

A GNN based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks

Tianrui Chen, Xinruo Zhang, *Member, IEEE*, Minglei You, Gan Zheng, *Fellow, IEEE*,
and Sangarapillai Lambotharan, *Senior Member, IEEE*

Abstract—The Internet of Things (IoT) allows physical devices to be connected over the wireless networks. Although device-to-device (D2D) communication has emerged as a promising technology for IoT, the conventional solutions for D2D resource allocation are usually computationally complex and time-consuming. The high complexity poses a significant challenge to the practical implementation of wireless IoT networks. A graph neural network (GNN) based framework is proposed to address this challenge in a supervised manner. Specifically, the wireless network is modeled as a directed graph, where the desirable communication links are modeled as nodes and the harmful interference links are modeled as edges. The effectiveness of the proposed framework is verified via two case studies, namely the link scheduling in D2D networks and the joint channel and power allocation in D2D underlaid cellular networks. Simulation results demonstrate that the proposed framework outperforms the benchmark schemes in terms of the average sum rate and the sample efficiency. It shows potential generalization to different system settings and robustness to corrupted input features. Besides, the proposed GNN approach accelerates the execution time of the conventional optimization algorithms to only a few milliseconds.

Index Terms—Resource allocation, Graph neural network (GNN), Link scheduling, Device-to-device (D2D), Internet of Things (IoT)

I. INTRODUCTION

Device-to-device (D2D) communication is considered as a key enabling technology for the Internet of Things (IoT) ecosystem, where the devices communicate with each other directly without the essential interventions of the central agents such as base stations (BSs) and access points (APs) [1]. The main advantages of D2D communications include the efficient use of network resources with traffic offloading and the robustness against single point of failure [2]. However, resource allocation problems in D2D communications such as channel allocation [3] and link scheduling [4] that involve integer variables are usually challenging to obtain global optimal solutions. Conventional algorithms such as the branch-and-bound (B&B) algorithm [5] are time-consuming and of high computational complexity. Hence these global optimization algorithms are usually inappropriate for solving practical problems in wireless IoT networks. Therefore, many studies in the literature focus on the sub-optimal algorithms that reduce

the computational complexity while achieving near-optimal results. The authors in [4] proposed a fractional programming based design called FPLinQ to find the sub-optimal solutions to the link scheduling problem in D2D communications. The authors in [6] demonstrated that the convex optimization based beamformer design can be efficiently implemented via approximation solutions. In recent years, the sub-optimal cross-entropy (CE) algorithm was proposed to solve the resource allocation problems such as joint antenna selection problem [7] and cache content placement problem [8] in wireless networks.

More recently, machine learning (ML) techniques have been introduced to solve the resource allocation problems in wireless communications which have the ability to accelerate the execution time of algorithms [9] [10]. The authors in [11] proposed a framework named learning to optimize for resource management (LORM) to accelerate the optimal pruning policy in the B&B algorithm for the mixed integer nonlinear programming problems, and verified it via a network power minimization problem in cloud radio access networks. In [12], an imitation learning method was proposed to accelerate the B&B algorithm for resource allocation in D2D underlaid cellular networks. Two ML techniques including classification and regression were utilized in [13] to speed up the generalized benders decomposition algorithm for wireless resource allocation. Although existing works (e.g., [11] - [13]) have made great efforts to accelerate the conventional algorithms, these techniques can reach at most 10^{-2} second completion time, which is still far longer than the millisecond level real-time requirement [14] in wireless networks.

Although the ML based schemes can improve the time complexity performance in wireless communication design problems, the integration of wireless network topologies is still a challenge. Fortunately, the graph theory can be adopted to address this challenge due to the natural similarities of topologies between the wireless networks and the graphs. Graph coloring algorithms have been successfully applied to solve the resource allocation tasks in femtocell networks [15], D2D communication in the long term evolution system [16] and D2D communication in cellular networks [17]. Besides, graph embedding techniques can be applied to node, edge and graph related applications, such as node classification and link scheduling [18]. The combination of the graph theory and the ML technologies has brought a lot of attention in the wireless research community, as it benefits from both the graph properties and performance acceleration [19] [20]. [An edge learning was considered in \[21\] with distributed implementation and the energy efficient federated learning](#)

Tianrui Chen, Minglei You, Gan Zheng and Sangarapillai Lambotharan are with the Signal Processing and Networks Research Group in the Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, LE11 3TU, UK (Email: {t.chen, m.you, g.zheng, s.lambotharan}@lboro.ac.uk), and Xinruo Zhang is with the School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ, UK (Email: xinruo.zhang@essex.ac.uk)

was studied over wireless communication networks. A spatial convolution method was proposed in [19] to solve a D2D link scheduling problem, wherein the convolution operation is applied to the density grid which is quantified based on the numbers of transmitters and receivers in each grid. Their proposed method, nevertheless, requires a large number of training samples. The acquisition of large training dataset in real-world wireless networks is expensive or even impractical. Accordingly, a graph embedding method with a multi-layer classifier was proposed in [20] to address this problem, where each D2D pair is represented by a low-dimensional vector with the features from itself and its neighbors, and only hundreds of training samples are required. However, both [19] and [20] only take distances into consideration and are not compatible with channel information, which may lead to performance degradation in small scale fading environment. Graph neural networks (GNNs) that have been proven to be successful in a wide range of applications including computer vision, natural language processing and chemistry [22], can effectively exploit non-Euclidean data such as channel state information (CSI). In [23], an interference graph convolutional neural network was proposed to learn the optimal power control in an unsupervised manner in a K-user interference channel, where the instantaneous CSI was incorporated. It was extended to solve the radio resource management problems through a message passing graph neural network in [24], and the proposed method was tested on both power control and beamforming design problems. Additionally, a random edge graph neural network was proposed in [25] to solve the power optimization problems in wireless ad-hoc networks and cellular networks. However, their proposed designs [23] - [25] are limited to homogeneous wireless systems and may not be compatible with the heterogeneous IoT systems. Besides, these works only studied continuous optimization problems and their proposed approaches may not handle discrete optimization problems. Different from them, our work proposes a general framework focusing on discrete resource optimization problems and it performs well for both homogeneous and heterogeneous networks.

Inspired by the previous works, a GNN based framework is proposed to tackle the resource allocation problems in wireless IoT networks in a supervised manner in this paper. The proposed framework has a layer-wise structure combining the convolutional neural network (CNN) with an average operation and the deep neural network (DNN) to aggregate and combine feature information iteratively. The main contributions are summarized as follows:

- The wireless IoT networks are modeled as directed graphs, where the communication links and interference links are treated as nodes and edges, respectively. A GNN based framework is proposed to solve several resource allocation problems involving integer parameters in wireless networks, where each node iteratively aggregates feature information from its adjacent nodes and edges, and combines its own feature with the aggregated information. The constrained cross-entropy (CCE) algorithm is employed for sample generation to further reduce the

computational complexity.

- The proposed framework is verified using two resource allocation problems, namely the link scheduling problem in the D2D networks, and the joint channel and power allocation for D2D underlaid cellular networks. The proposed framework is compared to three benchmark schemes: the unsupervised GNN [23], the graph embedding method [20] and the conventional DNN. Simulation results demonstrate that this framework outperforms the benchmark designs and maintains a stable end performance with various system settings and network scales.
- The proposed framework is sample efficient since it achieves near-optimal results with only hundreds of training samples. Besides, the execution time of solving the considered resource allocation problems is reduced to a few milliseconds by the proposed GNN based framework, making it attractive for real-time implementation of wireless IoT systems. Furthermore, simulation results suggest that it has good generalizability to different system settings such as pairwise distances and network sizes without further training. **It is also robust to missing input features.**

The remainder of this paper is organized as follows. Section II introduces a generalized resource allocation problem in wireless IoT networks. The proposed GNN based framework for resource allocations in wireless networks is presented in Section III, which includes a CE based algorithm for training samples generation, a graph modeling of wireless networks, and a GNN that is operated in a supervised manner. Sections IV and V present two applications of the proposed framework and their performance. Finally, conclusions are drawn in Section VI.

II. A GENERALIZED RESOURCE ALLOCATION PROBLEM

Many resource allocation problems such as link scheduling and channel selection problems in wireless IoT networks can be formulated as a discrete optimization problem, which is usually difficult to find the optimal solutions. A general formulation of this kind of problems can be written as follows,

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}) \\ & \text{subject to } x_i \in \mathbb{N}, \\ & \quad g_n(\mathbf{x}) \leq 0, \end{aligned} \quad (1)$$

where $f(\cdot)$ represents an objective function that measures the system performance such as the network capacity and the outage probability. $\mathbf{x} = \{x_i\}$ denotes the discrete optimization variable, which indicates the decision of the resource allocation, such as user association or channel allocation in the IoT networks. x_i denotes the i -th element of the optimization variables \mathbf{x} and \mathbb{N} refers to a set of non-negative integers. Besides, $g_n(\mathbf{x}) \leq 0$ represents a series of constraints involving the discrete variable \mathbf{x} , e.g., the number of devices that can be served by each AP and the quality-of-service constraint at each individual device.

This work proposes a GNN based framework to solve the optimization problems in (1) via end-to-end learning,

such that the time consumption is promising for real-time implementation in wireless IoT networks.

III. A GNN BASED FRAMEWORK FOR RESOURCE ALLOCATION IN WIRELESS NETWORKS

In this section, a general framework based on supervised GNN is proposed to approximate the optimization problem in (1) by learning directly the input-output mapping. Firstly, the CE method that can simplify the training sample generation is introduced. Then, the graph modeling of wireless networks is described, followed by a GNN that is operated in a supervised manner. An illustration of the proposed framework is given in Fig. 1.

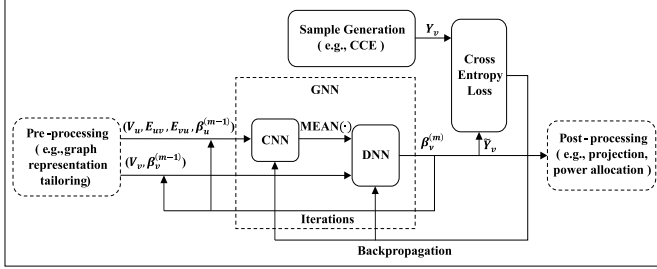


Fig. 1 The proposed GNN based framework for resource optimization in wireless IoT networks.

A. Training Samples Generation

The proposed GNN based framework is operated in a supervised manner, which requires sufficient labeled training samples. The optimal algorithms such as the B&B algorithm have an exponential computational complexity, which poses significant challenges to generate a large dataset for the training purpose. Therefore, in order to further reduce the computational complexity and time consumption for the sample generation, the CCE algorithm is employed for training samples generation.

The CE method is mainly based on Kullback-Leibler cross-entropy and importance sampling, and it involves an iterative procedure where each iteration can be divided into two phases [26]:

- 1) Generate random samples according to a specified mechanism.
- 2) Update the parameters of the mechanism based on the data for “better” samples in the next iteration.

In order to solve the general constrained resource allocation problem in (1), the CCE algorithm is adopted in the proposed framework. The steps of the CCE algorithm are summarized in Algorithm 1, where the independent Bernoulli distribution is utilized for generating random samples.

In Algorithm 1, ρ denotes the quantile and it typically ranges from 0.01 to 0.1 [26]. Any infeasible samples generated in Step 2 will be converted to the feasible ones via the projection in Step 3. After iterations, the near-optimal results can be obtained by the CCE algorithm and can be used as training labels for the GNN as will be introduced in the following subsections.

Algorithm 1: The General CCE Algorithm for Discrete Optimization Problem in (1)

- Step 1: Initialize the Bernoulli probability vector $\mathbf{P}^0 = \{P_i^0\}$, $P_i^0 = 0.5, \forall i$. Set the iteration index $t = 1$.
- Step 2: Randomly generate a large number M samples $\{\mathbf{x}^j\}_{j=1}^M$ according to the probability $\mathbf{P}^{(t-1)}$, where the i -th element of \mathbf{x}^j is denoted by x_i^j .
- Step 3: Project any infeasible samples into feasible samples.
- Step 4: Sort $\{\mathbf{x}^j\}_{j=1}^M$ in an ascending order as $\{\mathbf{x}^{\sigma_j}\}_{\sigma_j=1}^M$ with respect to the scale values calculated by the objective function $f(\mathbf{x}^j)$.
- Step 5: Select the best $\lceil \rho M \rceil$ samples from $\{\mathbf{x}^{\sigma_j}\}_{\sigma_j=1}^M$ and update the probability vector \mathbf{P}^t as follows,

$$P_i^t = \frac{\sum_{\sigma_j=1}^{\lceil \rho M \rceil} x_i^{\sigma_j}}{\lceil \rho M \rceil}, \forall i. \quad (2)$$

- Step 6: **if** \mathbf{P}^t does not converge to a binary vector **then**
Set $t = t + 1$ and go to Step 2.
else
 $\mathbf{x} = \mathbf{P}^t$.
end if
-

B. Graph Representation of Wireless Networks

Graphs provide a structured view of the abstract concepts, especially with regard to the relationships and interactions between the graph elements. This feature is favorable in modeling the transmitters and receivers in the wireless IoT networks as the geometrical information can be embedded in the graph features. In wireless networks, the links between communication agents can be generally categorized as beneficial and harmful links, which represent communication links and interference links, respectively. The functions of the communication links and interference links are completely contrary. Therefore, it is better to distinguish them in the graph modeling. By the nature of nodes and edges of a graph, edges can model the interactions between nodes. Thus, edges are rational to represent the interference links. Accordingly, the wireless communication system can be modeled as a directed graph, where the communication link between a transceiver pair can be treated as a node, and the interference link between two nodes can be treated as an edge. The properties such as the distance, channel information, weight and priority that are related to communication links can be taken as node features. The properties such as the distance and channel information that are related to interference links can be treated as edge features. After graph modeling of the wireless network, the advanced algorithms based on graphs, such as graph coloring and graph embedding, can be utilized to solve the challenging problems in the wireless networks effectively.

Let \mathcal{V} and \mathcal{E} denote a set of nodes and edges of a graph, respectively. The edge connecting the two nodes $u, v \in \mathcal{V}$ can be defined as $e(u, v) \in \mathcal{E}$. In a wireless network, edges are directional, for example, $e(u, v)$ and $e(v, u)$ denote respectively,

the interference from node u to node v , and vice versa. Let V and E represent node features and edge features, respectively. To differentiate the contributions of the communication links and the interfering links, the desired direct channel gains are modeled as the node features while the harmful interference channel gains are modeled as the edge features. Meanwhile, the CSI including both small scale fading and large scale fading is taken as the input features of the proposed GNN based framework.

C. Graph Neural Network

The GNN was firstly proposed to extend the existing neural network mechanisms for processing the data represented in graph domains [27]. GNNs have multi-layer structures. In each layer, each node aggregates the features from its neighborhood, and the central node will then combine its own features with the aggregated features. GNNs iteratively update the representation of each node by the aggregation and combination functions. The update rule of the m -th layer at the node v is given as follows [28],

$$\begin{aligned} \alpha_v^{(m)} &= \text{AGGREGATE}^{(m)}(\{\beta_u^{(m-1)} : u \in \mathcal{N}(v)\}), \\ \beta_v^{(m)} &= \text{COMBINE}^{(m)}(\beta_v^{(m-1)}, \alpha_v^{(m)}), \end{aligned} \quad (3)$$

where $\beta_v^{(m)}$ is the feature vector of the node v at the m -th iteration/layer. $\mathcal{N}(v)$ denotes the set of the adjacent nodes of the node v . $\alpha_v^{(m)}$ represents an intermediate variable to denote the features aggregated by node v from its neighboring nodes at the m -th layer. In brief, the variety of AGGREGATE and COMBINE functions forms different GNNs [28].

Based on the graph modeling introduced in the previous subsection, a GNN framework incorporating the node and edge features (e.g., CSI) is proposed to address the resource allocation problems in (1). Since the feature aggregation operation in GNNs is analogous to the behavior of the convolutional kernels in CNNs [29], our proposed GNN framework utilizes the CNN to aggregate feature information on a local graph-structured neighborhood. Additionally, the neighborhood aggregation is expected to possess a property of the permutation invariance where the aggregated feature is invariant no matter the order of the neighboring nodes. This property can be achieved by a permutation-invariant function, such as a sum, mean, maximum and minimum, to reduce a set of aggregated neighborhood features to a single vector [29]. The permutation-invariant operation on the aggregated neighborhood features can be different depending on specific problems. In the sequel, the mean operation is adopted as an example. Since the model is more powerful by combining the aggregated information with feed-forward neural networks [29], a DNN is adopted as a combination function after the aggregation operation. Accordingly, the update rule of the proposed GNN is given below,

$$\begin{aligned} \alpha_v^{(m)} &= \mathbb{E}_u(\text{CNN}(V_u, E_{uv}, E_{vu}, \beta_u^{(m-1)})), u \in \mathcal{N}(v), \\ \beta_v^{(m)} &= \text{DNN}(V_v, \beta_v^{(m-1)}, \alpha_v^{(m)}), \end{aligned} \quad (4)$$

where $\alpha_v^{(m)} \in \mathbb{R}^{1 \times d_1}$ represents the aggregated neighborhood feature vector of node v at the m -th layer, and d_1 is self-defined output size (out_channels) of the CNN. $\beta_v^{(m)} \in \mathbb{R}^{1 \times d_2}$

denotes the embedding feature vector of node v at the m -th layer, and d_2 shares the same size with the number of classes depending on specific problems. The $\beta_v^{(m)}$ at the last layer of the GNN also denotes the final output of the GNN. \mathbb{E} denotes a mean operation with respect to u , which provides the permutation invariance property for the aggregated features. In the DNN, the Softmax function is adopted as the last activation function. As aforementioned, the communication link between a transceiver pair is modeled as a node and the direct channel gain is taken as the node features, while the interference link between two nodes is modeled as an edge and the interference channel gain is taken as the edge feature. Hence, $V_u, u \in \mathcal{N}(v)$ denotes the node feature (direct channel gain) of node u which is a neighbor of node v . E_{uv} represents the edge feature (interference channel gain) from node u to node v , and similarly E_{vu} denotes the edge feature from node v to node u . $\beta_v^{(0)}$ is initialized with a zero vector, whose size varies depending on specific problems. Note that only α and β need to be updated at each layer of the GNN, and the other parameters (e.g., V_u and E_{uv}) are constant. Fig. 2 illustrates the update rule of one node at the m -th layer of the GNN, where node 1 aggregates information from its neighborhood (nodes 2-4 and edges) and then the aggregated information forms $\alpha_1^{(m)}$ which is combined with the local information of node 1 as formulated in (4). Since the properties related to communication and interference links can be mapped to the node and edge features in a graph domain, the proposed GNN framework can be generalized to the problems in wireless IoT networks following the similar mapping rules.

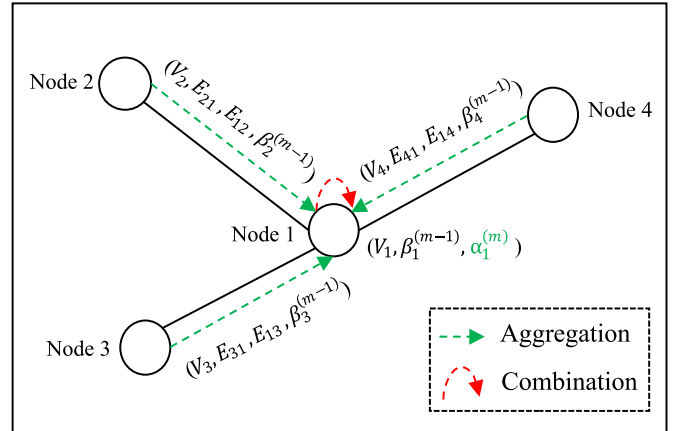


Fig. 2 An illustration of the aggregation and combination of one node at the m -th layer of GNN.

The resource allocation problems in wireless IoT networks as formulated in (1) can be viewed as multi-class classification problems. Let C denote the number of classes and let $\mathcal{S} = \{0, 1, \dots, C-1\}$ denote the indexes of classes. Accordingly, the output of the GNN consists of C neurons which indicate the class probabilities of each individual node. Let $\mathbf{x} = \{x_v\}$, $x_v \in \mathcal{S}$, $v \in \mathcal{V}$ denote the target labels generated by the CCE algorithm, wherein x_v indicates the resource allocation decision of node v . Let $Y_v = \{y_{vc}\}$, $c \in \mathcal{S}$ denote the one-hot classification of node v . For each node v , $y_{vc} = 1$ if node v is labelled as class c , $y_{vc} = 0$ otherwise. Let $\tilde{Y}_v = \{\tilde{y}_{vc}\}$, $c \in \mathcal{S}$

represent the output class probabilities of the GNN for node v , where \tilde{y}_{vc} denotes the probability of node v to be in class c . The cross-entropy is adopted as the loss function as follows,

$$\ell = - \sum_{v \in \mathcal{V}} \sum_{c \in \mathcal{S}} y_{vc} \ln \tilde{y}_{vc}. \quad (5)$$

By minimizing the loss function in (5), the parameters of the GNN are updated.

For adaptation to various problems, the proposed GNN based framework may involve pre-processing and post-processing steps, as shown in Fig. 1. The pre-processing and post-processing steps are optional depending on practical problems as well as the availability of the expert knowledge. For example, the graph representation can be pre-processed by setting a distance threshold or considering a fixed number of nearest neighboring nodes rather than using full connections to further reduce the complexity. For post-processing, the output of the GNN may not satisfy the constraints or may need further steps to achieve the final objective, hence expert knowledge is required to address these issues, such as projection algorithm, power allocation and recovery. The proposed framework has the potential to be adapted to the general resource allocation problem in wireless networks via the integration of expert knowledge in pre-processing and post-processing steps.

D. Complexity of GNN

In each layer of the GNN, an aggregation operation is applied to all the edges, and a combination operation is applied to all the nodes. Therefore, the time complexity of each layer is $\mathcal{O}(|\mathcal{V}| + |\mathcal{E}|)$, then the total time complexity of a M -layer GNN is $\mathcal{O}(M(|\mathcal{V}| + |\mathcal{E}|))$.

In the next two sections, the effectiveness of the proposed GNN based framework is verified by two case studies of resource allocation problems, namely the link scheduling in a homogeneous D2D network and the joint channel and power allocation in a heterogeneous D2D underlaid cellular network.

IV. APPLICATION ON LINK SCHEDULING PROBLEM IN D2D NETWORKS

This section presents the link scheduling problem in the D2D network. The proposed supervised GNN framework is applied to address this problem and its effectiveness is demonstrated by the simulation results at the end of this section.

A. System Model and Problem Formulation

Let us consider a wireless IoT network with L D2D pairs, where the D2D pairs are randomly located in a square region with an edge length of d_{area} . The set and the indexes of all D2D pairs are denoted by $\mathcal{D} = \{D_1, D_2, \dots, D_L\}$ and $\mathcal{L} = \{1, 2, \dots, L\}$, respectively. For each D2D pair $D_l \in \mathcal{D}$, its transmitter and receiver are denoted as T_l and R_l , $l \in \mathcal{L}$, respectively. It is assumed that each D2D pair is within a pairwise distance between d_{\min} and d_{\max} . Let us denote p_l the fixed transmit power of D2D pair l , $l \in \mathcal{L}$. All D2D pairs share the same spectrum. A simple network with three D2D pairs is shown as an example in Fig. 3a.

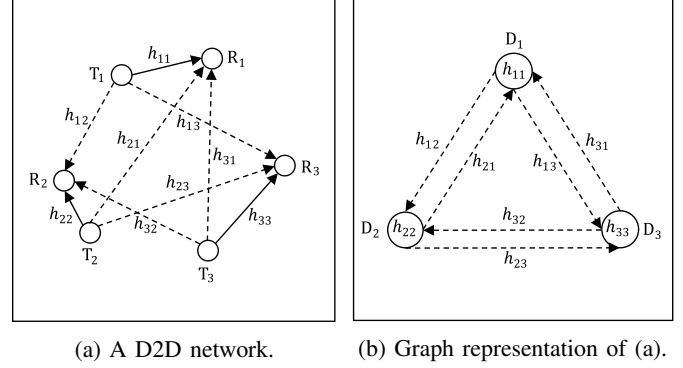


Fig. 3 A three-pair D2D network and its corresponding graph representation.

Let h_{ll} denote the communication channel between the transmitter and receiver of D_l , and h_{lk} denote the interference channel from T_l to R_k , $l, k \in \mathcal{L}$ and $l \neq k$. Let $\mathbf{x} = \{x_l\}$ denote the indicator vector of the status of the D2D pairs, where x_l , $l \in \mathcal{L}$ denote the binary decision variable of D_l , and $x_l = 1$ if D_l is active and $x_l = 0$ otherwise. Then the signal-to-interference-plus-noise ratio (SINR) of D_l denoted by ξ_l is given as

$$\xi_l = \frac{|h_{ll}|^2 p_l x_l}{\sum_{k \neq l} |h_{kl}|^2 p_k x_k + \sigma_N^2}, \quad l, k \in \mathcal{L}, \quad (6)$$

where σ_N^2 denotes the power of the additive white Gaussian noise (AWGN). Generally, the objective is to maximize the sum rate by finding the optimal link scheduling. This problem can be formulated as

$$\begin{aligned} \max_{\mathbf{x}} \quad & \sum_{l \in \mathcal{L}} \log_2(1 + \xi_l) \\ \text{subject to} \quad & x_l \in \{0, 1\}, \quad \forall l \in \mathcal{L}. \end{aligned} \quad (7)$$

Note that the data rate is normalized by the channel bandwidth, hence the unit is in bits per second per hertz.

B. Graph Representation

The D2D wireless network is modeled as a fully connected graph, where each D2D pair is treated as a node, and each interference link between D2D pairs is treated as an edge, as depicted in Fig. 3b. Given an example of feature mappings, for node $v = 1$, node $u = 2$ is one of its neighbors and then the mappings between the channel information and the node/edge features are as follows, $V_v = h_{11}$, $V_u = h_{22}$, $E_{uv} = h_{21}$ and $E_{vu} = h_{12}$. In this case study, the aim of the GNN is to map from the channel matrix to binary decisions of whether each individual D2D pair is active or not, therefore, $d_2 = C = 2$.

The CCE algorithm for link scheduling in D2D networks is described in Algorithm 1. In this considered scenario, the projection in Step 3 of Algorithm 1 can be ignored as there are no other constraints except the binary constraints. Besides, it does not require any pre-processing and post-processing steps for the learning stage.

C. Numerical Results

For both case studies, the simulation was conducted with processor Intel Core i5-9600KF CPU using PyTorch. The performance of the proposed supervised GNN based framework is compared against the following four benchmark schemes:

- CCE: The CCE algorithm with corresponding adaptations is utilized to generate training samples, and it also serves as an upper bound. The performances of the ML based schemes are given with respect to this CCE algorithm.
- Unsupervised GNN : The unsupervised GNN has the same structure and parameters as of the proposed supervised GNN, while the loss function is defined as the negative sum rate as in [23].
- Graph embedding: Distance with quantization is taken as the node and edge features for graph embedding. The embedding feature of nodes is learned by a 3-layer classifier in a supervised manner as in [20].
- DNN: A 4-layer conventional supervised DNN is adopted. The channel matrix is taken as the input of the DNN.

To ensure fair comparisons, the performance of the proposed framework and the benchmark designs are evaluated using the following settings. All samples are generated by the corresponding CCE algorithms. The size of the testing dataset is set to be 200 for all simulations. The adaptive moment estimation (ADAM) [30] optimizer is adopted to update network parameters for both problems.

For both cases, the performance comparisons between the proposed method and the benchmark schemes are presented in terms of the average classification accuracy, sum rate and time consumption. The classification accuracy is the first metric to measure the performance of the proposed design, which reflects the similarity between the classification results generated by the proposed design as well as the benchmark schemes and the target produced by the CCE method. Additionally, the average sum rate is taken as the second metric to measure the end performance, which is the normalized sum rate achieved by the proposed design as well as the benchmark schemes with respect to that generated by the CCE algorithm. Moreover, the time consumption is examined for running time comparisons between the proposed framework and all benchmark schemes.

In this case study, the transmitter of each D2D pair is generated randomly according to the uniform distribution in a square area, and the corresponding receiver is uniformly distributed with a specified pairwise distance away from the transmitter. A distance dependent path loss model is adopted as the large scale fading model, and the Rayleigh fading with zero mean and unit variance is modeled for the small scale fading. The main system and GNN parameters are listed in Table I.

1) Performance Comparisons with Different Number of Training Samples

The performance comparison results with the different number of training samples are summarized in Table II, for $L = 30$ D2D pairs.

As shown in Table II, the performance of the proposed supervised GNN is slightly increasing with the number of

TABLE I System and GNN Parameters

Parameters	Values
Edge length, d_{area}	500 m
D2D pairwise distance, $d_{\text{min}} - d_{\text{max}}$	2 - 65 m
Transmit power of activated link, p_l	0 dBm
Path loss model	$148 + 40 \log_{10}(d[\text{km}])$ [31]
The number of layers of GNN	5
Sizes of CNN in GNN	{5, 32, 32, 6}
Sizes of DNN in GNN	{9, 32, 32, 2}

training samples. The proposed supervised GNN approach achieves an accuracy of 0.9027 and a normalized sum rate of 0.9724 with only 100 training samples, wherein the gap of the end performance between the proposed supervised GNN and the CCE algorithm is only 2.76%. This feature of high sample efficiency is preferred for practical problems in wireless networks as the acquisition of sufficient training samples in wireless networks can be expensive or even impractical. The performance can be further improved with a larger number of training samples. As can be observed from Table II, with 1000 training samples, the accuracy and sum rate can reach 0.9277 and 0.9827, respectively. The gap of the end performance is further reduced to 1.73%. As a conclusion, the end performance of the proposed method is improved by approximately 0.01 with increasing the number of training samples from 100 to 1000.

It is indicated in Table II that the proposed supervised GNN method outperforms the benchmark schemes. The reason that the supervised GNN outperforms the unsupervised GNN is probably due to the fact that the D2D link scheduling is a discrete classification problem. Besides, the unsupervised GNN may need more samples to obtain a better performance. The proposed GNN based framework outperforms the graph embedding method. The reason is that the small scale fading information has been neglected by the nature of the graph embedding method. In other words, the full information of the fading channel has been included as the input feature of our proposed approach while the graph embedding method only considered the distance information as embedding features. In addition, the conventional supervised DNN has the worst performance amongst all designs. The reason is that the supervised DNN is a data-driven approach that normally requires a large training dataset. Besides, the conventional DNN, by its nature, ignores the node/edge features incorporated in the graph theory.

2) Performance Comparisons with Different Number of D2D Pairs

The performance of the proposed method is evaluated under the different number of D2D pairs with $L \in \{10, 30, 50\}$. For the performance evaluation, 500 training samples are generated by the CCE algorithm in each case. The simulation results are shown in Table III. The performance of the CCE algorithm is evaluated by comparing it with the FPLinQ algorithm [4]. The ratio between the sum rate achieved by the FPLinQ algorithm and that generated by the CCE algorithm is given, and it indicates that the performance of the CCE algorithm is close to the FPLinQ. Although the CCE algorithm can not beat the FPLinQ, it has much wider applications than the FPLinQ al-

TABLE II Performance Comparisons with Different Numbers of Training Samples

Number of Training Samples	100		200		500		1000	
	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate
GNN (Supervised)	0.9027	0.9724	0.9100	0.9758	0.9143	0.9780	0.9277	0.9827
GNN (Unsupervised)	0.7570	0.8385	0.7623	0.8478	0.7767	0.8485	0.7622	0.8393
Graph Embedding	0.7578	0.8328	0.7645	0.8452	0.7742	0.8519	0.7805	0.8642
DNN	0.6500	0.7309	0.7122	0.7976	0.7128	0.8005	0.7218	0.8147

TABLE III Performance Comparisons with Different Number of D2D Pairs

Number of D2D Pairs	10		30		50	
	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate
FPLinQ	-	1.0052	-	1.0099	-	1.0151
GNN (Supervised)	0.8955	0.9751	0.9143	0.9780	0.9232	0.9787
GNN (Unsupervised)	0.7385	0.8599	0.7767	0.8485	0.7966	0.8417
Graph Embedding	0.7955	0.9057	0.7742	0.8519	0.7902	0.8430
DNN	0.7945	0.8952	0.7128	0.8005	0.7044	0.7846

TABLE IV Performance Comparisons with Different Pairwise Distances

Pairwise Distance (m)	2 – 65		15 – 65		15 – 50		fixed 30	
	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate
GNN (Supervised)	0.9100	0.9758	0.9115	0.9691	0.9003	0.9676	0.8847	0.9608
GNN (Unsupervised)	0.7623	0.8478	0.7413	0.7712	0.7168	0.7922	0.6865	0.7623
Graph Embedding	0.7645	0.8452	0.7430	0.7730	0.7140	0.7779	0.6333	0.7012
DNN	0.7122	0.7976	0.6755	0.7014	0.6785	0.7479	0.6395	0.7366

gorithm. Table III shows that the proposed approach maintains the best and the most stable performance with the increasing system scale. Whereas, the performance of all benchmark designs either fluctuates or degrades for larger scale systems. This proves that the proposed supervised GNN framework can handle large scale systems with stable performances. The accuracy and sum rate of the proposed method for all considered cases remain over 0.89 and 0.97, respectively. By contrast, although the graph embedding method can achieve a normalized sum rate of 0.9057 at $L = 10$, and the performance degrades to 0.8430 at $L = 50$. The performance degradation of the graph embedding method is around 6% when the network size increases from 10 to 50.

3) Performance Comparisons with Different Pairwise Distances

The performance of the proposed approach and the benchmark schemes with varying D2D pairwise distances are compared in Table IV with 200 training samples and $L = 30$ D2D pairs. When the distribution of the pairwise distances changes, the accuracy of the proposed method can achieve at least 88% of the target scheduling results, and the average normalized sum rate can maintain above 96% of that achieved by the CCE algorithm. The performance of the proposed framework on the scenario with a fixed pairwise distance is the worst amongst all system settings. This is due to the fact that the channel gain largely depends on the distance, hence embedding the fixed pairwise distance into node features will lose the geometrical information of wireless network to some extent. As shown in Table IV, the proposed supervised GNN outperforms the three benchmark schemes in all four tested parameter settings.

4) Running Time Performance Comparisons

Three network settings with $L \in \{10, 30, 50\}$ at pairwise distance 2 – 65 m are considered to compare the running time performance of the supervised GNN method, the benchmark schemes and the CCE algorithm. The average results of the running time are shown in Table V.

TABLE V Average Running Time Comparisons

Number of D2D Pairs	10	30	50
CCE (ms)	1527.3	19567.6	58822.1
GNN (Supervised) (ms)	0.1645	0.6258	1.5053
GNN (Unsupervised) (ms)	0.1668	0.6426	1.5165
Graph Embedding (ms)	0.9046	6.3865	17.0046
DNN (ms)	0.0094	0.0135	0.0188

It can be observed from Table V that the conventional CCE method consumes significant time when the number of D2D pairs is increasing since the scheduling problem becomes more complicated with larger networks. The traditional algorithms for D2D link scheduling problems are usually time-consuming, which are not suitable for real-time applications, and may result in significant performance degradation for real-time implementation in wireless networks. By contrast, the supervised GNN method significantly accelerates the link scheduling problem in D2D networks. It is around 10^4 times faster than the conventional CE algorithm for $L = 10$, 3×10^4 times faster for $L = 30$, and 4×10^4 times faster for $L = 50$. Such significant acceleration by the proposed approach is very promising for real-time implementation in wireless networks. The proposed supervised GNN approach has a similar running time performance with the unsupervised GNN since they share similar network structure and input features. Whereas, it outperforms the graph embedding method on time consumption because the latter takes time to obtain the embedding features of nodes, where a distance quantization is required and each node iteratively updates its embedding feature from itself and all its adjacent nodes. Although the DNN achieves a better running time performance due to the negligence of the graph features, it has inferior end performance and sample efficiency as compared to the GNN based design.

5) Generalizability to Different System Settings

To demonstrate the generalization ability, the proposed GNN framework is trained with 1000 samples at $L = 30$ and

TABLE VI Generalizability to Different System Settings

Number of D2D Pairs	Pairwise Distance (m)	GNN (Supervised)		GNN (Unsupervised)		Graph Embedding		DNN	
		Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate
30	15 – 65	0.9158	0.9699	0.6385	0.4753	0.5477	0.0887	0.6810	0.7004
30	15 – 50	0.9065	0.9665	0.5983	0.5108	0.4818	0.0913	0.6868	0.7354
30	fixed 30	0.8822	0.9489	0.4828	0.2082	0.4485	0.0325	0.6522	0.7164
10	2 – 65	0.8960	0.9759	0.5605	0.6343	0.3985	0.2787	-	-
50	2 – 65	0.9168	0.9636	0.7567	0.7773	0.5403	0.0048	-	-

pairwise distance 2 – 65 m, then apply the trained GNN model directly to different system settings, such as pairwise distances and system scales, without any further training.

Table VI shows comparison results of the generalization ability, which indicates that the performance of the proposed GNN approach is stable on scenarios with different pairwise distances and system scales without any retraining. Regarding the scenarios with various pairwise distances, when the pairwise distances are 15 – 65 m and 15 – 50 m, the generalization can achieve almost the same performance with the training of 200 samples as shown in Table IV, and there is only 1.19% performance loss even for the worst case with the fixed pairwise distance. Regarding the scenarios with different system scales, the generalization achieves nearly the same performance at $L = 10$ and results in only 1.51% performance loss at $L = 50$ comparing to the training of 500 samples as illustrated in Table III. **By contrast, the unsupervised GNN and the graph embedding methods indicate significant performance degradations on the generalizability. Although the DNN shows a relatively good performance on the varying pairwise distances, it requires retraining when the network scales change.** In this case study, the size of the proposed GNN is independent of L , hence it is promising to be generalized to the network scales with different L where no further training is required. Comparing to the neural networks where retraining is needed once the system setting is changed, the generalization feature of the proposed GNN framework is desirable in wireless IoT networks to prevent expensive training cost.

6) Robustness

The situation with partial CSI is considered to test the robustness of the proposed GNN framework. The pre-trained model for $L = 30$ is adopted to test on the case where a fixed proportion of the interference CSI is missing. The ratio between the performance achieved by the proposed GNN with partial CSI and that achieved by the case with full CSI is reported in Fig. 4. It can be observed that the proposed GNN achieves an accuracy of 0.85 and a sum rate of 0.91 with respect to that of the case with full CSI when the CSI for even 50% of the links is unavailable. This demonstrates that the proposed GNN framework is robust to the corruption of input features.

V. APPLICATION ON JOINT CHANNEL AND POWER ALLOCATION PROBLEM IN D2D UNDERLAID CELLULAR NETWORKS

In this section, a joint channel and power allocation problem in the D2D underlaid cellular network is introduced, followed by a solution based on the proposed GNN framework and its performance.

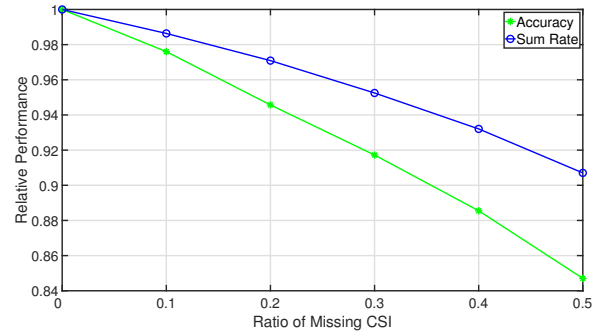


Fig. 4 Performance of GNN with missing CSI.

A. System Model and Problem Formulation

This case study considers an uplink single-cell system with K cellular users (CUs) and L D2D pairs. Let $\mathcal{K} = \{1, \dots, K\}$ and $\mathcal{L} = \{1, \dots, L\}$ denote the indexes of CUs and D2D pairs, respectively. The individual CUs transmit signals to the BS via orthogonal channels. It is assumed that D2D pairs transmit data by using the channels of CUs in the underlay mode. In the D2D underlaid cellular networks, the number of D2D pairs is usually smaller than that of CUs, hence we consider $K \geq L$. A simple system model is depicted in Fig. 5.

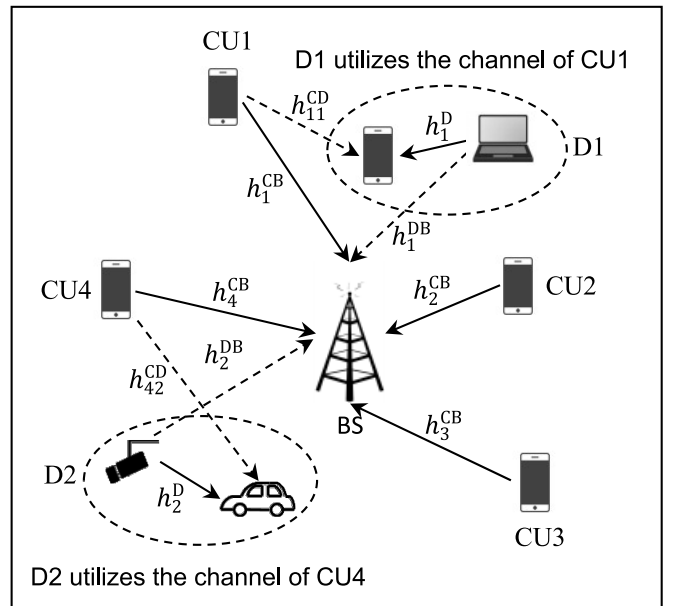


Fig. 5 An uplink single-cell with CUs and D2D pairs.

Let h_k^{CB} denote the channel gain between the k -th CU, $k \in \mathcal{K}$ and the BS, and let h_l^D denote the channel gain between the transmitter and the receiver of the l -th D2D pair, $l \in \mathcal{L}$. h_l^{DB}

denotes the channel gain of the interference link between the transmitter of the l -th D2D pair and the BS, and h_{kl}^{CD} represents the channel gain of the interference link between the k -th CU and the receiver of the l -th D2D pair.

Let $\mathbf{x} = \{x_{kl}\}$, $k \in \mathcal{K}$, $l \in \mathcal{L}$ denote the indicator vector of the channel allocation, where $x_{kl} = 1$ if the channel of the k -th CU is utilized by the l -th D2D pair, and $x_{kl} = 0$ otherwise. Let $\mathbf{p}^{\text{C}} = \{p_k^{\text{C}}\}$, $k \in \mathcal{K}$ denote the transmit power vector of the CUs, and let $\mathbf{p}^{\text{D}} = \{p_l^{\text{D}}\}$, $l \in \mathcal{L}$ represent the transmit power vector of D2D pairs. It is assumed that each CU channel can be accessed by at most one D2D pair. The SINR of the l -th D2D pair on the channel of the k -th CU is formulated as

$$\xi_{kl}^{\text{D}} = \frac{h_l^{\text{D}} p_l^{\text{D}} x_{kl}}{h_{kl}^{\text{CD}} p_k^{\text{C}} + \sigma_N^2}. \quad (8)$$

The SINR at the BS achieved by the k -th CU can be written as

$$\xi_k^{\text{C}} = \frac{h_k^{\text{CB}} p_k^{\text{C}}}{\sum_{l \in \mathcal{L}} h_l^{\text{DB}} p_l^{\text{D}} x_{kl} + \sigma_N^2}. \quad (9)$$

Note that the data rate is normalized by the channel bandwidth. Therefore the data rates of the l -th D2D pair and the k -th CU can be expressed, respectively, as

$$R_l^{\text{D}} = \log_2(1 + \xi_{kl}^{\text{D}}) \quad (10)$$

and

$$R_k^{\text{C}} = \log_2(1 + \xi_k^{\text{C}}). \quad (11)$$

The objective is to maximize the sum rate of both CUs and D2D pairs by optimizing the channel allocation decisions \mathbf{x} as well as the power allocation decisions \mathbf{p}^{C} and \mathbf{p}^{D} , which is formulated as

$$\begin{aligned} & \max_{\mathbf{x}, \mathbf{p}^{\text{C}}, \mathbf{p}^{\text{D}}} \sum_{l \in \mathcal{L}} R_l^{\text{D}} + \sum_{k \in \mathcal{K}} R_k^{\text{C}} \\ & \text{subject to } x_{kl} \in \{0, 1\}, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}, \\ & \sum_{l \in \mathcal{L}} x_{kl} \leq 1, \forall k \in \mathcal{K}, \\ & R_k^{\text{C}} \geq R_{\min}^{\text{C}}, \forall k \in \mathcal{K}, \\ & R_l^{\text{D}} \geq R_{\min}^{\text{D}}, \forall l \in \mathcal{L}, \\ & p_k^{\text{C}} \leq p_{\max}^{\text{C}}, \forall k \in \mathcal{K}, \\ & p_l^{\text{D}} \leq p_{\max}^{\text{D}}, \forall l \in \mathcal{L}, \end{aligned} \quad (12)$$

where R_{\min}^{C} represents the minimum data rate requirement of each CU. p_{\max}^{C} and p_{\max}^{D} denote the maximum transmit power of each CU and each D2D pair, respectively. The second constraint means that each CU channel can be utilized by at most one D2D pair. The third and fourth constraints represent the minimum data rate constraint of each CU and each D2D pair, respectively. The fifth constraint denotes that each CU has a limited maximum transmit power. The last constraint indicates that the transmit power of each D2D pair should not exceed a specific maximum value.

B. Graph Representation

The link between the BS and each CU is treated as a node, it is termed as a CU node. Each D2D pair is also treated as a node, it is termed as a D2D node. The interference links between each D2D node and each CU node are treated as edges. The graph representation of the D2D underlaid cellular network is illustrated in Fig. 6.

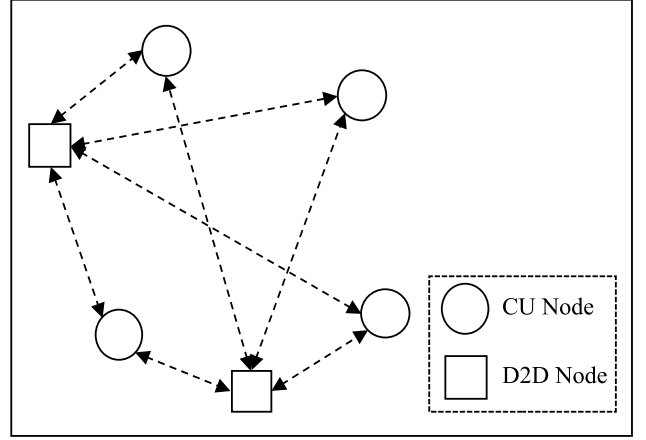


Fig. 6 Graph representation of D2D underlaid cellular network.

In this case study, only the channel allocation for D2D pairs are needed to be learned, so $\beta_u^{(m-1)}$ is removed in the aggregation function in (4). Since the target of the GNN is to learn which CU channel can be utilized by each D2D pair, this problem can be viewed as a K -class classification problem, where K classes correspond to K orthogonal channels from the CUs to the BS.

C. Adaptation

In this case, each CU channel can be accessed by at most one D2D pair. In the CCE algorithm as described in Algorithm 1, the randomly generated samples in Step 2 may not meet this constraint, therefore the projection algorithm is required to convert them into feasible samples in Step 3.

Since this case study involves power allocation with constraints, the following optimal power allocation will be integrated into Step 4 in Algorithm 1 as the power allocation solution for the sum rate calculation. Additionally, it will be also applied to the channel allocation results produced by the GNN in the post-processing steps of our proposed framework.

1) Optimal Power Allocation with Constraints

In the considered system, if the channel of the k -th CU is accessed by the l -th D2D pair, then the power allocation problem only involves one CU and one D2D pair in a shared channel. To maximize the sum rate, a closed-form solution of the power allocation for such optimization problem was provided in [32] and the authors proved that the optimal power must be in the set $p^* \in \{(p_{\max}^{\text{C}}, p_{\min}^{\text{D}(R_{\min}^{\text{D}})}), (p_{\max}^{\text{C}}, p_{\max}^{\text{D}(R_{\min}^{\text{C}})}), (p_{\min}^{\text{C}(R_{\min}^{\text{C}})}, p_{\max}^{\text{D}}), (p_{\max}^{\text{C}(R_{\min}^{\text{D}})}, p_{\max}^{\text{D}})\}$,

where

$$\begin{aligned}
p_{\min}^{D(R_{\min}^D)} &\triangleq \max\left\{\frac{(2^{R_{\min}^D} - 1)(p_{\max}^C h_{kl}^{CD} + \sigma_N^2)}{h_l^D}, 0\right\}, \\
p_{\max}^{D(R_{\min}^C)} &\triangleq \min\left\{\frac{1}{h_l^{DB}}\left(\frac{p_{\max}^C h_k^{CB}}{2^{R_{\min}^C} - 1} - \sigma_N^2\right), p_{\max}^D\right\}, \\
p_{\min}^{C(R_{\min}^C)} &\triangleq \max\left\{\frac{(2^{R_{\min}^C} - 1)(p_{\max}^D h_l^{DB} + \sigma_N^2)}{h_k^{CB}}, 0\right\}, \\
p_{\max}^{C(R_{\min}^D)} &\triangleq \min\left\{\frac{1}{h_{kl}^{CD}}\left(\frac{p_{\max}^D h_l^D}{2^{R_{\min}^D} - 1} - \sigma_N^2\right), p_{\max}^C\right\}.
\end{aligned} \tag{13}$$

If there is no minimum data rate requirement for D2D pair, i.e., $R_{\min}^D = 0$, then we have $p_{\min}^{D(R_{\min}^D)} = 0$ and $p_{\max}^{C(R_{\min}^D)} = p_{\max}^C$. The optimal power allocation will be the one in the set that maximizes the sum rate of both CU and D2D pair and satisfies the data rate and power constraints.

It is assumed that each channel can be utilized by at most one D2D pair. However, the channel allocation results generated by the GNN may not satisfy this requirement. Therefore, a projection step is necessary to be incorporated with the post-processing step of our proposed framework to mitigate the problem of infeasible results.

2) Projection on the infeasible learning output

In this case study, the indicator vector of the channel allocation $\mathbf{x} = \{x_{kl}\}$, $k \in \mathcal{K}$, $l \in \mathcal{L}$ is denoted by a $L \times K$ matrix. The output probabilities of the GNN also form a $L \times K$ matrix denoted by $\tilde{Y} = \{\tilde{y}_{lk}\}$, $k \in \mathcal{K}$, $l \in \mathcal{L}$, where the l -th row represents the probabilities of K classes for the l -th D2D pair. The key procedures of the projection method are summarized as follows. The algorithm will

- find the maximum value of \tilde{Y} , e.g., \tilde{y}_{lk} , and assign the corresponding k -th CU channel to the l -th D2D pair, e.g., $x_{kl} = 1$.
- set all elements of l -th row and k -th column of \tilde{Y} to be zeros to avoid that the same CU channel is allocated to multiple D2D pairs.
- continue to find the maximum value of the updated probability matrix, and repeat the above steps until all D2D pairs are allocated with different CU channels.

D. Numerical Results

This section evaluates the performance of the proposed framework on the joint channel and power allocation problem in the D2D underlaid cellular network. The locations of CUs and D2D pairs are randomly generated in a square area with an edge length of d_{area} . The BS locates in the centre of this region. The Rayleigh fading with zero mean and unit variance is modeled for the small scale fading. The main system and GNN parameters are listed in Table VII.

A 2-layer GNN is adopted for performance evaluations due to the simplicity of connections between D2D nodes and CU nodes. Unless otherwise stated, the accuracy is given by the value calculated after applying the projection step to the output of the neural network. For fair comparisons, the same post-processing steps (projection and closed-form power allocation) have been applied to all benchmark schemes.

TABLE VII System and GNN Parameters

Parameters	Values
Edge length, d_{area}	1000 m
D2D pairwise distance, $d_{\min} - d_{\max}$	15 – 50 m
Maximum power of each D2D pair, p_{\max}^D	20 dBm
Maximum power of each CU, p_{\max}^C	20 dBm
Minimum data rate of CUs, R_{\min}^C	2 bits/s/Hz
Minimum data rate of D2D pairs, R_{\min}^D	0 bits/s/Hz
Path loss of cellular links	$128.1 + 37.6 \log_{10}(d[\text{km}])$ [31]
Path loss of D2D links	$148 + 40 \log_{10}(d[\text{km}])$ [31]
The number of layers of GNN	2
Sizes of CNN in GNN	{3, 32, 32, 8}
Sizes of DNN in GNN	{9+K, 32, 32, K}

1) Performance Comparisons with Different Number of Training Samples

The performances of the supervised GNN with different number of training samples are investigated for $K = 5, L = 2$, and compared against the benchmark designs, as shown in Table VIII.

As indicated in Table VIII, the average accuracy and sum rate of the proposed supervised GNN method are increasing with the growing number of training samples. The normalized sum rate achieves approximately 0.98 with 500 training samples, and can be further improved to around 0.993 with doubled training samples. For the benchmark schemes, the performance of the normalized sum rate only fluctuates between 0.83 and 0.87. The comparison results demonstrate that the proposed supervised GNN approach has a better capability for handling the heterogeneous resource allocation problem than the three benchmark schemes.

2) Performance Comparisons with Different Number of CUs and D2D Pairs

The performance of the proposed framework with different network sizes is examined with 1000 training samples. The results are compared in Table IX and all results are given with respect to the CCE algorithm. The optimal solution is obtained by applying the exhaustive search. Since the considered network sizes with $K = 5, 7$ are small, it is easy to obtain the optimal solutions by the exhaustive search mechanism. Note that the comparison between the optimal solution and the CCE method is not given for the case $K = 10, L = 5$ due to the exponential computational complexity of the former. It is seen that the results of the CCE algorithm are very close to the optimal results. In all considered parameter settings, the normalized sum rates achieved by the supervised GNN remain above 0.975 while the benchmark schemes can achieve at most 0.91 approximately. It can be concluded that the proposed framework surpasses all the benchmark schemes. The possible reasons that the supervised GNN performs better than the unsupervised GNN include the supervised method is more suitable than the unsupervised method for the classification problems, and the unsupervised learning usually requires larger dataset for training to achieve better performances. The proposed GNN approach outperforms the graph embedding method since our approach utilizes more information (e.g., full CSI) than the graph embedding mechanism which only uses distance. Moreover, the DNN underperforms the proposed GNN because DNN by its nature is a data-driven approach and cannot learn the topology of the wireless network.

TABLE VIII Performance Comparisons with Different Numbers of Training Samples

Number of Training Samples	500		750		1000		1500	
	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate
GNN (Supervised)	0.7375	0.9791	0.8200	0.9914	0.8075	0.9905	0.7850	0.9940
GNN (Unsupervised)	0.2575	0.8397	0.2475	0.8448	0.2500	0.8454	0.2825	0.8674
Graph Embedding	0.2250	0.8414	0.2175	0.8550	0.2225	0.8569	0.1800	0.8557
DNN	0.1775	0.8426	0.2150	0.8333	0.2075	0.8461	0.2025	0.8580

TABLE IX Performance Comparisons with Different Number of CUs and D2D pairs

System Scales	K=5, L=2		K=5, L=3		K=7, L=2		K=7, L=3		K=10, L=5	
	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate	Accuracy	Sum Rate
Optimal Solution	-	1.0001	-	1.0100	-	1.0003	-	1.0062	-	-
GNN (Supervised)	0.8075	0.9905	0.6700	0.9948	0.7375	0.9849	0.5867	0.9814	0.4730	0.9755
GNN (Unsupervised)	0.2500	0.8454	0.2067	0.8815	0.1725	0.8632	0.1467	0.8756	0.1180	0.9094
Graph Embedding	0.2225	0.8569	0.2067	0.8290	0.1709	0.8614	0.1367	0.8309	0.1110	0.8263
DNN	0.2075	0.8461	0.2383	0.8322	0.1600	0.8552	0.1650	0.8336	0.1240	0.8206

TABLE X Average Running Time Comparisons (including post-processing steps)

System Scales	K=5, L=2	K=5, L=3	K=7, L=2	K=7, L=3	K=10, L=5
CCE (ms)	573.0	1394.4	682.9	1622.0	3905.0
GNN (Supervised) (ms)	1.0787	1.5524	1.1048	1.5841	2.4810
GNN (Unsupervised) (ms)	1.1627	1.6336	1.2148	1.6530	2.5773
Graph Embedding (ms)	1.7208	2.2610	1.9863	2.5718	4.3553
DNN (ms)	1.1316	1.5532	1.1823	1.6060	2.6187

TABLE XI Generalizability to Different System Scales

System Scales		K=5, L=3		K=7, L=3	
Metrics		Accuracy	Sum Rate	Accuracy	Sum Rate
GNN (Supervised)	Training	0.6700	0.9948	0.5867	0.9814
	Generalization	0.6140	0.9803	0.5987	0.9715
GNN (Unsupervised)	Training	0.2067	0.8815	0.1467	0.8756
	Generalization	0.2167	0.8178	0.1267	0.8257
Graph Embedding	Training	0.2067	0.8290	0.1367	0.8309
	Generalization	0.2047	0.8182	0.1520	0.8303
DNN	Training	0.2383	0.8322	0.1650	0.8336
	Generalization	0.2173	0.8145	0.1433	0.8333

3) Running Time Performance Comparisons

The time consumptions of the proposed framework and the benchmark designs are evaluated with different network scales. The comparison outcomes are illustrated in Table X.

As shown in Table X, the proposed GNN method significantly accelerates the conventional CCE algorithm from second level down to millisecond level, e.g., 1.58 ms for $K = 7, L = 3$. The proposed supervised GNN method has a similar time consumption with the unsupervised GNN and the DNN because the same post-processing steps, such as the projection algorithm and the closed-form power allocation, are applied to all methods. Additionally, it is faster than the graph embedding method since the graph embedding operation is time-consuming. Since the running time of the proposed framework is only a few milliseconds, it is very attractive for solving practical problems in wireless networks which usually have stringent real-time requirements.

4) Generalizability to Different System Settings

To evaluate the generalization capability of the proposed GNN framework and the benchmark designs, they are trained with 1000 samples at $K = 5, L = 2$ and $K = 7, L = 2$, then the trained models are tested on the systems with a larger number of D2D pairs at $K = 5, L = 3$ and $K = 7, L = 3$, respectively.

The results are shown in Table XI where the training refers that testing samples share the same system scales with the training samples, and the generalization means that

testing samples have different network scales with the training samples. The generalization of the proposed supervised GNN approach only generates 0.01 – 0.015 sum rate loss comparing to the scenario with training. Some benchmark methods also show good generalization performance in this case study, but their final performance still has a remarkable gap to the proposed supervised GNN method. The results suggest that the proposed GNN approach has potential generalizability to systems with larger L . In this case study, since the size of the proposed GNN is related to K but independent of L , it is possibly generalized to the network scales with larger L without retraining when K is invariant.

5) Robustness

The pre-trained model of $K = 5, L = 3$ is used to test the robustness of the proposed GNN approach in this case study. The performance of the GNN with missing CSI is shown in Fig. 7, when the CSI of half of the links are missing, the proposed GNN can still achieve 90% of the sum rate generated by the situation with full CSI. This robustness feature is desirable in practical wireless IoT networks where some of the CSI may be unavailable.

VI. CONCLUSION

In this work, wireless IoT systems are represented by graphs. With the aid of graph modeling, a general GNN based framework is proposed to solve the resource optimization

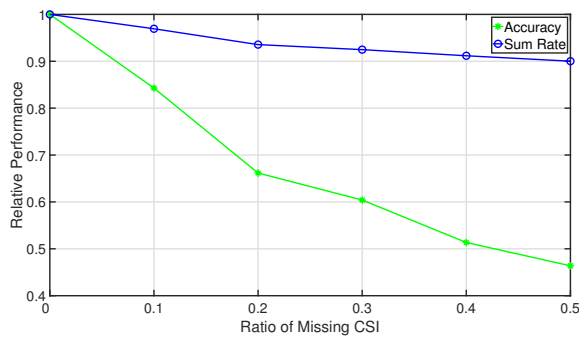


Fig. 7 Performance of GNN with missing CSI.

problems in wireless IoT networks. The proposed framework adopts a CNN with a mean operation for feature aggregations and a DNN for feature combinations to update the feature vector of each node in an iterative manner. The performance of our proposed framework is verified in two case studies of resource allocation in D2D wireless networks. Simulation results prove that the proposed framework works well for both the homogeneous systems and the heterogeneous networks. It outperforms all the considered benchmark schemes and is very promising for real-time implementation in wireless IoT networks. Additionally, the proposed GNN framework shows potential generalizability to various network settings and robustness to partial CSI.

REFERENCES

- [1] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, Jan. 2014.
- [2] B. Zhou, H. Hu, S.-Q. Huang, and H.-H. Chen, "Intracluster device-to-device relay algorithm with optimal resource utilization," *IEEE transactions on vehicular technology*, vol. 62, no. 5, pp. 2315–2326, Jan. 2013.
- [3] J. Wang, Y. Huang, S. Jin, R. Schober, X. You, and C. Zhao, "Resource management for device-to-device communication: A physical layer security perspective," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 4, pp. 946–960, Apr. 2018.
- [4] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *2017 IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, 2017, pp. 2323–2327.
- [5] R. Mathar and M. Schmeink, "Optimal base station positioning and channel assignment for 3G mobile networks by integer programming," *Annals of Operations Research*, vol. 107, no. 1-4, pp. 225–236, Oct. 2001.
- [6] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 62–75, Apr. 2010.
- [7] Y. Zhang, G. Zheng, C. Ji, K.-K. Wong, D. J. Edwards, and T. Cui, "Near-optimal joint antenna selection for amplify-and-forward relay networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, pp. 2401–2407, Jul. 2010.
- [8] X. Zhang, G. Zheng, S. Lambotharan, M. R. Nakhai, and K.-K. Wong, "A reinforcement learning-based user-assisted caching strategy for dynamic content library in small cell networks," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3627–3639, Jun. 2020.
- [9] X. Zhang, M. R. Nakhai, G. Zheng, S. Lambotharan, and B. Ottersten, "Calibrated learning for online distributed power allocation in small-cell networks," *IEEE Transactions on Communications*, vol. 67, no. 11, pp. 8124–8136, Aug. 2019.
- [10] L. Li, Y. Xu, J. Yin, W. Liang, X. Li, W. Chen, and Z. Han, "Deep reinforcement learning approaches for content caching in cache-enabled D2D networks," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 544–557, Jan. 2020.
- [11] 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 Transactions on Wireless Communications*, vol. 19, no. 1, pp. 665–679, Jan. 2020.
- [12] M. Lee, G. Yu, and G. Y. Li, "Learning to branch: Accelerating resource allocation in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 958–970, Nov. 2019.
- [13] M. Lee, N. Ma, G. Yu, and H. Dai, "Accelerating generalized benders decomposition for wireless resource allocation," *IEEE Transactions on Wireless Communications (Early Access)*, pp. 1–14, Oct. 2020.
- [14] Y. Yang, J. Xu, G. Shi, and C.-X. Wang, *5G wireless systems*. New York, NY, USA: Springer, 2018.
- [15] S. Uygungelen, G. Auer, and Z. Bharucha, "Graph-based dynamic frequency reuse in femtocell networks," in *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, Budapest, Hungary, 2011, pp. 1–6.
- [16] D. Tsolkas, E. Liotou, N. Passas, and L. Merakos, "A graph-coloring secondary resource allocation for D2D communications in LTE networks," in *2012 IEEE 17th international workshop on computer aided modeling and design of communication links and networks (CAMAD)*, Barcelona, Spain, 2012, pp. 56–60.
- [17] X. Cai, J. Zheng, and Y. Zhang, "A graph-coloring based resource allocation algorithm for D2D communication in cellular networks," in *2015 IEEE International Conference on Communications (ICC)*, London, UK, 2015, pp. 5429–5434.
- [18] H. Cai, V. W. Zheng, and K. C.-C. Chang, "A comprehensive survey of graph embedding: Problems, techniques, and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616–1637, Feb. 2018.
- [19] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1248–1261, Mar. 2019.
- [20] M. Lee, G. Yu, and G. Y. Li, "Wireless link scheduling for D2D communications with graph embedding technique," in *2020 IEEE International Conference on Communications (ICC)*, Virtual Conference, 2020, pp. 1–6.
- [21] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Transactions on Wireless Communications*, Oct. 2020.
- [22] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems (Early Access)*, pp. 1–21, Mar. 2020.
- [23] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *2019 IEEE Globecom Workshops (GC Wkshps)*, Hawaii, USA, 2019, pp. 1–6.
- [24] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "Graph neural networks for scalable radio resource management: Architecture design and theoretical analysis," *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 101–115, Nov. 2020.
- [25] M. Eisen and A. R. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, pp. 2977–2991, Apr. 2020.
- [26] P.-T. De Boer, D. P. Kroese, S. Mannor, and R. Y. Rubinstein, "A tutorial on the cross-entropy method," *Annals of operations research*, vol. 134, no. 1, pp. 19–67, Feb. 2005.
- [27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, Dec. 2008.
- [28] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *7th International Conference on Learning Representations*, New Orleans, LA, USA, 2019, pp. 1–17.
- [29] W. L. Hamilton, "Graph representation learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 14, no. 3, pp. 1–159, Sep. 2020.
- [30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015, pp. 1–15.
- [31] S. Hakola, T. Chen, J. Lehtomäki, and T. Koskela, "Device-to-device (D2D) communication in cellular network-performance analysis of optimum and practical communication mode selection," in *2010 IEEE wireless communication and networking conference*, Sydney, Australia, 2010, pp. 1–6.
- [32] C. S. Chen and G. E. Oien, "Optimal power allocation for two-cell sum rate maximization under minimum rate constraints," in *2008*

IEEE International Symposium on Wireless Communication Systems,
Reykjavik, Iceland, 2008, pp. 396–400.