# Knowledge-Driven Resource Allocation for Wireless Networks: A WMMSE Unrolled Graph Neural Network Approach

Hao Yang, *Student Member, IEEE*, Nan Cheng, *Senior Member, IEEE*, Ruijin Sun, *Member, IEEE*, Wei Quan, *Senior Member, IEEE*, Rong Chai, *Senior Member, IEEE*, Khalid Aldubaikhy, *Member, IEEE*, Abdullah Alqasir, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—This article proposes a novel knowledge-driven approach for resource allocation in wireless networks using the graph neural network (GNN) architecture. To meet the millisecond-level timeliness and scalability required for the dynamic network environment, our proposed approach, named UWGNN, incorporates the deep unrolling of the weighted minimum mean-square error (WMMSE) algorithm, referred to as domain knowledge, into GNN, thereby reducing computational delay and sample complexity while adapting to various data distributions. Specifically, by unrolling the WMMSE algorithm into a series of interconnected submodules, UWGNN aligns closely with the optimization steps of the algorithm. Our analysis reveals the effectiveness of the deep unrolling method within UWGNN, which decomposes complicated end-to-end mappings, leading to a reduction in model complexity and parameter count. Experimental results demonstrate that UWGNN maintains optimal performance with computation latency 3–4 orders of magnitude lower than the WMMSE algorithm and exhibits strong performance and generalization across diverse data distributions and communication topologies without the need for retraining. Our findings contribute to the development of efficient and scalable wireless resource management solutions for distributed and dynamic networks with strict latency requirements.**

***Index Terms*—Deep unrolling, graph neural network (GNN), knowledge-driven resource allocation, weighted minimum mean-square error (WMMSE) algorithm, wireless communication.**

Manuscript received 28 December 2023; revised 7 February 2024; accepted 19 February 2024. Date of publication 22 February 2024; date of current version 9 May 2024. This work was supported by the National Key Research and Development Program of China under Grant 2022YFB2901900. *(Corresponding author: Ruijin Sun.)*

Hao Yang, Nan Cheng, and Ruijin Sun are with the State Key Laboratory of ISN and the School of Telecommunications Engineering, Xidian University, Xi'an 710071, China (e-mail: yh_xd210@xidian.edu.cn; dr.nan.cheng@ieee.org; sunruijin@xidian.edu.cn).

Wei Quan is with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China (e-mail: weiquan@bjtu.edu.cn).

Rong Chai is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: chairong@cqupt.edu.cn).

Khalid Aldubaikhy and Abdullah Alqasir are with the Department of Electrical Engineering, College of Engineering, Qassim University, Buraydah 52389, Qassim, Saudi Arabia (e-mail: khalid@qec.edu.sa; a.alqasir@qec.edu.sa).

Xuemin Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: sshen@uwaterloo.ca).

Digital Object Identifier 10.1109/JIOT.2024.3368516

## I. Introduction

IN THE era of 6G, mobile communication networks are envisioned to provide a wide variety of services and applications, from data-intensive services such as extended reality, reliable and low-latency services, such as autonomous driving and remote surgery, to the soaring intelligent services, such as metaverse and ChatGPT [1]. Furthermore, 6G networks are becoming increasingly complex and dynamic as the emergence and fast development of space–air–ground integrated networks significantly enlarge the network scale and require efficient management of multidimensional resources [2]. This complexity poses a significant challenge to wireless network management, such as resource allocation schemes and task scheduling, to fulfill the service requirements, especially delay-sensitive and reliability services, where a fault or delayed decision may lead to fatal outcomes. Therefore, it is critical to design efficient, responsive, and scalable wireless network management schemes in 6G networks.

Wireless resource allocation plays a pivotal role in network management to allocate multidimensional wireless resources for certain goals, such as maximizing the transmission rate or minimizing the transmission delay or energy consumption. To address the wireless resource allocation problem, a plethora of model-based iterative algorithms, such as iterative water-filling type algorithms [3], weighted minimum mean-square error (WMMSE) algorithms [4], and successive convex approximation algorithms [5], [6], have been proposed based on the convex optimization theory. These algorithms have successfully solved classical resource allocation problems, often with small network scale and static network environments. However, as the network scale increases, the high computational complexity associated with multiple iterations of these algorithms can hardly meet the stringent millisecond-level service requirements.

Due to the efficient real-time computational capabilities, deep learning techniques have found application in diverse areas of wireless communication systems [7], such as UAV-assisted IoT applications [8], spectrum sharing [9], privacy protection [10], resource management [11], and mobile computing offloading [12]. Ye et al. [13] employed deep neural networks (DNNs) for channel estimation and signal detection, achieving efficient handling of channel distortion. He et al. investigated the utilization of convolutional neural network (CNN)-based architectures for channel estimation in beamspace millimeter-wave massive multiple-input–multiple-output (MIMO) systems, surpassing the most advanced compressed sensing-based algorithms. Tang and Wong [14] utilized bidirectional long short-term memory in mobile edge computing systems to handle user scheduling problems in ad hoc networks, effectively minimizing the average age of information. To solve the resource allocation problem for weighted sum rate maximization, multilayer perceptrons (MLPs) [15] and CNN [16] are employed to approximate the WMMSE algorithm, using outputs of algorithms as labels and reducing computational complexity. In [17], the objective function, i.e., the weighted sum rate, is regarded as the loss function, achieving better performance. These studies presented exhibit impressive performance and low inference complexity. However, applying them effectively to radio resource management for an arbitrary number of users faces a significant challenge. Widespread neural networks, including DNN, CNNs, recurrent neural networks, and attention-based Transformer models, are not readily scalable for an arbitrary number of users, as their input and output dimensions must remain constant. Therefore, designing scalable neural network architectures is crucial to effectively managing wireless resources given the dynamic fluctuations in user numbers within mobile applications.

In this context, a graph structure with expandable connecting nodes is more suitable for capturing the dynamic characteristics of wireless networks. By incorporating such graph structure into neural networks, graph neural networks (GNNs) are envisioned as a potential solution to realize scalable resource allocation [18], [19], [20]. A random edge GNN (REGNN) is proposed to enhance scalability and generalization for optimal power control in interference channels [21]. Addressing the limitations of REGNN in heterogeneous agents and multiantenna systems, the interference graph convolutional network is proposed in [22]. Furthermore, a message-passing GNN (MPGNN) is presented for tackling large-scale wireless resource management problems, such as beamforming, user association, and channel estimation [23]. The authors established the equivalence between MPGNN and distributed optimization algorithms, showcasing its performance and generalization capabilities. While GNNs demonstrate scalability, their intrinsic learning approach primarily relies on statistical distributions with poor interpretability, leading to struggles to accommodate varying distributions and necessitating a large amount of training data for a particular distribution. In the context of radio resource management, collecting identical distribution training data is time consuming and costly, and the dynamic nature of wireless networks causes a data set shift that degrades model performance.

Unlike deep learning, model-based iterative algorithms can consistently achieve solutions with theoretical performance guarantees. Integrating domain knowledge in model-based algorithms and neural networks, known as knowledge-driven methods, can simplify the architecture of machine learning systems, decrease training overhead, enhance the interpretability of decisions, and increase their practical utility [24], [25]. As a representative approach of knowledge-driven deep learning, deep unrolling [26] provides an effective solution for integrating domain knowledge with iterative algorithms. The main idea is to design neural networks by utilizing the structure of classical iterative algorithms, incorporating the iterative structure of the algorithm into each layer of the network. This approach treats the network layer as an iteration in the original iterative optimization algorithm and learns the network parameters from the data. Deep unrolling updates the benefits of data-driven learning with the domain knowledge embedded in the iterative algorithm, resulting in improved performance and generalization capabilities. In the field of image processing, deep unrolling has successfully addressed several challenging problems, including image restoration [27], deep image deblurring [28], and image super-resolution [29]. In the field of wireless communications, deep unrolling projected gradient descent algorithm into a neural network has shown better accuracy with lower and more flexible computational complexity in MIMO detection problems [30]. A low-complexity DNN-based MIMO detector was proposed using the multipliers algorithm's deep unrolling alternating direction approach [31]. In [32], the original iterative shrinkage thresholding algorithm is transformed into an unrolled RNN, maintaining the robustness of the algorithm and improving estimation accuracy.

For the WMMSE algorithm unrolled neural networks handling resource allocation problems, in [33], the iterative WMMSE algorithm is unrolled into a layer-by-layer CNN structure, introducing trainable parameters to replace the high-complexity operations in forward propagation, reducing computational complexity for efficient performance and enhancing neural network generalization. In addition to unrolling model-based iterative algorithms such as DNN and RNN, unrolled GNN has the advantages of scalability and interpretability. Hence, a deep unrolling architecture based on GNN is proposed in [34], which only learns key parameters of the WMMSE algorithm with GNN without unrolling iterations in the WMMSE algorithm as layers in GNN. As demonstrated in [35], aligning the GNN architecture with the algorithm can potentially enhance the representation of the GNN and thus reduce the sample complexity. Therefore, it is worthwhile to investigate the GNN unrolling with accurate alignment between GNN layers and algorithm iterations to further improve the performance.

In this article, we propose a novel knowledge-driven GNN architecture-based resource allocation approach for device-to-device (D2D) networks, guided by the WMMSE algorithm, aiming to maximize the weighted sum rate. Specifically, we unroll each iteration of the WMMSE algorithm as one neural network layer within our model. Within each layer, by borrowing the three alternative optimization blocks of the WMMSE algorithm, the proposed UWGNN consists of three specialized neural network modules, i.e., $\text{GNN}_u$, $\text{DNN}_w$, and $\text{GNN}_v$. The design of the $\text{GNN}_u$ and $\text{GNN}_v$ modules is inspired by the neighborhood information aggregation process in the WMMSE algorithm, while the $\text{DNN}_w$ module adopts the intermediate variable update strategy from the WMMSE algorithm. By adopting the structure and domain knowledge of the WMMSE algorithm, our proposed approach retains the algorithm's robustness while effectively reducing its computational delay, decreasing the sample complexity of the neural network, and adapting to various data distributions. The main contributions of the article can be summarized as follows.

1) We propose a novel GNN architecture inspired by the WMMSE algorithm, named UWGNN. This approach is constructed by unrolling the iterative WMMSE algorithms into multilayer neural networks. To align with the three iterative optimization blocks of the WMMSE algorithm, we design three neural network submodules and establish their interconnections. Furthermore, we design the part that requires summation of neighborhood information in the WMMSE algorithm as a GNN submodule, and the part of updating single node features as a DNN submodule.
2) We conduct an analysis to demonstrate the efficiency of the deep unrolling approach employed in the UWGNN. Our study suggests that by aligning the network architecture with the iterative steps of the WMMSE algorithm and decomposing the entire end-to-end learning process into a series of submodules specifically designed for algorithmic components, the UWGNN significantly reduces the model's complexity and the total number of parameters. This modular design not only reduces the sample complexity but also enhances the network's generalization capability, aligning with the theory on algorithm alignment presented in [35].
3) Our proposed UWGNN maintains high-performance in large-scale networks while its processing latency remains 3–4 orders of magnitude lower than that of WMMSE algorithm. Experimental results show that UWGNN performs well in terms of robustness and scalability. Furthermore, the architecture exhibits excellent generalization performance when dealing with diverse data distributions and communication topologies without necessitating retraining for new environments.

The remainder of this article is organized as follows. Section II introduces the communication model and the Formulas for resource allocation problems. In Section III, we present the WMMSE algorithm along with our proposed unrolling network architecture. In Section IV, we demonstrate the effectiveness of our proposed approach through numerical experiments. Finally, Section V summarizes and concludes this article. The notations are illustrated in Table I in this article.

## II. System Model and Problem Formulation

### A. System Model

We consider a D2D scenario consisting of $N$ single-antenna transceiver pairs. Let $p_i$ denote the transmission power that transmitter $i$ uses to send a baseband signal $s_i$ to receiver $i$. That is, the transmission signal $x_i = \sqrt{p_i} s_i$. Then, the received signal at receiver $i$ is

$$y_i = h_{ii} x_i + \sum_{j=1, j\neq i}^{N} h_{ij} x_j + n_i \ \forall i \tag{1}$$

where $h_{ii} \in \mathbb{R}$ represents the direct channel between transmitter $i$ and receivers $i$, $h_{ij} \in \mathbb{R}$ with $i \neq j$ interference channel from transmitter $j$ to receiver $i$, and $n_i \in \mathbb{R}$ denotes the additive noise following the complex Gaussian distribution $\mathcal{CN}(0, \sigma^2)$ with $\sigma^2$ representing the variance of the additive noise in the system. Based on receiver equalization $u_i$, the signal recovered by receiver $i$ can be obtained as $\hat{s}_i = u_i y_i$. Assuming that the signals of different users are independent of each other and receiver noises, the signal-to-interference-plus-noise ratio (SINR) of receiver $i$ is expressed as

TABLE I
Description of Notations

| Notations | Description |
|---|---|
| $N$ | The number of single-antenna transceiver pairs |
| $K$ | The the number of iterations or layers |
| $p_i$ | The power of transmitter $i$ |
| $h_{ii}$ | The direct channel gain from transmitter $i$ to receiver $i$ |
| $h_{ij}\ (i \neq j)$ | The interference channel gain from transmitter $j$ to receiver $i$ |
| $\sigma^2$ | The noise power or variance in the communication channel |
| $\lambda_i$ | Priority of transmitter $i$ in the sum rate problem |
| $\mathbf{z}_i$ | Feature vector of node $i$ |
| $v_i$ | The transmit power coefficient for node $i$ |
| $u_i$ | The receive beamforming coefficient for node $i$, expanded to a feature vector in GNN |
| $w_i$ | The weight coefficient for node $i$, expanded to a feature vector in GNN |
| $\mathbf{H}_{(i,j)}$ | Channel gain matrix |
| $\mathbf{A}_{(i,j)}$ | Edges adjacency feature matrix |
| $\mathbf{C}_{(j,i)}$ | Edge removal probability matrix |
| $\mathbf{D}_{(i,j)}$ | Distance matrix between nodes |

$$\text{SINR}_i = \frac{|h_{ii}|^2 p_i}{\sum_{j\neq i}^{N} |h_{ij}|^2 p_j + \sigma^2} \ \forall i, \tag{2}$$

where $0 \leq p_i \leq p_{\max}$ with $p_{\max}$ denoting as the max transmission power of the transmitter.

Our objective is to maximize the weighted sum rate by optimizing the transmission power, formulated as

$$\begin{aligned} \max_{\mathbf{p}} \ & \sum_{i=1}^{N} \lambda_i \log_2\left(1 + \frac{|h_{ii}|^2 p_i}{\sum_{j\neq i}^{N} |h_{ij}|^2 p_j + \sigma^2}\right) \\ \text{s.t.} \ & 0 \leq p_i \leq p_{\max} \ \forall i \end{aligned} \tag{3}$$

where the weight $\lambda_i$ represents the priority of transmitter $i$ in the sum rate problem, and the power vector is expressed as $\mathbf{p} = [p_1, \ldots, p_N]$.

### *B. WMMSE Algorithm*

Problem (3) is nonconvex, due to the nonconvex objective function. Many iterative algorithms have been proposed to solve it effectively, of which the WMMSE algorithm [4] is the most classical one. The main idea of the WMMSE algorithm is to equivalently transform the weighted sum-rate maximization problem into a problem of minimizing a weighted sum of mean-squared errors (MSEs). Mathematically, let $e_i \triangleq \mathbb{E}_{s,n}[(\hat{s}_i - s_i)^2]$ denote the MSE covariance of the transmission signal $s_i$ and the recovery signal $\hat{s}_i$, the formulated WMMSE problem is expressed as

$$\begin{aligned} \min_{\mathbf{u},\mathbf{v},\mathbf{w}} \quad & \sum_{i=1}^{N} \lambda_i (w_i e_i - \log w_i) \\ \text{s.t.} \quad & 0 \le v_i^2 \le p_{\max} \ \forall i \\ & e_i = (1 - u_i h_{ii} v_i)^2 \\ & \quad + \sum_{j \neq i} \left(u_i h_{ij} v_j\right)^2 + \sigma^2 u_i^2 \ \forall i \end{aligned} \tag{4}$$

where $w_i \ge 0$ is an introduced auxiliary variable indicating the weight for MSE of transmitter $i$, $v_i = \sqrt{p_i}$ and $\mathbf{u} = [u_1, \ldots, u_N]$, $\mathbf{v} = [v_1, \ldots, v_N]$ and $\mathbf{w} = [w_1, \ldots, w_N]$.

It has been demonstrated in [4] that the WMMSE problem presented in (4) is equivalent to the problem of maximizing the sum rate as depicted in (3), with both problems sharing an identical optimal solution denoted by $v_i$. Subsequently, the weighted sum-MSE minimization problem is decomposed into three separate optimization subproblems, each of which can be solved iteratively. Since the subproblems associated with the optimization variable vectors $\{\mathbf{u}, \mathbf{v}, \mathbf{w}\}$ are convex in nature, the algorithm utilizes a block coordinate descent approach to solve the WMMSE problem in (4). More specifically, by sequentially fixing two of the three variables $\{u_i, w_i, v_i\}$ and simultaneously updating the third variable, the WMMSE formula is as follows:

$$u_i^{(k)} = \frac{h_{ii} v_i^{(k-1)}}{\sigma^2 + \sum_j h_{ij}^2 v_j^{(k-1)} v_j^{(k-1)}} \ \forall i \tag{5}$$

$$w_i^{(k)} = \frac{1}{1 - u_i^{(k)} h_{ii} v_i^{(k-1)}} \ \forall i \tag{6}$$

$$v_i^{(k)} = \frac{\lambda_i u_i^{(k)} h_{ii} w_i^{(k)}}{\sum_j \lambda_j h_{ji}^2 u_j^{(k)} u_j^{(k)} w_j^{(k)}} \ \forall i, \tag{7}$$

where $k = 1, \ldots, K$ represents the number of iterations. The detailed WMMSE algorithm is outlined in Algorithm 1.

Although the WMMSE algorithm has demonstrated high performance in various wireless communication systems, some of its shortcomings limit its practical application. First, the algorithm is prone to get trapped in local optima. Additionally, the computational time required for the WMMSE algorithm to converge is significant, particularly in large-scale networks.

**Algorithm 1** WMMSE Algorithm

```
Initialization:$\{v_i\}$ to satisfy $0 \le v_i^2 \le p_{\max}$. The current
  iteration index $k = 1$.
repeat
    Update $u_i^{(k)}$ based on Equation (5);
    Update $w_i^{(k)}$ based on Equation (6);
    Update $v_i^{(k)}$ based on Equation (7);
    $k = k + 1$;
until the convergence condition is met,
Output: transmission power $p_i = v_i^2$.
```

## III. Neural Network Architecture Design Using WMMSE Algorithm

To enhance online computational efficiency while preserving the interpretability of the WMMSE algorithm, we propose a knowledge-driven GNN approach for transmission power allocation in D2D networks. Our proposed method incorporates the unrolled WMMSE algorithm as the message aggregation and combination functions within the GNN. In what follows, we first briefly introduce GNN and the unrolling technique, then present the knowledge-driven GNN.

### *A. Preliminaries*

*1) Graph Neural Networks:* GNNs were initially designed to process non-Euclidean structured graphs data [36]. Unlike traditional neural networks that operate on a fixed grid of inputs, GNNs can handle data with arbitrary connectivity, making them well suited for tasks, such as node classification, graph classification, and clustering detection. Particularly, GNNs operate by iteratively passing messages between nodes in the graph, updating the node representations based on the received information from neighboring nodes. This process can be thought of as a form of message passing, where each node can aggregate information from its local neighborhood and integrate it into its representation.

The aggregation function is primarily utilized to consolidate the neighborhood features of nodes from their neighboring nodes and connected edges. In contrast, the update function is responsible for updating the current node features based on the previous iteration node features and the neighborhood features. Formally, the aggregate and update rules of the $k$th layer at node $i$ in GNNs are, respectively, expressed as

$$\boldsymbol{\alpha}_i^{(k)} = \text{AGGREGATE}^{(k)}\left(\left\{\boldsymbol{\beta}_j^{(k-1)} : j \in \mathcal{N}(i)\right\}\right) \tag{8}$$

$$\boldsymbol{\beta}_i^{(k)} = \text{UPDATE}^{(k)}\left(\boldsymbol{\beta}_i^{(k-1)}, \boldsymbol{\alpha}_i^{(k)}\right) \tag{9}$$

where $\beta_i^{(k)}$ represents the feature vector of node $i$ at either the $k$th layer or after the $k$th iteration. $\mathcal{N}(i)$ is the set of neighbor nodes of $i$, and $\alpha_i^{(k)}$ is an intermediate variable.

*2) Algorithm Unrolling:* Algorithm unrolling, also referred to as deep unrolling or unfolding, represents a technique that bridges the gap between deep learning and traditional iterative models, enabling the amalgamation of domain knowledge and data-driven learning. The fundamental concept of deep unrolling is transforming an iterative inference algorithm into a hierarchical structure that mimics a neural network. Each layer of the neural network corresponds to each iteration of the algorithm. Gregor and LeCun [37] proposed deep unrolling seminal work, which has been used to connect various iterative

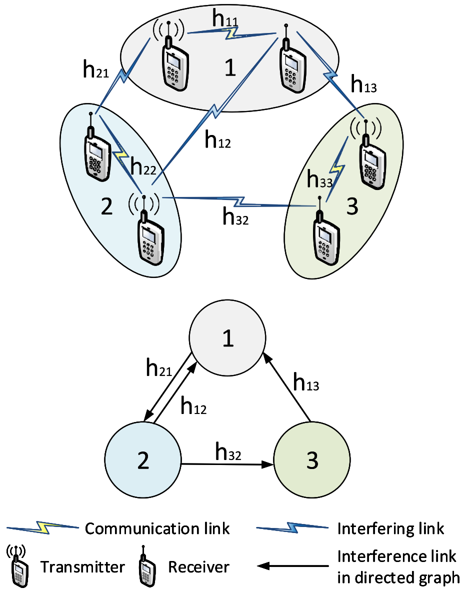


Fig. 1. Graph modeling of the D2D communication network. We consider a pair of D2D users as a node in the graph and the interference link between transmitter $j$ and receiver $i$ as an edge between node $j$ and node $i$.

algorithms, such as those used in sparse coding, to diverse neural network architectures. It is possible to unfold an N-step iterative inference algorithm into an N-layer neural network with trainable parameters. This aims to enhance the model performance by utilizing a computationally lighter neural network.

Unrolled networks boast high parameter efficiency and require less training data than popular neural networks. Therefore, this approach efficiently counters the lack of interpretability normally found in traditional neural networks. This approach provides a systematic link between traditional iterative algorithms and DNNs, leading to efficient, interpretable, and high-performance network architectures.

### *B. Graph Representation of the Weighted Sum Rate Maximization Problem*

Before presenting the knowledge-driven neural network architecture, we consider the D2D network as a directed graph with node and edge features. The modeled directed graph is mathematically represented as $G = (V, E)$, where $V$ is the set of nodes, and $E$ is the set of edges. As shown in Fig. 1, we consider a pair of D2D communication users as a node in the graph and the interference link between transmitter $j$ and receiver $i$ as an edge between node $j$ and node $i$. Typically, node features include attributes, such as node labels, node degrees, and node positions, while edge features may include attributes, such as edge weights, edge types, and edge directions.

For problem (4), features of node $i$ contain channel gain $h_{ii}$ between D2D pair, the weighted factor $\lambda_i$, transmission power $v_i$ of transmitter $i$. To enhance the feature extraction capability of GNN, we add two resource allocation intermediate features $u_i, w_i$ in node $i$. We denote the feature of node $i$ by notation vector $\mathbf{z}_i$, represented as

$$\mathbf{z}_i = [\lambda_i, h_{ii}, v_i, \mathbf{u_i}, \mathbf{w_i}]^\top,\ \mathbf{z}_i \in \mathbb{C}^{(3+2d_{uw})\times 1} \tag{10}$$

where $\lambda_i$, $h_{ii}$, $v_i$ are 1-D variables, $u_i$ and $w_i$ are a $d_{uw}$-dimension vector. The node feature matrix $\mathbf{Z}$ can be expressed as $\mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]$.

For the interference link edge in the graph, the edge feature of node $i$ to node $j$ includes the channel gain of the interfering channel $h_{ij}$. The edges adjacency feature matrix $\mathbf{A} \in \mathbb{C}^{N\times N}$ is given by

$$\mathbf{A}_{(i,j)} = \begin{cases} 0, & \text{if } \{i,j\} \notin E \\ h_{ij}, & \text{otherwise.} \end{cases} \tag{11}$$

By defining nodes feature matrix $\mathbf{Z}$ and edges feature matrix $\mathbf{A}$, the considered D2D scenario is converted as a directed graph

$$\max_{\{\mathbf{v}\}} \sum_{i=1}^{N} \boldsymbol{\alpha} \log_2\left(\mathbf{I} + \frac{|\mathbf{h} \circ \mathbf{v}|^2}{\mathbf{A}^2 \cdot \mathbf{v}^2 + \sigma^2 \mathbf{I}}\right). \tag{12}$$

Based on this, we will develop an effective algorithmic knowledge-inspired GNN architecture.

### *C. Proposed Knowledge-Driven GNN Architecture*

While the non-Euclidean data structure of GNN has the advantage of handling communication topological information compared to other deep learning models, its internal structure design is also essential. Most GNNs typically design their aggregate and update functions with consideration of graph data structures and features, such as permutation invariance and self-attentiveness mechanisms, etc. Although these data-driven design approaches have excellent end-to-end nonlinear mapping performance, they often lack interpretability, behaving like black-box models.

Inspired by the deep unrolling technique, we propose a novel GNN architecture based on the unrolling WMMSE algorithm, named UWGNN. The entire structure of the proposed UWGNN model is inspired by the WMMSE algorithm, which is constructed by unrolling the iterative WMMSE algorithms into multilayer neural networks. Specifically, as shown in Fig. 2, the proposed UWGNN model is a neural network of $K$ layers that mimics the WMMSE algorithm with $K$ iterations. This is achieved by the deep unrolling method, regarding one iteration in the WMMSE algorithm as one layer of the neural network. Then, in each iteration of the WWMSE algorithm, there are three formulas to, respectively, optimize variables $u_i$, $w_i$ and $v_i$. To maintain this interpretable structure, each layer of the UWGNN model also consists of three neural network modules to, respectively, learn corresponding variables. Notice that the updates of $u_i$ and $v_i$ in the WMMSE algorithm, as expressed in (5) and (7), require to collect information from neighbor transceiver pairs to calculate the interference in the denominator. This information collection process is a summation operation of information from neighbor nodes in the graph-structured network topology, which is similar to the aggregation process of GNNs. Motivated by this, therefore, we adopt GNNs to, respectively, learn $u_i$ and $v_i$ in each layer of the proposed UWGNN model, denoted by $\text{GNN}_u$ and $\text{GNN}_v$. As $w_i$ in the WMMSE algorithm is updated by each transceiver's own information, expressed in (6), we adopt MLP

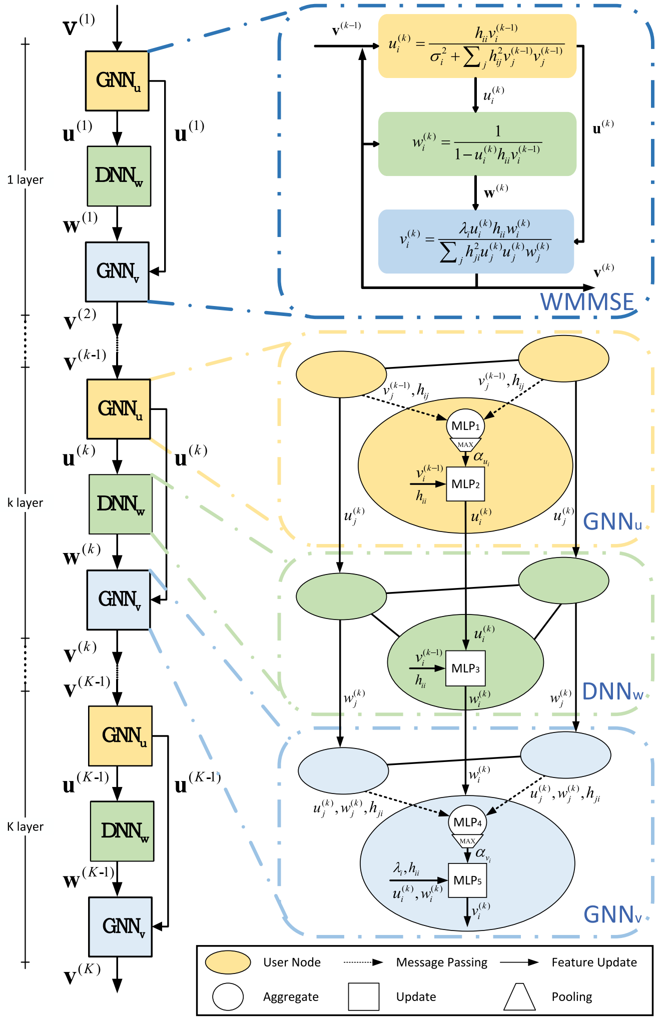


Fig. 2. Knowledge-driven GNN architecture. The proposed GNN model utilizes a three-module architecture inspired by the unrolling WMMSE algorithm. The GNN$_u$ module focuses on calculating $u_i$ and the DNN$_w$ module focuses on calculating $w_i$, while the GNN$_v$ module computes node power $v_i$. This hierarchical feature extraction and node update strategy allows for deep learning of data features and more effective cognition within the GNN architecture.

to fit this function in each layer of the proposed UWGNN model, denoted by DNN$_w$.

The specific design of neural network models in each layer of the proposed UWGNN model, i.e., GNN$_u$, DNN$_w$, and GNN$_v$, is also inspired by the three variable updates of the WMMSE algorithm. The right part of Fig. 2 provides a detailed description of the internal design of the three modules. In particular, the GNN$_u$ module employs designated aggregation and update functions to unroll the formula of the WMMSE algorithm as expressed in (5). Typically, data-driven GNNs pass all node and edge features to neighboring nodes during the message-passing process. This approach requires the GNN's aggregation function to extract useful features from the complete set of features, which may include superfluous inputs. In contrast, GNN$_u$ designs the aggregation function (13) based on the summation part of the WMMSE algorithm in (5), i.e., $\sum_j h_{ij}^2 v_j^{(k-1)} v_j^{(k-1)}$, aligning the input and output of the aggregation function with the WMMSE algorithm and selectively aggregating node power and interference channel features. This approach not only ensures consistency with the computational framework of the WMMSE algorithm but also avoids unnecessary input features. Following (5), the update function (14) integrates the aggregated feature information $\alpha_{u_i}$ with the node features $h_{ii}$, $v_i$ to update the node feature $u_i$. Since $u_i$ is an intermediate variable in the WMMSE algorithm, prioritizing its update simplifies the learning process for the GNN. Thus, the GNN model does not have to learn the complex mapping from input features to output power features directly, which enhances learning efficiency. The DNN$_w$ updates the node feature $w_i$ by using MLP$_3$ from (15) in place of (6) in the WMMSE algorithm. In line with the design philosophy of GNN$_u$, GNN$_v$ unrolls the WMMSE algorithms (7) via aggregation function (16) and update function (17). Corresponding to the summation part of (7) within the WMMSE algorithm, i.e., $\sum_j h_{ji}^2 u_j^{(k)} u_j^{(k)} w_j^{(k)}$, aggregation function (16) only aggregates the $u$ and $w$ features of neighboring nodes, along with the relevant interference channel information $h_{ji}$. Finally, update function (17) employs the node features $\lambda_i$, $h_{ii}$, $u_i$, $w_i$, and the aggregated neighborhood information $\alpha_{v_i}$ to update the node's power feature $v_i$.

Specifically, in (13) of the GNN$_u$ module, MLP$_1$ extracts neighborhood features from neighbor power $v_j$ and path loss $h_{ij}$. To cope with the lack of channel information, we adopt the MAX pooling operation to aggregate the neighborhood information $\alpha_{u_i}$. Feeding $\alpha_{u_i}$ and node feature $v_i$, $h_{ii}$ into the combination function MLP$_2$ in (14), we can obtain the $u_i$ features of the nodes. Similarly, the DNN$_w$ module feeds $u_i$, node feature $v_i$, and $h_{ii}$ into MLP$_3$ to calculate the node feature $w_i$ as indicated in (15)

$$\alpha_{u_i}^{(k)} = \text{MAX}\left\{\text{MLP}_1\left(h_{ij}, v_j^{(k-1)}\right)\right\}, j \in \mathcal{N}(i) \quad (13)$$

$$u_i^{(k)} = \text{MLP}_2\left(h_{ii}, v_i^{(k-1)}, \alpha_{u_i}^{(k)}\right) \quad (14)$$

$$w_i^{(k)} = \text{MLP}_3\left(h_{ii}, v_i^{(k-1)}, u_i^{(k)}\right). \quad (15)$$

In the GNN$_v$ module, we employ MLP$_4$ as the aggregation function, which enables us to gather information on neighboring nodes $\lambda_j$, $u_j$, $w_j$, and the edges $h_{ji}$ feature. The neighborhood information $\alpha_{v_i}$ is aggregated by MAX pooling operation. Finally, we use MLP$_5$ in (16) as the combining function to update the node power information $v_i$

$$\alpha_{v_i}^{(k)} = \text{MAX}\left\{\text{MLP}_4\left(\lambda_j, h_{ji}, u_j^{(k)}, w_j^{(k)}\right)\right\}, j \in \mathcal{N}(i), \quad (16)$$

$$v_i^{(k)} = \gamma\left(\text{MLP}_5\left(\lambda_i, h_{ii}, u_i^{(k)}, w_i^{(k)}, \alpha_{v_i}^{(k)}\right)\right) \quad (17)$$

where $\gamma(x)$ is a sigmoid function to constrain output power into 1, i.e., $\gamma(x) = [1/(1 + e^{-x})]$.

### *D. Training Approach*

The selection of the loss function for neural networks has a significant impact on the overall performance of the network. In supervised learning, the loss function is computed using the labels $\hat{v}_i$ derived from the WMMSE algorithm. However, in realistic scenarios, communication channel and network topology change quickly, and ground-truth labels required for training are difficult to collect within a limited period of time. And it is demonstrated in [17] that the algorithm's output constrains the upper limit of convergence performance.

The unsupervised loss function in (18) uses problem formation to optimize the neural network. Recent research on [17] and [21] has shown that unsupervised training approaches outperform the WMMSE algorithm. Therefore, for unsupervised training of our model, we adopt the optimization objective as the loss function in

$$\mathcal{L}_U(\theta) = -\mathbb{E}\left(\sum_{i=1}^{N}\lambda_i \log_2\left(1+\frac{|h_{ii}v_i(\theta)|^2}{\sum_{i\neq j}|h_{ji}v_j(\theta)|^2+\sigma^2}\right)\right) \qquad (18)$$

where $\theta$ is the learnable parameter of the neural network. In contrast to the WMMSE algorithm that may be trapped in local optima, our unsupervised training methodology employs a global optimization objective as the loss function, which holds the potential to escape local optima and achieve superior performance.

### *E. Efficiency of the Proposed UWGNN*

Hu et al. [38] demonstrated that model complexity influences the representational capacity of neural networks. Typically, to approximate complicated functions, DNNs require a large number of parameters and significant model complexity [39]. To capture the details of the entire iterative sequence of the WMMSE algorithm within a single model structure, an end-to-end MLP may necessitate an extensive number of neurons and nonlinear activation functions. As the number of iterations increases, the ability of the MLP to capture these additional behaviors could become limited. In contrast, a deep unfolding neural network treats each layer as an iterative process of the WMMSE algorithm and further decomposes each layer into three submodules, each corresponding to different parts of the learning algorithm. For the neighborhood summation process in the WMMSE algorithm, it utilizes the inherent message passing and pooling mechanisms of the GNN submodule. This design allows each submodule to learn only a part of the iterative process, significantly reducing the required model complexity and the number of parameters compared to an end-to-end MLP. With fewer parameters to adjust, not only is the deep unfolded neural network easier to train, but it also reduces the sample complexity, i.e., the number of training samples required to achieve a certain level of performance.

As proposed in [35] through the concept of algorithmic alignment, if a neural network can simulate an algorithm with a limited number of simple modules, i.e., it has low sample complexity, then the network is aligned with the optimization algorithm. Effective algorithmic alignment indicates that each iterative step in the algorithm aids learning. Reference [35, Th. 3.5] suggests that to reduce sample complexity and facilitate the training process, neural networks should not include complex cyclic structures. Moreover, [35] mathematically proves that neural networks with better algorithmic alignment are not only easier to train but also possess superior generalization capabilities. The deep unrolling approach of UWGNN cleverly utilizes this concept, decomposing a complex, end-to-end mapping that includes multiple iterations into a series of hierarchical, more easily learned subtasks. Each submodule is designed to learn a part of the mapping relationship in the optimization algorithm, thus effectively reduces both model complexity and sample complexity, enhancing training efficiency and generalization performance.

The computational complexity of neural networks is closely associated with the scale of the model. Herein, we provide a brief analysis of the computational complexity of the UWGNN. The complexity of the GNN is primarily associated with the number of edges $E$ and the number of nodes $N$ in the graph [40]. The complexity each layer of a traditional end-to-end GNN can be expressed as $O(EFH+NHF')$, where $F$ represents the dimension of node features, $H$ represents the width of the hidden layers, and $F'$ represents the size of the output feature representation for each node. Unlike the traditional end-to-end GNN architectures, UWGNN unrolls computations in each layer across three consecutive submodules: GNN$_u$, DNN$_w$, and GNN$_v$. Specifically, the first module GNN$_u$ performs message passing over the graph to learn an intermediate representation with complexity $O(EFH_u + NH_uF'_u)$, where $H_u$ denotes the hidden layers width of GNN$_u$, and $F'_u$ denotes the size of the GNN$_u$ output feature. The subsequent DNN$_w$ module, a dense neural network unit, transforms this representation with complexity proportional to the number of parameters, has the complexity of $O(NH_uF'_uH_w + NH_wF'_w)$, where $H_w$ denotes the hidden layers width of DNN$_w$, and $F'_w$ denotes the size of the DNN$_w$ output feature. Finally, the GNN$_v$ module further refines the representation through another round of message passing on the graph, adding a complexity of $O(EH_uF'_uH_v + NH_vF'_v)$, where $H_v$ denotes the hidden layers width of GNN$_v$ and $F'_v$ denotes the size of the GNN$_v$ output feature. Overall, the total complexity per layer of the UWGNN is given by $O(EFH_u + NH_uF'_u + NH_uF'_uH_w + NH_wF'_w + EH_uF'_uH_v + NH_vF'_v)$. Although the design of UWGNN introduces additional complexity due to the inclusion of multiple message-passing processes, the simplification of tasks within each submodule allows for a reduction in the width of the hidden layers, which in actually decreases the overall computational complexity.

## IV. Numerical Experiments

This section is dedicated to conducting comprehensive numerical tests to validate the effectiveness and generalization capabilities of the proposed knowledge-driven network architecture. We adopt an unsupervised learning method, using the optimization objective of maximizing sum rate as the loss function without additional label data. Therefore, our training data set only needs to construct the connection topology graph and corresponding channel data of the D2D network. In the default experimental scenario, the connection topology graph of the D2D network is a fully connected graph composed of ten nodes. The differences between training samples mainly comes from the random generation of channel data, and both the interference link $h_{ij}$ and the direct link $h_{ii}$ channel data follow the Rayleigh distribution, derived from the complex normal distribution $\mathcal{CN}(0,1)$. The background noise variance $\sigma^2$ is set to 0.1. The initial power characteristics ${v_i}^{(0)}$ of the nodes are randomly initialized based on a Gaussian distribution. Regarding neural network training under this default setup, we employ the Adam optimizer, set to a learning rate of 0.001, and opt for a batch size of 64 samples. Our default training data set comprises 10 000 samples, with the test set containing 2000 samples. We compare UWGNN with established benchmarks and cutting-edge approaches.

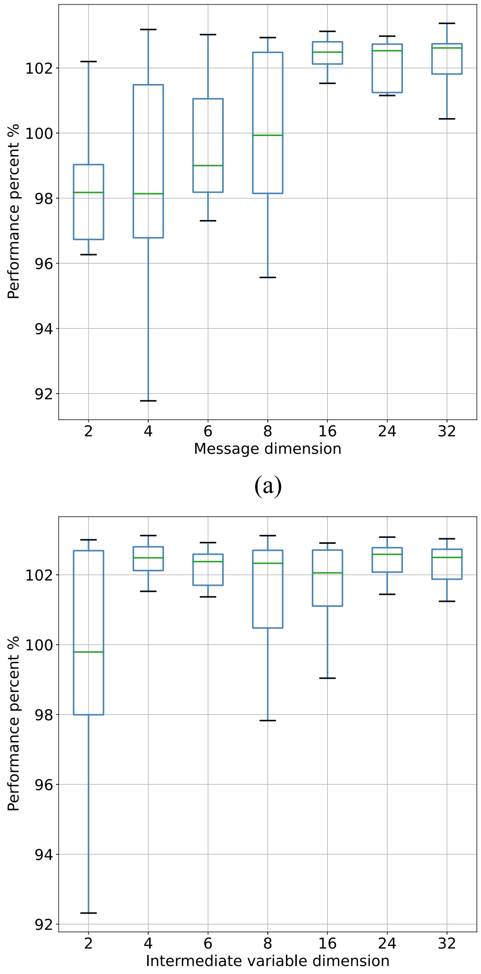
1) *WMMSE [4]:* This is a classical iterative optimization algorithm for weighted sum rate maximization in interference channels. We run WMMSE for 100 iterations with $p_{\max}$ as the initial power setting. The results obtained from this process served as our benchmark measurements.
2) *WCGCN [23]:* This is an unsupervised message passing GNN that uses two MLP networks to aggregate neighbor information and update its power information, and obtain a performance much better than the WMMSE algorithm.
3) *UWMMSE [34]:* UWMMSE is a deep unrolling architecture based on GNNs, primarily utilizing GNNs to learn the iterative step sizes and the weight parameters within the iterative WMMSE algorithm, with the aim of reducing the number of WMMSE iterations while achieving performance on par with established benchmarks. Although UWMMSE incorporates GNNs into iterative optimization, its core computational framework still relies on the WMMSE algorithm.
4) *MLP [15]:* MLP uses the WMMSE output as a training label to supervise and learn a function mapping between the channel state information and the corresponding resource allocation.

To balance the performance and control variables, we set UWGNN, WCGCN, and UWMMSE as 3-layer networks, corresponding to three iterations.

### *A. Selection of UWGNN Hyperparameters*

In this section, our study focuses on the impact of model width on model performance. A too narrow model width will result in too few neurons, making it difficult to extract sufficient sample features and affecting the output performance of the model. Excessively wide model width will lead to too many redundant neurons, increase the amount of model calculation, and require greater bandwidth to transmit intermediate layer information in distributed GNN.

In UWGNN, there are two main factors that affect the width of the model. The first is the aggregated neighborhood messages dimension of $\alpha_{u_i}$ and $\alpha_{v_i}$ in (13) and (16), that is, the dimension $d_\alpha$. It affects the output dimensions of $\text{MLP}_{1,4}$ and the input dimensions of $\text{MLP}_{2,5}$. The second is the updated node feature dimension of $u_i$ and $w_i$ in (14) and (15), that is, the dimension $d_{uw}$. It affects the output dimensions of $\text{MLP}_{2,3}$ and the input dimensions of $\text{MLP}_{3,4}$. In order to observe the impact of $d_\alpha$ and $d_{uw}$ on model performance, we conducted 20 times randomly repeated experiments and selected the WMMSE algorithm output as the baseline. As illustrated in Fig. 3(a), the effect of $d_\alpha$ on model performance was demonstrated. We observed that when the size of $d_\alpha$ was too small, UWGNN performance declined. When $d_\alpha$ increased to 16, UWGNN performance exceeded the WMMSE baseline. Further increases did not impact the model performance significantly. We hypothesized that $d_\alpha$ influences the ability of the node to extract information from neighboring nodes and concluded that as shown in the summation part of (7), $d_\alpha$ must be at least equal to the sum of the dimensions of $u_j$, $w_j$, and $h_{ij}$. Reducing $d_\alpha$ will lead to feature information compression of neighboring nodes that decrease network performance. However, in distributed GNNs, communication resources are utilized during the message passing process. An excessively wide neighborhood messages dimension requires a larger communication cost, such as transmission latency and bandwidth, thereby affecting the accuracy and delay of GNN inference.



(a)

(b)

Fig. 3. Impact of variation in width of GNN on performance. (a) Performance impact of neighborhood messages dimension, $d_\alpha$. (b) Performance impact of node feature dimension, $d_{uw}$.

The dimensions of our two new node features, $u_i$ and $w_i$, have little impact on the performance of UWGNN. As demonstrated in Fig. 3(b), slight performance degradation is observed only when the dimension of the intermediate variable is less than or equal to 2. Since $u_i$ and $w_i$ are 1-D variables

TABLE II
NETWORK COMPUTATION AND PARAMETERS

| Network | FLOPs(G) | Parameters (K) |
|---|---|---|
| UWGNN | 0.590 | 1.906 |
| WCGCN | 0.898 | 2.466 |

TABLE III
SUM RATE PERFORMANCE

| N | UWGNN | WCGCN | UWMMSE | MLP | WMMSE 100 times |
|---|---|---|---|---|---|
| 10 | 102.817% | 102.313% | 100.326% | 94.794% | 105.801% |
| 30 | 102.695% | 102.094% | 97.212% | 71.476% | 102.730% |
| 50 | 102.256% | 101.743% | 96.110% | 60.485% | 102.493% |
| 70 | 102.007% | 101.367% | 93.286% | 27.157% | 101.733% |

in the WMMSE algorithm [see (5)–(7)], adding $d_{uw}$ does not significantly influence the extraction of information from 1-D features. And when there are a large number of irrelevant or redundant features, the neural networks may have difficulty identifying the truly useful features. As stated above, we set $d_\alpha$ to 16 and $d_{uw}$ to 4, establishing the network unit sizes of $\text{MLP}_1$–$\text{MLP}_5$ in (13)–(17) as {5, 8, 16}, {19, 8, 4}, {7, 8, 4}, {10, 8, 16}, and {27, 8, 1}. UWGNN is constructed with three unrolled WMMSE layers, each of which shares parameters.

We use the thop library in Python to compare the computational load and the parameter count of networks, as detailed in Table II. Compared to end-to-end GNN proposed in [23], the deep unrolling operation in UWGNN requires two message-passing processes, increasing the computational complexity. However, the alignment of the algorithm's architecture allows for narrower intermediate layer dimensions, resulting in a reduced parameter count for UWGNN compared to WCGCN. As discussed in Section III-E, the deep unrolling technique, despite introducing additional modules and computational steps, reduces the scale of each submodule due to the simplification of the learning content. In each module of UWGNN, the width of the hidden layers has been decreased from 32 dimensions in WCGCN to eight dimensions. The optimization of node features and the width of hidden layers has enabled UWGNN to strike a delicate balance between computational efficiency and model performance. As shown in Table II, UWGNN not only maintains performance but also has fewer model parameters and requires less computational effort in terms of FLOPs compared to WCGCN.

### B. Sum Rate Performance

We compare the sum rate performance obtained by UWGNN with other approaches, as shown in Table III. To determine the upper bound, we ran the WMMSE algorithm 100 times for random power initialization and selected the best performance. As shown by the results, UWGNN and WCGCN consistently achieve performances near the WMMSE benchmark. Notably, UWMMSE depends on the WMMSE algorithm for its output power determination, which impacts its performance as the computational complexity increases. MLP to capture iterative processes can be complex and may necessitate a large number of training samples. To ensure effective model training in this research, we increased the training data of MLP tenfold. However, as the user numbers escalates, the performance of the MLP deteriorates sharply. This underscores the difficulty of encapsulating iterative optimization challenges within a direct end-to-end neural network, aligning with our earlier assumptions in Section IV. In comparison, both UWGNN and WCGCN display remarkable stability in performance, closely mirroring the WMMSE's optimal results irrespective of rising user numbers. Our findings suggest that message passing GNNs are preferable and more effective than other approaches for the iterative optimization problems.

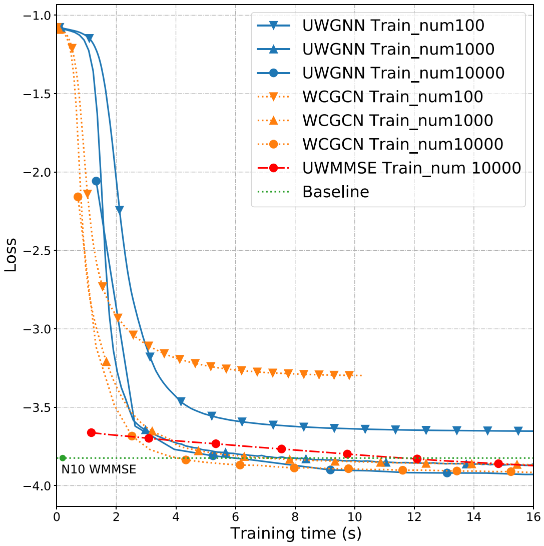


Fig. 4. Network convergence speed comparison.

### C. Time Efficiency Comparison

Model performance is essential, but so is its time efficiency. This section evaluates the convergence time of various models during training and computation. As depicted in Fig. 4, we examined the convergence time using varying training sample sizes. Each test was fixed at 2000 samples, and each experiment was conducted 50 times to calculate the average convergence time. Notably, UWGNN and WCGCN converge swiftly across all sample sizes. The UWMMSE model, which learns the step size parameter and utilizes the WMMSE algorithm for iterative output, shows promising initial performance but exhibits a slower linear convergence rate. When sample size is limited, our model outperforms the WCGCN in terms of convergence. This superiority arises because data-driven neural network models can be severely impacted when sample sizes inadequately represent the data distribution. Yet, our model, due to its algorithmic knowledge, remains consistent even with limited samples. Although our model aligns with WCGCN in terms of convergence time for larger sample sizes, it has a more intricate network structure, necessitating more backpropagation cycles. This characteristic is also reflected in our previous forward computation time table. Experimentally, our model converges as swiftly as WCGCN, underscoring the merits of our knowledge-centric approach that refines the model design and expedites convergence. Notably, our model's adaptability is evident as it demands fewer samples when

TABLE IV
COMPUTATION TIME (S) OF DIFFERENT METHODS

| $N$ | 10 | 30 | 50 | 70 |
|---|---|---|---|---|
| UWGNN | 0.01815 | 0.02280 | 0.02169 | 0.02599 |
| WCGCN | 0.00848 | 0.01061 | 0.00991 | 0.01646 |
| UWMMSE | 0.18949 | 0.24035 | 0.54055 | 1.16788 |
| WMMSE | 4.26259 | 40.52869 | 111.70453 | 188.00520 |

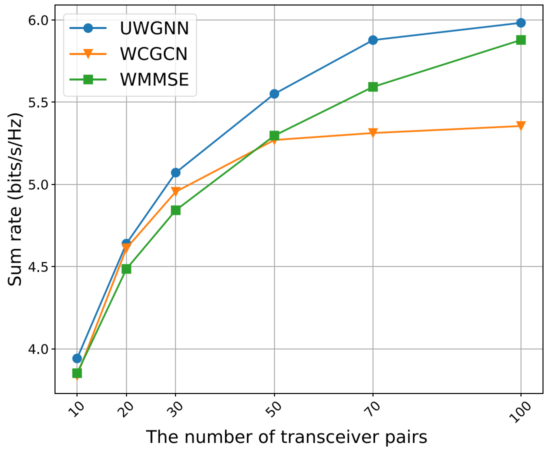


Fig. 5. Scalability comparison.

trained for novel scenarios, significantly cutting down sample collection.

Table IV presents the computation time of various methods across different network sizes. In this experiment, we utilized 10 000 sample groups. The table illustrates that as the network size expands, WMMSE's computation time rises markedly, while the growth for UWGNN and WCGCN remains comparatively modest. UWGNN's longer computation time, relative to WCGCN, can be attributed to its more intricate network architecture. Even though the UWMMSE model adopts an optimized iterative step-length WMMSE algorithm, it is still unable to curtail the rise in computation time with increasing user size. This implies that traditional optimization algorithm-based methods may be less efficient for larger networks. In contrast, UWGNN demonstrates its stability in adapting to various user sizes without the need to proportionally increase computational burdens.

### *D. Scalability Comparison*

In order to assess network scalability, both UWGNN and WCGCN were independently trained over 30 epochs until convergence was achieved in a scenario with 20 users. Subsequently, these trained networks were transferred to new scenarios featuring varying numbers of users, without the need for additional training. As shown in Fig. 5, both GNN networks have good convergence performance for a smaller number of ten user scenario. However, when the number of users increases to 50, the WCGCN network can no longer exceed the performance of the WMMSE baseline, while our network can still exceed the baseline performance. As the number of users reaches 100 and the user connection density intensifies, the WCGCN can no longer achieve the baseline performance, while our network architecture still achieves the baseline performance.

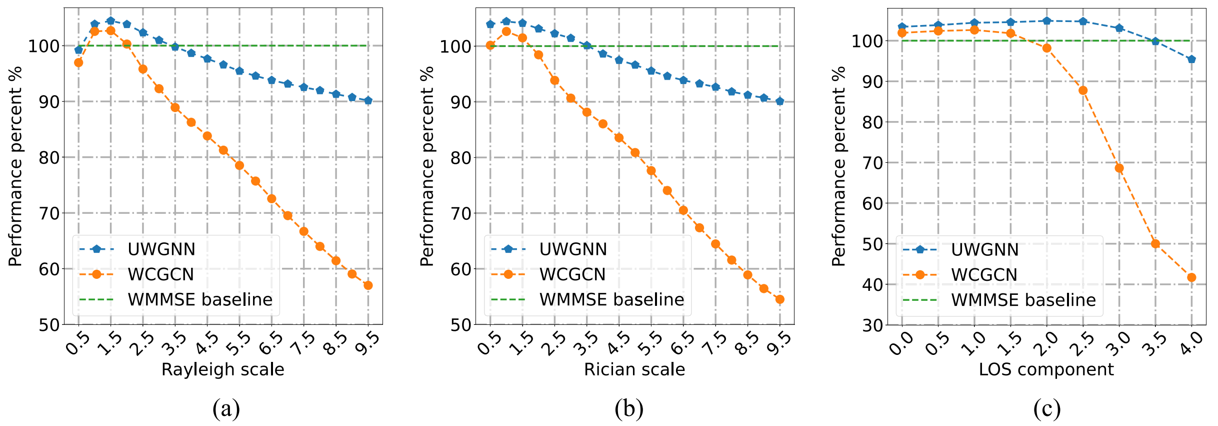
After our experiments, we found that the scalability of GNN comes from the smoothing operation of the max pooling layer on the features of neighboring nodes. When the user dimension changes, the pooling layer uses SUM(.), MAX(.), or MEAN(.) functions to extract the feature information of neighbor nodes and edges. Experimentally, it is found that the MAX function works best for the problem in this article. Moreover, our network architecture undergoes two rounds of MAX pooling because of two information aggregation operations; thus, it is better suited for scaling to diverse scenarios with varying user number densities.

### *E. Channel Distribution Generalization*

The generalization performance of the network is tested using data sets with varying channel distributions. As the user moves, the scattered channel may convert into a direct channel, leading to a potential shift in data distribution from Rayleigh to Rician distributions in the communication scenario. To ascertain the model's generalization ability, we adjusted the sample distribution of the channel in the test set. The initial training set consisted of Rayleigh channels with a mean of 0 and a variance of 1. As shown in Fig. 6(a), we modified the variance of the Rayleigh channel within the test set. During minor variance changes, UWGNN and WCGCN both exhibited generalization capabilities. Nonetheless, as the variance gap widened, WCGCN's performance deteriorated noticeably, signifying its limitations in adapting to data distribution shifts. With the integration of a knowledge-driven network grounded on the WMMSE algorithm, our network architecture demonstrated superior generalization compared to WCGCN, maintaining robust adaptability amidst diverse data distributions. Remarkably, our model sustained roughly 90% of its performance even amidst substantial variance alterations. In Fig. 6(b), we introduced the line-of-sight (LOS) component to the Rayleigh channel by increasing the channel mean to 1, thus transforming its distribution into a Rician distribution. Following this, we altered the variance of the Rician distribution. The experimental results underscored the continued robust generalization performance of our network. Fig. 6(c) illustrates the impact of modifying the strength of the LOS component within the Rician channel. Our model maintained satisfactory performance under varied direct path strengths, facilitating a seamless transition between scattered and direct channels—a critical feature ensuring the model's generalization ability in a mobile environment.

### *F. Communication Topology Generalization*

The topology of a communication network changes with the movement of users and the addition or removal of nodes, which affects the size of the node degree in the graph and thus impacts the performance of GNNs. To evaluate network generalization in adapting to communication topology changes, we generated a connection weight matrix as depicted in (19). The edges with weights lower than the probability of losing connection $\eta_{lc}$ were removed to simulate sparse connections, allowing us to convert between fully and sparsely



(a) (b) (c)

Fig. 6. Channel distribution generalization comparison. (a) Variance alteration in Rayleigh channel. (b) Transition from Rayleigh to Rician channel and variance shift. (c) Adjustment the strength of LOS component.

connected interference graph data sets

$$C_{(i,j)} = \begin{cases} \mathbf{0}, & \text{c}_{\text{ij}} < \eta_{\text{lc}} \\ \mathbf{1}, & \text{otherwise} \end{cases} \quad (19)$$

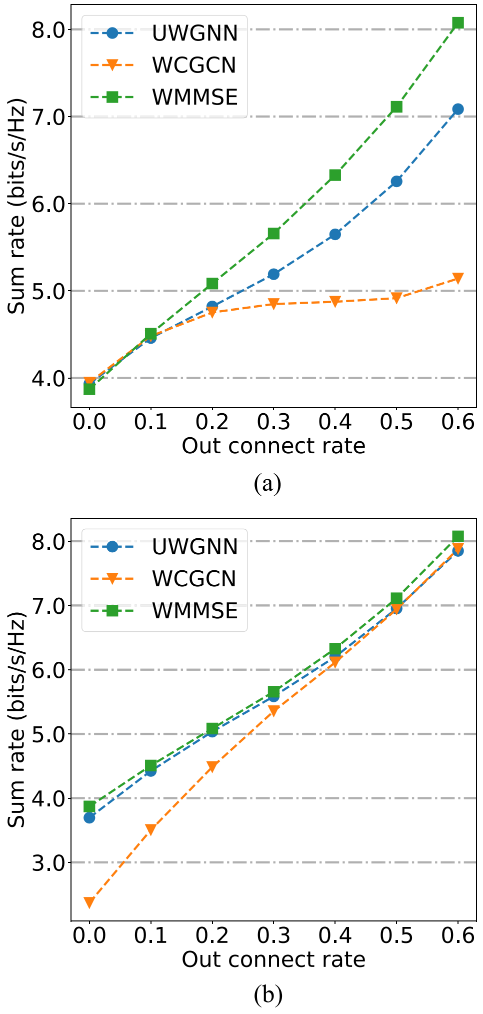
$$\hat{A}_{(i,j)} = A_{(i,j)} \odot C_{(i,j)} \quad (20)$$

where $c_{ij}$ followed standard Gaussian distribution. $\hat{A}_{(i,j)}$ is a new adjacency matrix of communication graph.

We experimented with two training and testing directions: from dense to sparse, and from sparse to dense. First, we trained the network on a fully connected graph data set with ten pairs of users, migrating the test to sparsely connected graph data. As shown in Fig. 7(a), our network degrades in performance as the communication topology becomes more sparse, but still maintains a generalization performance of over 85%. In contrast, the performance of WCGCN degrades more significantly as the topology of the communication network changes. Second, we train the neural network on a training sample with $\eta_{lc}$ of 0.6 and gradually decrease the $\eta_{lc}$ value on the test data, making the sparse connected graph into a fully connected graph. The results are shown in Fig. 7(b). With increasing communication topology density, our network performance still closely follows the WMMSE algorithm performance, but the traditional end-to-end GNN performance decreases as increasing interference density.

### G. Mobile Generalizability Performance

The previous experiment only changed the link structure of the communication topology, however, as wireless devices usually move, the communication channel gain changes as well. In this experiment, we distribute N-transmitters uniformly in the space of [1000 m × 1000 m], and the corresponding receivers are distributed around the transmitters with distances obeying uniform distribution $\mathbf{U}(30, 90)$. Then, we let each receiver device move randomly with speed $S$. The change of position of the receiver obeys a 2-D Gaussian distribution $N(0, 0, S, S, 0)$. At each time step, we update the position of the nodes according to the predetermined speed and direction and subsequently compute the distances between nodes. According to the logarithmic distance path-loss model, in the far-field region of the transmitter, the path loss at any distance $d$ is

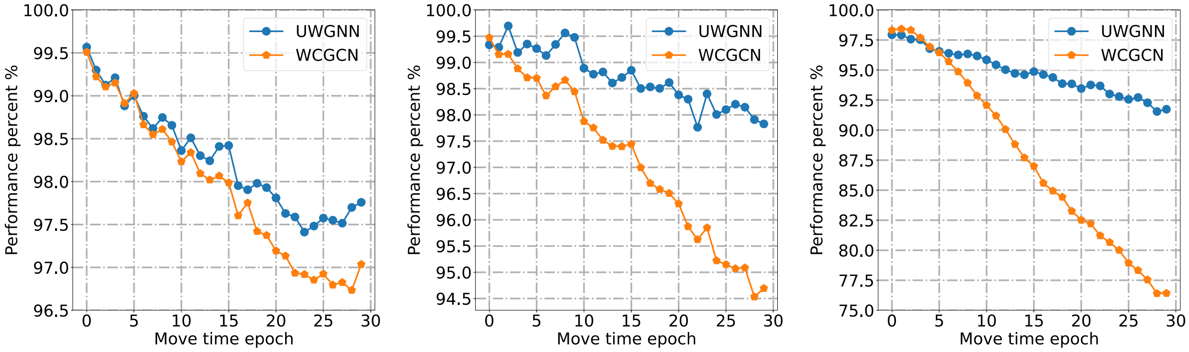


(a)

(b)

Fig. 7. Performance comparison of changing communication network topology. (a) Dense to sparse. (b) Sparse to dense.

$$\mathrm{PL(dB)} = \mathrm{PL}_0 + 10 \cdot n \cdot \log_{10}\left(\frac{d}{d_0}\right) + X_\sigma \quad (21)$$



(a)

(b)

(c)

Fig. 8. Performance comparison of changing communication network topology. (a) Speed 50 m/s. (b) Speed 100 m/s. (c) Speed 200 m/s.

where $\text{PL}_0$ is the path loss at a distance $d_0$ from the transmitter, $n$ is the path-loss index. In free space, the value of n is set to 2. $X_\sigma$ is a Gaussian random variable representing the shadowing effect. Notably, when the distance between any two nodes exceeds 1000 m, the interference link path loss is higher than 60 dB. This level of path loss means that the interference signal has little or no impact on the receiver. Therefore, considering the WMMSE algorithm based on matrix operations, we set the corresponding channel coefficient in matrix $H$ to 0, symbolically representing the disconnection of the link between these two nodes. When the user mobility makes the link distance within 1000 m, the channel coefficient matrix will be recovered as a connected link.

In Fig. 8, we test how the different user movement speeds affect the network generalization. When the device moves at low speed, the communication topology and channel gain change slowly, and the distribution of training and test data do not differ much. So, both GNNs can maintain more than 95% performance. However, when the speed of device movement gradually increases, and the test data changes more drastically, the end-to-end GNN has difficulty in adapting to the new distribution and the performance degrades. Due to the incorporation of algorithmic knowledge, our proposed GNN excels in its generalization capabilities, especially in dynamic environments involving user mobility. This advantage equips the UWGNN to seamlessly adapt to evolving communication scenarios and maintain the communication rates under varying conditions.

### *H. Performance Comparison of Partial CSI*

Although UWGNN shows certain robustness in mobility scenarios, it is still challenging to obtain complete CSI in highly dynamic or ultradense scenarios. Therefore, we conduct experiments to assess the sum rate performance of different approaches under the partial CSI (PC) scenario, where the CSI of the interference link is acquired incompletely. In experiments, the power allocation scheme computed by the WMMSE algorithm under the full CSI (FC) scenario served as the baseline, denoted by FC-WMMSE. The power allocation schemes inferred by other approaches under the PC scenario were, respectively, labeled as PC-UWGNN, PC-WCGCN, and PC-WMMSE. These schemes were then applied in the FC scenario to obtain the performance of real sum rate. For PC-UWGNN and PC-WCGCN, the neural networks are trained on the FC scenario and tested on the PC to learn the power allocation strategies.

In Fig. 9, within the PC scenario, we randomly discard the CSI of some interference links. As the proportion of CSI loss increases, the PC-WMMSE algorithm, based on iterative optimization, struggles to accurately estimate interference, leading to a decline in communication rate. This decline shows a linear relationship with the increasing proportion of loss CSI. The data-driven WCGCN continues to learn based on the statistical distribution of training samples. When confronted with PC, the distribution of model input will change. This results in a distribution shift in the model's output power allocation scheme, which no longer matches the CSI in the real scenario, thereby noticeably reducing the communication rate performance. In comparison, our UWGNN model in PC scenario demonstrates superior adaptability and robustness. On one hand, unlike traditional data-driven GNNs, PC-UWGNN, designed based on the deep unrolling method, does not rely entirely on data distribution for feature extraction. On the other hand, compared to the PC-WMMSE algorithm, PC-UWGNN utilizes the neural network's memory capability to infer and supplement missing parts of CSI to some extent. As a result, UWGNN experiences a slower performance decline in the face of PC. These characteristics significantly advantage UWGNN in scenarios with PC.

The performance of GNNs in the face of PC scenario surpasses that of traditional optimization algorithms, primarily due to their ability to effectively adapt to the incompleteness of information through pooling functions in neighborhood feature aggregation. For instance, the use of a MAX pooling strategy allows the selection of the most significant features from all neighboring nodes, enabling GNNs to maintain robustness even when PC data is missing. To further validate this theory, our experiment depicted in Fig. 10 specifically zeros out the smaller elements in each column of channel matrix $\mathbf{H}$, simulating real-world scenarios of missing CSI due to longer communication distances (or smaller channel gains). The WMMSE algorithm in PC scenario, which needs to sum up neighborhood information, struggles to maintain

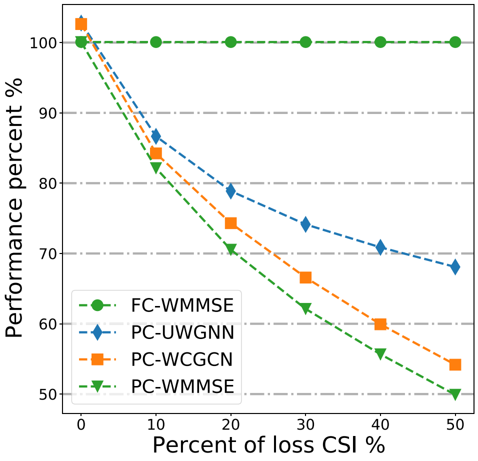


Fig. 9. Random removal of interfering link CSI.

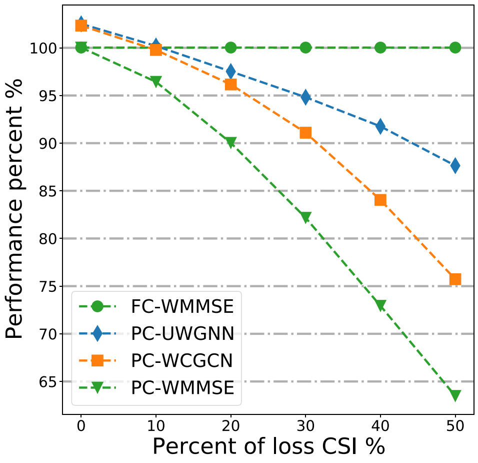


Fig. 10. Targeted removal of lower gain interfering links CSI.

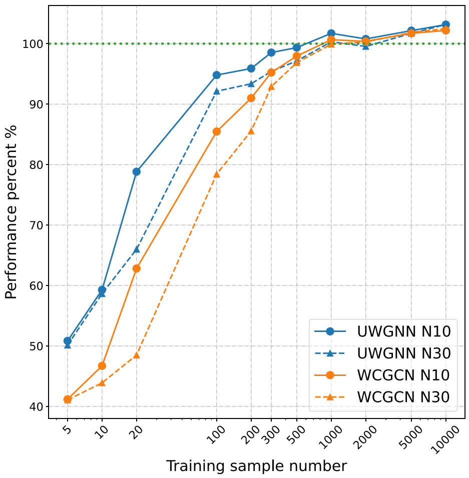


Fig. 11. Network sample complexity comparison.

performance at baseline levels even with the loss of only the smallest CSI in the neighborhood. However, PC-UWGNN and PC-WCGCN, both employing MAX pooling strategies, are able to maintain over 90% performance even with the loss of a small portion of CSI.

### *I. Sample Complexity*

In [35], the concept of network sample complexity and algorithm alignment is introduced, which shows that the higher algorithm alignment, the lower network sample complexity. To compare their network sample complexities, UWGNN and WCGCN are trained using different training samples until they attain convergence, and their performance is tested on 2000 test samples. Analysis of Fig. 11 shows that our network performs well on small training data sets with different numbers of users, indicating that it does not depend on a large sample size to learn the statistical distribution of the data, but to learn the structure and iterative calculation of the WMMSE algorithm. Additionally, the performance improves faster with an increase in sample size, indicating that our network aligns more effectively with the algorithm than the conventional GNN network.

## V. Conclusion

In this article, we have proposed a novel knowledge-driven approach based on the WMMSE algorithm-inspired GNN to solve the resource allocation problem in wireless networks. Compared with current approaches, our approach has exhibited unique advantages in scalability and data generalization. Moreover, we have introduced a hypothesis for the validity of the GNN unrolling approach. Going forward, we plan to extend this work to other wireless resource allocation problems, such as bandwidth allocation, beam assignment, etc., and further develop our results to better guide the design of unrolling approaches. We expect that neural networks with knowledge-driven architecture will be significant in the future of wireless communication networks.

## References

[1] C. Zhou et al., “Delay-aware IoT task scheduling in space–air–ground integrated network,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–6.

[2] N. Cheng et al., “6G service-oriented space–air–ground integrated network: A survey,” *Chin. J. Aeronaut.*, vol. 35, no. 9, pp. 1–18, 2022.

[3] W. Yu, G. Ginis, and J. Cioffi, “Distributed multiuser power control for digital subscriber lines,” *IEEE J. Select. Areas Commun.*, vol. 20, no. 5, pp. 1105–1115, Jun. 2002.

[4] 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.

[5] J. Papandriopoulos and J. S. Evans, “Scale: A low-complexity distributed protocol for spectrum balancing in multiuser DSL networks,” *IEEE Trans. Inform. Theory*, vol. 55, no. 8, pp. 3711–3724, Aug. 2009.

[6] R. Sun, N. Cheng, R. Zhang, Y. Wang, and C. Li, “Sum-rate maximization in IRS-assisted wireless-powered multiuser MIMO networks with practical phase shift,” *IEEE Internet Things J.*, vol. 10, no. 5, pp. 4292–4306, Mar. 2023.

[7] A. Zappone, M. Di Renzo, and M. Debbah, “Wireless networks design in the era of deep learning: Model-based, AI-based, or both?” *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.

[8] N. Cheng et al., “AI for UAV-assisted IoT applications: A comprehensive review,” *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14438–14461, Aug. 2023.

[9] M. Li, C. Chen, H. Wu, X. Guan, and X. Shen, “Edge-assisted spectrum sharing for freshness-aware industrial wireless networks: A learning-based approach,” *IEEE Trans. Wireless Commun.*, vol. 21, no. 9, pp. 7737–7752, Sep. 2022.

[10] J. Shen et al., “RingSFL: An adaptive split federated learning towards taming client heterogeneity,” *IEEE Trans. Mobile Comput.*, early access, Aug. 30, 2023, doi: 10.1109/TMC.2023.3309633.

[11] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Trans. Netw. Sci. Eng*, vol. 7, no. 4, pp. 2416–2428, Oct.–Dec. 2020.

[12] N. Cheng et al., "Space/aerial-assisted computing offloading for IoT applications: A learning-based approach," *IEEE J. Select. Areas Commun.*, vol. 37, no. 5, pp. 1117–1129, May 2019.

[13] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[14] M. Tang and V. W. S. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.

[15] 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.

[16] W. Lee, M. Kim, and D.-H. 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.

[17] 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.

[18] 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.

[19] X. Wang et al., "Joint flying relay location and routing optimization for 6G UAV-IoT networks: A graph neural network-based approach," *Remote Sens.*, vol. 14, no. 17, pp. 4377–4403, Aug. 2022.

[20] X. Wang et al., "Scalable resource management for dynamic MEC: An unsupervised link-output graph neural network approach," in *Proc. IEEE 34th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, 2023, pp. 1–6.

[21] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, Apr. 2020, doi: 10.1109/TSP.2020.2988255.

[22] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[23] 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. Select. Areas Commun.*, vol. 39, no. 1, pp. 101–115, Jan. 2021.

[24] L. von Rueden et al., "Informed machine learning—A taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 1, pp. 614–633, Jan. 2023.

[25] R. Sun, N. Cheng, C. Li, F. Chen, and W. Chen, "Knowledge-driven deep learning paradigms for wireless network optimization in 6G," *IEEE Netw.*, early access, Jan. 10, 2024, doi: 10.1109/MNET.2024.3352257.

[26] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[27] Y. Chen and T. Pock, "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1256–1272, Jun. 2017.

[28] Y. Li, M. Tofighi, V. Monga, and Y. C. Eldar, "An algorithm unrolling approach to deep image deblurring," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 7675–7679.

[29] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2015, pp. 370–378.

[30] N. Samuel, T. Diskin, and A. Wiesel, "Learning to detect," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2554–2564, May 2019.

[31] M.-W. Un, M. Shao, W.-K. Ma, and P. C. Ching, "Deep MIMO detection using ADMM unfolding," in *Proc. IEEE Data Sci. Workshop (DSW)*, 2019, pp. 333–337.

[32] Y. Shi, H. Choi, Y. Shi, and Y. Zhou, "Algorithm unrolling for massive access via deep neural networks with theoretical guarantee," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 945–959, Feb. 2022.

[33] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative algorithm induced deep-unfolding neural networks: Precoding design for multiuser MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394–1410, Feb. 2021.

[34] A. Chowdhury, G. Verma, C. Rao, A. Swami, and S. Segarra, "Unfolding WMMSE using graph neural networks for efficient power allocation," *IEEE Trans. Wireless Commun.*, vol. 20, no. 9, pp. 6004–6017, Sep. 2021.

[35] K. Xu, J. Li, M. Zhang, S. S. Du, K. Kawarabayashi, and S. Jegelka, "What can neural networks reason about?" in *Proc. Int. Conf. Learn. Represent. (ICLR) 2020*, 2020, pp. 1–18.

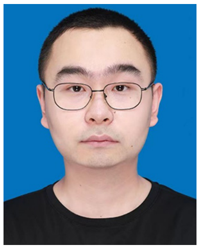
[36] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2009.

[37] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[38] X. Hu, L. Chu, J. Pei, W. Liu, and J. Bian, "Model complexity of deep learning: A survey," *Knowl. Inf. Syst.*, vol. 63, pp. 2585–2619, Aug. 2021.

[39] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the expressive power of deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2847–2854.

[40] 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.



**Hao Yang** (Student Member, IEEE) received the B.E. degree in electronic science and technology from Zhejiang University, Hangzhou, China, in 2016. He is currently pursuing the M.S. degree in communication engineering with Xidian University, Xi'an, China.

His research interests include intelligent networking for 6G, and knowledge-driven, on-demand scheduling of multidimensional resources in wireless networks.

**Nan Cheng** (Senior Member, IEEE) received the B.E. and M.S. degrees from the Department of Electronics and Information Engineering, Tongji University, Shanghai, China, in 2009 and 2012, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2016.

He was a Postdoctoral Fellow with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada, from 2017 to 2019. He is currently a Professor with the State Key Laboratory of ISN and the School of Telecommunications Engineering, Xidian University, Xi'an, Shaanxi, China. He has published over 90 journal articles in IEEE transactions and other top journals. His current research interests include B5G/6G, AI-driven future networks, and space–air–ground integrated networks.

Prof. Cheng serves as an Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and *Peer-to-Peer Networking and Applications*. He serves/served as the guest editor for several journals.

**Ruijin Sun** (Member, IEEE) received the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2019.

From September 2017 to September 2018, she was a visiting student with the University of Waterloo, Waterloo, ON, Canada. From 2019 to 2021, she was a joint Postdoctoral Fellow with Peng Cheng Laboratory, Shenzhen, China, and Tsinghua University, Beijing. She is currently a Lecturer with the School of Telecommunications Engineering and the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China. Her research interests include knowledge-driven wireless resource allocation.

**Wei Quan** (Senior Member, IEEE) received the Ph.D. degree in communication and information system from Beijing University of Posts and Telecommunications, Beijing, China, in 2014.

He is currently a Full Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing. He has coauthored more than 50 papers in prestigious international journals and conferences, including IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, *IEEE Communications Magazine*, IEEE WIRELESS COMMUNICATIONS, IEEE NETWORK, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE COMMUNICATIONS LETTERS. His research interests include reliable transmission in mobile networks, vehicular networks, and Industrial IoT.

Dr. Quan was the recipient of the 2022 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award and a Principle Investigator of the National Key Research and Development Program of China. He is an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Peer-to-Peer Networking and Applications*, *Journal of Internet Technology*, and IEEE ACCESS.

**Rong Chai** (Senior Member, IEEE) received the B.E. and M.S. degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1995 and 1998, respectively, and the Ph.D. degree in electrical engineering from McMaster University, Hamilton, ON, Canada, in 2008.

In 2008, she joined the School of Communication and Information Engineering, Chongqing University of Posts and Technology, Chongqing, China, where she is currently a Professor. She has authored or coauthored more than 90 research articles. Her research interests include wireless communication and network theory.

**Khalid Aldubaikhy** (Member, IEEE) received the B.E. degree from Qassim University, Buraydah, Saudi Arabia, in 2008, the M.A.Sc. degree in electrical and computer engineering from Dalhousie University, Halifax, NS, Canada, in 2012, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2019.

He is currently an Assistant Professor with Qassim University. His research interests include millimeter-wave wireless networks, medium access control, impulse radio ultra-wideband, and millimeter-wave 5G cellular networks.

**Abdullah Alqasir** (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Iowa State University, Ames, IA, USA, in 2019.

He is currently an Assistant Professor with the Department of Electrical Engineering, Qassim University, Buraydah, Qassim, Saudi Arabia. His research interests include wireless networks, next-gen networks, 6G, and green communication.

**Xuemin (Sherman) Shen** (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular networks.

Dr. Shen received the West Lake Friendship Award from Zhejiang Province in 2023, the President's Excellence in Research from the University of Waterloo in 2022, the Canadian Award for Telecommunications Research from the Canadian Society of Information Theory in 2021, the R.A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society (ComSoc), and the Technical Recognition Award from Wireless Communications Technical Committee in 2019 and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He serves/served as the General Chair for the 6G Global Conference'23 and ACM Mobihoc'15, the Technical Program Committee Chair/Co-Chair for IEEE Globecom'24, 16, and 07, IEEE Infocom'14, and IEEE VTC'10 Fall, and the Chair for the IEEE ComSoc Technical Committee on Wireless Communications. He is the President of the IEEE ComSoc. He was the Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a member of the IEEE Fellow Selection Committee of the ComSoc. He served as the Editor-in-Chief for the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, and *Peer-to-Peer Networking and Applications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Member, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.