allocation in non-orthogonal multiple access systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1432–1444, Mar. 2018.

[15] C. Kai, L. Xu, J. Zhang, and M. Peng, "Joint uplink and downlink resource allocation for D2D communication underlying cellular networks," in *Proc. 10th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, Oct. 2018, pp. 1–6.

[16] Y. Pan, C. Pan, Z. Yang, and M. Chen, "Resource allocation for D2D communications underlaying a NOMA-based cellular network," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 130–133, Feb. 2018.

[17] P. Mach, Z. Becvar, and M. Najla, "Resource allocation for D2D communication with multiple D2D pairs reusing multiple channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1008–1011, Aug. 2019.

[18] M. Najla, Z. Becvar, and P. Mach, "Reuse of multiple channels by multiple D2D pairs in dedicated mode: A game theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4313–4327, Jul. 2021.

[19] S. Liu, Y. Wu, L. Li, X. Liu, and W. Xu, "A two-stage energy-efficient approach for joint power control and channel allocation in D2D communication," *IEEE Access*, vol. 7, pp. 16940–16951, 2019.

[20] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[21] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "LORM: Learning to optimize for resource management in wireless networks with few training samples," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 665–679, Jan. 2020.

[22] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1760–1776, Mar. 2020.

[23] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, Oct. 2018.

[24] W. Lee, M. Kim, and D. Cho, "Deep power control: Transmit power control scheme based on convolutional neural network," *IEEE Commun. Lett.*, vol. 22, no. 6, pp. 1276–1279, Jun. 2018.

[25] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.

[26] S. He et al., "An overview on the application of graph neural networks in wireless networks," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 2547–2565, 2021.

[27] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: Architectures, stability, and transferability," *Proc. IEEE*, vol. 109, no. 5, pp. 660–682, May 2021.

[28] M. Lee, G. Yu, and G. Y. Li, "Learning to branch: Accelerating resource allocation in wireless networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 958–970, Jan. 2020.

[29] Z. Zhang and M. Tao, "Learning-based branch-and-bound for non-convex complex modulus constrained problems with applications in wireless communications," *IEEE Trans. Wireless Commun.*, vol. 21, no. 6, pp. 3752–3763, Jun. 2022.

[30] T. Lin and Y. Zhu, "Beamforming design for large-scale antenna arrays using deep learning," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 103–107, Jan. 2020.

[31] W. Lee and R. Schober, "Deep learning-based resource allocation for device-to-device communication," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5235–5250, Jul. 2022.

[32] K. Nakashima, S. Kamiya, K. Ohtsu, K. Yamamoto, T. Nishio, and M. Morikura, "Deep reinforcement learning-based channel allocation for wireless lans with graph convolutional networks," *IEEE Access*, vol. 8, pp. 31823–31834, 2020.

[33] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.

[34] J. Guo and C. Yang, "Learning power control for cellular systems with heterogeneous graph neural network," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–6.

[35] X. Zhang, H. Zhao, J. Xiong, X. Liu, L. Zhou, and J. Wei, "Scalable power control/beamforming in heterogeneous wireless networks with graph neural networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[36] V. Ranasinghe, N. Rajatheva, and M. Latva-aho, "Graph neural network based access point selection for cell-free massive MIMO systems," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2021, pp. 1–6.

[37] X. Zhang, Z. Zhang, and L. Yang, "Joint user association and power allocation in heterogeneous ultra dense network via semi-supervised representation learning," 2021, *arXiv:2103.15367*.

[38] A. Chowdhury, G. Verma, C. Rao, A. Swami, and S. Segarra, "Efficient power allocation using graph neural networks and deep algorithm unfolding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 4725–4729.

[39] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[40] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Learning resilient radio resource management policies with graph neural networks," *IEEE Trans. Signal Process.*, vol. 71, pp. 995–1009, 2023.

[41] Y. Wang, Y. Li, Q. Shi, and Y.-C. Wu, "ENGNN: A general edge-update empowered GNN architecture for radio resource management in wireless networks," *IEEE Trans. Wireless Commun.*, p. 1, 2023, doi: 10.1109/TWC.2023.3325735.

[42] Z. Wang, M. Eisen, and A. Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 1850–1863, 2022.

[43] Z. He, L. Wang, H. Ye, G. Y. Li, and B. F. Juang, "Resource allocation based on graph neural networks in vehicular communications," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–5.

[44] Y. Gu, C. She, Z. Quan, C. Qiu, and X. Xu, "Graph neural networks for distributed power allocation in wireless networks: Aggregation over-the-air," *IEEE Trans. Wireless Commun.*, vol. 22, no. 11, pp. 7551–7564, Nov. 2023.

[45] Y. Shen, J. Zhang, S. H. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Trans. Wireless Commun.*, vol. 22, no. 5, pp. 3554–3569, May 2023.

[46] N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "State-augmented learnable algorithms for resource management in wireless networks," *IEEE Trans. Signal Process.*, vol. 70, pp. 5898–5912, 2022.

[47] P. Cheng, G. Chen, and Z. Han, "Graph neural networks based resource allocation in heterogeneous wireless networks," in *Proc. 7th Int. Conf. Intell. Inf. Process.* New York, NY, USA: Association for Computing Machinery, 2023, doi: 10.1145/3570236.3570293.

[48] T. Chen, X. Zhang, M. You, G. Zheng, and S. Lambotharan, "A GNN-based supervised learning framework for resource allocation in wireless IoT networks," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1712–1724, Feb. 2022.

[49] Y. Peng, J. Guo, and C. Yang, "Learning resource allocation policy: Vertex-GNN or edge-GNN?" *IEEE Trans. Mach. Learn. Commun. Netw.*, vol. 2, pp. 190–209, 2024.

[50] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Jan. 2021.

[51] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, in Proceedings of Machine Learning Research, vol. 70, D. Precup and Y. W. Teh, Eds., 2017, pp. 1263–1272. [Online]. Available: https://proceedings.mlr.press/v70/gilmer17a.html

[52] Z. Zhang, P. Cui, and W. Zhu, "Deep learning on graphs: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 249–270, Jan. 2022.

[53] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," 2012, *arXiv:1212.5701*.

[54] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[55] (2009). *Propagation Data and Prediction Methods for the Planning of Short-range Outdoor Radiocommunication Systems and Radio Local Area Networks in the Frequency Range 300 Mhz to 100 Ghz*. [Online]. Available: https://api.semanticscholar.org/CorpusID

[56] J. Blank and K. Deb, "PYMOO: Multi-objective optimization in Python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.

[57] M. Wang et al., "Deep graph library: A graph-centric, highly-performant package for graph neural networks," 2019, *arXiv:1909.01315*.

[58] J. Nocedal and S. J. Wright, *Numerical Optimization*. Cham, Switzerland: Springer, 1999.

[59] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4281–4298, Jun. 2019.

[60] C. He, C. Tian, C. Zhang, D. Feng, C. Pan, and F.-C. Zheng, "Energy efficiency optimization for distributed antenna systems with D2D communications under channel uncertainty," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 4, pp. 1037–1047, Dec. 2020.

**MAHER MARWANI** (Student Member, IEEE) received the B.S. degree in engineering from École Polytechnique de Tunisie (EPT), Tunisia, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering with École de Technologie Supérieure. His research interests include intelligent solutions for radio resource management (RRM) and leveraging advancements in graph neural networks (GNNs).

**GEORGES KADDOUM** (Senior Member, IEEE) received the bachelor's degree in electrical engineering from École Nationale Supérieure de Techniques Avancées (ENSTA Bretagne), Brest, France, the M.S. degree in telecommunications and signal processing (circuits, systems, and signal processing) from Université de Bretagne Occidentale and Telecom Bretagne (ENSTB), Brest, in 2005, and the Ph.D. degree (Hons.) in signal processing and telecommunications from the National Institute of Applied Sciences (INSA), University of Toulouse, Toulouse, France, in 2009. He is currently a Professor and the Research Director of the Resilient Machine Learning Institute (ReMI), and the Tier 2 Canada Research Chair of École de Technologie Supérieure (ÉTS), Université du Québec, Montréal, Canada. He has published more than 300 journal articles, conference papers, and two chapters in books, and has eight pending patents. His research interests include wireless communication networks, tactical communications, resource allocations, and network security. He received the Best Papers Award from the 2014 IEEE International Conference on Wireless and Mobile Computing, Networking, Communications (WIMOB), the 2017 IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), and the 2023 IEEE International Wireless Communications and Mobile Computing Conference (IWCMC). He received the IEEE Transactions on Communications Exemplary Reviewer Award in 2015, 2017, and 2019. He received the Research Excellence Award from Université du Québec in 2018. In 2019, he received the Research Excellence Award from ÉTS in recognition of his outstanding research outcomes. He also won the 2022 IEEE Technical Committee on Scalable Computing (TCSC) Award for Excellence (Middle Career Researcher). He has received the prestigious 2023 MITACS Award for Exceptional Leadership. He served as an Associate Editor for IEEE Transactions on Information Forensics and Security and IEEE Communications Letters. He is also serving as an Area Editor for IEEE Transactions on Machine Learning in Communications and Networking and an Editor for IEEE Transactions on Communications.