Graph Neural Network based Beamforming in D2D Wireless Networks

Tianrui Chen[1], Minglei You[2], Gan Zheng[1], and Sangarapillai Lambotharan[1]

[1]Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University
[2]Department of Electrical & Electronic Engineering, University of Nottingham
[1]Email: {t.chen, g.zheng, s.lambotharan}@lboro.ac.uk
[2]Email: minglei.you@nottingham.ac.uk

***Abstract*—An unsupervised graph neural network (GNN) approach is proposed to solve the beamforming design problem in device-to-device (D2D) wireless networks. Instead of directly learning the beamforming, the GNN is utilized to learn primal power and dual variables, and then a beamforming recovery module is applied to convert them to the beamforming. In this way, the overall problem dimension is decreased by a factor of the number of antennas. Additionally, the proposed GNN approach is potential to be generalized to different system settings without retraining when the number of antennas remains unchanged. Simulation results demonstrate that the proposed GNN based beamforming approach achieves superior performance with 10 times fewer samples than the benchmarks, and the running time is reduced down to millisecond-level for 50 pairs of D2D users which is promising for practical applications in D2D wireless networks.**

***Index Terms*—Beamforming, Graph neural network (GNN), Device-to-device (D2D), Resource allocation**

I. Introduction

Spatial beamforming design is an important technique for interference management in multi-antenna scenarios in wireless networks. Beamforming design problems are usually non-convex and mainly rely on numerical algorithms, e.g., the weighted minimum mean square error (WMMSE) algorithm [1], which is proposed to solve the weighted sum rate maximization problem for a multiple-input multiple-output interference channel via iterative minimization of mean square error. The WMMSE algorithm is commonly used for beamforming designs in wireless networks. However such algorithms are computationally complex and time-consuming, hence they cannot meet the real-time requirement in practical wireless networks.

To mitigate these challenges, machine learning (ML) based mechanisms have been proposed to accelerate the approximation of the beamforming optimization problems. A deep neural network (DNN) method was developed in [2] to solve the beamforming designs in multi-cell scenarios. Besides, a deep learning based fast beamforming method was proposed in [3], where the downlink beamforming problem was divided into power allocation and virtual uplink beamforming design. These works cannot embed the network topologies, and they require a large dataset for training, which is usually difficult or even impossible to acquire in real-world wireless networks.

To address these challenges, graph based learning approaches were proposed to approximate the beamforming problems in unsupervised manners. A wireless channel graph convolution network (WCGCN) was proposed in [4] for scalable radio resource management and it was tested on a device-to-device (D2D) beamforming problem. As [4] directly learns the beamforming, it results in high training complexity. In the existing ML based beamforming designs, in order to improve the learning efficiency, the expert knowledge specifically the structure of the optimal downlink beamforming was utilized to convert the beamforming problems to power allocation problems in single-cell scenario [5] [6] and multi-cell scenario [6].

Inspired by the previous works, an unsupervised graph neural network (GNN) approach is employed to approximate the beamforming solutions in D2D networks combining the structure of the optimal beamforming. To be specific, the beamforming is transformed to the primal power and dual variables, then the GNN is applied to learn the mapping from the channel information to these variables rather than the direct beamforming. The main contributions are summarized as follows:

- The domain knowledge is utilized to transform beamforming to primal power and dual variables which enables the GNN to learn these variables instead of directly learning the beamforming. A recovery module is then applied to transform these variables back to the beamforming.
- Compared with directly learning beamforming, the proposed method reduces the number of parameters required to be learned by a factor of the number of antennas.
- Simulation results show that the proposed approach achieves superior performance with fewer training samples than the benchmarks, and it outperforms the state-of-the-art WCGCN by a 20% margin in the studied cases.

The remainder of this paper is organized as follows. Section II introduces the system model and problem formulation for a D2D beamforming problem. The proposed GNN approach for the D2D beamforming is presented in Section III, which includes a graph modeling of D2D networks, an unsupervised GNN approach and a beamforming recovery module. Sections IV demonstrates the simulation results of the proposed approach. Conclusion for this work is drawn in Section V.

II. System Model and Problem Formulation

A wireless network with L D2D pairs is considered, where all D2D pairs share the same spectrum. All D2D pairs are randomly located in a square area with an edge length of d_{area} and each D2D pair is located within a pairwise distance between d_{min} and d_{max}. The set and the indexes of all D2D pairs are denoted by $\mathcal{D} = \{D_1, \ldots, D_L\}$ and $\mathcal{L} = \{1, \ldots, L\}$, respectively. The transmitter and receiver of D_l are represented by T_l and R_l, respectively. Each transmitter T_l is equipped with N_t antennas and each receiver R_l is equipped with a single antenna. A simple network with three D2D pairs is illustrated in Fig. 1.

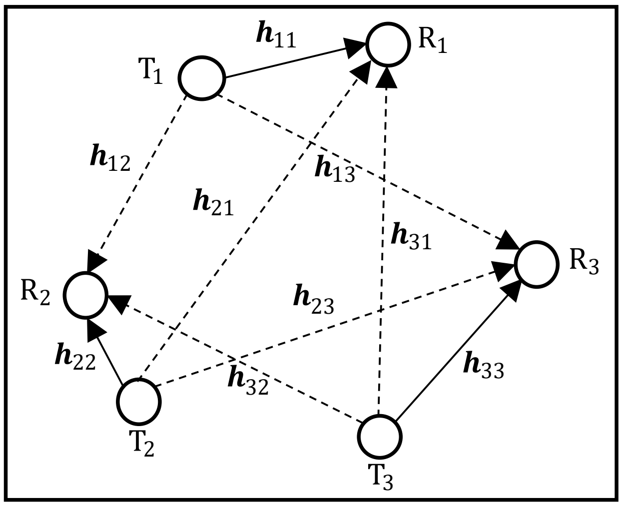

Fig. 1 A 3-pair D2D network.

Let $\boldsymbol{x}_l$ and s_l denote the beamforming and the signal of T_l, respectively. The received signal at R_l is written as

$$y_l = \boldsymbol{h}_{ll}^H \boldsymbol{x}_l s_l + \sum_{k=1, k\neq l}^{L} \boldsymbol{h}_{kl}^H \boldsymbol{x}_k s_k + n_l, \tag{1}$$

where $\boldsymbol{h}_{kl} \in \mathbb{C}^{N_t}$ denotes the channel state from T_k to R_l. $n_l \sim \mathcal{N}(0, \sigma_l^2)$ represents the additive white Gaussian noise. The signal-to-interference-plus-noise ratio (SINR) of D_l denoted by ξ_l is written as

$$\xi_l = \frac{|\boldsymbol{h}_{ll}^H \boldsymbol{x}_l|^2}{\sum_{k=1, k\neq l}^{L} |\boldsymbol{h}_{kl}^H \boldsymbol{x}_k|^2 + \sigma_l^2}, \ \forall l \in \mathcal{L}\,. \tag{2}$$

Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L]^T \in \mathbb{C}^{L\times N_t}$ denote the beamforming matrix. In this paper, a beamforming optimization problem for maximizing the overall sum rate is studied, which is formulated as

$$\begin{aligned} \max_{\mathbf{X}} \quad & \sum_{l=1}^{L} \log_2(1+\xi_l) \\ \text{subject to} \quad & ||\boldsymbol{x}_l||_2^2 \leq P_{\text{max}}, \ \forall l \in \mathcal{L}\,. \end{aligned} \tag{3}$$

where the constraint denotes that each D2D pair has a maximum transmit power of P_{max}. Note that the sum rate is normalized by the channel bandwidth.

III. Graph Neural Networks for Beamforming in D2D Wireless Networks

In this section, the proposed approach is introduced, which mainly consists of a graph representation of the D2D network, an unsupervised GNN, and a beamforming recovery module.

A. Graph Modeling of D2D Networks

A D2D wireless network can be modeled as a fully connected graph where the communication links can be treated as nodes and interference links can be treated as edges, as shown in Fig. 2. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a graph where $\mathcal{V}$ denotes a set of nodes and $\mathcal{E}$ denotes a set of edges. The edge connecting from node v to node u, $v, u \in \mathcal{V}$ can be denoted as $e(v,u) \in \mathcal{E}$. Note that edges are directional in wireless networks.

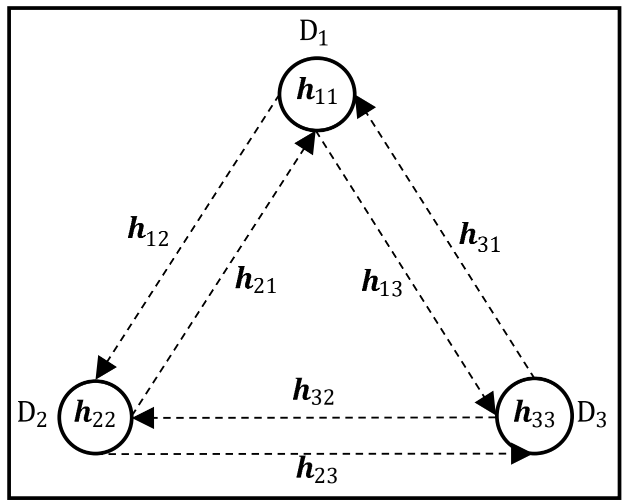

Fig. 2 Graph modeling of the 3-pair D2D network illustrated in Fig. 1.

B. Graph Neural Network

A GNN is utilized to approximate the beamforming optimization problem by learning the mapping from the channel information to the primal power and the dual variables, and then a beamforming recovery module is applied to the learned variables.

The GNN was firstly invented to extend the existing neural networks for handling the data in graph domains [7]. A GNN has layerwise structures and all layers share the same structure, which mainly consists of an aggregation operation and a combination operation. Each node aggregates the features from its adjacent nodes, and it then combines its own node features with the aggregated neighborhood features in each layer. In this work, the channels are modeled as the node and edge features. Let V_v and E_{uv} denote the node feature of node v and the edge feature from node u to node v, respectively. The representation of node v at the m-th layer of the GNN is updated as formulated below [8]:

$$\begin{aligned} \alpha_v^{(m)} &= \varphi_{u\in\mathcal{N}(v)}(\phi(V_u,\ E_{uv},\ E_{vu},\ \beta_u^{(m-1)}))\,, \\ \beta_v^{(m)} &= \psi(V_v,\ \beta_v^{(m-1)},\ \alpha_v^{(m)})\,, \end{aligned} \tag{4}$$

where $\alpha_v^{(m)}$ denotes the aggregated neighborhood feature of node v at the m-th layer. ϕ represents an aggregation operation, e.g., a convolutional neural network (CNN). φ denotes a permutation invariant function, e.g., sum, mean or max, and $\mathcal{N}(v)$ denotes the neighbors of node v. $\beta_v^{(m)}$ refers to the embedding feature of node v at the m-th layer. ψ is a combination operation, e.g., a deep neural network (DNN). The aggregation and combination operations of one node (D_1) at the m-th layer of the GNN is illustrated in Fig. 3 as an example.

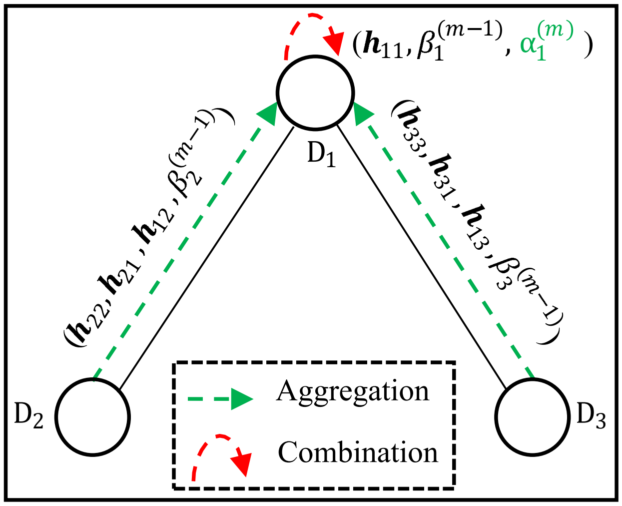

Fig. 3 An illustration of the aggregation and combination of one node at the m-th layer of GNN.

As the dimensions of the GNN is only related to the number of antennas N_t and independent with the number of D2D pairs L, the trained model with small L is potential to be generalized to larger system scales without further training when N_t is invariant.

C. Beamforming Recovery Module

This module aims to recover the D2D beamforming from the learned primal power and dual variables. The beamforming problem in (3) is non-convex and usually difficult to obtain the optimal solution. The optimal beamforming $\boldsymbol{x}_l^*$ is highly structured which is written as [9]

$$\boldsymbol{x}_l^* = \sqrt{p_l}\frac{(\boldsymbol{I}_{N_t} + \sum_{k=1,k\neq l}^{L} \frac{q_k}{N_0}\boldsymbol{h}_{kl}\boldsymbol{h}_{kl}^H)^{-1}\boldsymbol{h}_{ll}}{||(\boldsymbol{I}_{N_t} + \sum_{k=1,k\neq l}^{L} \frac{q_k}{N_0}\boldsymbol{h}_{kl}\boldsymbol{h}_{kl}^H)^{-1}\boldsymbol{h}_{ll}||}, \forall l \in \mathcal{L}, \tag{5}$$

where p_l denotes the primal power of D_l and satisfies $p_l \leq P_{\text{max}}$, which is achieved by the Sigmoid function in the GNN, and q_k denotes the dual variable of D_k and satisfies $\sum_{k=1}^{L} q_k = P_{\text{max}}$, which is achieved by the Softmax function in the GNN. $\boldsymbol{I}_{N_t}$ refers to a $N_t \times N_t$ identity matrix. Through (5), learning beamforming can be transferred to learning the primal power and the dual variables, thus the overall problem dimension is reduced from $2N_tL$ to $2L$.

D. Loss Function

For the unsupervised learning, a negative sum rate is adopted as the loss function which is written as

$$\ell(\theta) = -\sum_{l=1}^{L}\log_2(1+\frac{|\boldsymbol{h}_{ll}^H\boldsymbol{x}_l(\theta)|^2}{\sum_{k=1,k\neq l}^{L}|\boldsymbol{h}_{kl}^H\boldsymbol{x}_k(\theta)|^2+\sigma_l^2}), \tag{6}$$

where θ denotes the parameters of the GNN and $\boldsymbol{x}_l(\theta)$ denotes the beamforming vector obtained by substituting the learned primal power $p_l(\theta)$ and dual variable $q_k(\theta)$ to (5).

IV. Numerical Results

A distance dependent path loss is adopted to model the large scale fading, and the Rayleigh fading with zero mean and unit variance is employed to model the small scale fading. The main system parameters are given in Table I.

TABLE I System parameters.

Parameters	Values
Edge length, d_{area}	200 m
D2D pairwise distance, $d_{\text{min}} - d_{\text{max}}$	2 – 50 m
Maximum transmit power of D_l, P_{max}	20 dBm
Path loss model	$148 + 40\log_{10}(d[\text{km}])$ [10]
Number of antennas, N_t	3

For the GNN, the real part and imaginary part of the channel are treated as two real numbers, and are fed into the neural network. The adaptive moment estimation (ADAM) [11] optimizer is applied to update parameters of the GNN. The main parameters used for the GNN and training process are listed in Table II.

TABLE II GNN and training parameters.

Parameters	Values
Number of layers of GNN	3
Sizes of ϕ (CNN) in GNN	$\{6N_t$+2, 32, 32, 6$\}$
Sizes of ψ (DNN) in GNN	$\{2N_t$+8, 32, 32, 2$\}$
Learning rate	0.001
Batch size	10

The generation of training samples is implemented by Matlab, and the neural networks are implemented by PyTorch. The performance of the proposed approach is compared against the following benchmarks:

- **WMMSE** [1]: The WMMSE algorithm with 100 iterations is adopted. It also serves as an upper bound. The performances of the ML based schemes are normalized with respect to the WMMSE algorithm.
- **Supervised GNN**: This method employs the same GNN structure and recovery module as the proposed method, but a supervised manner is adopted.
- **WCGCN** [4]: This method directly learns the mapping from the channel information to the beamforming in an unsupervised manner.

A. The Number of Training Samples

The performance comparisons with different number of training samples for $L = 20$ D2D pairs are summarized in Fig. 4.

As indicated in Fig. 4, the proposed unsupervised GNN approach achieves a sum rate of approximately 97% of the WMMSE algorithm with only 1000 training samples, and it maintains stable with increasing number of training samples. The proposed method is sample efficient since the searching space has been reduced by converting the beamforming to the primal power and the dual variables. As observed from Fig. 4, the proposed approach outperforms its corresponding supervised method and the unsupervised WCGCN method by margins of at least 10% and 20%, respectively, in all considered settings. The unsupervised WCGCN with direct beamforming learning requires a larger training dataset for better performance.

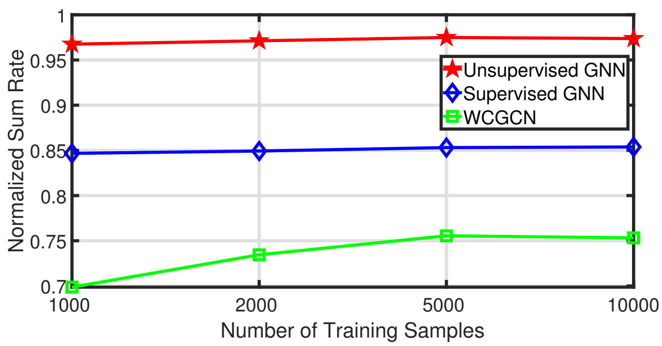

Fig. 4 Performance comparison with different numbers of training samples when $L = 20$.

B. The Number of D2D Pairs

The performance of the proposed approach and benchmarks is evaluated with the scenarios $L \in \{5, 10, 15, 20\}$ as shown in Fig. 5, where 10000 training samples are adopted for each setting.

As observed from Fig. 5, the proposed unsupervised GNN approach achieves a normalized sum rate of around 0.94 for $L = 5$, and it still outperforms the supervised approach and the WCGCN by margins of approximately 9% and 25%, respectively. Besides, the proposed approach maintains good performance with sum rate between 0.96 and 0.975 for $L = 10, 15, 20$.

C. Different Pairwise Distances

The performance of the proposed approach is evaluated on varying pairwise distances with 10000 training samples when $L = 20$.

Table III indicates that the proposed GNN approach can handle different pairwise distances, where it achieves a normalized sum rate of at least 0.95 when the pairwise distances are (2, 50), (10, 50) and (10, 40), while it performs the worst on the scenario with a fixed pairwise distance among all tested settings, which achieves a sum rate of around 0.91. The reason is that the channels are modeled as the node features which are mainly determined by the distances, thus the similar node features will decrease the performance of the proposed method.

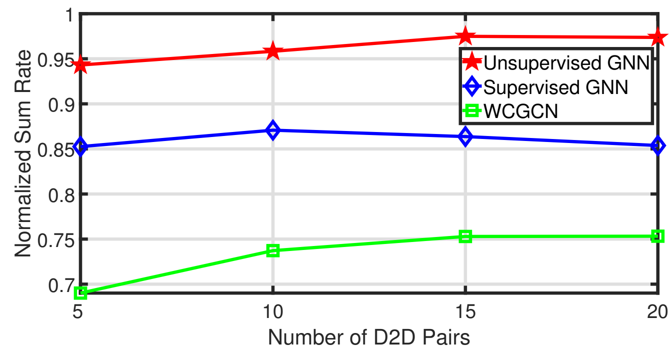

Fig. 5 Performance comparison with different numbers of D2D pairs.

TABLE III Performance of the proposed approach with different pairwise distances.

Pairwise Distances (m)	2 – 50	10 – 50	10 – 40	fixed 30
Normalized Sum Rate	0.9737	0.9503	0.9556	0.9108

D. Scaled-up Systems

The GNN approach is expected to handle large system scales. It is tested on scaled-up networks with $L \in \{50, 75, 100\}$, and the results are presented in Table IV, where the GNN model is trained with 1000 samples for each system scale. The performance of the proposed unsupervised GNN method remains stable with increasing number of D2D pairs, and it achieves a normalized sum rate of approximately 0.98 for $L = 50$ and around 0.97 for $L = 75$ and $L = 100$. The overall performance loss is around 2% - 3% compared to the WMMSE algorithm.

TABLE IV Performance of the proposed approach with scaled-up systems $L \in \{50, 75, 100\}$.

System Scales	50	75	100
Normalized Sum Rate	0.9796	0.9678	0.9689

E. Generalization

The trained model with 10000 samples for $L = 20$ and pairwise distance 2 – 50 m is generalized to larger network scales with the same pairwise distance and the same network scale with different pairwise distances, respectively, without retraining.

Firstly, the trained model is generalized to larger system scales with $L \in \{30, 40, 50\}$ and the performance is given in Table V, which indicates that the proposed method can generalize to larger networks, e.g., the generalization results in 1% and 4% performance loss compared to the training for $L = 30$ and $L = 40$, respectively. It may requires retraining for better performance for larger system sizes.

Secondly, the generalization performance is tested on different pairwise distances as shown in Table VI, where the generalization can achieve almost the same performance with the training of 10000 samples for all considered pairwise distances.

TABLE V Generalizability of the trained model with $L = 20$ to larger network scales $L \in \{30, 40, 50\}$.

System Settings	L=30, 2–50m	L=40, 2–50m	L=50, 2–50m
Training	0.9767	0.9749	0.9796
Generalization	0.9655	0.9350	0.9085

TABLE VI Generalizability of the trained model with $L = 20$ and pairwise distance 2 – 50 m to different pairwise distances.

System Settings	L=20, 10–50m	L=20, 10–40m	L=20, fixed 30m
Training	0.9503	0.9556	0.9108
Generalization	0.9491	0.9553	0.9061

F. Running Time

The inference time performance is evaluated on processor Intel Core i5-9600KF CPU. The time performance of the proposed unsupervised GNN approach with 3 layers is compared to the conventional WMMSE algorithm as shown in Table VII, where the proposed GNN approach is around 1.2×10^4, 2.1×10^4 and 2.2×10^4 times faster than the conventional WMMSE algorithm for $L = 10$, $L = 30$ and $L = 50$, respectively. The proposed method significantly reduces the running time of the WMMSE algorithm from second-level to millisecond-level. Therefore, it is applicable to real-time implementations in wireless networks.

TABLE VII Comparison of average running time.

Number of D2D Pairs	10	30	50
WMMSE (s)	1.5075	12.1037	32.7157
Unsupervised GNN (ms)	0.1290	0.5862	1.4586

V. Conclusion

This work develops an unsupervised GNN approach to approximate the D2D beamforming optimization problem, where the expert knowledge is utilized to convert the beamforming to primal power and dual variables so that the GNN can be adopted to learn the variables rather than the beamforming. Thus, the learning complexity is reduced which leads to higher sample efficiency comparing to learning direct beamforming. Simulation results prove that the proposed method with fewer samples outperforms the considered benchmarks including the corresponding supervised method and the unsupervised WCGCN. Besides, the proposed design shows the promising capability to handle larger network scales. It also demonstrates the potential of generalizability to different system settings without further training required.

VI. Acknowledgement

This work was supported by the CELTIC-NEXT European collaborative project AIMM (ID: C2019/2-5) under the UK Research and Innovation (UKRI) Innovate UK grant number 48160 and the Leverhulme Trust Research Project Grant under grant number RPG-2017-129.

References

[1] 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 Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, Apr. 2011.

[2] 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 Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, Aug. 2018.

[3] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, “Fast beamforming design via deep learning,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1065–1069, Oct. 2019.

[4] 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, no. 1, pp. 101–115, Nov. 2020.

[5] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, “A deep learning framework for optimization of MISO downlink beamforming,” *IEEE Transactions on Communications*, vol. 68, no. 3, pp. 1866–1880, Dec. 2019.

[6] J. Zhang, M. You, G. Zheng, I. Krikidis, and L. Zhao, “Model-driven learning for generic MIMO downlink beamforming with uplink channel information,” *IEEE Transactions on Wireless Communications*, to appear.

[7] 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.

[8] 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 of Things Journal (Early Access)*, pp. 1–1, Jun. 2021.

[9] E. Björnson and E. Jorswieck, *Optimal resource allocation in coordinated multi-cell systems*. Now Publishers Inc, 2013.

[10] 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.

[11] 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.