Scalable Spatial–Geometric–Temporal Graph Learning for Radio Resource Management in Wireless Communication

by

MAHER MARWANI

MANUSCRIPT-BASED THESIS PRESENTEDTO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE IN PARTIAL FULFILLMENT FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
Ph.D.

MONTREAL, 15 MAY 2026

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Prof Georges Kaddoum, Thesis supervisor
Department of Electrical Engineering, École de technologie supérieure (ÉTS)

Prof Georges Ghazi, Chair, Board of Examiners
Department of Systems Engineering, École de technologie supérieure (ÉTS)

Prof Marcos Dias De Assunção, Member of the Jury
Department of Software Engineering and IT, École de technologie supérieure (ÉTS)

Prof Mai Vu, External Examiner
Department of Computer Science, Tufts University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON 1 MAY 2026

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This doctoral journey began with a simple observation: the wireless networks we rely on every day are becoming more complex than ever. As devices multiply and expectations for high-quality connectivity grow, conventional tools for radio resource management become increasingly strained. This motivated a central question that guided my work throughout this PhD: How can modern machine learning, particularly graph neural networks, be used to design scalable and efficient resource allocation strategies in interference-coupled wireless systems?

The research presented in this thesis was conducted between 2022 and 2026 at the Department of Electrical Engineering of École de technologie supérieure (ÉTS), Montréal, under the supervision of Prof. Georges Kaddoum. The thesis is structured as a thesis by articles: each core chapter corresponds to a published peer-reviewed or submitted publication, and that jointly contribute to a unified objective: developing scalable, adaptive, and robust learning-based frameworks for joint power control and spectrum/channel allocation.

This work would not have been possible without the support and guidance of my advisor, as well as the stimulating environment provided by ÉTS. I am equally grateful to my colleagues and collaborators for the valuable discussions, feedback, and shared efforts that shaped many aspects of this research. Above all, I thank my family and friends for their patience, encouragement, and steady presence throughout the inevitable challenges of doctoral study.

I hope that the methods and perspectives developed in this thesis will be useful to the research community and, more broadly, contribute to the ongoing effort towards creating more efficient, reliable, and inclusive communication networks.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Georges Kaddoum, for his invaluable guidance, encouragement, and trust throughout this doctoral journey. His expertise, vision, and constructive feedback have been my constant source of motivation and have shaped the direction and quality of this work.

I also wish to acknowledge the École de technologie supérieure for providing the infrastructure, resources, and financial support that made this research possible.

On a personal note, I owe my deepest appreciation to my family for their unconditional love, patience, and unwavering support. Their belief in me has been a constant source of strength throughout this journey.

I am equally grateful to my friends, who offered both welcome distractions and valuable perspective when needed. I would like to extend special thanks to my dear roommate, Princy, who shared this Ph.D. journey with me side by side, from its beginning to its completion.

I also wish to express a heartfelt thanks to someone dear to my life, who, although unable to be with me at the end of this journey, played an important role in helping me reach it. None of this would have been possible without your emotional support, Panda.

To everyone who, in ways big or small, contributed to the completion of this thesis: thank you.

Apprentissage de graphes spatial géométrique temporel évolutif pour la gestion des ressources radio dans les réseaux sans fil

MAHER MARWANI

RÉSUMÉ

Les réseaux sans fil de nouvelle génération, tels que la 6G et les communications appareil à appareil (D2D), doivent prendre en charge un nombre toujours croissant d'utilisateurs, de canaux et d'exigences de qualité de service (QoS). Cette complexité croissante rend les approches traditionnelles fondées sur l'optimisation inadaptées au fonctionnement des réseaux à grande échelle en temps réel.

Cette thèse, présentée sous forme d'articles de recherche, étudie l'utilisation des réseaux de neurones à graphes (GNN) pour les tâches conjointes de contrôle de puissance et d'allocation de spectre dans les réseaux sans fil. Trois contributions principales sont proposées :

1. Un modèle GNN CNN capable de traiter des graphes d'interférence non euclidiens afin de permettre une allocation conjointe efficace et évolutive, tout en restant robuste en présence d'informations d'état de canal (CSI) imparfaites.
2. Un cadre d'apprentissage spatial et géométrique non supervisé combinant un autoencodeur variationnel spatial et un autoencodeur variationnel de graphes, permettant la généralisation à des réseaux de taille variable sans réentraînement.
3. Un réseau de neurones à graphes temporel basé sur les événements, qui modélise des graphes dynamiques continus dans le temps (CTDG), permettant l'adaptation aux environnements à forte mobilité et aux topologies de réseau changeant rapidement.

Les résultats expérimentaux montrent que les méthodes proposées surpassent les approches heuristiques ainsi que les références d'apprentissage profond conventionnelles en termes de débit moyen, de respect des contraintes de QoS et de capacité de généralisation, tout en réduisant la complexité de calcul.

Mots-clés: réseaux de neurones à graphes, gestion des ressources radio, allocation de spectre, contrôle de puissance, 6G

Scalable Spatial–Geometric–Temporal Graph Learning for Radio Resource Management in Wireless Communication

MAHER MARWANI

ABSTRACT

Next-generation wireless networks, such as 6G and device to device (D2D) communications, must accommodate an ever growing number of users, channels, and quality of service (QoS) requirements. This increasing complexity makes traditional optimization based approaches inadequate for real-time, large-scale network operation.

This thesis, presented in the form of research articles, investigates the use of graph neural networks (GNNs) for the joint tasks of power control and spectrum allocation in wireless networks. The following three main contributions are proposed:

1. A GNN CNN model capable of processing non-Euclidean interference graphs to enable efficient, scalable joint allocation, while maintaining robustness under imperfect channel state information (CSI);
2. An unsupervised spatial and geometric learning framework combining a variational spatial autoencoder and a variational graph autoencoder, which enables generalization to variable size networks without retraining;
3. An event-based temporal graph neural network that models continuous time dynamic graphs (CTDGs), allowing adaptation to high mobility environments and rapidly changing network topologies.

Experimental results show that the proposed methods outperform both heuristic and conventional deep learning baselines in terms of average throughput, QoS satisfaction, and generalization capability, while also reducing computational complexity.

Keywords: Graph neural networks, radio resource management, spectrum allocation, power control, 6G

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Page

LIST OF ABBREVIATIONS

6G	Sixth Generation of Wireless Systems
5G	Fifth Generation of Wireless Systems
B5G	Beyond 5G
IMT-2030	International Mobile Telecommunications towards 2030 and beyond
RRM	Radio Resource Management
QoS	Quality of Service
CSI	Channel State Information
SINR	Signal-to-Interference-plus-Noise Ratio
RB	Resource Block
D2D	Device-to-Device
OFDMA	Orthogonal Frequency Division Multiple Access
NOMA	Non-Orthogonal Multiple Access
URLLC	Ultra-Reliable Low-Latency Communications
ML	Machine Learning
DL	Deep Learning
DNN	Deep Neural Network
MLP	Multi-Layer Perceptron
CNN	Convolutional Neural Network
RL	Reinforcement Learning

DRL	Deep Reinforcement Learning
GNN	Graph Neural Network
GCN	Graph Convolutional Network
GAT	Graph Attention Network
VAE	Variational Autoencoder
VSAE	Variational Spatial Autoencoder
VGAE	Variational Graph Autoencoder
LRGAT	Left-Right Graph Attention (layer)
TGNN	Temporal Graph Neural Network
TGN	Temporal Graph Network
CTDG	Continuous-Time Dynamic Graph
GA	Genetic Algorithm
MINLP	Mixed-Integer Nonlinear Programming
WMMSE	Weighted Minimum Mean Square Error
MIMO	Multiple-Input Multiple-Output
SISO	Single-Input Single-Output
MRT	Maximum Ratio Transmission
MRC	Maximum Ratio Combining
SVD	Singular Value Decomposition
AP	Access Point

IoT	Internet of Things
UAV	Uncrewed Aerial Vehicle
MEC	Multi-Access Edge Computing
SUMO	Simulation of Urban MObility
CCU	Centralized Control Unit
BS	Base Station

LIST OF SYMBOLS AND UNITS OF MEASUREMENTS

t, $\mathcal{T}$	Time-slot index; time horizon / set of time slots
N / $D(t)$	Number of links (static setting) / number of active links at time t (dynamic setting)
$\mathcal{N}$ / $\mathcal{D}(t)$	Set of links (static) / set of active links at time t
K / $\mathcal{K}$	Number of resource blocks (RBs) / set of RB indices
B / W	Bandwidth of one RB (same concept; the thesis uses both notations)
$L_{\max}$	Maximum number of RBs that a link may occupy per time slot
x_{ik} / ψ_i^k	RB-assignment indicator: 1 if link i uses RB k, else 0
$\mathbf{X}(t)$ / Ψ	RB-allocation matrix (binary); time-varying / generic notation
p_i / p_{ik} / $p_{ik}^{(t)}$	Transmit power (per-link / per-RB / time-varying per-RB)
$\mathbf{P}$ / $\mathbf{P}(t)$	Power vector/matrix collecting powers; time-varying version
$p_{\max}$ / $p_i^{\max}$	Maximum transmit power (common limit / per-link limit)
$\mathbf{1}$	All-ones vector (dimension implied by context)
$\mathrm{SINR}_i^k(t)$	SINR of link i on RB k at time t
$r_i(t)$ / γ_i / R_i	Achievable rate/throughput of link i (same concept across chapters)
$r_{\min}$ / $\gamma_{\min}$ / $R_{\min}$	Minimum-rate (QoS) requirement (same concept across chapters)
σ^2 / $(N_0 B)$	Noise power per RB (also written as noise PSD N_0 times RB bandwidth)
N_0	Noise power spectral density (PSD)
$\mathbf{G}(t)$ / G	CSI tensor (time-varying / generic notation)
$g_{ij,k}(t)$ / g_{ij}^k	Channel power gain from transmitter j to receiver i on RB k

$h_{ij,k}(t)$ / h_{ij}^{k} / $\kappa_{ij,k}(t)$	Small-scale fading coefficient (complex; notation varies across papers)
$\tilde{h}$ / $\tilde{h}$	Estimated (noisy) small-scale fading coefficient
σ_e / γ	CSI estimation error coefficient (both notations appear in the thesis)
$\xi_{ij,k}(t)$ / β_{ij}^{k}	Path-loss / large-scale attenuation term (notation varies)
$\tau_{ij,k}(t)$ / α_{ij}^{k}	Shadowing term (notation varies)
σ / σ_{sh}	Shadowing standard deviation (in dB; notation varies)
$d_{\min}$, $d_{\max}$	Minimum / maximum Tx–Rx distance used in simulations / pairing
w_x, w_y	Width / height of the simulated 2D area (when used)
$v_i(t)$	Velocity of node/link i at time t
f_c, c	Carrier frequency; speed of light
$f_{D,n}$	Doppler shift (e.g., for multipath cluster/component n)
ρ_c, ρ_e	Temporal correlation coefficients (fading process / CSI-error process)
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	Interference graph (nodes $\mathcal{V}$, edges $\mathcal{E}$)
$\mathcal{G}_k$	RB-specific interference graph (when modeling one graph per RB)
$\mathcal{V}(t)$, $\mathcal{E}(t)$	Time-varying node and edge sets (dynamic topology)
$\mathbf{v}_i(t)$ / h_i	Node feature vector (derived from CSI; notation varies)
$\mathbf{e}_{ij}(t)$ / e_{ij}	Edge feature vector (interference features; notation varies)
$\mathbf{z}_i(t)$ / $\mathbf{z}_i$	Learned node/link embedding (time-varying / generic)
$\mathbf{Z}$	Matrix/tensor collecting node embeddings
Z^S, $\hat{Z}^S$	Spatial latent representation (VSAE) / size-aligned spatial embedding

$Z^{\mathcal{V}},\ Z^{\mathcal{E}},\ \hat{Z}^{\mathcal{E}}$	Geometric node/edge latents (VGAE) / pooled/size-aligned edge embedding
$\mu_s,\ \sigma_s$	VSAE latent mean and standard deviation (encoder outputs)
$\mu_{\mathcal{V}},\ \sigma_{\mathcal{V}}$	VGAE node-latent mean and standard deviation
$\mu_{\mathcal{E}},\ \sigma_{\mathcal{E}}$	VGAE edge-latent mean and standard deviation
$d,\ d_S,\ d_{\mathcal{V}},\ d_{\mathcal{E}}$	Embedding/latent dimensions (generic, spatial, node, edge)
$\mathcal{L}(t)$	Set of interaction events at time t
ρ	Event type (e.g., add/update/delete)
$l(t)$	One event tuple (source, destination, features, time, type)
$\mathcal{M}(t),\ \mathcal{M}_i(t)$	Set of messages at time t / messages received by node i
$m_i(t),\ \bar{m}_i(t)$	Message to node i / aggregated message for node i
$\mathcal{S}(t)$	Set of node memory states at time t
$s_i(t),\ s_{\mathcal{G}}(t)$	Node-wise memory / graph-wise (global) memory
T_h	Temporal history window length
$\Delta t_1,\ \Delta t_2$	Time since last interaction / cumulative interaction duration (temporal encoding inputs)
P	Number of temporal embedding update iterations/layers
α	Attention weight (generic notation)
$\phi(\cdot)$	Temporal encoding function
$\omega,\ \omega',\ \omega''$	Learnable temporal-encoding parameters (when used)
$\lambda,\ \lambda_1,\ \lambda_2$	Loss weights (constraint penalty / fairness / throughput weights)

CHAPTER 1

INTRODUCTION

Today, wireless systems are progressing toward the IMT-2030 framework and the 5G-Advanced/6G vision, where densification, aggressive spectrum reuse, and heterogeneous services must coexist under stringent requirements on capacity, reliability, and latency International Telecommunication Union (ITU) (2023). In interference-limited regimes—particularly in device-to-device (D2D) and dense reuse scenarios—the performance of each link becomes tightly coupled to the actions of many other links sharing the same time–frequency resources. Consequently, radio resource management (RRM) must continuously coordinate spectrum/resource-block allocation (i.e., who transmits where) and power control (i.e., how strongly) in a way that scales with network size and remains robust to channel uncertainty and mobility.

On a modeling level, joint power control and spectrum allocation naturally yields mixed-integer, non-convex formulations: discrete RB/channel selections interact with continuous power variables through nonlinear interference-coupled rates. While classical optimization and carefully designed heuristics can provide strong baselines, they typically rely on iterative procedures, scenario-specific tuning, and repeated re-optimization as the environment changes. In emerging regimes where decisions must be refreshed under tight control-loop budgets, this repeated optimization can become incompatible with real-time operation. These limitations have motivated learning-based RRM, where expensive computation is shifted offline and where online decisions are produced by fast inference. However, conventional Euclidean deep networks (e.g., MLP/CNN pipelines) frequently assume fixed input sizes and lack an explicit representation of interference relations, which can hinder scalability and generalization when the number of links, topology, or channel statistics differ from those observed during training.

Graph learning offers a natural inductive bias for interference management: Wireless interactions can be represented as interference graphs where nodes correspond to links and edges capture cross-link interference. Graph neural networks (GNNs) provide permutation-equivariant mappings and support variable-size inference, which makes GNNs well suited to scalable RRM.

However, from a deployment perspective, the following three practical gaps must be addressed, each motivating one step of the thesis progression:

1) Scalability gap: Conventional Euclidean deep networks (e.g., MLP/CNN pipelines) typically assume fixed input/output sizes and do not explicitly encode interference relations, which limits generalization when the number of links or the topology changes. This motivates a graph-based foundation that would enable size-invariant inference for joint allocation.

2) Multi-RB representation gap: Even with graph learning, multi-RB CSI is naturally available as a tensor indexed by transmitter–receiver pairs and RBs. Flattening this tensor into concatenated node/edge features (or reducing it by averaging) can obscure spatial correlations and geometric patterns informative for allocation. This motivates *structure-preserving* representations that would jointly exploit tensor (spatial) and graph (geometric) views.

3) temporal gap: Most graph-based RRM solutions still operate per time slot and treat the network as a sequence of independent snapshots. Real systems evolve continuously due to fading, mobility, and sporadic link activity; ignoring temporal correlation can lead to unstable decisions and degraded robustness under dynamics. This motivates *time-aware* graph learning with event-driven modeling and memory, so historical context informs current decisions.

Together, these three gaps define the scope and the structure of the presented thesis: We progress from scalable graph learning for joint allocation to preserving multi-RB CSI structure and, finally, to temporal memory for dynamic network evolution.

1.1 Motivation

Overall, three trends make scalable and adaptive RRM increasingly essential. First, densification and spectrum reuse strengthen interference coupling: local actions (RB reuse or power boosts) can have network-wide consequences. Second, heterogeneity in deployments implies that a practical solution must generalize across different network sizes and geometries without redesign.

Third, time variation (mobility, fading, and link churn) requires quickly and consistently adaptable decisions, rather than restarting optimization from scratch at every scheduling instant.

In next-generation interference-limited networks, in order to remain valid under mobility and fast channel variations, , resource decisions must be refreshed on tight control loops. When the channel coherence time becomes comparable to the scheduling interval, repeatedly running iterative solvers (frequently with many inner iterations and scenario-dependent tuning) can become incompatible with real-time operation. This motivates a shift in perspective: instead of solving a fresh optimization problem at every decision epoch, we seek policies capable of amortizing computational effort offline and producing high-quality allocations online via fixed-cost inference. In these settings, the key question is no longer only *how optimal* a solution is, but whether it can be generated *fast* and *consistently enough* to track the evolving interference environment.

This thesis adopts the viewpoint that *structure should be built into the learning model*. Interference is relational and naturally graph-structured; multi-RB CSI contains spatial and geometric patterns beyond pure topology; and wireless dynamics are event-driven over time. A learning framework respecting these properties is positioned to improve scalability, robustness, and real-time feasibility.

1.2 Problem Statement

We consider interference limited multi-link wireless networks, with a particular emphasis on D2D network under shared spectrum. At each decision epoch, the controller determines (i) transmit powers for active links and (ii) a spectrum/RB allocation policy (e.g., RB selection per link) to maximize a network utility (e.g., mean rate) while satisfying per-link QoS constraints (minimum rate requirements) and feasibility constraints (power budgets and RB limits). The difficulty stems from the following two sources: (a) mixed decision types (continuous power and discrete RB assignment) and (b) nonlinear interference coupling in achievable rates.

In realistic settings, the problem is further complicated by imperfect CSI, distribution shifts between training and deployment, and time variation caused by mobility and sporadic activity. Therefore, a practical solution must simultaneously satisfy the following conditions:

- **Scalability:** Operating over variable numbers of links and interference graphs without retraining or architectural redesign;
- **Structure preservation:** Using interference topology while retaining spatial/geometric information encoded in tensorized multi-RB CSI;
- **Temporal adaptivity:** Incorporate time evolution (events, mobility, fading) to produce stable decisions as the network changes.

Accordingly, the central problem addressed in this thesis is designing structure-aware learning frameworks that would map time-varying wireless observations (CSI, topology, and events) to joint power and spectrum/RB decisions, to achieve high throughput with QoS compliance while remaining scalable and robust across network sizes and dynamics.

1.3 Research Objectives

To address this problem, the thesis pursues the following objectives:

1. **Size-invariant joint allocation:** Development of learning architectures for joint power control and spectrum/RB allocation that generalize across network sizes and topologies;
2. **CSI-structure preservation:** Learning representations that would retain spatial and geometric correlations in CSI, rather than rely solely on graph reduction;
3. **Temporal modeling beyond snapshots:** Incorporation of time evolution through event-driven or memory-based modeling to support dynamic, continuously changing networks;
4. **Robustness to CSI imperfections and shifts:** Maintaining reliable QoS-aware performance under estimation errors, mobility-induced variability, and deployment mismatch;
5. **Rigorous evaluation:** Benchmarking against optimization and learning baselines, quantifying throughput/QoS trade-offs, and validating scalability and ablation evidence.

1.4 Contributions and Outline

While this thesis is presented as three articles, it is written to be read as a single research story rather than three independent papers. The story begins with a practical requirement: joint power and spectrum/RB allocation must be computed *fast* and *repeatedly* under tight control-loop budgets, while remaining reliable under interference coupling and CSI uncertainty. This motivates learning-based policies with fixed-cost inference. However, making such policies deployable requires the following three missing ingredients that the thesis addresses in sequence: *1) scalability* across variable network sizes; *2) structure preservation* for multi-RB CSI; and *3) temporal consistency* under dynamic network evolution.

Accordingly, Chapter 3 establishes a graph-learning foundation that enables scalable, size-invariant inference for joint allocation by modeling interference as a graph and learning permutation-equivariant mappings. This provides the first deployable step: fast inference that generalizes as the number of links changes. Chapter 4 then resolves a key limitation exposed by this graph-based approach: namely, in multi-RB systems, CSI is naturally tensorized, and collapsing it into flat graph features (or averaging it across RBs) can hide spatial and geometric correlations that matter for allocation. Therefore, we introduce a spatial-geometric framework that preserves complementary tensor and graph structure in the learned representation. Finally, Chapter 5 resolves a limitation shared by both chapter 3 and 4: snapshot-based inference treats each time slot independently. We introduce an event-based continuous-time dynamic graph (CTDG) formulation and a memory-aware temporal GNN, enabling stable and adaptive decisions under dynamic evolution.

The contribution of the present thesis can be briefly summarized as follows:

- **Chapter 3 (Scalability):** We propose a GNN-based framework for joint power control and spectrum allocation that supports permutation-equivariant, variable-size inference with fast online execution;

- **Chapter 4 (Structure preservation):** We outline a spatial–geometric learning approach that preserves multi-RB CSI structure by combining complementary tensor (spatial) and graph (geometric) embeddings for robust generalization;
- **Chapter 5 (Temporal consistency):** We present an event-based temporal graph learning framework (CTDG + memory-aware TGNN) capturing continuous-time evolution and link churn for dynamic RRM.

Thesis-level novelty: The novelty of this thesis lies in developing a progressive graph-learning framework for RRM that addresses three deployment-oriented limitations in the literature. First, it moves from fixed-size Euclidean learning models toward permutation-aware graph inference for scalable joint power and spectrum allocation. Second, it extends graph-based RRM by preserving the complementary spatial and geometric structure of multi-RB CSI, rather than relying only on flattened or aggregated graph features. Third, it moves beyond independent snapshot-based allocation by modeling wireless network evolution through event-based temporal graphs and memory-aware GNNs. Therefore, the thesis contribution is not limited to applying GNNs to RRM, but consists of advancing graph learning for wireless resource allocation along three axes: scalability, CSI-structure preservation, and temporal adaptivity.

The remainder of this thesis is structured as follows Chapter 2 provides the *background and literature review*. In this chapter, we introduce the interference-network setting and CSI fundamentals, which is followed by a survey of classical optimization and learning-based RRM to identify three gaps that motivate the thesis progression, scalability across variable network sizes, preservation of multi-RB CSI structure, and temporal modeling beyond snapshot-based decisions. Chapters 3–5 address these gaps in sequence through the three thesis contributions. Chapter 6 concludes the thesis and discusses deployment-oriented recommendations and future research directions aligned with next-generation requirements.

1.5 Author's Publications

The author's Ph.D. research resulted in the following journal papers, conference publications, and submitted manuscripts.

Journal Papers

1. M. Marwani and G. Kaddoum, "Graph Neural Networks Approach for Joint Wireless Power Control and Spectrum Allocation," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 717–732, 2024, doi: 10.1109/TMLCN.2024.3408723.
2. M. Marwani and G. Kaddoum, "Scalable Spatial and Geometric Learning Approach for Joint Power Control and Channel Allocation," *IEEE Transactions on Wireless Communications*, vol. 23, no. 11, pp. 16976–16991, Nov. 2024, doi: 10.1109/TWC.2024.3449036.
3. M. Marwani and G. Kaddoum, "Event-Based Temporal Graph Neural Network for Radio Resource Management," *IEEE Transactions on Wireless Communications*, 2026, doi: 10.1109/TWC.2026.3654880.
4. M. Marwani and G. Kaddoum, "A Unified Learning Framework for Robust Spectrum and Power Control under Imperfect CSI," *IEEE Transactions on Wireless Communications*, submitted, 2026.

Related Conference Publications

5. M. Marwani and G. Kaddoum, "Predictive Temporal Graph Neural Networks for Power and Spectrum Allocation in D2D Networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2025, doi: 10.1109/ICC52391.2025.11162029.
6. M. Marwani and G. Kaddoum, "Variational Graph Autoencoder-Driven Initialization for Genetic Algorithms in D2D Resource Allocation," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2025.

7. M. Marwani and G. Kaddoum, “Joint Beamforming and Channel Assignment via Complex-Valued Graph Neural Networks in MU-MISO Systems,” in *Proc. IEEE International Conference on Communications (ICC)*, 2026.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

This chapter provides the background required to contextualize the contributions of the present thesis and position them within the literature on radio resource management (RRM) and learning-based optimization for wireless networks. To this end, we first introduce the interference-limited multi-link setting (with emphasis on D2D) and summarize CSI fundamentals, including imperfect CSI. We then formalize the coupled RRM tasks studied throughout the thesis. Next, we review classical optimization families and highlight why their iterative nature becomes problematic under scale, uncertainty, and tight control loops. This is followed by a survey of learning-based RRM (supervised, RL, and unsupervised paradigms) and graph learning and temporal/dynamic graph modeling. Building on this and the thesis narrative introduced in Chapter 1, this chapter concludes with a synthesis of the literature into the key gaps that motivate the research reported in Chapters 3–5.

2.1 Wireless Networks and Interference Setting

This section introduces the wireless setting considered in this thesis and defines key notions required for the subsequent RRM formulations. We focus on interference-limited networks and summarize CSI fundamentals, including the impact of imperfect CSI.

2.1.1 Interference-Limited Networks and D2D Context

Next-generation wireless networks are increasingly shaped by densification and aggressive spectrum reuse, that jointly improve spectral efficiency, but also intensify interference coupling. In interference-limited regimes, the performance of a link is strongly affected by concurrent transmissions, making resource allocation a network-wide problem, rather than an isolated per-link decision Akhtar, Tselios & Politis (2021). This challenge becomes particularly prominent in device-to-device (D2D) communications, where nearby devices communicate directly and often reuse shared resources, leading to strong co-channel interference and tighter coupling between

spectrum allocation and power control Jameel, Hamid, Jabeen, Zeadally & Javed (2018); Gupta, Patel, Gupta, Tanwar & Patel (2022). Therefore, an extensive body of work has used D2D as a representative testbed for studying joint resource allocation under interference constraints Huang, Huang, Xing & Qian (2018); Ramezani-Kebrya, Dong, Liang, Boudreau & Seyedmehdi (2017); Najla, Becvar & Mach (2021).

2.1.2 CSI Fundamentals and Imperfect CSI

RRM decisions rely on channel state information (CSI), which captures propagation effects between transmitters and receivers. A standard view separates large-scale attenuation (path loss and shadowing) from small-scale fading induced by multipath propagation Tse & Viswanath (2005); ITU-R (2009). In multi-link and multi-RB settings, CSI is frequently represented in structured forms (e.g., tensors indexed by transmitter-receiver pairs and RBs), which may contain informative spatial and geometric patterns.

In practical systems, CSI is acquired through a pilot-based procedure. Transmitters periodically send known reference signals (pilots) on the available RBs; receivers estimate the channel coefficients (or suitable summaries such as gains) and then report them to the controller when centralized decisions are required. Depending on the duplexing mode and system design, this reporting can be explicit (feedback) or implicit through uplink measurements. Since pilots consume time-frequency resources and feedback incurs latency and quantization, the controller typically operates with a delayed and noisy CSI view, rather than the instantaneous true channel.

To capture estimation errors, we model the estimated small-scale channel as a noisy version of the true channel: $\tilde{h} = \sqrt{1 - \sigma_e^2}\, h + \sigma_e n$, where n is complex Gaussian noise and $\sigma_e \in [0, 1]$ controls CSI accuracy. This model reflects that estimation quality depends on pilot SNR, pilot overhead, and channel time variation. If the allocation policy is not robust, imperfect CSI, which arises from estimation noise, limited pilot budgets, feedback quantization, and mobility-driven aging, can lead to QoS violations. Therefore, a common evaluation methodology is to compute

allocation decisions using imperfect CSI while measuring performance on the underlying (clean) channels, thereby quantifying robustness to estimation errors He *et al.* (2020a).

With the interference setting and CSI acquisition in place, we next summarize the key RRM tasks studied in this thesis and explain why their joint design is challenging.

2.2 RRM Problem Fundamentals

RRM allocates limited radio resources (spectrum and power) to satisfy QoS requirements while optimizing a network utility (e.g., mean throughput). Here, two foundational and tightly coupled tasks are *power control* and *spectrum/RB allocation*. Their coupling is central in interference-limited networks, since spectrum reuse determines who interferes with whom, and power decisions determine both the desired signal quality and interference leakage Akhtar *et al.* (2021).

2.2.1 Power Control

Power control regulates transmit powers to manage interference and stabilize network performance. In D2D settings, power control has been extensively studied using model-based formulations such as game-theoretic approaches and interference-aware optimization, often aiming to satisfy link-level requirements while controlling cross-link interference Huang *et al.* (2018); Ramezani-Kebrya *et al.* (2017). In broader wireless contexts, power allocation is also coupled with other physical-layer degrees of freedom (e.g., beamforming), where iterative optimization baselines are commonly used for sum-utility maximization Shi, Razaviyayn, Luo & He (2011).

2.2.2 Spectrum/Channel Allocation

Spectrum/channel allocation assigns RBs to links and directly shapes interference topology. In D2D, multi-channel reuse has been studied under both optimization and game-theoretic perspectives. This body of work revealed that, while allowing flexible reuse can improve throughput, it also increases interference complexity Mach, Becvar & Najla (2019); Najla *et al.*

(2021). Practical systems impose resource constraints such as limiting each link to one RB, or allowing only a bounded number of RBs per link per scheduling interval.

2.2.3 Joint Allocation: Coupling, Constraints, and Complexity

Joint power and spectrum allocation is challenging because it combines discrete decisions (RB assignments) with continuous decisions (powers) under interference coupling. Typical formulations aim to maximize a network utility while enforcing the following: (i) per-link QoS constraints (minimum rate); (ii) per-link power budgets; and (iii) RB-usage constraints (e.g., at most one RB per link or a small RB budget). These problems are generally non-convex and computationally hard in dense interference regimes, which motivates approximation methods and learning-based policies for scalability Akhtar *et al.* (2021); Mach *et al.* (2019); Najla *et al.* (2021).

Since exact solutions are impractical at scale and under tight latency, in the next section, we review classical optimization families used to obtain near-optimal RRM solutions and highlight their limitations in dynamic settings.

2.3 Classical Optimization for RRM

Classical RRM approaches rely on explicit models and algorithmic solvers. For D2D and interference networks, the literature includes game-theoretic methods, matching and assignment strategies, convex approximations, and metaheuristics for mixed-variable problems.

2.3.1 Principle and Representative Families

Representative optimization approaches include iterative matching/game formulations for joint channel and power allocation Yuan, Yang, Feng & Hu (2018), optimization-based designs for power control and channel allocation in underlaid/overlay D2D systems Abdallah, Mansour & Chehab (2018), and two-stage schemes separating RB assignment from power optimization to simplify the search space Liu, Wu, Li, Liu & Xu (2019). Metaheuristic methods

(including genetic-algorithm families) are also frequently employed when discrete RB assignment must be optimized jointly with continuous power variables Lee, Kim & Cho (2019).

Recent graph-based RRM surveys further classify optimization-oriented approaches into graph coloring, matching, scheduling, decomposition, and combinatorial optimization families. These methods remain important baselines because they explicitly model interference and feasibility constraints, but they usually require repeated optimization and problem-specific reformulation when the network size, topology, or resource structure changes Dai *et al.* (2025a).

2.3.2 Use in Power/Channel Allocation

Optimization-based methods, which are widely used as baselines, are also a practical source of offline labels for supervised learning. Evolutionary optimization frameworks are particularly convenient for mixed-variable problems and can use parallel evaluation to accelerate dataset generation Blank & Deb (2020). However, while these methods can produce strong solutions, they often require iterative procedures and may be sensitive to scenario-dependent design choices Akhtar *et al.* (2021).

2.3.3 Limitations and Motivation for Learning

The following two limitations are central for next-generation systems: (i) *runtime and scalability*, since iterative optimization can be too slow to re-run at each decision epoch in dense networks; and (ii) *robustness*, since solutions optimized under estimated CSI can degrade under uncertainty and mobility. These challenges motivate learning-based RRM, where expensive computation is shifted offline and where online decisions are produced via fast inference Zhang, Patras & Haddadi (2019).

Learning-based methods promise fast online decisions, yet raise new questions about generalization, constraint satisfaction, and scalability, which are reviewed below.

2.4 Learning-Based RRM

Learning-based RRM replaces repeated per-instance optimization with a learned policy. The main paradigms include supervised imitation learning, reinforcement learning (RL), and unsupervised/self-supervised optimization.

2.4.1 Principle (Supervised, RL, Unsupervised)

Supervised learning trains a model to imitate an optimizer using labeled solutions, thus enabling fast inference, but requiring expensive label generation Shen, Shi, Zhang & Letaief (2020); Lee, Yu & Li (2020); Zhang & Tao (2022). **Reinforcement learning** frames RRM as sequential decision making under uncertainty and can learn adaptive policies, but requires careful reward design and extensive interaction Wang *et al.* (2021); Sun, Mei, Wang & Jin (2023b); Tan, Liang, Zhang & Feng (2021); Sun & Nakhai (2021). **Unsupervised/self-supervised learning** directly optimizes a differentiable surrogate of the network objective (e.g., throughput), thus reducing dependence on labels, but requiring explicit mechanisms to respect QoS and feasibility constraints Zhang *et al.* (2019).

More recent surveys on AI-native and graph-centric wireless networks show that learning-based RRM is increasingly viewed as a core component of future 6G and open, intelligent radio access networks. In this view, learning models are not only used to approximate optimization solvers, but also to support scalable, adaptive, and topology-aware control in heterogeneous wireless environments Rahmani *et al.* (2026); Dai *et al.* (2025b).

2.4.2 Use in Resource Allocation Policies

Euclidean architectures such as DNNs and CNNs have been widely applied to power control and resource allocation by mapping CSI to allocation decisions. For instance, CNN-based power control was explored to capture local structure in input representations Lee, Kim & Cho (2018), and DL-based resource allocation was proposed for D2D communications Lee & Schober (2022).

Once trained, these approaches can provide very fast inference, which is attractive for real-time operation.

2.4.3 Limitations and Motivation For Graph Learning

A key limitation of many Euclidean models is that they assume fixed input/output sizes and do not naturally encode the relational structure of interference. This can hinder scalability when the number of links changes, as well as, degrade generalization across different network topologies Zhang *et al.* (2019). These observations motivate graph learning, which explicitly models interference relations and supports permutation-aware, size-invariant inference Shen, Shi, Zhang & Letaief (2021); Shen, Zhang, Song & Letaief (2023); He *et al.* (2021).

Graph learning injects wireless structure directly into the model by operating on interference graphs. In the next section, we summarize graph modeling and GNN methodology, which is followed by an outline of the remaining gaps addressed by this thesis.

2.5 Graph Learning for Wireless Networks

2.5.1 Graph Modeling Principle (Interference Graphs and Features)

Interference networks can be represented as graphs where nodes correspond to communication links, while edges represent interference coupling. Node and edge features are typically derived from CSI (e.g., direct-link quality and cross-link interference strength). For multi-RB settings, common constructions include per-RB graphs or a single graph with multi-dimensional features concatenated across RBs. This modeling principle is widely used in graph-based RRM to encode relational dependencies potentially ignored by Euclidean models Shen *et al.* (2021, 2023); He *et al.* (2021). Recent works further confirm this trend for joint channel and power allocation, QoS-aware multi-channel resource allocation, compact GNN inference for RRM, and outage-aware power control in dense spectrum-sharing networks Chen, Zhu & Evans (2024); Chen, She, Zhu & Evans (2025); Ghasemi & Pishro-Nik (2024); Han, Shi & Lu (2025). A recent

survey also identifies graph learning as a major direction for wireless resource management, covering power control, spectrum management, beamforming, scheduling, task offloading, and aerial coverage planning Dai *et al.* (2025b).

2.5.2 GNN Methodology for Scalable RRM

GNNs compute node representations by message passing and neighborhood aggregation, using permutation-invariant operators that yield permutation-equivariant mappings. This supports size-invariant inference and transfer across network scales Wu *et al.* (2021); Zhang, Cui & Zhu (2022); Gilmer, Schoenholz, Riley, Vinyals & Dahl (2017); Ruiz, Gama & Ribeiro (2021). Attention mechanisms can further improve expressiveness by learning which neighbors (interferers) matter most, making GNNs particularly suitable for heterogeneous interference conditions Veličković *et al.* (2018). These properties underpin modern scalable RRM architectures Shen *et al.* (2021, 2023), including the first thesis article that learns joint allocation directly from interference graphs Marwani & Kaddoum (2024b).

Recent studies have also started to move beyond standard message-passing architectures by considering lightweight GNN models for efficient RRM inference and graph-based generative policies for wireless resource allocation Ghasemi & Pishro-Nik (2024). These directions reinforce the importance of graph-structured inductive bias for scalable wireless optimization.

2.5.3 Limitations and Thesis Gaps (Multi-RB Structure and Temporal Evolution)

Despite the advantages of GNNs, two limitations recur in much of the GNN-based RRM literature and motivate the present thesis contributions:

- **Multi-RB reduction choices can hide CSI structure:** Many previous studies handled multi-RB CSI by concatenating RB-wise gains into high-dimensional node/edge features or by aggregating across RBs to build a single summary graph. Recent GNN-based RRM works have extended graph learning to joint channel and power allocation and QoS-aware multi-channel resource allocation Chen *et al.* (2024, 2025). However, most graph-learning

formulations still emphasize relational interference modeling, while the explicit preservation of complementary tensor-level CSI structure and graph-level interference geometry remains less developed Dai *et al.* (2025b). Albeit convenient, these reductions can weaken the model's ability to exploit spatial correlations and RB-dependent structure present in the original CSI tensor, thus motivating approaches that would preserve complementary tensor (spatial) and graph (geometric) information Marwani & Kaddoum (2024a).

- **Per-slot snapshot inference ignores temporal correlation:** Numerous graph-based policies compute decisions independently at each time slot. Under mobility and link churn, this snapshot assumption discards temporal context that can stabilize decisions and improve robustness. Recent dynamic-GNN surveys emphasize that temporal graph learning can model evolving nodes, edges, and interactions through recurrent updates, attention, memory, or event-driven mechanisms Zheng, Yi & Wei (2025). In wireless resource allocation, recent state-augmented GNN models also show the importance of retaining state information for long-term scheduling and constraint-aware decisions Garcia Camargo, Wang, NaderiAlizadeh & Ribeiro (2025). These observations motivate temporal/dynamic graph learning with memory and time-aware representations for RRM Marwani & Kaddoum (2026).

In the next section, we summarize the research gaps highlighted by this chapter and position the present thesis contributions relative to the reviewed literature.

2.6 Summary, Research Gaps, and Thesis Positioning

The reviewed literature points to a clear progression. Classical optimization methods provide strong baselines, but can be too slow for large-scale real-time operation in dense, dynamic networks Akhtar *et al.* (2021). Recent surveys further confirm that graph-based optimization remains a major family of RRM methods, while also highlighting its dependence on repeated optimization and problem-specific reformulation Dai *et al.* (2025a). Euclidean deep learning enables fast inference, but struggles with variable network sizes and does not inherently encode interference relations Zhang *et al.* (2019). Graph learning provides a principled way to capture

relational interference structure and supports scalable inference Shen *et al.* (2021, 2023); He *et al.* (2021). Recent graph-learning and AI-native 6G surveys further confirm that GNNs are becoming central tools for scalable, topology-aware, and adaptive wireless resource management Dai *et al.* (2025b); Rahmani *et al.* (2026). However, deployment-oriented RRM still requires (i) handling multi-RB CSI without discarding informative structure and (ii) accounting for temporal evolution, rather than relying on independent snapshots. This latter limitation is aligned with recent dynamic graph learning literature, which emphasizes the need for models that explicitly capture evolving nodes, edges, states, and events Zheng *et al.* (2025).

Accordingly, the novelty of the thesis is best understood as a progression from scalable static graph inference, to multi-RB spatial–geometric representation learning, and finally to event-based temporal graph modeling for dynamic RRM.

Table 2.1 Positioning of the thesis contributions with respect to the main limitations identified in the literature

Literature limitation	**Common treatment in prior work**	**Remaining gap**	**Thesis contribution**
Scalable joint allocation	Optimization, Euclidean DL, or GNN-based methods mainly designed for fixed settings or individual RRM tasks	Limited generalization across variable network sizes and joint power/RB decisions	GNN-based joint power control and spectrum allocation with permutation-aware scalable inference
Multi-RB CSI representation	Flattened CSI features, aggregated graph summaries, or independent RB-wise graph processing	Loss of complementary spatial and geometric CSI structure	Spatial–geometric learning that combines tensor-level and graph-level representations
Temporal network evolution	Independent per-slot inference or snapshot-based graph learning	Limited use of historical context, mobility, and link activity evolution	Event-based temporal GNN using continuous-time dynamic graphs and memory-aware representations

Taken together, the literature motivates three coupled needs that define the scope and structure of the present thesis:

- **Scalability:** Size-invariant, permutation-aware inference for joint allocation via graph learning Marwani & Kaddoum (2024b).
- **Multi-RB structure preservation:** Integration of spatial and geometric embeddings to retain tensor and graph information Marwani & Kaddoum (2024a).
- **Temporal adaptivity:** Event-driven and memory-based modeling for dynamic networks beyond snapshots Marwani & Kaddoum (2026).

CHAPTER 3

GRAPH NEURAL NETWORKS APPROACH FOR JOINT WIRELESS POWER CONTROL AND SPECTRUM ALLOCATION

Maher Marwani[1] , Georges Kaddoum[1,2]

[1] Electrical Engineering Department, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Artificial Intelligence and Cyber Systems Research Center, Lebanese American University (LAU), Lebanon

Article published in *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 717–732, 2024.

Abstract

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

be solved jointly. The proposed GNN-based framework presents a promising avenue for tackling this complex challenge. Our experimental results demonstrate that our proposed approach yields significant improvements compared to other existing methods in terms of convergence, generalization, performance, and robustness, particularly in the context of an imperfect channel.

3.1 Introduction

The sixth generation (6G) of wireless communications is expected to feature heterogeneous networks capable of supporting a vast number of connected devices while delivering high data rates, low latencies, and energy efficiency. Several technologies have been developed to meet these requirements, including those referenced in Zhao *et al.* (2021); Alsabah *et al.* (2021); Alwis *et al.* (2021); Moussaoui, Bertin & Crespi (2022). However, the increasing complexity of radio resource management (RRM) has emerged as a significant challenge with the proliferation of new technologies and diverse demands Akhtar *et al.* (2021); Qamar, Siddiqui, Hindia, Hassan & Nguyen (2020). Previous RRM solutions are insufficient in adapting to the novel heterogeneous wireless environment in terms of convergence time, generalization to different wireless contexts, and maintaining satisfactory performance while scaling up the number of devices. Therefore, novel approaches are required to address these challenges and pave the way for the efficient management of wireless resources in the upcoming 6G era.

The focus of our research is on the N-link interference environment with shared channels, a wireless network architecture characterized by multiple communication links sharing the same available bandwidth. In this setting, the co-existence of multiple links causes significant interference and performance degradation, which necessitate effective management of transmission power control and spectrum allocation. This network structure can be observed in various wireless scenarios, including device-to-device (D2D) communication Huang *et al.* (2018); Ramezani-Kebrya *et al.* (2017); Jameel *et al.* (2018); Gupta *et al.* (2022), where multiple devices communicate directly without a network infrastructure, and uplink/downlink Lin & Liu (2019); Shehata, Gasser, El-Badawy & Khedr (2015); Nguyen, Le & Han (2016); Ruby, Zhong, Yang & Wu (2018); Kai, Xu, Zhang & Peng (2018) scenarios, where a base station

communicates with multiple users using the same spectrum, also known as non-orthogonal multiple access (NOMA). To maximize the network's sum rate, solving the mixed-integer, non-convex optimization problem involving power control and channel allocation is essential. However, obtaining a globally optimal solution within the required time is challenging. Therefore, researchers have proposed several near-optimal solutions for specific cases Pan, Pan, Yang & Chen (2018); Mach *et al.* (2019); Najla *et al.* (2021); Liu *et al.* (2019), which tend to have high computational complexity and are impractical for real-time scenarios.

In recent years, researchers have explored the use of machine learning (ML) techniques to address wireless network optimization problems. Specifically, there has been interest in incorporating deep learning (DL) approaches, which have shown promise in a variety of applications. Two primary approaches have been pursued in this integration: (1) constructing end-to-end learnable architectures that can capture complex relationships between inputs and outputs O'Shea & Hoydis (2017); Shen *et al.* (2020); Liang, Shen, Yu & Wu (2020), and (2) replacing computational blocks within existing solutions with DL architectures to reduce computational costs Sun *et al.* (2018); Lee *et al.* (2018). Despite promising results, existing DL-based approaches have primarily focused on addressing isolated RRM tasks such as power control, user association, and link scheduling. Moreover, their scalability to large wireless networks is a concern as they scale linearly with respect to the size of the input data. Furthermore, techniques such as multi-layer perceptrons (MLPs) and convolutional neural networks (CNNs) can be subjected to overfitting and, thus, require large amounts of training data. Additionally, these methods rely on tabular data, such as channel state information (CSI), which ignores the network's underlying topology. Therefore, there is a need for further research to explore more effective ways to integrate DL approaches into wireless network optimization problems. Recent research has demonstrated the potential for improving the scalability and generalization of DL-based RRM solutions by integrating the target task's structure into the neural network architecture Shen *et al.* (2021). Given that wireless networks can be intuitively modeled as graph topologies, there is a growing interest in leveraging graph representation learning techniques to enhance the performance of RRM algorithms He *et al.* (2021). One such approach is Graph Neural Networks (GNNs),

which possess several attractive properties, including permutation equivariance, scalability, generalization, high computational efficiency, and the ability to train efficiently on relatively small datasets Ruiz *et al.* (2021). The application of GNNs has yielded promising results in various domains, indicating its potential as an effective technique for enhancing the performance of RRM algorithms in wireless networks.

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

This paper is structured as follows. In Section 2, a comprehensive literature review is presented to explore the previous work related to our research. In Section 3, we introduce the N-link interference environment with shared channels, which is the problem setting that our proposed solution is designed to address. In Section 4, we present the optimization problem that we aim to solve. In Section 5, we provide a detailed description of our proposed end-to-end solution architecture, which consists of various components, including CSI preprocessing, the GNN

feature extractor, the MLP component, the loss function design, and the training process. In Section 6, we conduct extensive simulations to evaluate our proposed solution's performance in terms of stability, generalization, and robustness compared to the state-of-the-art methods. Finally, in Section 7, we present our conclusions and future research directions.

3.2 Literature Review

Numerous studies have focused on solving the power control and spectrum allocation problems in different network topologies, either independently or concurrently. For example, in Pan *et al.* (2018), the authors proposed a dual-based iterative algorithm that allocates resources to D2D pairs while maintaining the quality of service requirements. Another study Liu *et al.* (2019) developed a two-stage algorithm to maximize the energy efficiency of D2D communication under cellular constraints, assuming that each D2D link could use one sub-channel at most to decrease the computational complexity. Conversely, this research Mach *et al.* (2019) proposed a channel and power allocation scheme with channel reuse based on the Hungarian algorithm and a prioritizing method. Moreover, this work Najla *et al.* (2021) employed a game-theory approach to manage the reuse of multiple channels by multiple D2D pairs. Despite the various proposed solutions, most of them were heuristic approaches or tended to convexify the RRM problems, resulting in a high computational complexity. Additionally, they did not provide complete flexibility in allocating multiple channels to multiple links, mainly because of convergence issues.

Given the limitations of model-based and heuristic solutions, researchers have turned to learnable approaches by integrating DL architecture to tackle RRM problems. For example, these works Shen *et al.* (2020); Lee *et al.* (2020); Zhang & Tao (2022) integrated a DL component to learn the optimal pruning policy for the branch-and-bound (B&B) algorithm to solve mixed-integer nonlinear programming (MINLP) problems. While this approach simplifies the problem significantly and reduces the exponential computation of the traditional B&B algorithm, intense sampling is required to train the DL architecture since the training is supervised. To alleviate the need for training data, unsupervised learning approaches have been explored Lin & Zhu (2020).

This work considers constructing a DNN framework to solve beamforming problems over an imperfect channel, which is trained in an unsupervised fashion using the negative of the sum rate. Similarly, Liang *et al.* (2020) constructed a DNN that takes CSI and computes the power of each user. However, this work did not integrate the minimum rate constraints into the training process, which raises questions about the solution's feasibility. Another approach to training DL models is combining supervised and unsupervised losses. For instance, Lee & Schober (2022) proposed an end-to-end DL framework to solve resource allocation in multi-channel cellular systems with D2D links. Moreover, the approach can be implemented in a centralized manner, with full knowledge of the CSI, or distributed manner with partial CSI. However, the authors transform the continuous power variable into a set of discrete levels in order to use the cross-entropy loss. Following the same training approach in Lee *et al.* (2018), a CNN model is employed to learn the patterns from CSI and output the power control that maximizes the energy or spectrum efficiency of the network. Despite the promising results achieved by the current DNN and CNN approaches, their lack of flexibility with input sizes is a significant limitation. Any alteration in input shape necessitates architectural modification. Furthermore, they prove inadequate in large wireless scenarios with a substantial number of connected devices. This deficiency stems from their heavy reliance on the quality of training data, which can be challenging to obtain in real-life situations. Moreover, the training process for these models is often time-consuming and typically conducted offline. Another drawback of DNN and CNN approaches is their disregard for the geometric information inherent in the input data.

To facilitate the incorporation of input data structures into DL models for RRM tasks, several solutions based on GNNs have been introduced He *et al.* (2021). These approaches have exhibited promise in tackling various RRM tasks, encompassing channel allocation, power control, and user association. For instance, in Nakashima *et al.* (2020), a framework combining Deep Reinforcement Learning (DRL) and Graph Convolution Networks (GCNs) was proposed for channel allocation. This method enabled the agent to learn optimal channel assignments to access points using features extracted from the wireless environment as a state space. However, the model's testing was confined to a wireless setting with perfect channels and a relatively small

number of devices. Similarly, Eisen & Ribeiro (2020) introduced a GNN-based framework for learning resource allocation strategies in wireless networks, offering reduced training times and improved scalability compared to conventional MLPs. Nonetheless, this framework was not well-suited for heterogeneous wireless devices or systems with single or multiple antennas. To address these limitations, Shen *et al.* (2021) presented a more flexible GNN-based solution for constrained power allocation in a heterogeneous MIMO-interfering environment. Leveraging the permutation-invariant properties of RRM problems, this GNN architecture demonstrated excellent generalization across different problem scales with minimal training data. In addition, in Guo & Yang (2021), researchers aimed to find the optimal power control strategy in an uplink multi-cell network by combining DNNs with knowledge from the wireless network's topology, reducing training complexity and model parameters. However, this approach was tailored exclusively to the power control task. In contrast, Zhang *et al.* (2021b) proposed employing GNNs to tackle power control and beamforming issues in heterogeneous D2D networks. Here, communication and interference links were represented as vertices in the wireless graph, and an unsupervised learning process was employed for the graph convolutional model. This method demonstrated favorable properties such as scalability and reduced execution time compared to alternative approaches. Similarly, Ranasinghe, Rajatheva & Latva-aho (2021) introduced an Access Point (AP) selection strategy for massive cell-free Multiple Input, Multiple Output (MIMO) systems based on GNNs. The authors constructed two graphs: a homogeneous one representing only AP nodes and a heterogeneous one containing both user equipments and AP nodes. However, these methods modeled the wireless network as a single graph, assuming that all communication links interfered with each other. GNNs can serve as end-to-end learnable solutions or feature extractors. For example, Zhang, Zhang & Yang (2021a) proposed a joint optimization framework for user association and power control in a heterogeneous ultra-dense network. Similarly, Chowdhury, Verma, Rao, Swami & Segarra (2021) improved the Iteratively Weighted Minimum Mean Square Error (WMMSE) algorithm Shi *et al.* (2011) by incorporating trainable components parametrized by GNNs. Simulations illustrated that the proposed method, unfolded WMMSE, delivered a comparable performance to WMMSE but with significantly lower time complexity.

The work in NaderiAlizadeh, Eisen & Ribeiro (2023) introduced a trainable resilient RRM policy using an unsupervised primal-dual approach for power control and user association. Another paper Wang, Li, Shi & Wu (2023) presented an edge-update empowered GNN architecture, enhancing GNNs' ability to handle node and edge variables and validating its Permutation Equivariance in power allocation scenarios. Additionally, Wang, Eisen & Ribeiro (2022) introduced Aggregation GNNs for decentralized resource allocation in wireless networks, utilizing a model-free primal-dual approach for asynchronous local information processing. The study in He, Wang, Ye, Li & Juang (2020b) proposed a distributed spectrum allocation scheme for vehicle-to-everything (V2X) networks using GNNs and multi-agent RL to optimize the network capacity. Furthermore, Gu, She, Quan, Qiu & Xu (2023) discussed GNN-based frameworks for distributed power allocation in wireless networks, aimed at minimizing signaling overhead by incorporating Recurrent Neural Networks (RNNs) to capture temporal dynamics. The work in Shen *et al.* (2023) offered a GNN framework to enhance power control and hybrid precoding in wireless systems, demonstrating scalability and efficiency. The work in NaderiAlizadeh, Eisen & Ribeiro (2022) proposed a state-augmented algorithm for RRM in multi-user networks, ensuring feasible and nearly optimal decisions. In addition, Cheng, Chen & Han (2023) introduced a Heterogeneous GNN model for resource allocation in heterogeneous networks. The study in Chen, Zhang, You, Zheng & Lambotharan (2022) presented a GNN-based scheme for RRM in wireless IoT networks, optimizing resources in D2D communications. Lastly, Peng, Guo & Yang (2023) explored the expressive power of GNNs in learning wireless policies, highlighting the limitations of Vertex-GNNs and the advantages of Edge-GNNs in resource allocation tasks.

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

3.3 System Model

We denote $\mathcal{N} = \{1, 2, \ldots, N\}$ a set of active (scheduled) links distributed randomly in a two-dimensional environment. The distance between transmitter-receiver pairs varies across links. We adopt a non-orthogonal scheme for all communication links, where $\mathcal{K} = \{1, 2, \ldots, K\}$ is the set of resource blocks (RBs) with constant bandwidth W that can be assigned to any link ($K \leq N$). In this environment, a centralized control unit (CCU) controls the transmission power of each link and allocates the required bandwidth to ensure effective communication, as illustrated in 3.1.

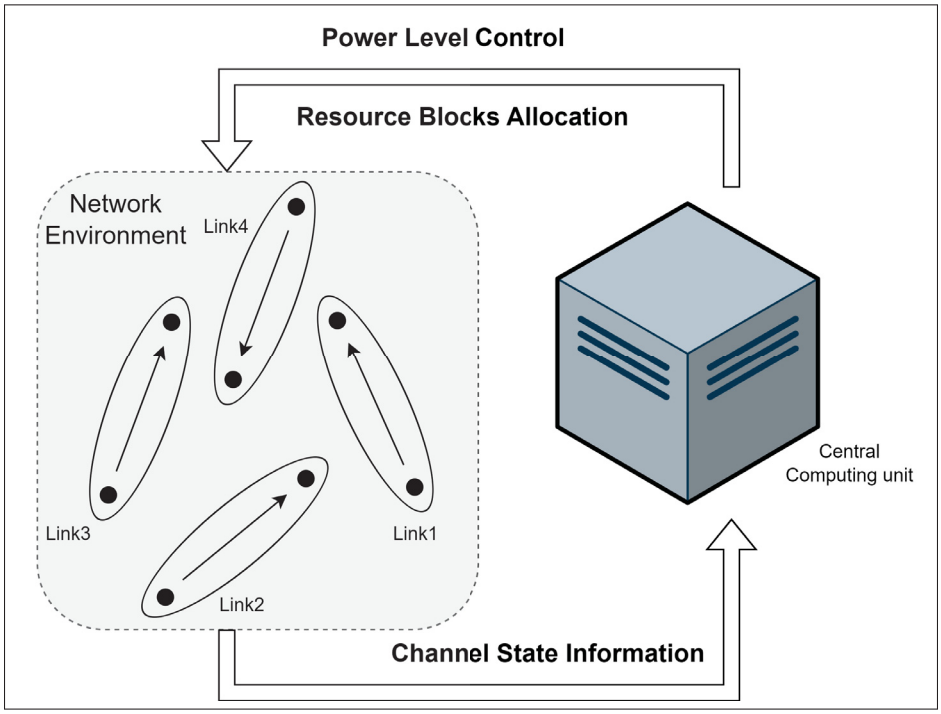


Figure 3.1 N-link interference environment with shared channels

We operate in a time slot scenario where the CCU obtains CSI from scheduled links, performs resource management, and communicates decisions to all transmitters. Although the CCU possesses full CSI knowledge, it may still be subject to noise and errors, leading us to evaluate our approach under noisy CSI conditions to assess its robustness in Section V.

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

$$G^k = \begin{pmatrix} g_{11}^k & g_{12}^k & \cdots & g_{1N}^k \\ g_{21}^k & g_{22}^k & \cdots & g_{2N}^k \\ \vdots & \vdots & \ddots & \vdots \\ g_{N1}^k & g_{N2}^k & \cdots & g_{NN}^k \end{pmatrix}. \tag{3.1}$$

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

$$g_{ij}^k = \beta_{ij}^k \alpha_{ij}^k |h_{ij}^k|^2 \quad \forall k \in \mathcal{K}, \quad \forall (i,j) \in \mathcal{N} \times \mathcal{N}. \tag{3.2}$$

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

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

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

defined as:

$$\text{SINR}_i^k = \frac{g_{ii}^k p_i}{\sum_{j \neq i}^{N} \psi_j^k g_{ij}^k p_j + N_0 W}, \forall (i,k) \in \mathcal{N} \times \mathcal{K}. \tag{3.3}$$

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

$$\gamma_i = W \sum_{k=1}^{K} \psi_i^k \log_2 \left(1 + \text{SINR}_i^k\right), \forall i \in \mathcal{N}. \tag{3.4}$$

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

$$\begin{aligned} \max_{P,\Psi} \quad & \frac{1}{N} \sum_{i=1}^{N} \gamma_i & \\ \text{s.t} \quad & \gamma_i \geq \gamma_{min} & \forall i \in \mathcal{N} \\ & \sum_{k=1}^{K} \psi_i^k \leq 1 & \forall i \in \mathcal{N} \\ & 0 \leq p_i \leq p_{max} & \forall i \in \mathcal{N} \\ & \psi_i^k \in \{0,1\} & \forall (k,i) \in \mathcal{K} \times \mathcal{N} \end{aligned} \tag{3.5}$$

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

The problem at hand presents significant challenges due to the complexity of the objective function and the inclusion of mixed variables in the optimization process. Furthermore, the presence of time constraints, specifically related to the channel state, necessitates the adoption of a solver with a convergence time that is shorter than the coherence time to ensure the validity of the obtained solution. As such, we seek a universal approach capable of producing efficient solutions within the necessary timeframe while adhering to the imposed constraints. In the

subsequent section, we introduce our novel GNN-based model, which offers a generalizable solution and yields promising outcomes, thereby fulfilling the aforementioned objectives.

3.4 Solution Architecture

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

3.4.1 Channel State Information Preprocessing

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

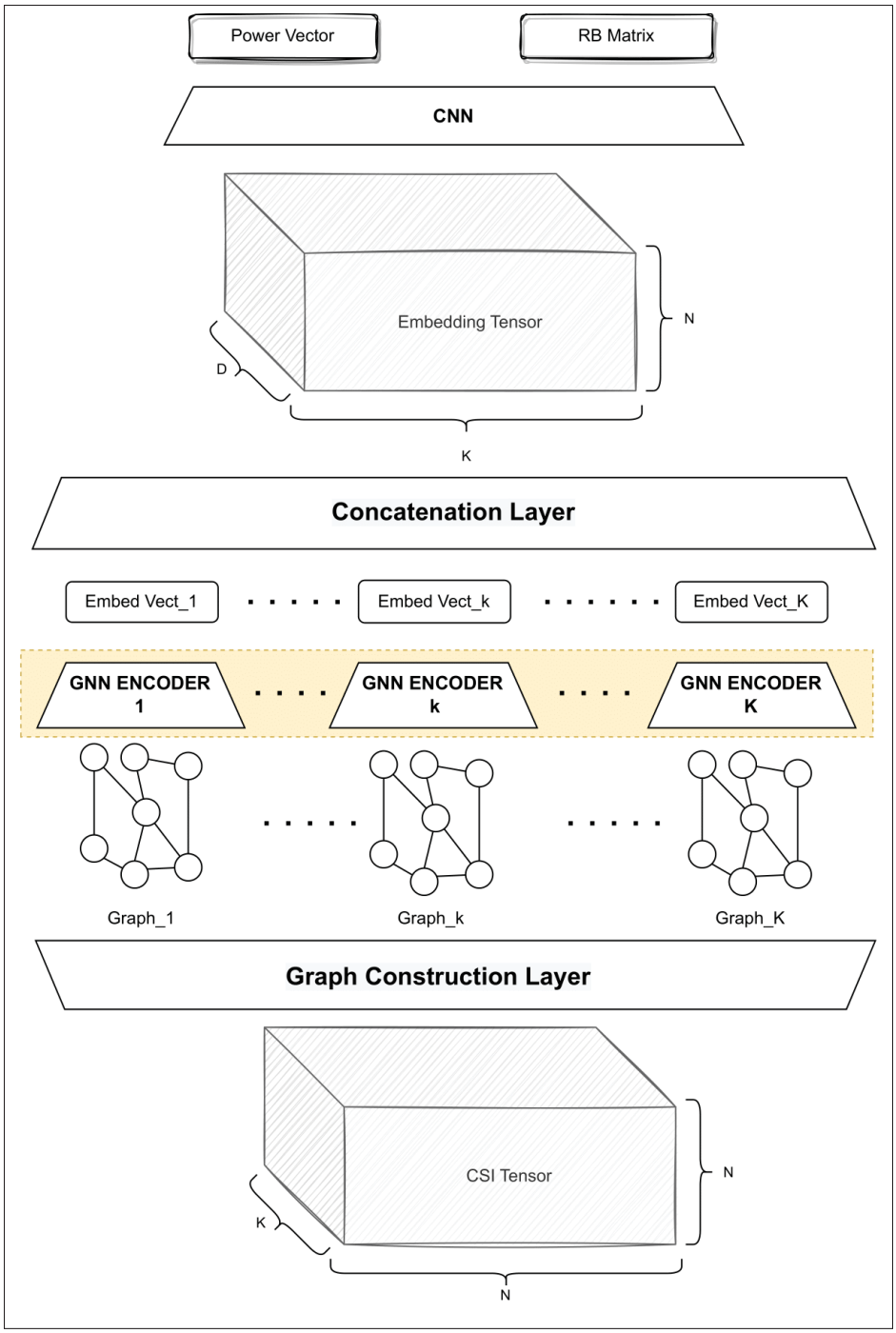


Figure 3.2 Model architecture

and each edge is labeled by $e_{ij} = G_{ij}^{k} \ \ \forall (i, j) \in \mathcal{N} \times \mathcal{N}$, representing the interference between links. This representation allows for a simplified yet effective understanding of the interactions

and interference patterns within the network, leveraging the geometric properties of the graphs to facilitate analysis and optimization. Fig. 3 illustrates an example of a three-interference graph for three communication links.

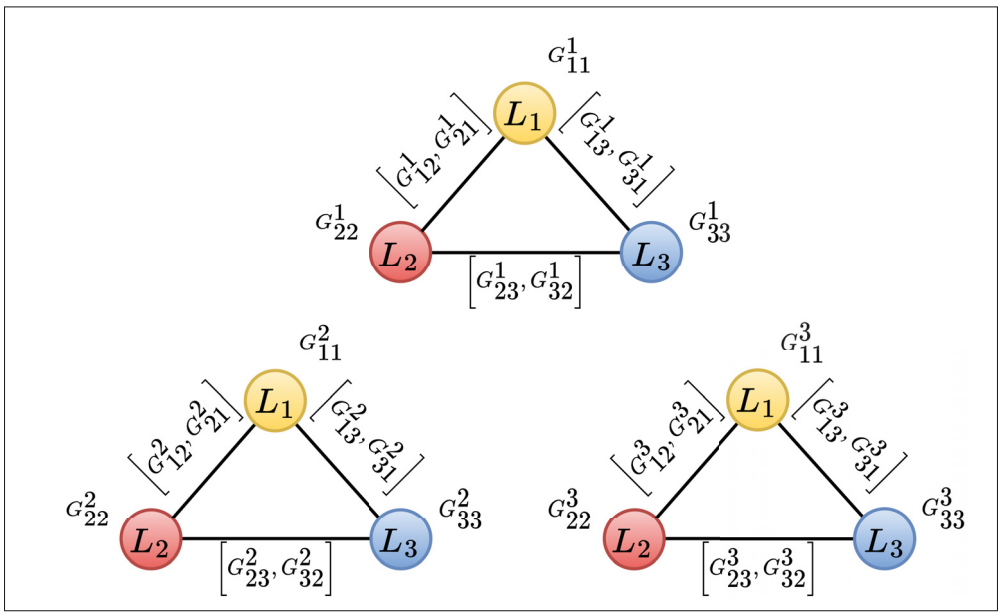


Figure 3.3 Interference graphs of 3 communication links using 3 RBs

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

3.4.2 GNN Feature Extractor

GNNs are a specialized type of neural network designed to operate on graph-structured data Wu *et al.* (2021). They share a multi-layer structure akin to DNNs, where each node within the graph combines its individual features with an aggregation of the features of its neighboring nodes. Furthermore, GNNs update the embedding of each node through iterative aggregation and combination operations. This iterative process relies on a message-passing mechanism

Gilmer *et al.* (2017), where information is exchanged among nodes in the graph through their connecting edges to capture relationships between nodes. In essence, the t-th GNN layer for a node $v \in \mathcal{V}$ can be succinctly summarized through two key iterative equations (6):

$$\begin{aligned} m_{vu}^{t+1} &= \phi(z_v^t, w_{vu}^t, z_u^t) \\ z_v^{t+1} &= \sigma(z_v^t, \rho(\{m_{vu}^{t+1}, u \in \mathcal{N}(v)\})) \end{aligned} \tag{3.6}$$

Here, $z_v^t \in \mathbb{R}^{d_1}$ represents the hidden state of node $v \in \mathcal{V}$, and $w_{vu}^t \in \mathbb{R}^{d_2}$ denotes the feature vector associated with the edge $(v, u) \in \mathcal{E}$ at time t. In the $(t+1)$-th iteration, the edge features w_{vu}^t are fused with the features of their incident nodes $\{z_u^t, z_v^t\}$ via the message function ϕ. Subsequently, within each node v, messages from all neighboring nodes in $\mathcal{N}(v)$ are aggregated using the reduce function ρ. Finally, the node feature z_v^{t+1} is updated using the update function σ. Different choices for combining and aggregation functions can lead to various types of GNNs Zhang *et al.* (2022), but the reduce function must always be a permutation-invariant operation, such as sum, mean, or max, to ensure that the input graph's global structure is preserved. Following a similar pattern, our computation equation (6) is defined as:

$$\begin{aligned} m_{vu}^{t+1} &= \mathrm{MLP}_1^t(z_u^t, g_{uv}, g_{vu}) \\ z_v^{t+1} &= \mathrm{MLP}_2^t(z_v^t, \max\{m_{vu}^{t+1}, u \in \mathcal{N}(v))\}) \end{aligned} \tag{3.7}$$

In this equation, the message function is represented as an MLP block, named MLP_1, which consists of layers of neurons, non-linear activation functions, and batch normalization. It operates on the concatenation of node u's hidden state and the features of edge (v, u). The reduce function is essentially a max aggregation operation, which combines messages received from all neighbors of node v. The update function for node v is similar to MLP_1 but with a distinct number of neurons and is referred to as MLP_2. It learns patterns from the aggregation of messages m_{vu}^{t+1} from neighboring nodes and the node's previous hidden state. Notably, as the size of the hidden state z_v^t of each node varies for each iteration t due to the concatenation operations, each message-passing iteration has its specific message and update functions denoted as MLP_1^t and MLP_2^t. MLPs are preferred in our context for stability, efficiency, and performance.

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

3.4.3 CNN Component

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

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

$$a_i^k = \frac{e^{x_{ik}}}{\sum_{s=1}^{K} e^{x_{ik}}} \tag{3.8}$$

Consequently, the channel allocation is calculated as follows:

$$\Psi_i^k = \begin{cases} 1 & \text{if } j = \mathrm{argmax}_k(\{a_i^k, k \in \mathcal{K}\}) \\ 0 & \text{otherwise} \end{cases} \tag{3.9}$$

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

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

$$p_i = \frac{1}{K} \sum_{k=1}^{K} \frac{p_{max}}{1 + e^{x_{ik}}} \tag{3.10}$$

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

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

3.4.4 Loss Function Design

The loss function of our model is composed of three parts: a supervised segment, an unsupervised segment, and a regulation loss designed to maintain the required minimum data rate. Initially, the neural networks in the model leverage supervised learning to acquire a generalized strategy, drawing from the diverse array of solutions within our dataset generated by PYMOO. This dataset, rich in variety due to alterations in minimum data rates, the number of links, and link

distance variations, equips the model with a wide-ranging understanding of potential wireless configurations. Such a broad perspective is crucial for the model's ability to effectively adapt and fine-tune to specific scenarios during the inference phase. Following the supervised learning stage, the networks further refine their ability to optimize the objective through unsupervised learning, all the while adhering to optimization constraints enforced by the regulation loss. This holistic approach ensures that the model not only learns generalizable strategies but also enhances its objective maximization capabilities and compliance with necessary constraints.

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

$$\mathcal{L}_{sup}(\hat{P}, P, \hat{\Psi}, \Psi) = \sum_{i=0}^{N} \sum_{k=0}^{K} (\hat{p}_i - p_i)^2 + \psi_i^k \log(\hat{\psi}_i^k) \tag{3.11}$$

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

$$\mathcal{L}_{rate}(\hat{\Psi}, \hat{P}) = -\sum_{i=0}^{N} \sum_{k=0}^{K} \frac{W\hat{\psi}_i^k}{N} \log_2\left(1 + \mathrm{SINR}_i^k(\hat{p}_i^k, \hat{\psi}_i^k)\right) \tag{3.12}$$

In order to ensure that every link retains the necessary minimum capacity, we incorporate a regulation loss into our model, strategically managing the rate of each link. Contrary to the conventional approach that penalizes the model when rates drop below $\gamma_{\min}$ Lee & Schober (2022), our method imposes a penalty when certain rates are excessively elevated, as expressed in equation (13). This strategy pushes the model to generate rates as near to $\gamma_{\min}$ as possible. Consequently, while rates that are excessively high are brought down, those that are too low are also increased due to the shared radio resources. Mathematically,

$$\mathcal{L}_{\mathrm{reg}}(\hat{\Psi}, \hat{P}) = \frac{1}{N} \sum_{i=0}^{N} \max(0, \gamma_i - \gamma_{\min}) \tag{3.13}$$

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

3.4.5 Training and Deployment Process

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

In both training and deployment, we first forward propagate to compute the prediction of $\hat{P}$ and $\hat{\Psi}$ to assess the loss value. Afterward, we back-propagate to calculate the gradients and update the model's weights accordingly using the Adadelta optimization technique Zeiler (2012). For each epoch, we preprocess each sample's CSI and construct K interference graphs. Then, we parallelly compute the node representations by evaluating (7) T times for all the K graphs. This parallel computation decreases the execution time by roughly $1/K$. Following that, we calculate $\hat{P}$ and $\hat{\Psi}$ accordingly. Lastly, we evaluate the loss and update the learnable weights of the model.

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

While constructing the training dataset, we have the option of utilizing any MINLP optimization technique to obtain near-optimal solutions to our optimization problem. The exhaustive search is not feasible taking into account the number of possible solutions. Therefore in our case, we

Algorithm 3.1 Supervised training process overview

Input: Dataset $\mathcal{D}$ containing tuples (G, P, Ψ)
Output: Optimal model parameters Θ

Initialize $\hat{P}, \hat{\Psi}$;
for *epoch* $\leftarrow 1$ **to** *MaxEpochs* **do**
  foreach *sample* $(G_l, P_l, \Psi_l) \in \mathcal{D}$, $l = 1, \ldots, L$ **do**
    PreProcess G_l;
    for $k \leftarrow 1$ **to** K ; // in parallel
    do
      foreach *node* $i \in \mathcal{V}^k$, $i = 1, \ldots, N$ **do**
        $\mathbf{z}^0_{k,i} \leftarrow g_{ii}$;
        for $t \leftarrow 1$ **to** T **do**
          Compute $\mathbf{z}^t_{k,i}$;
        end for
      end foreach
    end for
    Compute $\hat{\Psi}, \hat{P}$;
    Compute $\mathcal{L}_{\text{sup}}(\hat{\Psi}, \hat{P})$;
    Update weights using $\nabla_\Theta \mathcal{L}_{\text{sup}}(\hat{\Psi}, \hat{P})$;
  end foreach
end for

Algorithm 3.2 Deployment process overview

Input: CSI G, pre-trained parameters Θ^*
Output: P^* and Ψ^*

for *epoch* $\leftarrow 1$ **to** *MaxEpochs* **do**
  PreProcess G;
  Compute Z;
  Compute $\hat{\Psi}, \hat{P}$; // (8), (10)
  Compute $\mathcal{L}_{\text{rate}}(\hat{\Psi}, \hat{P}) + \mathcal{L}_{\text{reg}}(\hat{\Psi}, \hat{P})$;
  Update weights using $\nabla_\Theta\big(\mathcal{L}_{\text{rate}} + \mathcal{L}_{\text{reg}}\big)$;
end for
Compute Ψ^* ; // (9)

have chosen to utilize PYMOO Cui, Shen & Yu (2018) due to its widespread applicability and demonstrated efficacy in various fields. PYMOO provides a range of flexible genetic algorithm

techniques that include evaluation features capable of assessing the solutions obtained, as well as parallel computation functionalities that serve to expedite the dataset construction process.

3.5 Performance Evaluation

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

3.5.1 Simulation Parameters

We construct a rectangular 2D layout with width $w_x = 200\ m$ and height $w_y = 100\ m$ that represents a wireless environment, and we randomly distributed the N transmitters in the area. Consequently, we spread the receivers to be randomly distant from their corresponding transmitters in a range between $d_{\min}$ and $d_{\max}$. We adopted the channel model from the short-range outdoor model ITU-1411 with a distance-dependent path-loss ITU-R (2009), with 2.4 GHz carrier frequency, 1.5 m antenna height and 2.5 dBi antenna gain. The transmit power's maximum level is 4 dBm, and the background noise level is -169 dBm/Hz. We model the shadowing using the normal distribution, $\alpha_{ij}^k = 10^{\frac{S}{10}}, S \sim \mathcal{N}(0, \sigma)$ where σ is the shadowing deviation in dB. We consider an urban outdoor environment, where σ is between 4 dB and 12 dB. As for the fast fading channel, we use the Rayleigh fading model, $h_{ij}^k = R + jI; I, R \sim \frac{\mathcal{N}(0,1)}{\sqrt{2}}$. The number of channels is $K = 10$, where each has 500 Hz of bandwidth.

It is important to note that our model is **trained only once** with 10000 training samples over 30 epochs where $N = 50$, $d_{max} = 50$ m, $d_{min} = 5$ m, $\gamma_{min} = 200$ bps, and $\sigma = 4$ dB generated by PYMOO Single-Objective Optimization With Mixed Variables API Blank & Deb (2020).

We have also generated 1000 samples for testing the convergence of the supervised training following the same procedure. All the simulations and training have been conducted on the same hardware, processor Intel Intel(R) Xeon(R) W-1270 CPU, 16.0 GB memory, and 3.40 GHz. The code [1] is implemented using Python 3.9 with Deep Graph Learning library (DGL Wang *et al.* (2019)) and PyTorch as a backend.

Regarding network parameters, we configure the number of GNN layers to $T = 4$, and the embedding dimension is established at $D = 10$. We design MLP_1 and MLP_2 with three hidden layers, between these layers, we apply a ReLU activation function and incorporated batch normalization. Our CNN block comprised 3 convolution filters, with ReLU activation functions interposed. The parameters for the convolution filters, including strides, padding, and dilation, were all set to 1. The kernel size is defined as (3×3), ensuring that the spatial dimensions of the input tensor remain unchanged. The channel input-output pairs are configured as $(10, 5, 2, 1)$. For the training and the deployment process, we opt for a fixed learning rate of 1.0 without any decay over epochs, as Adadelta would adapt it accordingly.

3.5.2 Complexity Analysis

Assuming sequential processing, the complexity of GNN encoders is:

$$o(KTN[(N-1)\left((D+2)D + AD^2\right) + BD^2]), \tag{3.14}$$

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

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

3.5.3 Benchmarking Schemes

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

- **RANDOM:** We generate 40000 power and channel allocation solutions at random. Subsequently, we select the solution that minimizes QoS violations while maximizing the mean network rate.
- **PYMOO Blank & Deb (2020):** The problem is addressed utilizing a genetic algorithm provided by PYMOO. Notably, this solution is identical to the one used to generate the training dataset.
- **SLSQPNocedal & Wright (1999):** Initially, greedy channel allocation Sun, Xu, Ng, Dai & Schober (2019) is assigned to all links. Following this, we resolve the power control task by employing the Sequential Least Squares Programming (SLSQP) technique.
- **DNN Lee & Schober (2022):** The CSI is reshaped and fed into two DNN architectures. The first DNN architecture is responsible for power control, while the second DNN architecture handles channel allocation. However, due to the DNN architecture's inflexibility towards variations in the number of links, we adjust the number of neurons in the DNN to be suitable for the selected N and retrain it with an adequate dataset.
- **REGNN NaderiAlizadeh *et al.* (2023):** The problem is addressed using a Resilient GNN policy, trained using an unsupervised primal-dual approach. We average and normalize the CSI across resource blocks to construct a graph topology similar to the one used in the paper.

3.5.4 Training Convergence Analysis

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

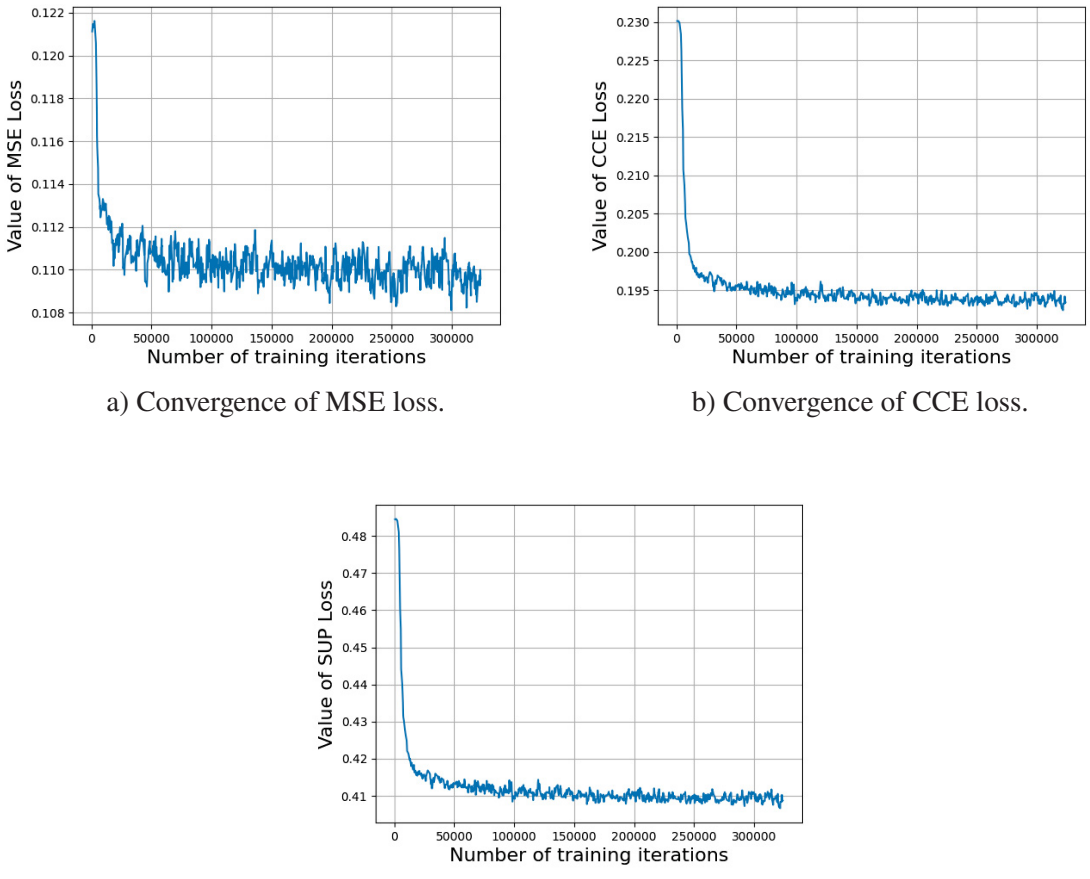
3.5.4.1 Supervised Convergence

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

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

3.5.4.2 Unsupervised Convergence

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



a) Convergence of MSE loss.

b) Convergence of CCE loss.

c) Convergence of supervised (SUP) loss.

Figure 3.4 Convergence of supervised training losses: (a) MSE (power), (b) CCE (RB assignment), and (c) total supervised loss

Fig. 8 illustrates the impact of varying the number of links on the average sum rate and QoS probability. We set $N = \{50, 75, 100, 125\}$, $d_{max} = 50$ m, $\gamma_{min} = 300$ bps, and $\sigma = 4$ dB. The top graph indicates a convergence trend in average sum rates for all the considered N values, with a rise of around 20% after 1000 iterations. Higher N values produce lower rates since the radio resources are finite and invariant. The bottom graph underscores the improvement in QoS adherence over the iterations. For instance, $N = 125$ decreases its initial violation probability from 17.5% to nearly 1%, while $N = 50$ reduces it from 5% to almost 0%. This demonstrates that our approach can achieve improved performances while maintaining strict QoS compliance for

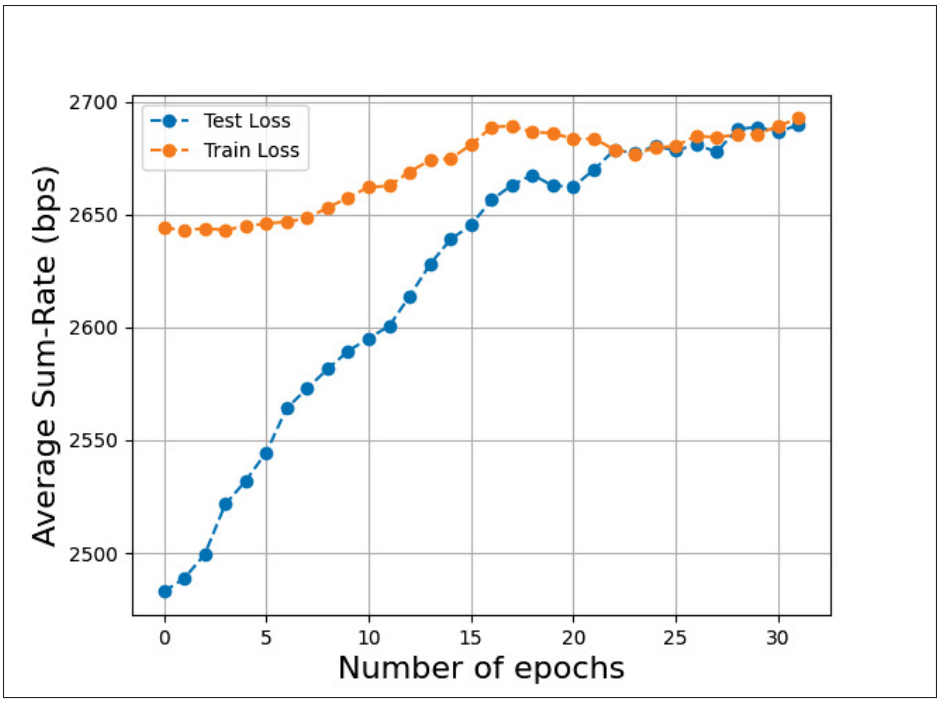


Figure 3.5 Training vs. test model's average sum rate evolution over the supervised learning process

various numbers of links. Fig. 9, on the other hand, evaluates the model's performance against changing QoS values, specifically at $\gamma_{min} = \{250, 500, 750, 1000\}$ bps, $N = 50$, $d_{max} = 50$ m, and $\sigma = 4$ dB. We kept the same CSI in this analysis for a fair assessment. The Average Sum Rate displays an increased performance across the examined values, stabilizing near 2500 bps, a boost of approximately 31% from the initial supervised outcome. Meanwhile, the QoS Violation Probability segment reveals a sharp decrease, especially for the $\gamma_{min} = 1000$ bps curve, and by the 800-th iteration, all curves converge to under 5%, with most nearing zero. This observation underscores the model's aptitude to adjust to fluctuating QoS values, constantly optimizing rates and reducing QoS breaches.

Fig. 10 explores the effects of different link distances, where $d_{max} = \{50, 60, 70, 80\}$ m, $\gamma_{min} = 300$ bps, $N = 50$, and $\sigma = 4$ dB. Similarly, we observe a clear trend: the Average Sum Rate increases over iterations by 30 %, and higher d_{max} values produce lower rates due to higher channel attenuations. However, although initially high, the QoS Violation Probability for all d_{max}

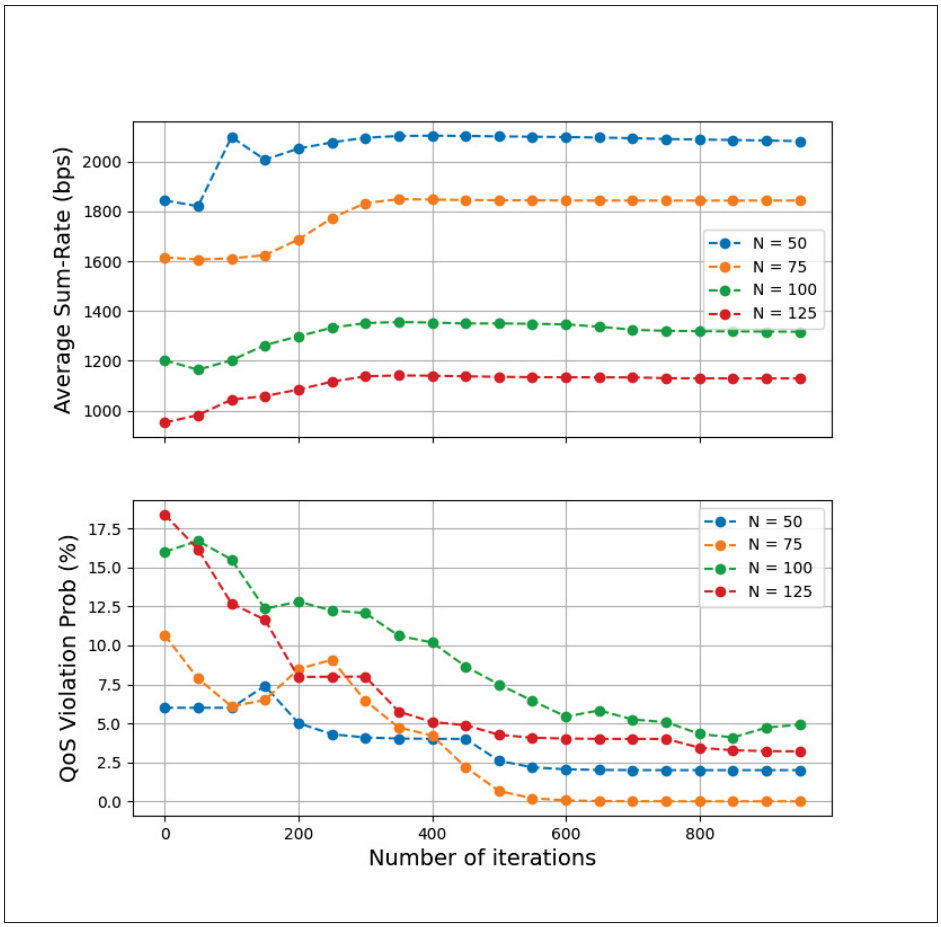


Figure 3.6 Model's average sum rate and QoS probability over unsupervised learning in different N values

values declines rapidly, converging to 2% by the 800-th iteration. This emphasizes the model's increased performance versatility across varying link distances. Finally, Fig. 11 showcases the influence of shadowing, where $\gamma_{min} = 300$ bps, $N = 50$, $d_{max} = 50$ m, and $\sigma = \{4, 6, 9, 12\}$ dB. The top graph reveals that all curves consistently increase the Average Sum Rate as iterations continue. Higher values of σ lead to slightly higher rates. This observation can be attributed to the fact that if shadowing results in a positive deviation (i.e., the signal strength is higher than expected), the SNR would increase, potentially leading to a higher Shannon capacity. However, it's important to note that this doesn't imply shadowing is inherently "beneficial." Rather, the random nature of shadowing can occasionally produce signal strengths that surpass deterministic predictions. The bottom graph, depicting QoS Violation Probability, suggests that the system is

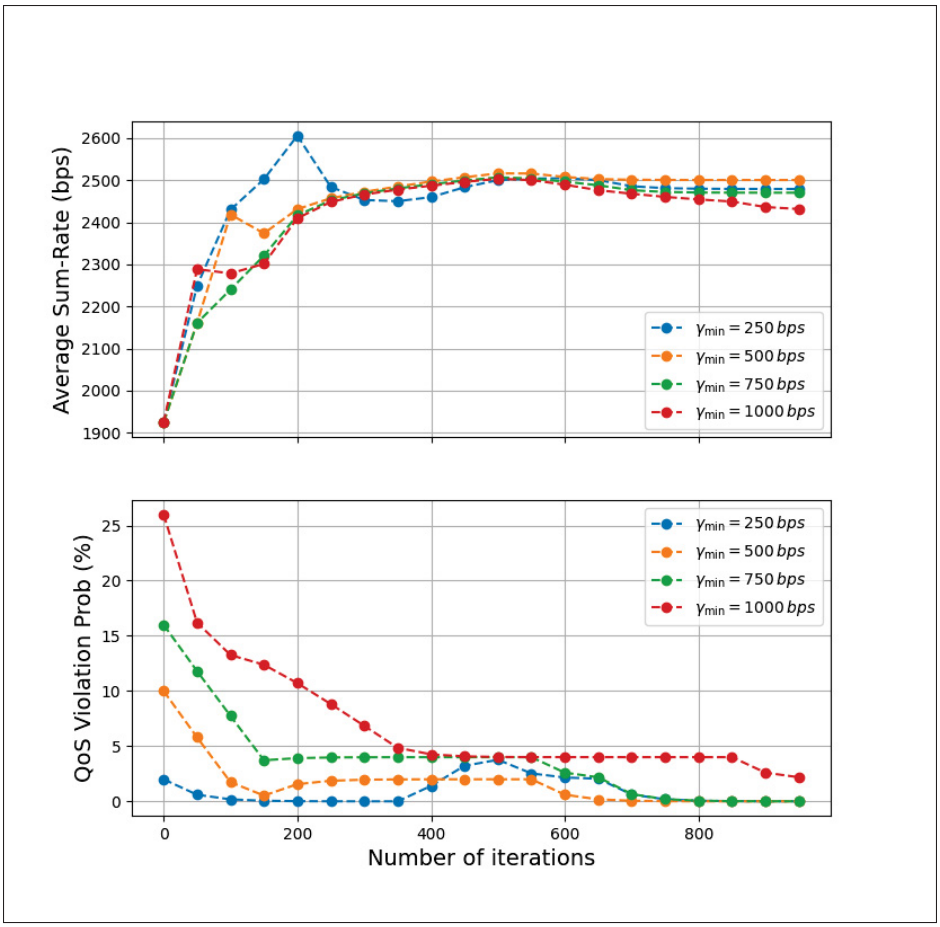


Figure 3.7 Model's average sum rate and QoS probability over unsupervised learning in different $\gamma_{\min}$ values

robust across diverse shadowing scenarios, with higher and lower σ values witnessing significant violation reductions over iterations, reaching nearly 0 violations. This highlights the model's increased performance across varying shadowing deviations.

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

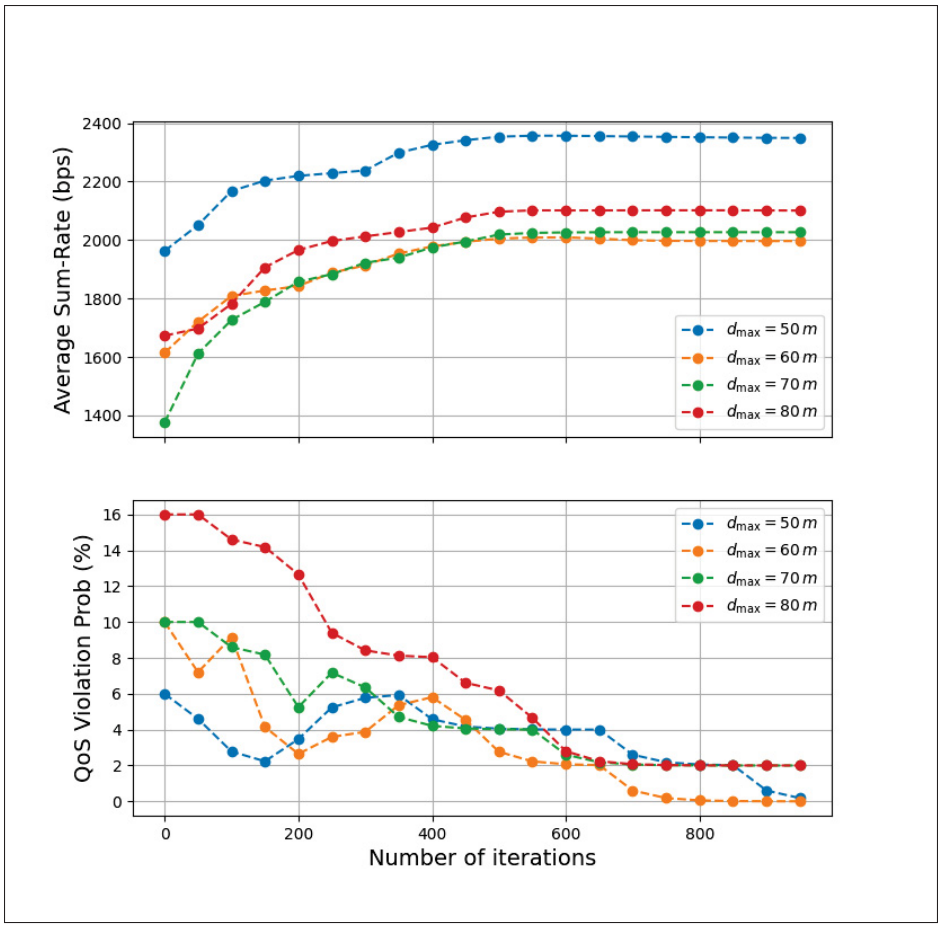


Figure 3.8 Model's average sum rate and QoS probability over unsupervised learning in different $d_{\max}$ values

3.5.5 Impact of QoS Constraints on Performance

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

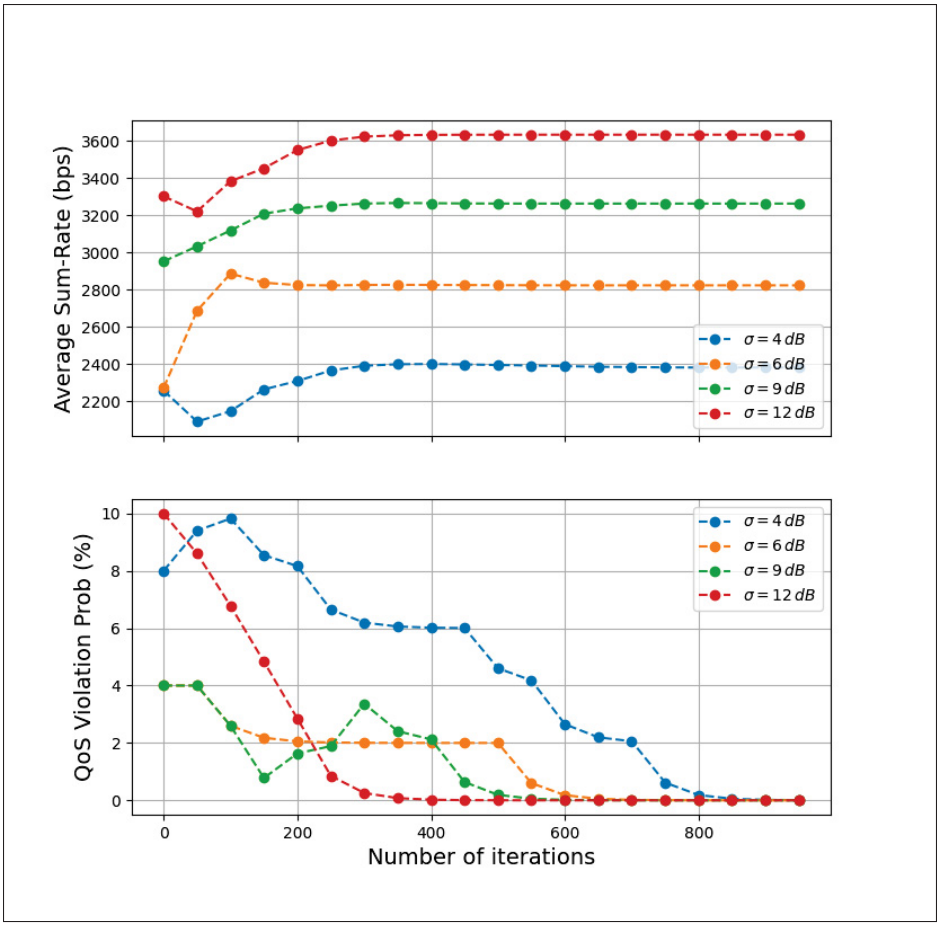
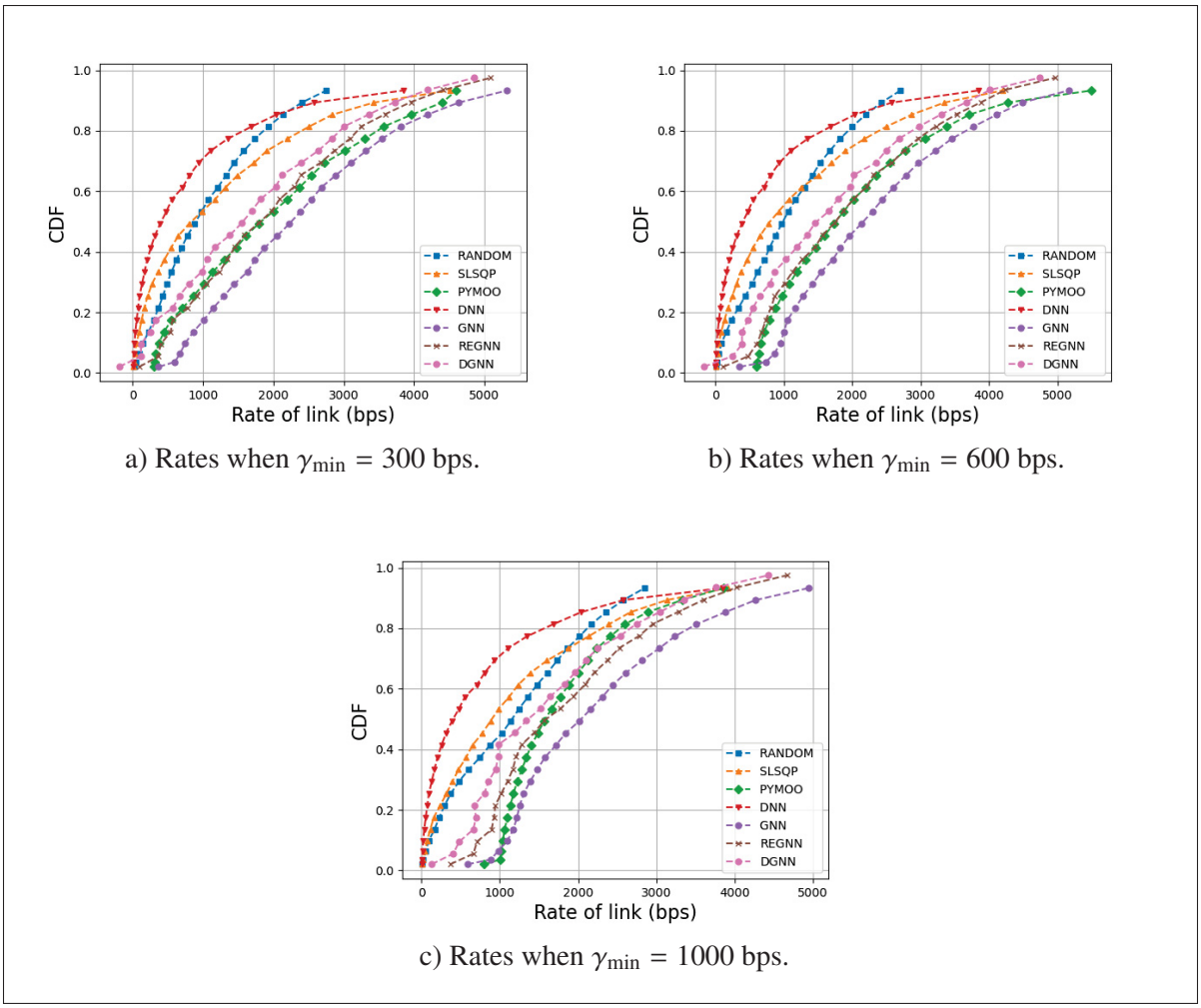


Figure 3.9 Model's average sum rate and QoS probability over unsupervised learning in different σ values

phase. This superiority is especially evident at extreme γ_{min} values where most schemes find it challenging to uphold the QoS requirements. It's worth noting that consistently maintaining extreme γ_{min} can be unfeasible for various channel realizations. Our model, as shown in (c), registers approximately 5% QoS violations, whereas other schemes hover around 50%, except for the PYMOO and REGNN approach.

Next, we show the average sum rate of the links, the probability of QoS constraint violation, and the level of QoS violation, which is defined as the difference between the QoS requirement and the rate when the QoS constraint is violated, i.e., $\mathbb{E}\left[\gamma_{min} - \gamma \mid \gamma_{min} > \gamma\right]$, as a function of γ_{min} in Figs. 13(a-c), respectively. In Fig. 13(a), the average sum rate is observed to decrease as γ_{min}



a) Rates when $\gamma_{\min} = 300$ bps.

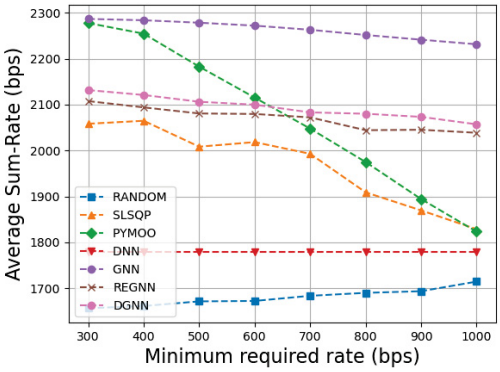
b) Rates when $\gamma_{\min} = 600$ bps.

c) Rates when $\gamma_{\min} = 1000$ bps.

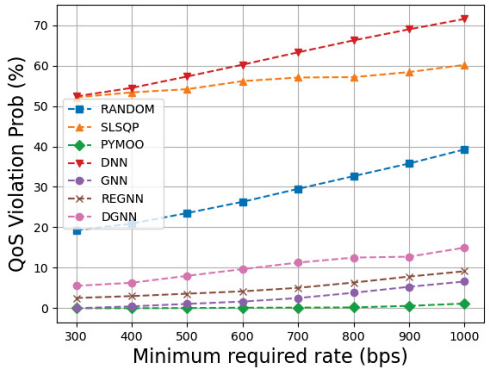
Figure 3.10 Rate CDF of benchmarking schemes for different $\gamma_{\min}$ values

increases. This decline is attributable to the restrictions on all links' transmit power and channel usage, which are necessary to minimize interference and, thus, to meet the QoS constraints. Notably, the proposed scheme maintains superior performance even as its average sum rate marginally declines. Moreover, this behavior widens the performance disparity with the PYMOO scheme as γ_{min} escalates. In contrast, other benchmark schemes consistently underperform relative to the proposed strategy. Regarding Fig. 13(b), the QoS violation probability for our model remains commendably low. Specifically, it hovers close to 0 when γ_{min} is minimal and reaches approximately 5% at higher γ_{min} values. This performance is comparable with the PYMOO, REGNN, and DGNN approaches. In contrast, the RANDOM, SLSQP, and DNN

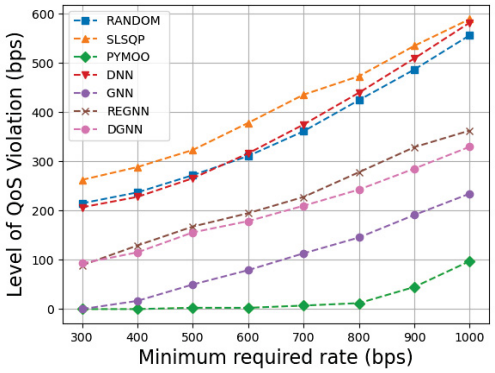
schemes exhibit substantially elevated violation probabilities, underscoring the efficacy of the proposed scheme. Lastly, Fig. 13(c) highlights the proposed scheme's performance in terms of QoS violation levels. Although the QoS constraint may occasionally be breached, the deviation of the link's rate from γ_{min} is still minimal (around 10%). This ensures that even when violations occur, their impact remains largely inconsequential.



a) Average sum-rate of all links.



b) Probability of QoS violation.



c) Level of QoS violation.

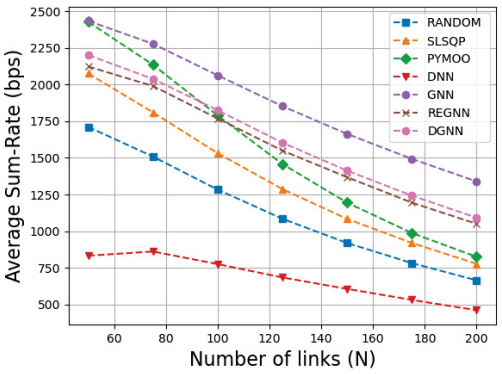
Figure 3.11 Performance of benchmarking schemes with respect to $\gamma_{\min}$

3.5.6 Impact of the Number of Links

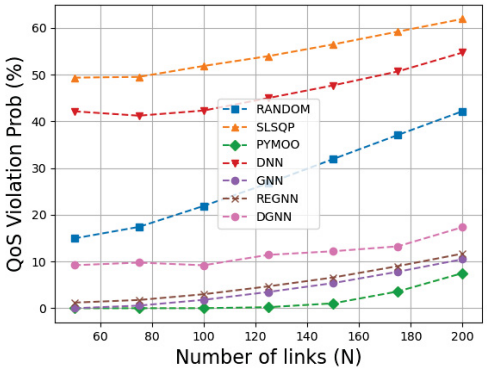
In Figs. 14(a) - 14(c), we illustrate the average sum rate of the network, the probability of QoS violation, and the level of QoS violation against the number of links, N. Fig. 14(a) highlights

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

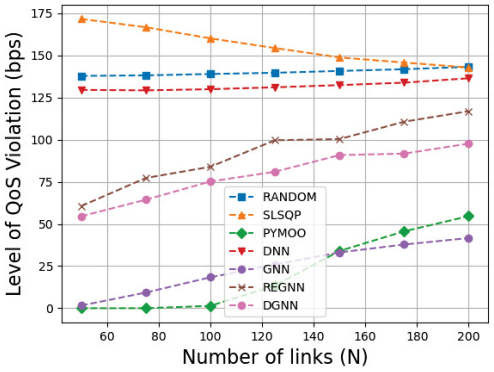
In Fig. 15, we assess the computation time required for resource allocation as a function of N. The simulation results focus solely on the computation time of the deployment phase, mainly because the GNN's training occurs just once and can be conducted offline prior to its actual implementation. It is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



a) Average sum-rate of all links.



b) Probability of QoS violation.



c) Level of QoS violation.

Figure 3.12 Performance of benchmarking schemes with respect to N

stemming from our graph modeling, has decreased the execution time by a factor of K. As a result, considering a higher K value renders our approach much more efficient than the other schemes.

3.5.7 Impact of Link Locations

In Figs. 16(a) - 16(c), we illustrate the average sum rate of the network, the probability of QoS violation, and the level of QoS violation against the maximum distance between each link, d_{max}. In Fig. 16(a), the GNN curve starts at approximately 2200 bps at 50 m and sees a steady decline,

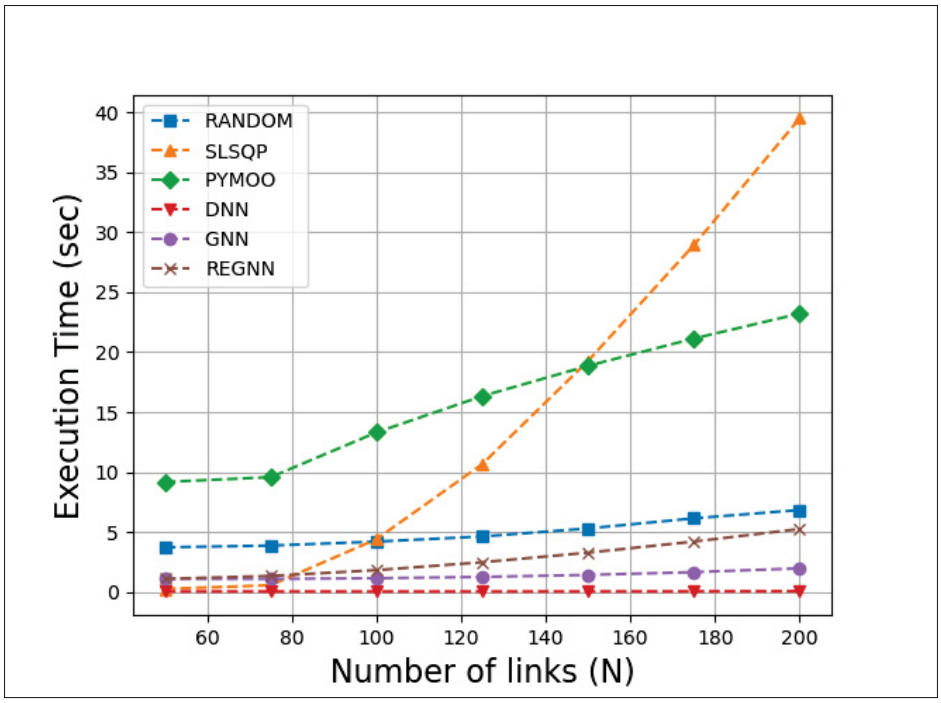
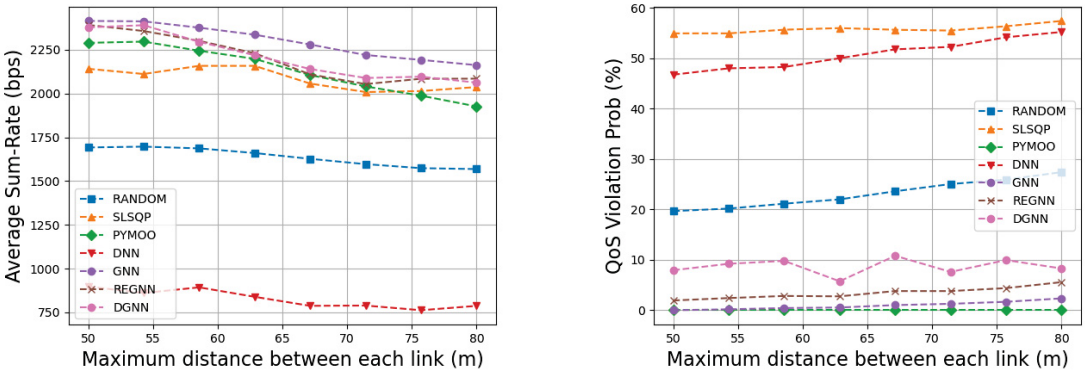


Figure 3.13 Execution time for different N values

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

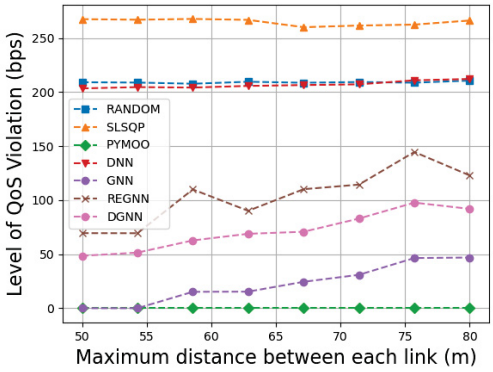
3.5.8 Impact of Fading Effects

In Figs. 17(a) - 17(c), we demonstrate the average sum rate of the network, the probability of QoS violation, and the level of QoS violation against the shadowing deviation, σ. Among the tested schemes, the GNN consistently outperforms its counterparts, achieving the highest



a) Average sum-rate of all links.

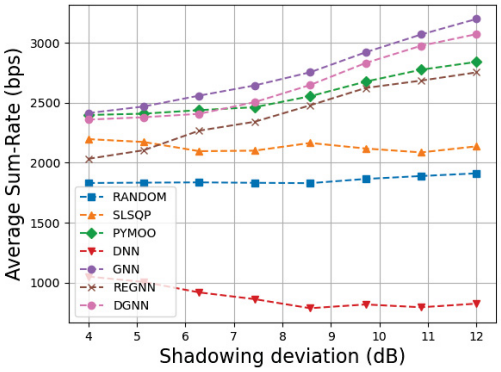
b) Probability of QoS violation.



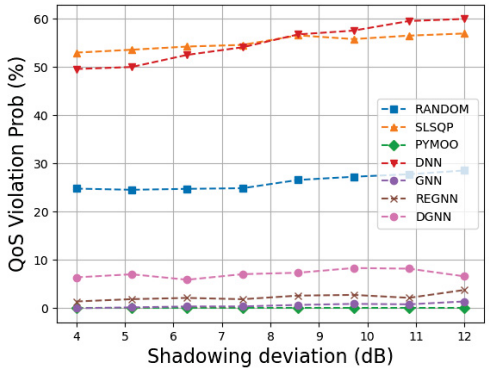
c) Level of QoS violation.

Figure 3.14 Performance of benchmarking schemes with respect to $d_{\max}$

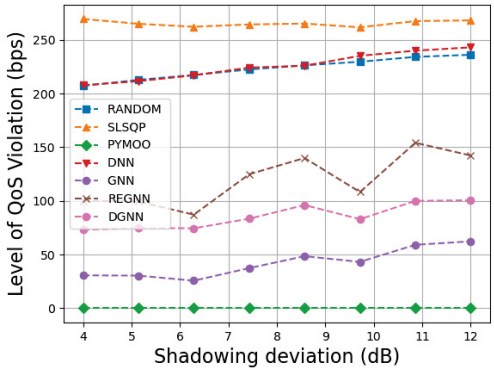
average sum rate across all shadowing deviations, as seen in Fig. 17(a). Furthermore, when it comes to ensuring the quality of service, the GNN demonstrated resilience, exhibiting the lowest probability and level of QoS violations, as shown in Fig. 17(b-c). This consistent superiority of the GNN emphasizes its potential as a highly reliable solution in environments with varying shadowing deviations.



a) Average sum-rate of all links.



b) Probability of QoS violation.



c) Level of QoS violation.

Figure 3.15 Performance of benchmarking schemes with respect to σ

3.5.9 Impact of Noisy CSI

In this subsection, we evaluate the resilience of our model when subjected to channel imperfections. These imperfections are quantified using a specific expression that defines the discrepancy between the estimated and actual multipath fading effects. The relationship between the estimated multipath fading values, h_{ij}^k, and their true counterparts, $\tilde{h}_{ij}^k$, is described by the first-order Gauss-Markov processHe *et al.* (2020a), expressed as follows:

$$\tilde{h}_{ij}^k = \sqrt{1-\sigma_e^2}h_{ij}^k + \sigma_e n_{ij}^k \tag{3.15}$$

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

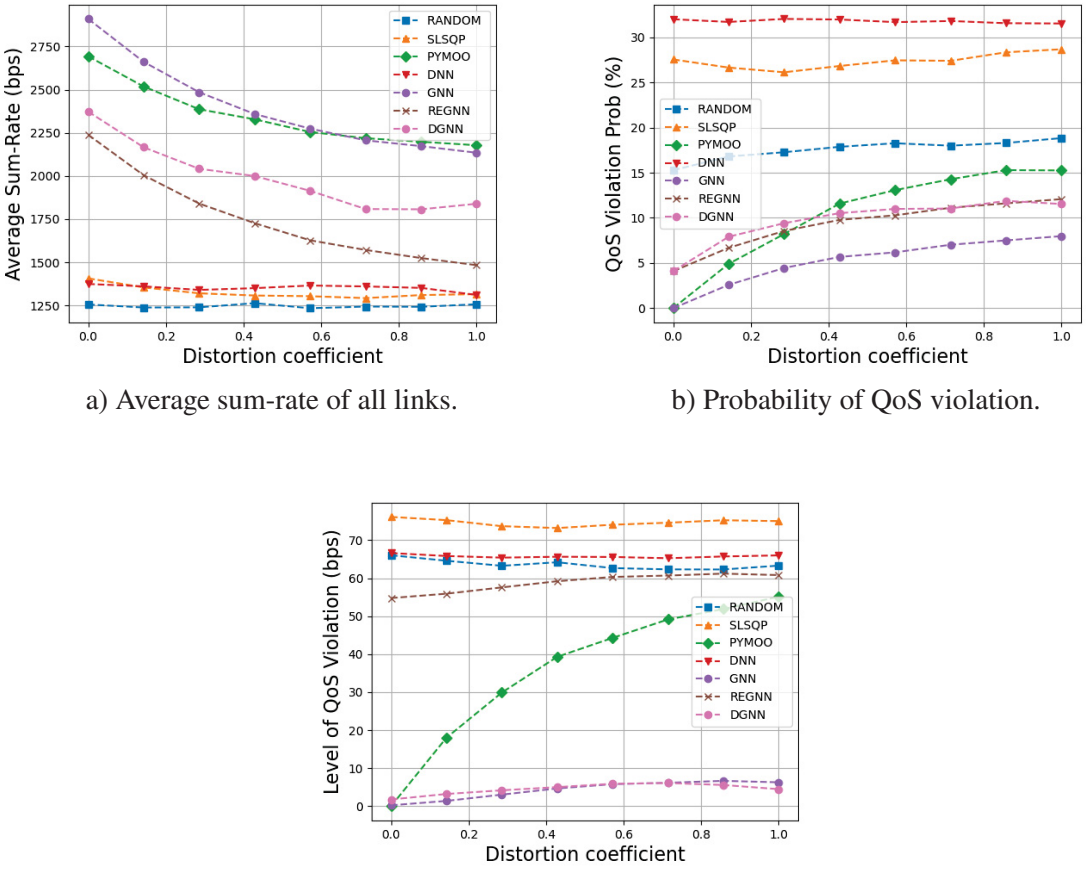
$$\tilde{g}_{ij}^k = \beta_{ij}^k \alpha_{ij}^k |\tilde{h}_{ij}^k|^2 \tag{3.16}$$

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

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

3.6 Conclusion

In conclusion, we presented a novel GNN-based framework for jointly solving power control and spectrum allocation in a non-orthogonal wireless environment. Our approach demonstrated superior performance in terms of average sum rate and Qos preservation in different wireless setups compared to other heuristic and learnable approaches and achieved robustness over an



a) Average sum-rate of all links.

b) Probability of QoS violation.

c) Level of QoS violation.

Figure 3.16 Performance of benchmarking schemes with respect to σ_e

imperfect channel. Additionally, our approach exhibited scalability, stability, and generalization, making it suitable for various network structures with different setups, such as D2D networks and Downlink-Uplink cellular scenarios. This study establishes a foundation for advanced RRM in future wireless networks. Future research should delve into GNNs' capabilities for dynamic spectrum allocation, interference management, and network optimization in changing conditions. Additionally, in-depth theoretical analysis is needed to pinpoint the best graph representations of wireless networks and fine-tune the GNN embedding layer. Lastly, incorporating temporal dynamics into GNN training could further improve RRM outcomes.

CHAPTER 4

SCALABLE SPATIAL AND GEOMETRIC LEARNING FOR JOINT POWER CONTROL AND CHANNEL ALLOCATION

Maher Marwani[1] , Georges Kaddoum[1,2]

[1] Electrical Engineering Department, École de Technologie Supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3
[2] Artificial Intelligence and Cyber Systems Research Center, Lebanese American University (LAU), Lebanon

Article published in *IEEE Transactions on Wireless Communications*, vol. 23, no. 11, pp. 16976–16991, 2024.

abstract

This research paper introduces an unsupervised scalable probabilistic approach for radio resource management in device-to-device (D2D) communication networks, essential for enhancing wireless data service capacity. We propose a joint optimization framework for spectrum allocation and power control, aiming to optimize the network's mean rate while meeting minimum data rate requirements. Although deep learning (DL) models have been explored for this purpose, their scalability is constrained by the fixed sizes of their input/output features, and their effectiveness is often limited by an insufficient understanding of the network's geometric structure. Consequently, Graph Neural Networks (GNNs) were introduced to integrate the wireless network's topology into the learning process. However, GNNs typically lose spatial correlation data when converting the tensorized channel state information (CSI) into a graph structure. To overcome this limitation, our solution combines GNNs, convolutional neural networks (CNNs), and variational autoencoders to extract meaningful embeddings from the CSI, preserving spatial and geometric features. We also introduce an innovative graph attention mechanism that enhances the model's focus on crucial node and edge features. Our holistic approach exploits the wireless network's topological and spatial relationships, offering a scalable, unsupervised, and generalizable solution without the need for retraining or

architectural adjustments across various wireless setups. Our findings confirm our method's superior performance and adaptability to different wireless environments.

4.1 Introduction

Device-to-device (D2D) communication is an evolving technology that facilitates direct data exchange between devices, bypassing the need for a base station or an intermediate network infrastructure. This innovative approach significantly reduces cellular system data traffic and shows great potential in revolutionizing connectivity landscapes and finding applications in diverse areas. Our research focuses on the challenges of managing D2D communication in environments with N-link interference, where multiple communication links operate on the same bandwidth resources. Such scenarios often lead to considerable interference and performance issues, stressing the need for efficient transmission power and spectrum allocation management. Tackling these issues requires solving a complex mixed-integer, non-convex utility maximization problem centered on power control and channel allocation. The difficulty lies in finding a globally optimal solution in a reasonable timeframe, a topic that has been extensively explored in the literature. This research is applicable to various communication scenarios, including Wireless Sensor Networks (WSN), Vehicular Networks, the Internet of Things (IoT), and traditional cellular communication. In WSN communication, resource blocks are shared and allocated to device pairs, enhancing the network's efficiency. In Vehicular Networks, vehicles communicate directly to support collision avoidance, traffic management, and autonomous driving. The IoT involves devices communicating with each other and central systems for tasks like home automation, industrial monitoring, and smart city applications, requiring efficient resource allocation. Additionally, traditional cellular communication benefits from optimized resource allocation in both uplink communication, where multiple devices communicate with a base station, and downlink communication, where a base station manages the resource allocation among devices. These optimizations, including power control and spectrum allocation, are necessary to improve communication efficiency, reduce interference, and boost network performance.

In the following, we first review deterministic-based resource allocation schemes, examining their limitations. We then discuss deep learning (DL)-based methods as an emerging learnable approach, discussing their potential and practical limitations. Next, we explore the recently proposed graph neural network (GNN) approaches, which are particularly intriguing as they incorporate the geometric properties of wireless networks, and we analyze their efficiency and unique characteristics in this context. Lastly, we present our research contributions and provide an overview of this paper's structure.

4.1.1 Deterministic-based Resource Allocation

Deterministic (Non-learnable) frameworks for D2D network resource allocation have been extensively studied. The work Yuan *et al.* (2018) proposed the use of iterative matching and Stackelberg game theory for dynamic channel and power control to reduce interference and enhance throughput. Additionally, the study Yang & Kuo (2017) introduced the Largest Aggregated Interference First Algorithm (LIFA) to maximize spatial reuse in OFDMA systems, improving throughput while reducing computation time. Research in Abdallah *et al.* (2018) proposed a stochastic geometry-based scheme allowing D2D links to share resources with cellular users, reducing interference and improving coverage. Also, the authors of Kai *et al.* (2018) devised a strategy for optimizing uplink and downlink resources, combining optimal power control with subcarrier allocation via maximum weight bipartite matching to enhance network performance. Similarly, the paper Liu *et al.* (2019) presented a two-stage approach merging optimal power control with a Gale-Shapley-based channel allocation algorithm, focusing on energy efficiency and quality of service. For UAV networks, the study Sun *et al.* (2021) optimized communication quality with the PSO-Kmeans algorithm, utilizing Adaptive Mutation Salp Swarm Algorithm for D2D optimization. For energy harvesting, the paperSu & Zhu (2019) employed the Kuhn-Munkres algorithm for optimal power and channel allocation under energy constraints. Also, a combined Hungarian algorithm with hybrid optimization for power allocation in multihop D2D communication, enhancing energy efficiency was proposed in A, PS, P & N (2023). These methods, though effective, are encumbered by reliance on

heuristic principles, significant computational demand, limited flexibility, and difficulties in contending with non-convex optimization problems. In contrast, our approach introduces a novel combination of GNNs, CNNs, and VAEs, addressing the limitations of these deterministic methods. By leveraging a scalable and unsupervised framework, we overcome the computational and flexibility challenges, providing a more adaptable and efficient solution for dynamic wireless environments.

4.1.2 DL-Based Resource Allocation

DL for resource allocation in D2D communications has seen advancements in multiple Resource Allocation problems, including channel selection and power control while maintaining quality of service (QoS) and minimizing the complexity and overhead associated with resource allocation. Deep Neural Networks (DNNs) were used in Zheng, Chi, Ding & Yu (2022) to improve resource allocation in D2D networks, particularly for URLLC in 5G, utilizing short packet coding and game theory algorithms to increase time efficiency. Also, a DNN approach was applied to multi-channel cellular networks with D2D communication in Lee & Schober (2022), combining supervised and unsupervised learning to efficiently manage mixed-integer optimization and achieve near-optimal results. Conversely, Convolutional Neural Networks (CNNs) were employed for channel allocation in D2D communications Li, Shen & Zhang (2022), effectively reducing interference and computational complexity while preserving QoS, demonstrating their viability in dynamic resource allocation contexts. Reinforcement Learning (RL) has also been applied to resource allocation, combining deep Q-learning with a CNN architecture to improve the system capacity and resource allocation efficiency Wang *et al.* (2021). Additionally, policy-based RL, combined with unsupervised learning, was used for channel allocation and power control, focusing on optimizing energy efficiency and transmission rate Sun *et al.* (2023b). Furthermore, an Online Distributed Deep RL (OD-DRL) algorithm was investigated in Sun & Nakhai (2021) to manage channel selection and power control in dynamic cellular network scenarios, ensuring QoS for D2D and cellular users. However, DNNs and CNNs face limitations like input/ output size inflexibility, overfitting, training data dependency, and convergence instability. In

comparison, our unsupervised framework reduces dependency on labeled data and mitigates issues of overfitting and convergence instability. It is also size invariant, making it more robust and adaptable to varying network sizes without retraining. GNNs emerge as a robust alternative for dynamic wireless networks, handling diverse data structures and adapting to complexity

4.1.3 GNN-Based Resource Allocation

Recent GNN advancements have contributed to enhancing resource allocation and network optimization. Various GNN-based models targeting specific wireless network challenges have been proposed. In Nakashima *et al.* (2020), an RL-based channel allocation method with graph convolutional networks was shown to improve system throughput but faces scalability issues. Eisen & Ribeiro (2020) proposed Random Edge Graph Neural Networks (REGNNs) for scalability and adaptability but struggled in dynamic environments. Guo & Yang (2021) employed Heterogeneous Graph Neural Networks (HetGNNs) for efficient power control in cellular networks. Chowdhury *et al.* (2021) introduced unfolded weighted minimum mean square error (UWMMSE) for robust and generalizable power allocation. Shen *et al.* (2021) focused on scalable RRM using Message Passing Graph Neural Networks (MPGNNs) and Wireless Channel Graph Convolution Networks (WCGCN) for practical applications. In Zhang *et al.* (2021a), semi-supervised learning with GNNs was proposed to optimize user association and power control in Ultra-Dense Networks; However, the solution suffers performance degradation in the face of a dynamic environment. Zhang *et al.* (2021b) introduced Heterogeneous Interference Graph Neural Network (HIGNN) for scalable and adaptable power control and beamforming. Ranasinghe *et al.* (2021) proposed an access point selection algorithm using GraphSAGE Hamilton, Ying & Leskovec (2017) for efficiency in cell-free massive MIMO systems. Chen *et al.* (2022) presented a GNN-based framework for resource allocation in wireless IoT networks that is adaptable in various operational environments. He *et al.* (2022) optimized beam selection and link activation in mmWave networks with a GNN-based model, effective in ultra-dense networks. Shen *et al.* (2023) analyzed GNN use in wireless communications, bridging the gap between theoretical expectation and practical implementation in terms of effectiveness, adaptability, and

reliability. These studies highlight GNNs' potential in optimizing radio resource management problems, but also point out several challenges in dynamic environments. GNNs must adapt to changing topologies, manage large networks effectively, ensure real-time processing, generalize across diverse network characteristics, function with limited training data, and remain robust against uncertainty and noise. In contrast, our approach combines GNNs with CNNs and VAEs to preserve spatial and geometric CSI features. We introduce a novel graph attention mechanism to focus on crucial node and edge features, addressing GNNs' typical loss of spatial correlation data. This method provides a scalable, unsupervised, and generalizable solution that maintains high performance and robustness across various wireless environments, overcoming traditional GNN limitations.

4.1.4 Motivation and Contributions

In this paper, we introduce an innovative approach to power and channel allocation that addresses the limitations of the above-mentioned methods. Firstly, we recognize that heuristic and iterative solutions, while comprehensive, are time-consuming and not well-suited for rapidly changing wireless scenarios. Secondly, supervised DL models, while fast, face challenges, such as limited training data, inflexibility in handling varying input/output sizes, and susceptibility to overfitting. To overcome these obstacles, we propose an unsupervised, scalable, and generalizable solution. Our solution leverages GNN as an embedding block, offering a scalable and size-invariant method particularly adapted to addressing wireless communication problems. GNN also provides a geometric perspective on CSI by learning from the interactions of nodes and their edges. However, one challenge is the potential loss of information when transforming the CSI into a graph and not considering its tensor form. To address this, our proposed approach integrates spatial and geometric learning components to embed the CSI input comprehensively. This ensures that we harness the full potential of the input data, including its tensorized form. Additionally, our model is designed to be unsupervised, enabling it to learn without labeled data, which is a significant advancement over traditional supervised models. Moreover, our model maintains a fixed parameter count regardless of the number of links, ensuring scalability.

Moreover, we integrate a probabilistic framework into our approach, allowing it to learn a general CSI distribution, thereby ensuring generalization across various scenarios. In summary, our approach offers an effective solution for power and channel allocation in wireless networks. It is characterized by its unsupervised and scalable nature and capability to learn from both graph and tensor representations of CSI. The key contributions of our research can be comprehensively detailed as follows:

1. **Development of a Spacial Variational Auto-Encoder (VSAE):** We introduces a CNN-based VSAE designed to learn the spatial distribution of a given CSI. Notably, the VSAE is constructed with a size-invariant architecture to ensure scalability across different scenarios.
2. **Graph Structured Modeling of multi-channel CSI:** We represent CSI as a graph structure by incorporating the channel gains between links in the node and edge features. This approach offers another perspective for analyzing and learning from the CSI data, adding a new dimension to our analysis.
3. **Creation of a Geometric Variational Auto-Encoder (VGAE):** We develop a GNN-based VGAE that specializes in learning the geometric distribution of the CSI. A key feature of the VGAE is its encoder, which utilizes a novel graph left and right attention mechanism, namely, LRGAT. This mechanism is adept at understanding the significance of both node features and edge features, considering their importance in both source and destination nodes.
4. **Designing a Task-Specific Projection Mechanism:** We construct a projection strategy that integrates both spatial and geometric embeddings. This strategy effectively outputs power and channel allocation solutions. Importantly, like other components of our system, this architecture is designed to be scalable. We highlight that although we are focusing on solving a particular problem, this strategy can be modified to accommodate various RRM tasks.
5. **Comprehensive Performance Evaluation:** The proposed method is rigorously tested under a variety of wireless network setups. Initial evaluations highlight the critical role of spatial and geometric embeddings. Simulation results affirm that our proposed scheme is capable of achieving near-optimal performance while maintaining low computational

requirements. When compared to heuristic approaches and solutions based on DNN, CNN, and GNN, our method demonstrates superior performance, especially in terms of constraint preservation and improvement of the network mean rate across different wireless setups. Additionally, our results show that our model maintains robustness even in the presence of imperfect CSI.

4.1.5 Organization

The remainder of this paper is organized as follows: Section II introduces the system model and problem formulation under consideration. Section III details the proposed solution structure for resource allocation, including the architectures of the VGAE, the VSAE, and the projection mechanism. Section IV is dedicated to explaining the training methodology. In Section V, we present the simulation results. Finally, Section VI concludes the paper.

4.2 Power Control, and Spectrum Allocation in D2D Networks

We first present the adopted system model. Then, the problem statement is provided, and the considered resource allocation problems are formulated.

4.2.1 System Model

We're investigating an N-link interference environment, which consists of N links, otherwise known as D2D pairs. As depicted in Figure 1, these D2D pairs communicate and interfere with each other over shared channels. A key assumption in our setup is the presence of a centralized control unit (CCU), similar to Yuan *et al.* (2018); Yang & Kuo (2017). The CCU manages the radio resources of all the links, including the spectrum allocation and the power control for each time slot. Thus, the CCU ensures smooth and efficient communication by mitigating interference and effectively managing resources. In this context, we're working within an environment where the CCU is fully aware of the CSI. This level of awareness is achieved through a two-step process involving pilot transmission and CSI reporting across all D2D pairs. The process begins when

the transmitters of D2D pairs broadcast reference signals across all the channels. Following this transmission, the receivers estimate the CSI on each channel. These estimations are then reported back to the CCU. In our system model, we assume perfect channel estimation Lee *et al.* (2020); Kai *et al.* (2018); Shen *et al.* (2021, 2023). However, we demonstrate later that our approach is resilient in the face of imperfect CSI.

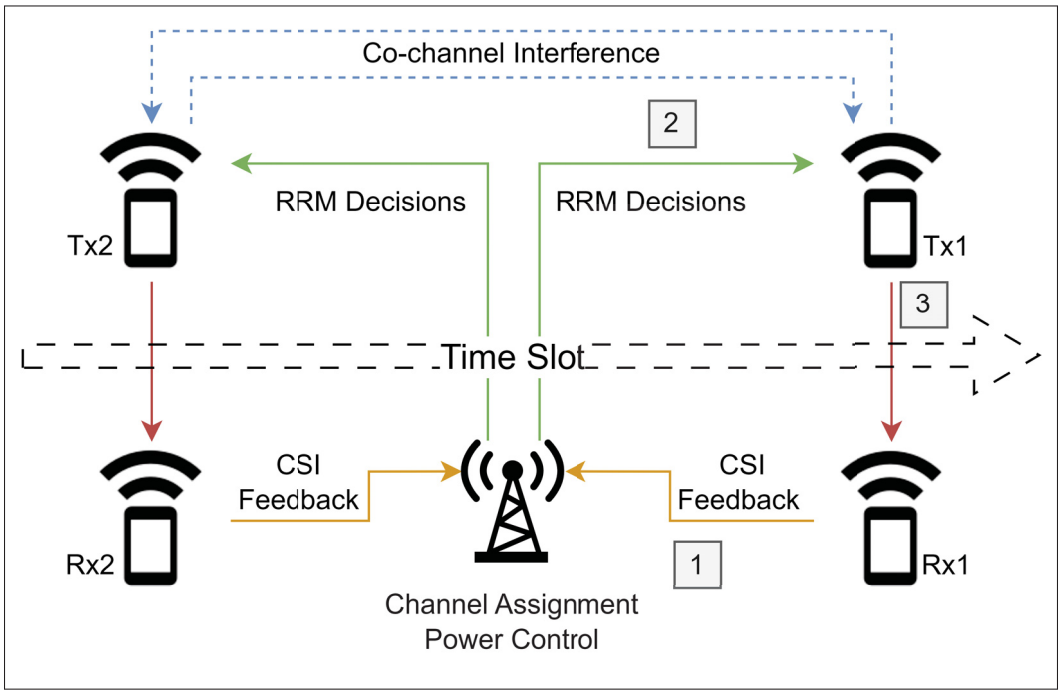


Figure 4.1 System model of the centralized RRM of 2 D2D links communicating and interfering with each other for a given time slot

Considering that RRM tasks are interconnected, our goal is to identify a joint optimal strategy that seamlessly integrates power control and channel allocation. The strategy we're aiming to develop needs to not only maintain the QoS requirements but also comply with the given constraints.

4.2.2 Problem formulation

We define $\mathcal{D} = \{1, 2, \ldots, N\}$ as a set of D2D pairs, randomly distributed within a two-dimensional environment. The distance between each transmitter-receiver pair can vary from

one link to another. We employ a non-orthogonal scheme for all communication links and define $\mathcal{K} = \{1, 2, \ldots, K\}$ as the set of resource blocks (RBs). Each RB has a constant bandwidth B and can be allocated to any link. The binary variable $x_i \in \{0, 1\}$ is introduced to signify the activation status of the i-th link. If $x_i = 1$, the i-th link is activated, and if $x_i = 0$, it is not. The variable $p_i^k \in [0, p_{max}]$ represents the transmission power on the k-th RB of the i-th link. Lastly, we introduce the binary variable $\psi_i^k \in \{0, 1\}$ to signify the RB assignment for the i-th link. When $\psi_i^k = 1$, the i-th link is utilizing the k-th RB. Conversely, $\psi_i^k = 0$ indicates that the k-th RB is not in use by the i-th link.

We adopt the block fading model such that all channel gains can change independently from one frame to the next. We introduce $g_{ii}^k \in \mathbb{R}$ to represent the direct channel gain between the transmitter and receiver of the k-th RB in the i-th link, and $g_{ij}^k \in \mathbb{R}$ to denote the channel gain between the transmitter of the j-th link and the receiver of the i-th link. To represent all channel gains, we define $G \in \mathbb{R}^{N \times N \times K}$ as the CSI tensor across all links in all resource blocks. Taking into consideration all types of fading, we have:

$$g_{ij}^k = \xi_{ij}^k \, \tau_{ij}^k \, |\kappa_{ij}^k|^2 \quad \forall k \in \mathcal{K}, \quad \forall (i, j) \in \mathcal{D} \times \mathcal{D}, \tag{4.1}$$

where ξ_{ij}^k is the path loss, τ_{ij}^k is the shadowing, and κ_{ij}^k represents the small scale Rayleigh fading. We assume that there is a dedicated bandwidth only used for D2D communication (an overlay configuration), therefore the interference can only happen between the D2D pairs. Based on that, the signal-to-interference-plus-noise ratio (SINR) for the k-th RB at the i-th receiver can be obtained as:

$$\text{SINR}_i^k = \frac{g_{ii}^k p_i}{\sum_{j \neq i}^{N} \psi_j^k g_{ij}^k p_j + N_0 B}, \forall (i, k) \in \mathcal{D} \times \mathcal{K}, \tag{4.2}$$

where N_0 is the noise density per unit bandwidth. Consequently, we calculate each link's achievable rate using Shanon's theorem as follows:

$$R_i = \sum_{k=1}^{K} B \psi_i^k \log \left(1 + \text{SINR}_i^k\right), \forall i \in \mathcal{D}. \tag{4.3}$$

Our objective in this study is to discern the optimal configuration for the transmission power vector, represented as $P \in [0, p_{max}]^N$, and the RBs matrix signified as $\Psi \in \{0,1\}^{K\times N}$. The optimization of these variables is driven by the primary aim of maximizing the network mean rate for each discrete time slot. This critical objective gives rise to our optimization problem, which can be comprehensively described as follows:

$$
\begin{aligned}
\max_{P,\Psi} \quad & \frac{1}{N}\sum_{i=1}^{N} R_i \\
\text{s.t} \quad & R_i \geq R_{min} && \forall i \in \mathcal{D} \\
& \sum_{k=1}^{K} \psi_i^k \leq 1 && \forall i \in \mathcal{D} \\
& 0 \leq p_i \leq p_{max} && \forall i \in \mathcal{D} \\
& \psi_i^k \in \{0,1\} && \forall (k,i) \in \mathcal{K}\times\mathcal{D}
\end{aligned}
\tag{4.4}
$$

This optimization problem is a Mixed Integer Nonlinear Programming (MINLP) problem, aiming to maximize the network's average rate while respecting the quality of service. In (4), the first constraint ensures a minimum required data rate is met, the second constraint limits each link to using only one channel, and the third and fourth constraints restrict transmission powers within the defined maximum power and enforce binary channel indicators.

To prove that the given optimization problem is NP-hard, we reduce the well-known NP-hard 0-1 Knapsack problem to our problem in polynomial time. The 0-1 Knapsack problem involves maximizing the total value of items selected without exceeding a weight limit. In our problem, each D2D pair can be mapped to an item in the knapsack, the achievable rate corresponds to the item's value, and the transmission power corresponds to the item's weight. The total power constraint matches the knapsack's capacity. Maximizing the network mean rate is akin to maximizing the total value, and constraints ensure each selected item meets a minimum value and adheres to power limits, aligning with the knapsack's binary nature. Since this reduction can

be done in polynomial time and the 0-1 Knapsack problem is NP-hard, our D2D communication problem is also NP-hard.

To solve this problem, we propose applying GNN, CNN, and VAE approaches. GNN layers are designed to perceive CSI as a graph, leveraging permutation equivariance properties for enhanced generalization. GNNs focus on all neighbors of each node, capturing intricate relationships in graph-structured data. This approach is common in network optimization tasks and provides a deeper understanding of the problem's dynamics. Conversely, CNN layers apply the same filter across different parts of the input to detect local spatial correlations by perceiving CSI in a tensorized form. These spatial correlations are generated between D2D pairs when the channel conditions of one D2D pair may relate to another, especially if they are geographically close or experience similar fading. Additionally, correlations occur between RBs, where channel conditions across different time slots can reveal correlated patterns. CNNs' ability to extract and process key features from large, complex CSI enables a more manageable representation of high-dimensional problems, simplifying the optimization process. While CNNs focus on local neighborhoods, the GNN encoder aggregates information from all neighboring nodes because the interference graph is complete. Using a combination of GNN and CNN layers provides meaningful geometric and spatial embeddings. Moreover, we include a VAE approach to enhance the generalization of our model by learning an approximation of the CSI distribution. This approach allows for better generalization and robustness in varying conditions.

4.3 Solution Architecture

In this section, we outline the main elements of our proposed framework. We also provide a step-by-step description of the model, from the initial CSI input, through the spacial and geometric embeddings, to the final optimization variables.

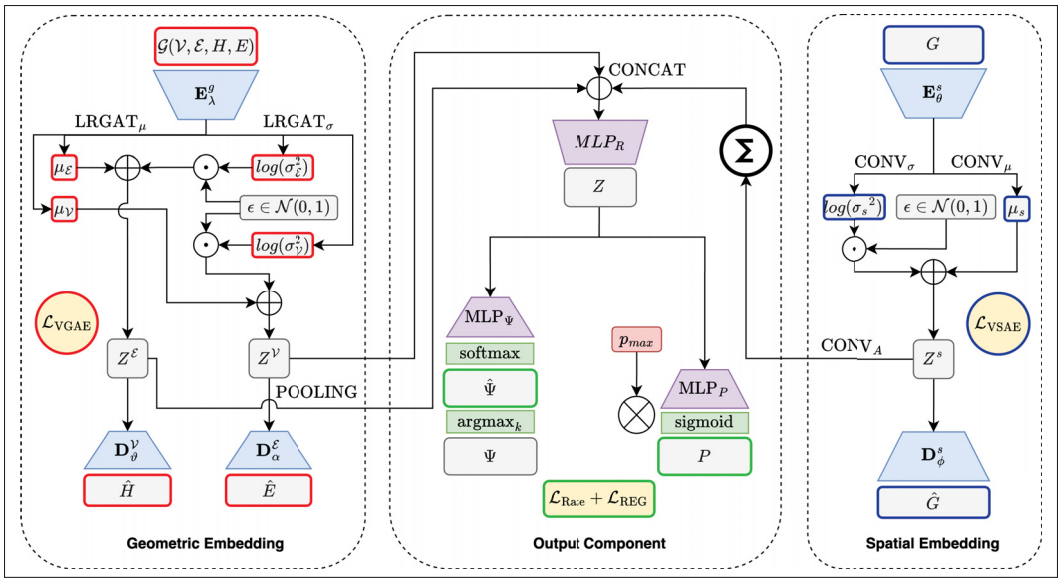


Figure 4.2 Our solution architecture leverages geometric and spatial embeddings from CSI to derive RRM variables. It includes three main components: Geometric Embedding, Spatial Embedding, and Output Component. The Geometric Embedding uses the VGAE to process the CSI-derived graph, generating geometric embedding matrices $Z^\mathcal{V}$ and $Z^\mathcal{E}$ through LRGAT layers. The Spatial Embedding employs the VSAE to process the CSI tensor, producing a spatial embedding matrix Z^S via CNN layers. These embeddings are concatenated and refined through an MLP block, yielding a global embedding matrix Z. This matrix is further processed to derive the optimal variables P and Ψ

4.3.1 Overview

Our model consists of three key parts: spatial learning, geometric learning, and output generation. These elements work together to embed the CSI input G and generate the optimal variables P and Ψ for the problem. Figure 2 shows the main components of the model, including the data flow, and the loss calculation. The VGAE takes a graph modeled from the CSI and learns to construct the graph's nodes and edges features, H and E, respectively. This learning phase yields two geometric embedding matrices, $Z^\mathcal{V}$ and $Z^\mathcal{E}$. In a similar manner, the variational spatial autoencoder (VSAE) inputs the CSI tensor G, learning its construction process. This learning phase results in a spatial embedding matrix that encapsulates the CSI's spatial details. Following this, we use an adaptive convolutional layer to project the intermediate latent matrix

Z^S into a spatial embedding matrix that has the same first dimension as the geometric embedding matrices. We concatenate both the spatial and geometric embeddings and feed them into an MLP block, MLP_R, for further refinement. The final product is a global embedding matrix Z that encapsulates various CSI details, including the correlation among its components and the connections between the links. These global embeddings are later employed to extract the two variables using the two MLP blocks, MLP_P and MLP_Ψ, while maintaining the inter-variable relationships and the problem constraints. Our approach is focused on three primary aspects:
1) **It is entirely unsupervised**, negating the need for any labeled data. This aspect is critical considering the scarcity of wireless data. Multiple existing solutions have trained their DL architectures in a supervised way, attempting to emulate a specific optimization process. However, this approach has a limitation: the resultant solution tends to mirror the training data, performing at or below the level of the training data in practice.
2) **It is architecturally scalable**, featuring a constant number of parameters that remain unaffected by the number of links, demonstrating size invariance. This is in contrast to DNN architectures where the number of parameters increases linearly with the input size. This characteristic is important as it enables us to train our architecture on a limited number of links, and then extend it to larger numbers without any alteration to the architecture. Later, we demonstrate our model's capacity to generalize and scale with only a few fine-tuning iterations.
3) **It is intrinsically probabilistic**, aiming to understand the overall distribution of CSI rather than pinpointing specific instances. This perspective is paramount due to the CSI's dynamic and unpredictable characteristics. Leveraging the capabilities of VGAE and VSAE, the model undergoes training with a wide range of samples, which facilitates the learning of a comprehensive CSI distribution. Such a method guarantees a robust comprehension of the various CSI fluctuations.

The next subsections explain each component of the proposed architecture in detail.

4.3.2 Spatial Learning

The principal idea of the proposed architecture is to create a Variational Autoencoder (VAE) Kingma & Welling (2022) that ensures a constant size bottleneck regardless of the input CSI size. To achieve this, the architecture consists of a spatial encoder $\mathbf{E}^s_\theta$ and a decoder $\mathbf{D}^s_\phi$, with parameters θ and ϕ, respectively. To maintain flexibility over varying input sizes, both the encoder and decoder employ multiple two-dimensional convolutional filters, which possess input-size invariant properties. The settings for these filters have been standardized with a kernel size of 3 and a stride of 1, and padding or dilation is intentionally not used. This ensures that the output size remains the same as the input, though the number of channels may vary. The process starts with reducing the number of input channels, which is equivalent to the number of RB (K), to 2. These 2 channels are then convolved to produce mean and variance matrices. The latent matrix is sampled from these mean and variance matrices. Finally, the latent matrix is upsampled to restore the original number of input channels, completing the generation process.

The VSAE operates by creating a bottleneck that utilizes two matrices. These matrices are produced by the encoder, $\mathbf{E}^s_\theta$, with its final layer containing two separate convolutional layers: $\text{CONV}^{\text{s}}_\mu$ and $\text{CONV}^{\text{s}}_\sigma$. The matrix of means is generated by the operation $\mu_s = \text{CONV}^{\text{s}}_\mu(\mathbf{E}^s_\theta(G))$, resulting in a matrix $\mu_s \in \mathbb{R}^{N\times N}$. The encoder also produces a matrix of log variances, $\log({\sigma_s}^2) \in \mathbb{R}^{N\times N}$, via the operation $\log({\sigma_s}^2) = \text{CONV}^{\text{s}}_\sigma(\mathbf{E}^s_\theta(G))$. The use of $\log(\sigma^2_s)$ instead of σ^2_s directly is a strategic choice that enhances the VSAE's stability and efficiency. This approach offers numerical stability, facilitating smoother optimization during training. It plays a crucial role in implementing the reparameterization trick, a key component for enabling gradient-based optimization. Additionally, $\log(\sigma^2_s)$ effectively captures a broad range of data uncertainties, providing a more flexible representation of the model's confidence. Lastly, this logarithmic representation naturally ensures the positivity of the standard deviation, eliminating the need for imposing extra constraints in the optimization process. Given these matrices, we can proceed to sample a latent matrix Z^S. Each component in Z^S is sampled from a normal distribution. The parameters of this distribution,i.e., mean and variance, come from the

corresponding elements in the matrices μ_s and σ_s. This process is represented by the equation:

$$Z_{ij}^S \sim \mathcal{N}\left((\mu_s)_{ij}, (\sigma_s)_{ij}^2\right) \longrightarrow Z_{ij}^S = (\mu_s)_{ij} + (\sigma_s)_{ij} \odot \epsilon, \tag{4.5}$$

where ϵ is drawn from $\mathcal{N}(0, 1)$ and $\odot$ is an element-wise multiplication. Upon obtaining Z^S, we calculate the reconstructed output $\hat{G}$ by decoding it through $\hat{G} = \mathbf{D}_\phi^s(Z^S)$. This output enables us to compute the reconstruction and KL divergence losses, leading to the subsequent update of the model parameters ϕ and θ.

The spatial embeddings of the VSAE, represented as Z^S, have an initial dimensionality of (N, N). This size is directly dependent on the parameter N, presenting a scalability issue. To resolve this and create a size-invariant embedding matrix, we aim to transform Z^S into a matrix of size (N, d_s) where d_s is the spacial embedding dimension. This transformation is achieved through a convolution operation, denoted as CONV_A, which upsamples Z^S to d_s channels. After upsampling, we average the first dimension of Z^S, thereby obtaining an output with the desired dimensions of (N, d_s). This process is mathematically expressed as $\hat{Z}_{ik}^S = \frac{1}{N}\sum_{i=0}^{N} \text{CONV}_\text{A}(\text{Z}^\text{S})_{(ijk)}$, where each element of the updated Z^S is the average of the CONV_A operation applied across its spatial dimensions.

Later on, we concatenate these transformed spatial embeddings $\hat{Z}^S$ with geometric embeddings to produce a global embedding of the CSI, which is a key component in our model.

4.3.3 Geometric Learning

The VGAE shares a similar architectural foundation with the VSAE explained above, albeit with distinct encoder and decoder structures tailored for graph data topologies instead of tensors. This study focuses on the learning process from graphs, specifically modeled from the CSI. The encoder component of the VGAE takes a graph as input and transforms it into a latent vector obtained through sampling from mean and variance vectors. Subsequently, the decoder reconstructs the adjacency matrix for the graph using this latent representation. The objective of the VGAE is to effectively learn the construction of the adjacency matrix, ultimately generating

a meaningful latent vector that holds utility for subsequent tasks. Furthermore, particular attention is given to maintaining input size invariance concerning the number of parameters of the VGAE, ensuring adaptability to graphs of varying sizes without compromising performance or computational efficiency. Through this framework, the VGAE facilitates the acquisition of meaningful graph representations from CSI data, enabling graph embedding and exploration while capturing essential relationships and patterns inherent in the network topology.

Our approach draws inspiration from two prominent concepts in graph representation learning: Graph Attention Network (GAT) Veličković *et al.* (2018) and VGAE Chen & Chen (2021). VGAEs adopt the principle of variational inference to approximate complex probability distributions. Typically, VGAEs comprise two key components: an encoder and a decoder. The encoder employs a graph convolution network (GCN) Kipf & Welling (2017) to reduce the dimensionality of the input graph data, transforming it into a latent space that captures essential features. Subsequently, the decoder reconstructs the adjacency matrix using an inner-product encoder, enabling the extraction and comprehension of intricate patterns within the graph-structured data.

Conversely, the GAT presents a powerful message-passing algorithm Wu *et al.* (2021) for graph embedding, specifically designed to overcome limitations associated with traditional graph convolutional networks. GATs introduce attention mechanisms into the aggregation process, enabling nodes to selectively focus on relevant neighbors while disregarding less relevant ones. In each iteration, every node computes attention coefficients for its neighbors, leveraging learned weights to determine the significance of each neighbor's contribution. These attention coefficients are then utilized to compute a weighted sum of neighboring node embeddings, which, when combined with the node's own embedding, generates its updated representation. The incorporation of attention in the GAT allows it to effectively capture complex relationships and variations in graph structures, making it highly proficient in node embeddings.

However, the GAT framework is designed exclusively for graphs with only node features, rendering it unsuitable for our specific case. Our work seeks to enhance the GAT framework

by incorporating edge features, introducing an additional edge attention coefficient, and implementing an updated rule for edge features. Moreover, we aim to integrate this modified encoder into the VGAE architecture, replacing the conventional GCN. Through these advancements, we expect to achieve improved performance and effectiveness in handling our graph-structured data, which contains both node and edge features.

In the next subsections, we describe the graph modeling process, our LRGAT convolution layer, which serves as a building block for the VGAE encoder, and finally, the general structure of the VGAE.

C.1) Multi-channel CSI Graph Modelling:

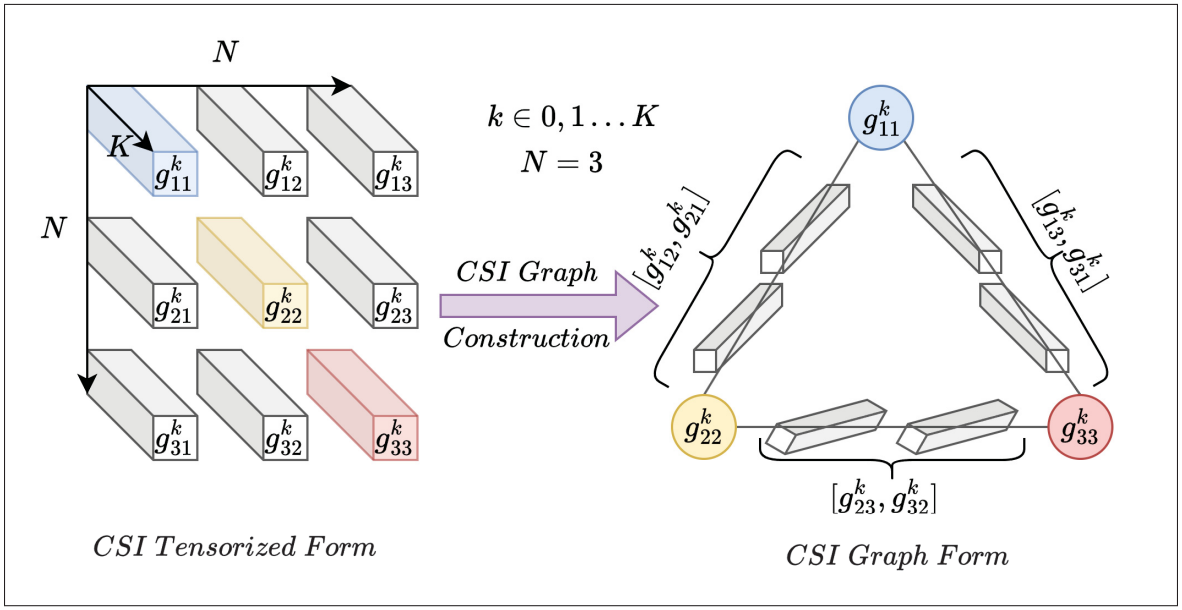


Figure 4.3 Transformation of the CSI tensor into an interference graph. Nodes in blue, yellow, and red represent communication links across K channels with CSI (g^k_{11}, g^k_{22}, g^k_{33}) as node features. Gray structures represent CSI of interference links as edge features. The graph is a complete directed graph with 3 nodes and 3 edges

We describe the process of transforming the CSI tensor G into a graph structure to gain insights into their geometric properties. We denote $\mathcal{G}(\mathcal{V}, \mathcal{E}, E, H)$ as the interference graph, where the node set $\mathcal{V}$ denote the active communication links, and the edges set $\mathcal{E}$ represent the interference links. Each node has a feature vector $h_i = \|_{k=0}^{K} G_{ii}^k, \quad \forall i \in \mathcal{V}$, and each edge has a feature

vector $e_{ij} = \|_{k=0}^{K} (G_{ij}^{k} \parallel G_{ji}^{k}), \quad \forall (i,j) \in \mathcal{V} \times \mathcal{V}$. The operator $\parallel (.,.)$ refers to a two-vector concatenation operation. Since any number of links can use any channel, the constructed graph is a complete undirected graph with N nodes and $N(N-1)/2$ edges.

C.2) LRGAT Convolution Layer:

Here, we provide a comprehensive description of our encoding layer, specifically termed the Left Right Graph Attention Network (LRGAT). The essence of LRGAT is its ability to employ both node and edge attention mechanisms in the learning process of the GNN, thereby leveraging the structure of the graph data to its fullest extent.

Starting with the nodes, we transform the node features using a feed-forward layer, denoted by $W_h \in \mathbb{R}^{K \times F}$. This transformation equips the network with a higher-order learning capability, thus enabling the discernment and integration of more complex patterns within the data. After that, a learnable weighted vector, $\vec{a} \in \mathbb{R}^{F}$, is used to compute the attention coefficients α_{ij}, given by:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\vec{a}[W_h h_i || W_h h_j]))}{\sum_{k \in \mathcal{A}_i \cup i} \exp(\text{LeakyReLU}(\vec{a}[W_h h_i || W_h h_k]))}, \tag{4.6}$$

The LeakyReLU(x) = $\max(0.001x, x)$ function mitigates the "dying ReLU" issue Maas (2013) by permitting a slight gradient for $x < 0$, facilitating continuous neuron learning even with negative inputs. In this equation, the product of $\vec{a}$ and the concatenation of transformed node features is passed through a LeakyReLU activation function. The resulting value is then normalized over the neighborhood of node i, denoted as $\mathcal{A}_i$, (including node i itself) using a softmax function. The node features are subsequently updated according to the equation:

$$h_i^{(t+1)} = \alpha_{ii} W_h h_i^{(t)} + \sum_{j \in \mathcal{A}_i} \alpha_{ij} W_h h_j^{(t)} \tag{4.7}$$

This process involves a weighted sum of the transformed features of the neighboring nodes, resulting in an effective aggregation of neighborhood information.

Alongside the node attention mechanism, our LRGAT also extends this process to the edges of the graph. To accomplish this, we utilize a distinct learnable weighted vector, $\vec{b} \in \mathbb{R}^{F+F'}$, which

is instrumental in the calculation of left and right attention coefficients, β_{il}^{L} and β_{rj}^{R}, respectively. Just as with the node features, the edge features are transformed using a distinct single-layer feed-forward network, represented by $W_e \in \mathbb{R}^{2K \times F'}$. The left edge attention coefficients are computed as:

$$\beta_{il}^{L} = \frac{\exp(\text{LeakyReLU}(\vec{b}[W_e e_{il} || W_h h_l]))}{\sum_{k \in \mathcal{A}_i} \exp(\text{LeakyReLU}(\vec{b}[W_e e_{ik} || W_h h_k]))} \tag{4.8}$$

The process is similar to the node attention coefficient, but here we concatenate the right edges features with their corresponding node features. The idea here is that an edge contains information within its features, its connections, and the features of the two connected nodes. Which is all that has been used to calculate β_{il}^{L}. Similarly, the right edge attention coefficients are computed as:

$$\beta_{rj}^{R} = \frac{\exp(\text{LeakyReLU}(\vec{b}[W_e e_{rj} || W_h h_r])}{\sum_{k \in \mathcal{A}_j} \exp(\text{LeakyReLU}(\vec{b}[W_e e_{kj} || W_h h_k]))} \tag{4.9}$$

Once these edge attention coefficients are calculated, the edge features are updated as follows:

$$e_{ij}^{(t+1)} = \sum_{l \in \mathcal{A}_i} \beta_{il}^{L} W_e e_{il}^{(t)} + \sum_{r \in \mathcal{A}_j} \beta_{rj}^{R} W_e e_{rj}^{(t)}, \tag{4.10}$$

This equation represents an aggregate of the weighted edge features from the neighboring nodes of each edge, mirroring the process used for the node features. We denote here that we don't actually calculate the attention coefficient of all edges between each other. Our mechanism is rather focusing on the first edge neighbors on the left and the right sides. Moreover, self-attention is implicit in the update rule.

Through iterative updates of the learnable parameters $\vec{a}$ and $\vec{b}$, along with the transformation matrices W_h and W_e during training, the network dynamically adjusts the relative importance of different nodes and edges, effectively amplifying its learning capability from the underlying graph structure.

C.3) VGAE Architecture:

In the context of the VGAE, the process begins by establishing two bottlenecks for the embedding dimensions of nodes and edges, denoted as $d_{\mathcal{V}}$ and $d_{\mathcal{E}}$ respectively, each using two parameters. The encoder, represented by $\mathbf{E}_{\lambda}^{g}$, operates through multiple LRGAT convolution layers, culminating in two distinct LRGAT layers: LRGAT_{μ} and LRGAT_{σ}. These layers generate mean and log variance vectors for both nodes and edges. Specifically, the mean vectors $\mu_{\mathcal{V}}$ and $\mu_{\mathcal{E}}$ are produced by $\text{LRGAT}_{\mu}(\mathbf{E}_{\lambda}^{g}(\mathcal{G}))$, resulting in $\mu_{\mathcal{E}}$ and $\mu_{\mathcal{V}}$ being of dimensions $(N(N-1)/2, d_{\mathcal{E}})$ and $(N, d_{\mathcal{V}})$, respectively. Concurrently, log variance vectors $\log(\sigma_{\mathcal{V}}^2)$ and $\log(\sigma_{\mathcal{E}}^2)$ are derived from $\text{LRGAT}_{\sigma}(\mathbf{E}_{\lambda}^{g}(\mathcal{G}))$, with sizes of $(N, d_{\mathcal{V}})$ and $(N(N-1)/2, d_{\mathcal{E}})$, respectively. These vectors are instrumental in sampling the latent matrices $Z^{\mathcal{V}}$ and $Z^{\mathcal{E}}$, where each element is sampled from a normal distribution, $\mathcal{N}(0,1)$. The means and variances for this distribution are sourced from $\mu_{\mathcal{V}}$, $\sigma_{\mathcal{V}}$, $\mu_{\mathcal{E}}$, and $\sigma_{\mathcal{E}}$. The sampling process is encapsulated in the following equations:

$$\begin{aligned} Z_{ij}^{\mathcal{V}} &\sim \mathcal{N}\left((\mu_{\mathcal{V}})_{ij}, (\sigma_{\mathcal{V}})_{ij}^2\right) \rightarrow Z_{ij}^{\mathcal{V}} = (\mu_{\mathcal{V}})_{ij} + (\sigma_{\mathcal{V}})_{ij} \odot \epsilon \\ Z_{ij}^{\mathcal{E}} &\sim \mathcal{N}\left((\mu_{\mathcal{E}})_{ij}, (\sigma_{\mathcal{E}})_{ij}^2\right) \rightarrow Z_{ij}^{\mathcal{E}} = (\mu_{\mathcal{E}})_{ij} + (\sigma_{\mathcal{E}})_{ij} \odot \epsilon \end{aligned} \tag{4.11}$$

In traditional VGAE, the decoding process typically involves using the inner product to reconstruct an adjacency matrix. This approach is effective for many graph structures, but it encounters limitations when dealing with interference graphs. Interference graphs are often complete graphs, meaning each node is connected to all the other nodes. In such cases, the adjacency matrix generated by the inner-product decoder becomes less informative because it doesn't provide distinctive insights into the unique connections or interactions between nodes, as the connections are uniform across the graph. To address this limitation and extract more meaningful information from interference graphs, instead of relying solely on the reconstructed adjacency matrix, we focus on constructing and utilizing node and edge features. By doing so, we can capture more nuanced and specific characteristics of the graph, such as the nature and strength of connections between nodes, as well as the attributes of the nodes themselves. This shift from a purely structural interpretation of the graph to a feature-based analysis allows for a richer and more insightful understanding of the interference graph, leading to more effective learning and

inference. Thus, upon obtaining $Z^{\mathcal{V}}$ and $Z^{\mathcal{E}}$, we calculate the reconstructed edges and nodes feature matrices E and H by decoding $Z^{\mathcal{V}}$ and $Z^{\mathcal{E}}$ using the 2 MLP decoders, $\hat{E} = \mathbf{D}_{\alpha}^{\mathcal{E}}(Z^{\mathcal{V}})$ and $\hat{H} = \mathbf{D}_{\vartheta}^{\mathcal{V}}(Z^{\mathcal{V}})$, respectively. Similar to the VSAE, this output enables us to compute the reconstruction and KL divergence losses, leading to the subsequent update of the model parameters λ, ϑ, and α.

After learning the edges and node embedding matrices, we address the challenge of aligning their dimensions, denoted as $Z^{\mathcal{V}}$ and $Z^{\mathcal{E}}$, which are initially of sizes $(N, d_{\mathcal{V}})$ and $(N(N-1)/2, d_{\mathcal{E}})$, respectively. To ensure dimensional consistency, we employ adaptive pooling to transform the edge embedding matrix $Z^{\mathcal{E}}$ into the desired size of $(N, d_{\mathcal{E}})$. This transformation is achieved through a systematic process: First, we partition $Z^{\mathcal{E}}$ into N groups, each representing the set of $N-1$ edge embeddings emanating from a specific node. Subsequently, we apply an average pooling operation within each group, calculating the mean of the $N-1$ edge embeddings. This step is essential for aggregating the comprehensive edge information associated with each node into a singular, representative embedding. Formally, for each node i, this aggregation is expressed as $\hat{Z}_i^{\mathcal{E}} = \frac{1}{N-1} \sum_{z \in Z_i^{\mathcal{E}}} z$, where $\hat{Z}_i^{\mathcal{E}}$ denotes the pooled embedding for node i. The final act involves concatenating these pooled embeddings to construct the revised edge embedding matrix $\hat{Z}^{\mathcal{E}}$ of size $(N, d_{\mathcal{E}})$. By executing this method, we effectively condense the rich edge information from each node, ensuring that both the node and edge embeddings are dimensionally aligned and suitable for the subsequent steps in our analytical process.

4.3.4 Output Component

In the output component of our model, we effectively integrate both geometric and spatial embeddings to compute a global embedding Z. The size of Z is $(N, d_{\mathcal{V}} + d_{\mathcal{E}} + d_s)$, and it is calculated through the concatenation of the spatial embeddings Z^S and geometric embeddings $Z^{\mathcal{V}}$ and $Z^{\mathcal{E}}$, as shown in the equation:

$$Z = \text{MLP}_R(\hat{Z}^S \parallel Z^{\mathcal{V}} \parallel \hat{Z}^{\mathcal{E}}), \tag{4.12}$$

Where MLP_R represents an MLP block that processes the concatenated embeddings. After that, Z is utilized in two distinct MLP blocks. To compute the first output P, we first pass the output of $\text{MLP}_P(Z)$, through the sigmoid function $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$. This step ensures P falls between 0 and 1. We then scale the result by p_{max}, as shown in the equation:

$$P = \text{sigmoid}(\text{MLP}_P(Z)) \cdot p_{max}. \tag{4.13}$$

The computation of Ψ unfolds in two main steps. Initially, the softmax function, $\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$, is applied to the outputs of $\text{MLP}_\Psi(Z)$ to obtain the channel selection probability for each link, denoted as $\hat{\Psi}$:

$$\hat{\Psi} = \text{softmax}(\text{MLP}_\Psi(Z)). \tag{4.14}$$

During the inference phase, the decision Ψ for each link i over channel k is made by selecting the channel with the highest probability from the vector $\hat{\Psi}_{ik}$. Specifically, Ψ_i^k is assigned a value of 1 if channel k has the maximum probability for link i, and 0 otherwise, ensuring adherence to channel allocation constraints. This decision process is encapsulated as:

$$\Psi_i^k = \begin{cases} 1 & \text{if } k = \text{argmax}_k(\hat{\Psi}_{ik}) \\ 0 & \text{otherwise} \end{cases} \tag{4.15}$$

In summary, the output component of the model adeptly combines and processes spatial and geometric data, resulting in optimized variables P and Ψ through distinct neural network structures and computational techniques suitable for both the training and inference phases. Since the trainable parameters of all the neural network blocks in the output component are going to be updated at the same time using the gradient of a loss function, we refer to them as ω. It is worth noting that while this solution is tailored to our specific case, it is possible to adapt it to different problems where the use of wireless network embeddings is useful.

4.4 Training Process

4.4.1 Loss functions

Generally, in VAE frameworks, the goal is to learn an effective latent representation of the input data that can be used for various tasks, including data generation and reconstruction. The VAE loss, denoted by $\mathcal{L}_{\text{VAE}}$, is a combination of two essential components: the reconstruction loss, $\mathcal{L}_{\text{REC}}$, and the regularization loss, $\mathcal{L}_{\text{KL}}$.

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{REC}} + \mathcal{L}_{\text{KL}} \tag{4.16}$$

The reconstruction loss measures the dissimilarity between the input data and their corresponding reconstructions produced by the VAE. On the other hand, the regularization loss, aims to encourage the latent space to follow a standard normal distribution. This is achieved by comparing the approximate posterior distribution, represented by a mean and a variance, with a standard normal distribution $\mathcal{N}(0, 1)$. The KL divergence term penalizes the deviation from the standard normal distribution and encourages the model to learn a continuous and disentangled representation. By combining these two components, as in Equation (16), the VAE is trained to minimize the reconstruction error while simultaneously learning a meaningful and structured latent space. The balance between these terms plays a crucial role in the success of the VAE, as it determines the trade-off between reconstruction fidelity and latent space regularization.

For the VSAE, the reconstruction loss is quantified using the normalized mean square error (MSE). This calculation is performed element-wise between the input and the reconstructed CSI, given by the formula:

$$\mathcal{L}^{s}_{\text{REC}} = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{(G^{k}_{ij} - \hat{G}^{k}_{ij})^2}{KN^2} \tag{4.17}$$

Additionally, the KL divergence loss is computed as follows:

$$\mathcal{L}_{\mathrm{KL}}^{s} = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{d_S}\left(1 + \log((\sigma_s)_{ij}^2) - (\mu_s)_{ij}^2 - (\sigma_s)_{ij}^2\right) \tag{4.18}$$

The comprehensive training loss for the VSAE then becomes:

$$\mathcal{L}_{\mathrm{VSAE}} = \mathcal{L}_{\mathrm{REC}}^{s} + \mathcal{L}_{\mathrm{KL}}^{s} \tag{4.19}$$

Similarly, the reconstruction loss for the VGAE adopts the normalized MSE, calculated as:

$$\mathcal{L}_{\mathrm{REC}}^{g} = \sum_{k=1}^{K}\sum_{i=1}^{N}\frac{(H_{ik} - \hat{H}_{ik})^2}{KN} + \sum_{k=1}^{K}\sum_{i=1}^{\frac{N(N-1)}{2}}\frac{2(E_{ik} - \hat{E}_{ik})^2}{KN(N-1)} \tag{4.20}$$

Similar to the VSAE, the KL divergence loss for the VGAE takes the form:

$$\begin{aligned}\mathcal{L}_{\mathrm{KL}}^{g} = -\frac{1}{2}\Bigg[&\sum_{i=1}^{N}\sum_{j=1}^{d_{\mathcal{V}}}\left(1 + \log((\sigma_{\mathcal{V}})_{ij}^2) - (\mu_{\mathcal{V}})_{ij}^2 - (\sigma_{\mathcal{V}})_{ij}^2\right) \\ &+ \sum_{k=1}^{\frac{N(N-1)}{2}}\sum_{l=1}^{d_{\mathcal{E}}}\left(1 + \log((\sigma_{\mathcal{E}})_{kl}^2) - (\mu_{\mathcal{E}})_{kl}^2 - (\sigma_{\mathcal{E}})_{kl}^2\right)\Bigg]\end{aligned} \tag{4.21}$$

Leading to the following complete training loss for the VGAE:

$$\mathcal{L}_{\mathrm{VGAE}} = \mathcal{L}_{\mathrm{REC}}^{g} + \mathcal{L}_{\mathrm{KL}}^{g} \tag{4.22}$$

Incorporating further elements, the mean rate loss is introduced, functioning as the negation of the objective function. This loss is formulated as:

$$\mathcal{L}_{\mathrm{Rate}} = -\frac{1}{N}\sum_{i=1}^{N} R_i(\psi_i, p_i) \tag{4.23}$$

Moreover, to guide training towards adhering to minimum QoS threshold constraints for each link, another loss is introduced. The regularization loss function, denoted as $\mathcal{L}_{\mathrm{REG}}$, is formulated

as follows:

$$\mathcal{L}_{\text{REG}} = \sum_{i=0}^{N} \max(0, R_i(\psi_i, p_i) - R_{\min}) - \min(\{R_i, i \in \mathcal{D}\}) \tag{4.24}$$

This loss function operates by imposing a penalty on the model's learning process whenever there is a QoS violation, specifically if the QoS threshold is not met. Additionally, the function emphasizes enhancing the minimum rate within the network. This minimum rate is subject to change in each iteration, leading to a consistent focus on increasing it to optimize network performance. After stating the loss functions, the next subsection explains the training and inference algorithms.

4.4.2 Learning and Training Algorithms

The presented algorithms offer a comprehensive framework for the embedding training and inference stages.

Algorithm 1, depicted below, outlines the procedure for training the VSAE and VGAE. Both models are jointly trained iteratively until the loss ceases to exhibit significant changes. Starting with the random generation of input graphs and CSIs with various simulation parameters, such as the distance between transmitters and receivers and shadowing parameters. More importantly, we can also train on different network sizes, i.e., the number of links N, since the VGAE and VSAE architectures are tailored to be input size invariant. Thus, enabling greater generalization performance. Subsequently, latent parameters μ_s, σ_s, $\mu_{\mathcal{V}}$, $\sigma_{\mathcal{V}}$, $\mu_{\mathcal{E}}$, and $\sigma_{\mathcal{E}}$ are calculated using their corresponding encoders. This lays the groundwork for the generation of latent variable samples, Z^S, $Z^{\mathcal{V}}$, and $Z^{\mathcal{E}}$. These latent variables then play a pivotal role in calculating the reconstructed $\hat{G}$, $\hat{H}$, and $\hat{E}$ using their respective decoders. The losses $\mathcal{L}_{\text{VSAE}}$ and $\mathcal{L}_{\text{VGAE}}$ are computed and serve as indicators of the model's performance since it is self-supervised. The optimization process involves updating the model's parameters θ, ϕ, λ, α, and ϑ through stochastic gradient descent, and we make use of the reparameterization trick Kingma & Welling (2022) for stable convergence as well.

Algorithm 4.1 Pre-training VSAE and VGAE in parallel

Input: Randomly generated CSI tensor G and corresponding interference graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, H, E)$
Output: Pre-trained parameters $\theta^*, \phi^*, \lambda^*, \alpha^*, \vartheta^*$

while *not converged* **do**
  Randomly generate G and $\mathcal{G}(\mathcal{V}, \mathcal{E}, H, E)$;
  Compute (μ_s, σ_s) and $(\mu_{\mathcal{V}}, \sigma_{\mathcal{V}})$, $(\mu_{\mathcal{E}}, \sigma_{\mathcal{E}})$;
  Sample Z^S, $Z^{\mathcal{V}}$, $Z^{\mathcal{E}}$;
  Reconstruct $\hat{G}$, $\hat{H}$, $\hat{E}$;
  Compute $\mathcal{L}_{\text{VSAE}}$ and $\mathcal{L}_{\text{VGAE}}$;
  Update (θ, ϕ) using $\nabla_{\theta/\phi}\mathcal{L}_{\text{VSAE}}$;
  Update $(\lambda, \alpha, \vartheta)$ using $\nabla_{\lambda/\alpha/\vartheta}\mathcal{L}_{\text{VGAE}}$;
end while
Save $\theta^*, \phi^*, \lambda^*, \alpha^*, \vartheta^*$;

Algorithm 4.2 Inference of P and Ψ

Input: CSI tensor G and pre-trained parameters $\theta^*, \phi^*, \lambda^*, \alpha^*, \vartheta^*$
Output: Optimized power vector P and RB allocation matrix Ψ

while *not converged* **do**
  Preprocess to build $\mathcal{G}(\mathcal{V}, \mathcal{E}, H, E)$;
  Compute (μ_s, σ_s), $(\mu_{\mathcal{V}}, \sigma_{\mathcal{V}})$, $(\mu_{\mathcal{E}}, \sigma_{\mathcal{E}})$;
  Sample $\hat{Z}^S$, $Z^{\mathcal{V}}$, $\hat{Z}^{\mathcal{E}}$, then Z;
  Compute $\hat{\Psi}$ and P;
  Compute $\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Rate}} + \mathcal{L}_{\text{REG}} + \mathcal{L}_{\text{VSAE}} + \mathcal{L}_{\text{VGAE}}$;
  Update $(\theta, \phi, \lambda, \alpha, \vartheta, \omega)$ using $\nabla_{\theta/\phi/\lambda/\alpha/\vartheta/\omega}\mathcal{L}_{\text{Total}}$;
end while
Compute final Ψ and P;

In the inference phase in algorithm 2, the process involves employing and refining the pre-trained VGAE and VSAE for latent variable computation. This is carried out using CSI data and a set of pre-trained parameters: θ^*, ϕ^*, λ^*, α^*, and ϑ^*, along with an initial ω. The methodology involves several steps similar to algorithm 1: initially, the CSI data G is transformed into a graph $\mathcal{G}$. Next, using the respective encoders, the means and variances are calculated. This is followed by the sampling of the spatial embedding and its transformation $\hat{Z}^S$ using the VSAE's

pre-trained encoder and sampling the nodes embeddings $Z^{\mathcal{V}}$ and edges embeddings and its transformation $\hat{Z}^{\mathcal{E}}$ using the VGAE's pre-trained encoders. After obtaining these embeddings, the global embedding vectors Z are computed. The process then involves calculating $\hat{\Psi}$ and P. Subsequently, losses are computed and all model parameters are updated accordingly for optimization. The final step in this iterative process, which aims for convergence within a few iterations, is the calculation of Ψ to ensure the preservation of the channel selection constraint.

4.5 Performance Test Results

In this section, we evaluate our proposed approach, including convergence analysis, scalability, and performance across different wireless setups and under an imperfect CSI.

4.5.1 Simulation Parameters

A 2D rectangular layout representing a wireless environment of 200 m in width and 100 m in height is created. Within this area, N transmitters are positioned randomly. The receivers are also randomly placed, maintaining a distance from their corresponding transmitters within the range of $[d_{\min}, d_{\max}]$. The channel model used is based on the ITU-1411 ITU-R (2009) short-range outdoor model, which factors in distance-dependent path loss. This model operates at a carrier frequency of 2.4 GHz, has an antenna height of 1.5 meters, and an antenna gain of 2.5 dBi. The maximum transmit power is set to 4 dBm, with a background noise level is -169 dBm/Hz. A normal distribution models the shadowing with a standard deviation σ_{sh} ranging from 4 dB to 12 dB, which is typical for urban outdoor environments. Concurrently, Rayleigh fading with zero mean and unit variance characterizes the fast-fading channel. The system comprises 5 channels ($K = 5$), each with a bandwidth of 500 Hz. In many wireless communication systems, the number of channels is fixed. Accordingly, our model is pre-trained with, and therefore dependent on, $K = 5$. However, our ablation study demonstrates that the model exhibits comparable performances without pre-training, indicating its ability to work effectively with different values of K.

Table 4.1 Summary of model variables categorized by VSAE, VGAE, and Projection components

Component	Variable (Value/Description)
VSAE	K: Channels available
	$d_S = 100$: Spatial dimensions
	E^s_θ: Conv layers ($K \rightarrow 16 \rightarrow 24 \rightarrow 32$)
	$\text{CONV}^s_{\mu/\sigma}$: Channels ($32 \rightarrow 1$)
	D^s_ϕ: Conv-transpose layers ($1 \rightarrow 32 \rightarrow 24 \rightarrow 16 \rightarrow K$)
	CONV_A: Channels ($1 \rightarrow d_S$)
VGAE	$d_{\mathcal{V}/\mathcal{E}} = 100$: Node/edge geometric dimensions
	$F, F' = 100$: Hidden dimensions
	$\mathbf{E}^g_\lambda$: LRGAT layers (3 layers)
	$\text{LRGAT}_{\mu/\sigma}$: LRGAT layers
	$\mathbf{D}^{\mathcal{E}/\mathcal{V}}_{\alpha/\vartheta}$: Fully connected layers (100×4, ends at K)
Projection	MLP_R: Fully connected layers (200×3)
	MLP_P: Fully connected layers (200×3, ends at 1)
	MLP_Ψ: Fully connected layers (200×3, ends at K)

Table I outlines the model's architecture parameters, which depend solely on the number of channels K and remain unaffected by the number of links N. The architecture leverages the ADAM optimizer with a uniform learning rate of 0.002 across all stages, as established by extensive testing.

4.5.2 Benchmark Schemes

We evaluate our model against four distinct methodologies, each with its unique approach to the problem. These include a genetic algorithm (GA), and techniques based on CNN, DNN, and GNN. The specifics of these approaches are as follows:

- **GA** Sun *et al.* (2021): Employs a Quantity-Weight Adaptive Salp Swarm Algorithm for D2D optimization.
- **CNN** Li *et al.* (2022): Uses a CNN architecture to emulate traditional optimization, processing CSI into power and channel outputs, requiring size-dependent supervised training.

- **DNN** Lee & Schober (2022): Applies two DNNs for power control and channel allocation, using brute force simulated data, with size-dependent supervised training.
- **GNN** Marwani & Kaddoum (2024b): Implements a GNN policy for channel and power management, iteratively maximizing network mean rates using a combination of supervised and unsupervised learning.

4.5.3 Convergence Analysis

C.1) Pre-training VSAE and VGAE: The training curves for the VSAE and VGAE components, presented in Figures 4 and 5 on a logarithmic scale, illustrate their learning performance on datasets comprising 30,000 samples from networks with sizes ranging from 20 to 50 links, with $d_{\max} = 50$m and $\sigma_{\text{sh}} = 4$dB. This size-invariant methodology enhances the models' versatility by enabling them to adapt to various network structures. However, it also introduces additional complexity due to the diverse nature of these network structures. Despite this, both VSAE and VGAE adeptly minimize their losses to an almost negligible 10^{-8}, indicating precise data reconstruction and regularization. The VGAE's edge losses are slightly higher, about 10^{-4}, primarily because edge losses involve more terms than node losses, leading to naturally higher aggregate loss values.

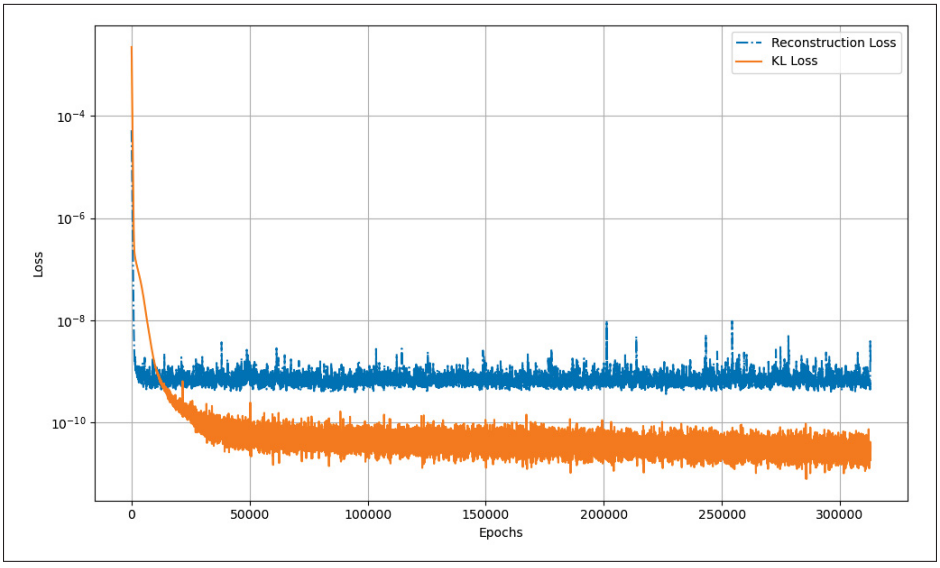


Figure 4.4 VSAE training analysis over epochs

The VSAE achieves faster convergence compared to the VGAE, likely due to its convolutional operations that process information within a fixed neighborhood, enhancing stability. The VGAE, with its comprehensive LRGAT-based learning and attention mechanisms, captures more complex patterns but requires longer to converge. Nonetheless, the VGAE's in-depth learning approach and graph data's data augmentation capabilities enable it to reach loss levels comparable to that of the VSAE, underscoring both models' efficacy in complex graph-structured data environments.

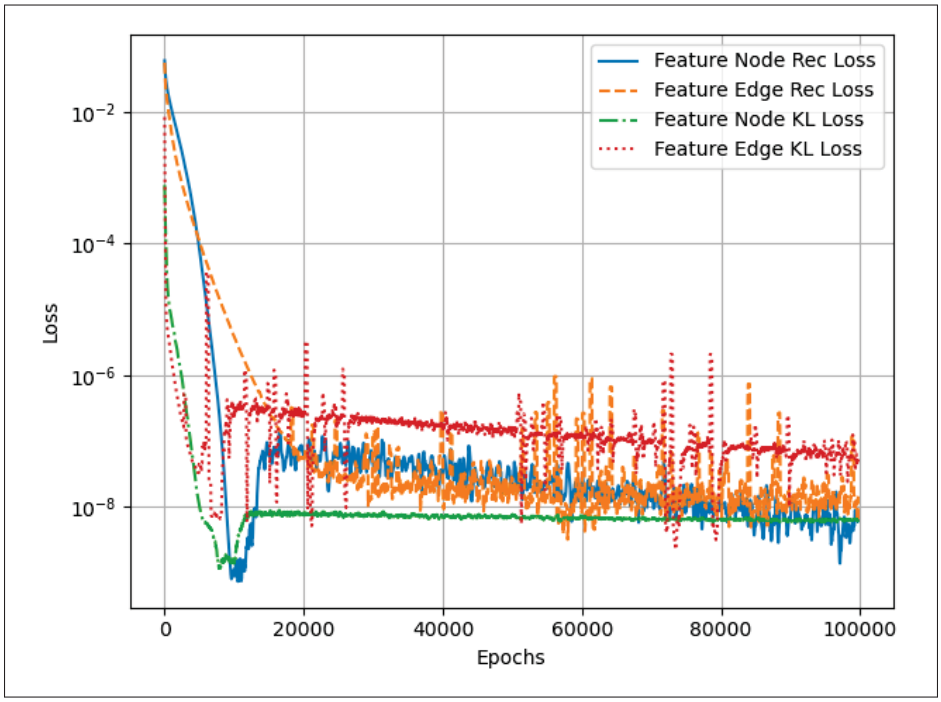
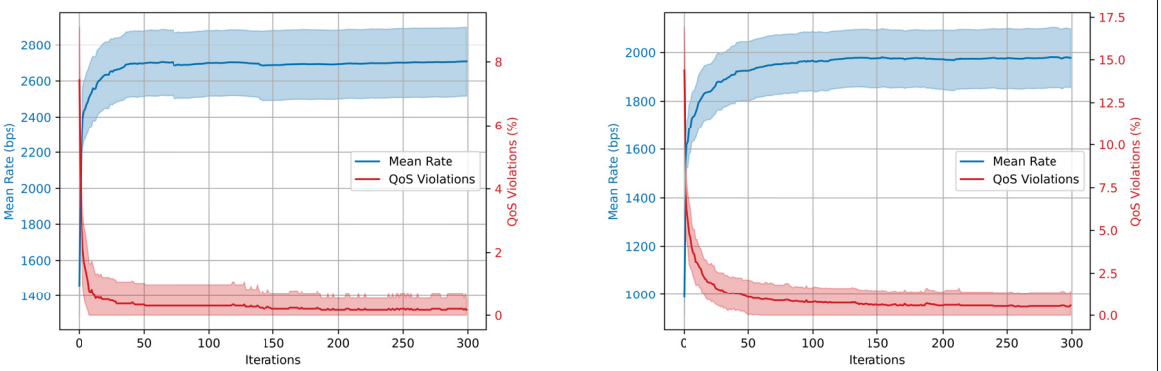


Figure 4.5 VGAE training analysis over epochs

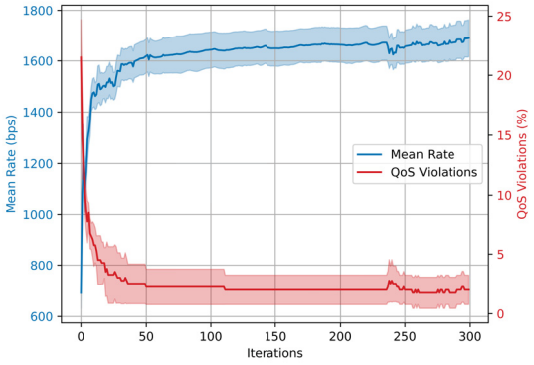
C.2) Inference: Figure 5 displays results from 10000 simulation scenarios, illustrating our approach's performance with varying numbers of users, denoted by N. The shaded area around the mean lines indicates the standard deviation, demonstrating the variability in the network's performance across simulations. We observe the same trend over the different values of N, where each graph shows a convergence and a stabilization of the network performance toward lower QoS violations and higher data rates over 300 iterations. The QoS violations initially spike for higher values of N, attributable to the limited radio resources being spread thin across

more users, but ultimately trend downward, indicating that the network adapts to the allocated resources efficiently, even under increased load.



a) $N = 20$.

b) $N = 30$.



c) $N = 40$.

Figure 4.6 Network mean rate and QoS violation over iterations for different N, with $\gamma_{\min} = 300$ bps

4.5.4 Ablation Study

In this test, we compare the performance (mean rate and QoS violation) of different versions of our model to evaluate the importance of each component. We generate 10,000 samples of CSI with constant wireless parameters: $N = 30$, $d_{max} = 50$, $\sigma_{sh} = 4$ dB, and $R_{min} = 500$ bps. The test involves different LRGAT, CNN, and MLP layer configurations, including configurations

with only VGAE and VSAE, a version without pre-training the VGAE and VSAE, and a version without sampling. Figure 7 clearly illustrates the superiority of approaches that incorporate LRGAT, CNN, and sampling across both metrics, validating the importance of these components. Additionally, both VGAE and VSAE achieve comparable performances, indicating that each component effectively learns significant information from the CSI. The models without sampling exhibit reduced performance, likely because they learn point embeddings rather than a more general distribution, which limits their effectiveness. Furthermore, the results without pre-training demonstrate that while pre-training the VGAE and VSAE helps the network become more familiar with the nature of the CSI, it is not significantly necessary. This implies that the model can perform well across any number of RBs even without pre-training. Overall, these findings confirm the critical role of LRGAT, CNN, and sampling in enhancing model performance while also highlighting the model's adaptability and robustness, even without pre-training.

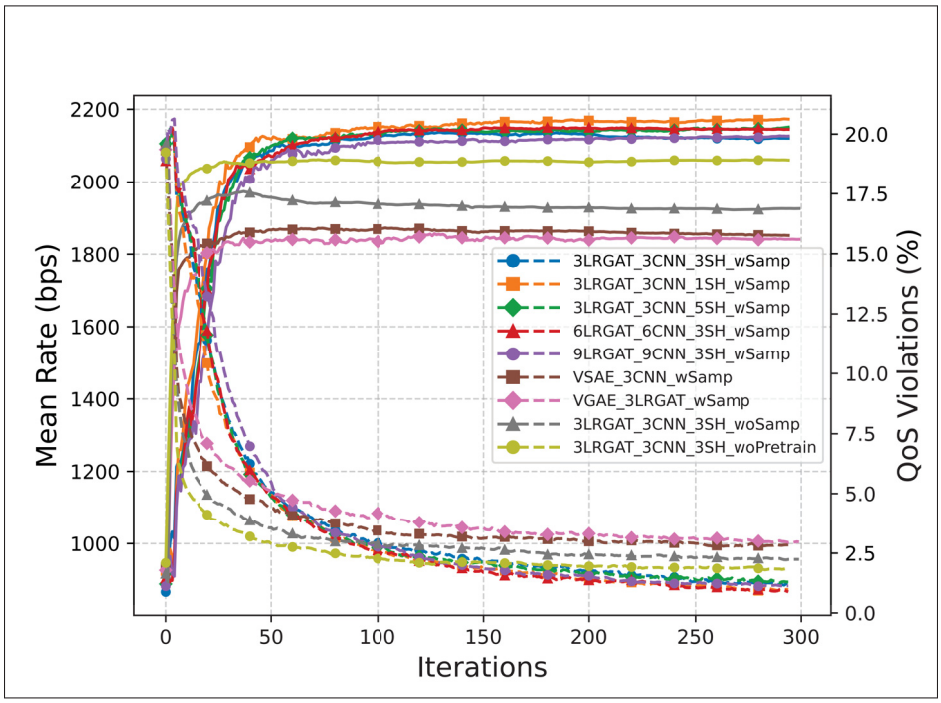


Figure 4.7 Mean rate and QoS violation comparison for different versions of the proposed model

To test the permutation equivariance of our model, we generate a CSI tensor ($N = 30$) and randomly permute the order of links and RBs. As well as P and Ψ in a similar manner. Formally, we want to test that, given multiple permutations π_1 (for indices $\{1, 2, \ldots, K\}$) and π_2 (for indices $\{1, 2, \ldots, N\}$), permuting the CSI tensor as $\mathbf{G}(\pi_1(K), \pi_2(N), \pi_2(N))$ causes a similar permutation to the solution, i.e., $P(\pi_1(K))$ and $\Psi(\pi_1(K), \pi_2(N))$. We generate 20 distinct permutations and compute the pairwise cosine similarity matrix between the results of different permutations as follows. For each permutation, we obtain the vectors P and Ψ. These vectors are then permuted according to π_1 and π_2. The pairwise cosine similarity matrix $\mathbf{S}$ for P is computed as: $\mathbf{S}_{i,j} = \frac{\mathbf{P}_i \cdot \mathbf{P}_j}{\|\mathbf{P}_i\| \|\mathbf{P}_j\|}$, where $\mathbf{P}_i$ and $\mathbf{P}_j$ are the permuted P vectors for the i-th and j-th permutations, respectively. Similarly, the pairwise cosine similarity matrix $\mathbf{S}$ for Ψ is computed as: $\mathbf{S}_{i,j} = \frac{\mathbf{\Psi}_i \cdot \mathbf{\Psi}_j}{\|\mathbf{\Psi}_i\| \|\mathbf{\Psi}_j\|}$, where $\mathbf{\Psi}_i$ and $\mathbf{\Psi}_j$ are the permuted Ψ vectors for the i-th and j-th permutations, respectively. By analyzing the cosine similarity between different permutations, we can assess the model's permutation equivariance, ensuring that the performance remains consistent regardless of the order of the D2D links or the RBs. Finally, we visualize the cosine similarity matrices in Figure 8 for P and Ψ to illustrate the model's equivariance properties. The heat maps of these matrices provide a clear depiction of the similarity (above 0.8) between different permutations.

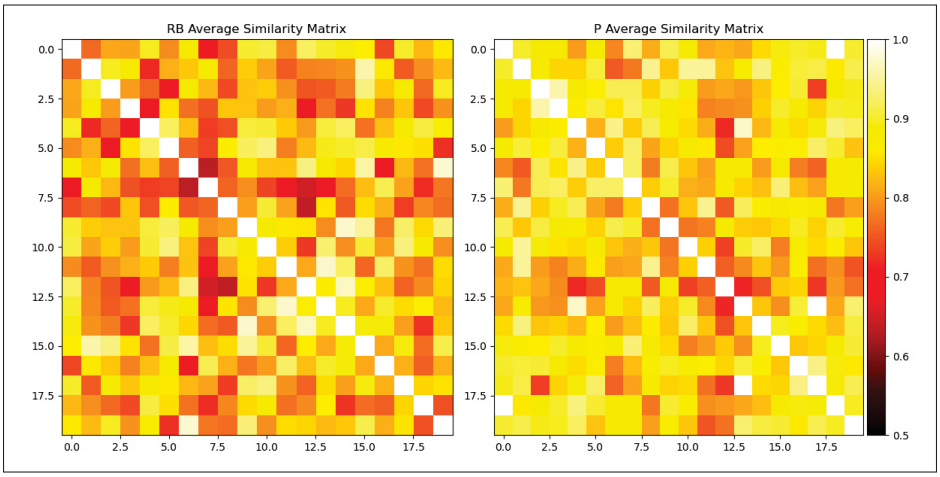
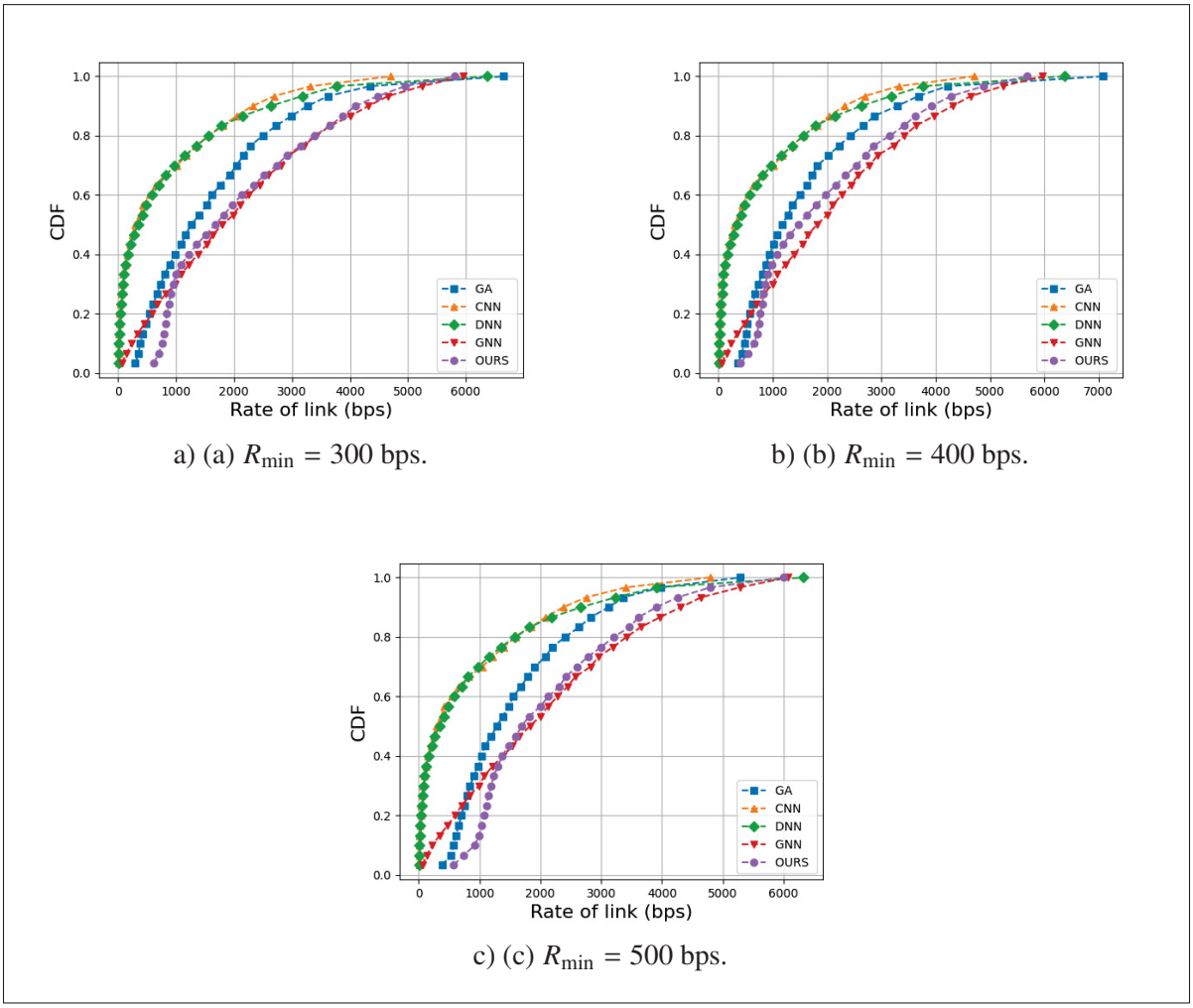


Figure 4.8 Average similarity matrix of P and Ψ for different permutations of the CSI

4.5.5 Impact of QoS Constraints on Performance



a) (a) $R_{\min} = 300$ bps.

b) (b) $R_{\min} = 400$ bps.

c) (c) $R_{\min} = 500$ bps.

Figure 4.9 Rate CDF of benchmarking schemes for different $R_{\min}$ values

This subsection compares the proposed model's performance to the established benchmark schemes considering different QoS values. Here, the CSI remains the same while R_{min} is changed for a fair comparison. The Cumulative Distribution Function (CDF) plots, presented in Figure 9, provide a comparative analysis of the link rate performances of four benchmark schemes for $R_{min} = \{300, 400, 500\}$ bps. The three CDF plots indicate that our model outperforms the others for all the threshold values, achieving higher link rates with greater probability, followed by the GNN scheme. The performance of the GA, CNN, and DNN are similar and lower than that of the GNN and proposed scheme. As the minimum rate threshold increases, all schemes

tend to achieve higher link rates, with our model consistently leading, indicating its robustness and potential for applications requiring reliable minimum data rates.

Table 4.2 Comparative analysis for different $R_{\min}$ values ($N = 30$, $d_{\max} = 50$ m, $\sigma_{\text{sh}} = 8$ dB)

$R_{\min}$	300 bps		400 bps	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1847 ± 32.6	2.18 ± 0.08	1711 ± 26.2	1.59 ± 0.06
CNN	869 ± 19.9	57.77 ± 1.14	966 ± 28.6	52.68 ± 0.93
DNN	911 ± 20.5	50.22 ± 0.91	961 ± 24.4	58.47 ± 1.06
GNN	2210 ± 20.2	6.38 ± 0.42	2054 ± 18.7	8.69 ± 0.53
Our model	**2156 ± 41.4**	**1.15 ± 0.07**	**2323 ± 18.1**	**1.91 ± 0.11**

$R_{\min}$	500 bps		600 bps	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1597 ± 30.8	3.32 ± 0.12	1665 ± 53.4	5.55 ± 0.41
CNN	853 ± 18.4	58.24 ± 0.59	917 ± 14.0	62.55 ± 0.57
DNN	933 ± 22.1	59.84 ± 1.32	962 ± 26.6	64.94 ± 1.06
GNN	1988 ± 16.9	13.60 ± 0.63	2041 ± 30.7	16.72 ± 0.80
Our model	**2210 ± 11.4**	**2.38 ± 0.13**	**2231 ± 26.4**	**2.89 ± 0.13**

Next, we compare benchmark algorithms across varying minimum rate thresholds, demonstrating that the proposed framework exhibits a higher performance in both mean rate and QoS breach metrics. Table II shows the results when assessed at R_{min} thresholds of 300, 400, 500, and 600 bps. Our model consistently reports the highest mean rates, starting at 2156 ± 41.4 bps and demonstrating a slight decline to 2231 ± 26.4 bps as the threshold increases. This represents a minimal reduction in the mean rate from the lowest to the highest threshold, underscoring the algorithm's robust throughput capabilities. In terms of service reliability, our model maintains a QoS breach percentage well below the considered benchmarks, increasing only marginally from 1.15 ± 0.07% to 2.89 ± 0.13% as the threshold, R_{min}, increases. This consistency in QoS breach percentages, even with high rate requirements, highlights the reliability of our model. In contrast, the CNN and DNN approaches show significantly higher QoS breach percentages, exceeding 50%, across all thresholds, which raises concerns about their suitability for high-reliability

applications. The GA scheme, despite exhibiting the lowest QoS breaches at the lowest threshold, shows diminished reliability with increasing R_{min} values. Meanwhile, the GNN maintains moderate performance metrics yet fails to achieve our model's performance.

4.5.6 Impact of the Number of Links

Table 4.3 Comparative analysis for different N values
($R_{\min}$ = 300 bps, $d_{\max}$ = 50 m, σ_{sh} = 8 dB)

N	50		100	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1367 ± 28.5	1.79 ± 0.06	540 ± 12.0	2.04 ± 0.02
CNN	892 ± 37.5	32.13 ± 1.72	439 ± 22.3	46.92 ± 0.78
DNN	938 ± 39.4	35.90 ± 1.87	469 ± 23.2	51.31 ± 0.92
GNN	1203 ± 24.5	1.33 ± 0.05	479 ± 10.3	3.65 ± 0.11
Our model	**1438 ± 21.5**	**0.77 ± 0.09**	**682 ± 17.4**	**2.72 ± 0.18**

N	200		300	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	291 ± 8.9	26.52 ± 0.13	230 ± 2.6	32.64 ± 0.26
CNN	238 ± 12.8	58.65 ± 1.20	202 ± 9.7	61.20 ± 1.94
DNN	255 ± 13.5	63.34 ± 1.34	214 ± 10.3	66.86 ± 2.09
GNN	259 ± 7.7	27.76 ± 0.10	204 ± 2.3	38.97 ± 0.23
Our model	**373 ± 14.1**	**21.04 ± 0.30**	**255 ± 4.3**	**29.07 ± 0.16**

In this section, we assess the scalability of our model. Table III provides a succinct comparison of five algorithms over increasing network sizes (N ranging from 50 to 300) at a constant R_{min} of 300 bps. The analysis reveals that our model exhibits superior scalability. At $N = 50$, our model maintains a high mean rate of 1438 ± 21.5 bps with minimal QoS breaches of $0.77 \pm 0.09\%$. As N increases to 300, the mean rate slightly decreases to 255 ± 4.3 bps, with QoS breaches rising to $29.07 \pm 0.16\%$. In contrast, the GA starts with a mean rate of 1367 ± 28.5 bps and QoS breaches of $1.79 \pm 0.06\%$ at $N = 50$, but its performance significantly deteriorates as N increases. At $N = 300$, the mean rate drops to 230 ± 2.6 bps, and QoS breaches increase to $32.64 \pm 0.26\%$. The CNN and DNN algorithms show poor scalability. For CNN, the mean rate decreases

from 892 ± 37.5 bps at $N = 50$ to 202 ± 9.7 bps at $N = 300$, with QoS breaches escalating from 32.13 ± 1.72% to 61.20 ± 1.94%. Similarly, DNN's mean rate falls from 938 ± 39.4 bps to 214 ± 10.3 bps, and QoS breaches increase from 35.90 ± 1.87% to 66.86 ± 2.09%. The GNN shows moderate performance, starting with a mean rate of 1203 ± 24.5 bps and QoS breaches of 1.33 ± 0.05% at $N = 50$, but also experiences a decline, reaching 204 ± 2.3 bps and 38.97 ± 0.23% QoS breaches at $N = 300$. These findings underscore our proposed scheme as the most robust algorithm in the face of increasing network sizes, maintaining high mean rates while minimizing QoS breaches.

Execution time (ms)						
N	20	30	40	50	60	70
GA	24.21	40.17	67.89	105.04	167.07	184.98
CNN	1.07	1.15	1.15	1.19	1.23	1.27
DNN	1.02	1.02	1.02	1.02	1.03	1.04
GNN	43.29	45.35	50.13	56.22	63.19	74.29
Our model	**17.84**	**19.34**	**21.05**	**23.11**	**24.94**	**27.22**

Table 4.4 Execution time comparisons with different N

Next, we focus on the execution time in milliseconds, as detailed in Table IV, using an Intel® Xeon® W-1270 CPU for our hardware setup. It's important to note that our evaluation considers only the online inference time since the training phase can be precomputed offline. Straightforward functional approaches like DNN and CNN do not require repeated optimization iterations, resulting in notably shorter execution times. In contrast, iterative models like the GA and GNN, necessitate more time to reach convergence, which we define as the point where performance stabilizes across 20 iterations. The data clearly illustrates that our model achieves significantly faster convergence compared to the GA and the GNN approaches. This suggests that, with better hardware setup, i.e., GPUs and RAM, our model could deliver efficient performance in fast-fading environments where the coherence time is relatively low.

Table 4.5 Comparative analysis for different $d_{\max}$ values
(N = 30, $R_{\min}$ = 300 bps, σ_{sh} = 8 dB)

$d_{\max}$	50 m		60 m	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1849 ± 24.0	0.00 ± 0.00	1593 ± 26.5	0.67 ± 0.15
CNN	899 ± 26.6	46.79 ± 1.01	846 ± 22.0	52.52 ± 0.96
DNN	917 ± 31.4	46.46 ± 0.91	883 ± 18.2	52.52 ± 0.51
GNN	2032 ± 34.7	5.39 ± 0.27	1810 ± 30.6	4.37 ± 0.18
Our model	**2030 ± 31.3**	**1.01 ± 0.09**	**1872 ± 32.3**	**2.02 ± 0.25**

$d_{\max}$	70 m		80 m	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1706 ± 35.7	1.33 ± 0.35	1250 ± 36.5	1.67 ± 0.35
CNN	793 ± 19.7	52.19 ± 1.26	630 ± 11.2	59.59 ± 0.97
DNN	884 ± 42.0	53.87 ± 1.11	776 ± 26.0	60.93 ± 1.03
GNN	1600 ± 39.6	8.50 ± 0.66	1419 ± 32.2	9.09 ± 0.73
Our model	**1784 ± 14.9**	**3.03 ± 0.31**	**1748 ± 33.9**	**4.37 ± 0.29**

4.5.7 Impact of the Links Locations

Table V provides a comparative analysis across different maximum distances (50 m to 80 m) with N = 30, R_{min} = 300 bps, and σ_{sh} = 8 dB, evaluating our model against GA, CNN, DNN, and GNN. Our model consistently matches or exceeds the mean rate of other top-performing models while maintaining a significantly lower QoS breach percentage. At 50 m, our model exhibits only a 1.01% QoS breach, which is substantially lower than the CNN and DNN models, both exceeding 46%. This low QoS breach rate is maintained even as the distances increase, demonstrating our model's robustness in challenging conditions. This analysis highlights our model's efficacy and reliability over varying distances, particularly in maintaining QoS constraints, thereby outperforming conventional methods.

Table 4.6 Comparative analysis for different σ_{sh} values ($N = 30$, $R_{\min} = 300$ bps, $d_{\max} = 50$ m)

σ_{sh}	4 dB		7 dB	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1192 ± 23.5	1.02 ± 0.32	1407 ± 48.2	2.38 ± 0.45
CNN	656 ± 24.3	53.72 ± 1.73	784 ± 29.6	53.38 ± 1.09
DNN	720 ± 32.8	57.80 ± 0.99	891 ± 24.3	53.38 ± 0.95
GNN	1086 ± 37.2	4.08 ± 0.50	1390 ± 28.8	4.08 ± 0.36
Our model	**1298 ± 32.0**	**0.68 ± 0.06**	**1637 ± 27.6**	**2.72 ± 0.37**

σ_{sh}	10 dB		12 dB	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1644 ± 63.4	2.04 ± 0.46	1423 ± 53.9	3.75 ± 0.68
CNN	760 ± 29.3	62.90 ± 0.98	659 ± 31.0	65.61 ± 1.30
DNN	731 ± 38.0	62.22 ± 1.39	882 ± 40.2	67.66 ± 0.72
GNN	1571 ± 65.3	5.78 ± 0.63	1515 ± 47.8	8.49 ± 1.12
Our model	**2146 ± 36.6**	**3.75 ± 0.24**	**2068 ± 34.1**	**4.42 ± 0.32**

4.5.8 Impact of the Shadowing Effects

Table VI offers a comparative analysis considering different shadowing standard deviation values, σ_{sh}, where $N = 30$, $R_{min} = 300$ bps, and $d_{max} = 50$ m. As the value of σ_{sh} increases from 4 dB to 12 dB, the mean rate and QoS breach of each scheme vary. Our model consistently delivers a superior mean rate across all σ_{sh} values, showcasing the best performance at 10 dB with a mean rate of 2146 ± 36.6 bps. Moreover, it maintains a low QoS breach percentage, notably at 4 dB with just 0.68 ± 0.06%, suggesting high reliability in maintaining QoS under different shadowing conditions. This test implies that our scheme not only provides higher data rates but also ensures a more consistent service quality, outperforming other benchmark models as the shadowing variation increases, which is critical for robust wireless communication systems.

Table 4.7 Comparative analysis for different γ levels
($N = 30$, $R_{\min} = 300$ bps, $d_{\max} = 50$ m, $\sigma_{\text{sh}} = 8$ dB)

γ	0.00		0.33	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	2088 ± 33.8	0.00 ± 0.00	2054 ± 36.0	7.44 ± 0.57
CNN	1013 ± 30.3	43.99 ± 1.06	1013 ± 30.3	43.99 ± 1.06
DNN	972 ± 20.9	49.05 ± 1.23	1032 ± 19.9	47.71 ± 0.84
GNN	2403 ± 24.9	9.82 ± 0.56	2361 ± 25.2	10.15 ± 0.48
Our model	**2197 ± 24.6**	**1.74 ± 0.15**	**2145 ± 42.5**	**5.75 ± 0.60**

γ	0.67		1.00	
Metrics	Mean rate	QoS breach	Mean rate	QoS breach
GA	1772 ± 25.6	16.24 ± 0.65	1776 ± 37.8	17.94 ± 0.93
CNN	1013 ± 30.3	43.99 ± 1.06	1013 ± 30.3	43.99 ± 1.06
DNN	1003 ± 41.1	50.41 ± 1.26	1101 ± 28.8	48.72 ± 0.85
GNN	2150 ± 25.8	16.57 ± 0.56	2037 ± 41.0	20.29 ± 0.97
Our model	**2042 ± 18.2**	**10.49 ± 0.95**	**1906 ± 24.9**	**16.58 ± 0.95**

4.5.9 Impact of Imperfect CSI

Our model evaluation in this section centers on its resilience against channel imperfections, specifically addressing the disparity between estimated and actual small-scale fading effects, modeled using a first-order Gauss-Markov process. Estimated small-scale fading values κ_{ij}^k are related to actual counterparts $\tilde{\kappa}_{ij}^k$ as $\tilde{\kappa}_{ij}^k = \sqrt{1-\gamma^2}\kappa_{ij}^k + \gamma n_{ij}^k$, where n_{ij}^k follows a complex Gaussian distribution, and γ is the error coefficient measuring CSI precision. Lower γ values indicate higher CSI estimation accuracy, improving as γ approaches zero. We evaluate the model robustness by initially using clean CSI and then introducing noise by varying γ. We compute the power and channel allocation solutions using this noisy CSI and calculate actual link rates using the clean CSI. Our model exhibits substantial resilience to varying channel distortion levels, maintaining high performance irrespective of distortion, as shown in Table VII. This comprehensive approach underscores the model's theoretical robustness and practical effectiveness in addressing typical wireless communication channel imperfections. Table VII

highlights the model's resilience against channel distortions, γ, ranging from 0.00 to 1.00, simulating the difference between estimated and actual small-scale fading effects. At no distortion (0.00), the model achieves a mean rate of 2197 ± 24.6 bps and a minimal QoS breach of only $1.74 \pm 0.15\%$. Even at a significant distortion level of 0.67, the model maintains effectiveness with a mean rate of 2042 ± 18.2 bps and a QoS breach of $10.49 \pm 0.95\%$. This performance remains superior compared to other models across varying distortion levels, indicating robust and reliable operation under different conditions of channel inaccuracies. This analysis highlights the model's consistent Mean Rate and effective QoS maintenance despite increasing distortion, emphasizing its robust capability to handle real-world wireless communication channel imperfections.

4.6 Conclusion

This study addresses power control and channel allocation in D2D wireless networks by proposing an unsupervised, scalable probabilistic framework that integrates GNNs, CNNs, and variational auto-encoders. Our contributions include the development of VSAE for capturing the CSI's spatial aspects in a size-invariant manner, a multi-channel graph-structured modeling of CSI to incorporate channel gains into node and edge features, and a VGAE equipped with a graph attention mechanism (LRGAT) for understanding the CSI's geometric distribution. Additionally, we introduce a task-specific projection mechanism that merges spatial and geometric embeddings for adequate power and channel allocation solutions. Rigorous evaluations across varied wireless setups demonstrate our method's superior performance in understanding network geometry, managing size variance, and preserving spatial correlations, thus outperforming existing methods in constraint preservation and network sum rate improvement, even with imperfect CSI. In future work, we will explore the permutation equivariance property of RBs, which is crucial in scenarios where the number of RBs can vary across different time slots or durations. This is particularly important in network slicing, where the network dynamically allocates a large number of RBs into different groups based on service requirements and demand. Addressing equivariance will enable our models to efficiently manage flexible and dynamic RB allocations. Additionally, we will extend our model to support partial CSI, using a combination of VGAE

and VSAE decoders to estimate the network's geometric and spatial features from partial CSI and reconstruct the complete CSI. We will also enhance our model to support varying numbers of RBs and develop a distributed version to address the overhead associated with full CSI knowledge. By sampling missing values from the VGAE and VSAE latent space, we aim to complete partial CSI. Furthermore, we will explore additional wireless scenarios, including IoT networks, Uplink/Downlink scenarios, and high mobility management.

CHAPTER 5

EVENT-BASED TEMPORAL GRAPH NEURAL NETWORK FOR RADIO RESOURCE MANAGEMENT

Maher Marwani[1] , Georges Kaddoum[1]

[1] Electrical Engineering Department, École de technologie supérieure,
1100 Notre-Dame Ouest, Montréal, Québec, Canada H3C 1K3

Article published in *IEEE Transactions on Wireless Communications*, Early Access, 2026,
doi: 10.1109/TWC.2026.3654880.

abstract

This paper addresses radio resource management (RRM) in highly dynamic device-to-device (D2D) networks. Existing heuristic and deep learning (DL) approaches often overlook the network's time-varying nature and struggle with variable link counts and channel conditions. We propose a Continuous-Time Dynamic Graph (CTDG) model that captures network events (activations, updates, and deactivations) in real time, rather than relying on discrete snapshots. Our Temporal Graph Neural Network (TGNN) processes these events, updating node-wise and graph-wise memories to track historical context. The resulting temporal embeddings drive power and channel allocation decisions that adapt to changing network topologies and mobility. We evaluate this TGNN-based solution in a realistic D2D setting using mobility traces from SUMO. Results show near-optimal throughput under stringent constraints and significant performance gains over conventional DL networks and memoryless GNN-based methods. This work underscores the importance of continuous-time graph modeling for scalable, efficient RRM in next-generation wireless systems.

5.1 Introduction

Radio Resource Management (RRM) is at the core of modern wireless systems, ensuring efficient allocation of spectrum and power resources to meet evolving quality-of-service (QoS)

demands. In many scenarios, particularly in device-to-device (D2D) communications where devices directly exchange data, the associated optimization problem becomes mixed-integer and non-convex. The complexity of the problem further increases when channel conditions are imperfect or time-varying, and the number of users fluctuates alongside their activities. Such dynamics, often observed in wireless sensor networks, vehicular networks, the Internet of Things (IoT), and dense cellular deployments, create significant interference challenges whenever multiple D2D links contend for the same bandwidth. While numerous power and channel allocation algorithms have been proposed, their computational complexity can become prohibitive, highlighting the need for more scalable, data-driven, and adaptive solutions.

We first review deterministic approaches and examine their computational bottlenecks, and then discuss deep learning (DL) strategies that can speed up inference but may struggle with limited training data, overfitting, and rigidity in input–output formats. Subsequently, we explore graph neural networks (GNNs), which leverage the inherent graph structure of wireless networks yet often fail to capture temporal correlations. We conclude with our research motivation, key contributions, and an outline of the paper's organization.

5.1.1 Optimization (Non-learnable) Solutions

Deterministic approaches to D2D resource allocation have been extensively explored. For instance, iterative matching combined with Stackelberg game theory was employed to dynamically manage channel and power allocation, thereby mitigating interference while boosting throughput Yuan *et al.* (2018). Similarly, the largest aggregated interference first (LIFA) algorithm demonstrated how spatial reuse could be maximized in orthogonal frequency division multiple access (OFDMA) systems while curtailing computational overhead Yang & Kuo (2017). Stochastic geometry-based methods enabled effective resource sharing between D2D links and cellular users, thereby improving overall coverage Abdallah *et al.* (2018). Other efforts focused on simultaneously optimizing power control and subcarrier assignment for both uplink and downlink Kai *et al.* (2018), or leveraged Gale–Shapley matching to reduce energy consumption Liu *et al.* (2019). In UAV-based networks, metaheuristic techniques, such as Particle Swarm

Optimization (PSO)–K-means and adaptive-mutation Salp Swarm methods, were introduced to balance communication quality with power efficiency Sun *et al.* (2021). In energy-harvesting scenarios, classical algorithms, such as the Kuhn–Munkres algorithm, were used for joint power and channel allocation Su & Zhu (2019). Additionally, hybrid methods combining power allocation with the Hungarian algorithm were applied to complex multi-hop D2D settings A *et al.* (2023). While each of these deterministic frameworks has its merits, they commonly rely on heuristics, face scalability issues, and grapple with non-convex optimization formulations, making them less suitable for rapidly evolving wireless environments.

5.1.2 Deep Learning (DL)-based Solutions

Deep learning (DL) made notable strides in the area of D2D resource allocation, primarily by approximating complex optimization functions through neural networks (NNs) Zheng *et al.* (2022). Early work explored branching-based methods to accelerate resource allocation, where NNs reduced inference overhead compared to exhaustive or iterative solvers Lee *et al.* (2020). Likewise, deep reinforcement learning (DRL) was employed for channel selection and power allocation, showing significant throughput gains by dynamically adapting to varying interference patterns Zhou (2021); Wang *et al.* (2021); Tan *et al.* (2021). In particular, deep Q-learning methods that integrated CNNs boosted system capacity and better handled continuous variations in user demands Wang *et al.* (2021), while policy-gradient reinforcement learning (RL) approaches combined with unsupervised learning were shown to improve both energy efficiency and transmission rates Sun *et al.* (2023b). Beyond reinforcement learning, deep neural networks (DNNs) were applied to mixed-integer optimization in multi-channel D2D systems. One notable example was a supervised–unsupervised hybrid approach that achieved near-optimal results in real-time Lee & Schober (2022). Similarly, a time-sensitive DL framework leveraged short-packet coding and game theory, achieving a suboptimal solution for channel selection and power control for 5G ultra-reliable low-latency communications (URLLC) scenarios Zheng *et al.* (2022). Meanwhile, CNN-based solutions proved adept at mitigating interference while preserving QoS requirements for channel allocation Li *et al.* (2022). Additional efforts included

Online Distributed Deep RL (OD-DRL), which focused on efficient channel assignment and power control in dynamic cellular environments Sun & Nakhai (2021). Furthermore, several recent studies have applied DRL to unmanned aerial vehicle (UAV) and mobile edge computing (MEC) settings. For instance, Li & Aghvami (2023); Li, Sellathurai, Chu, Xiao & Aghvami (2023) proposed DRL-based schemes for joint resource block allocation and beamforming in UAV networks, while Li *et al.* (2025) introduced a distributed multi-agent DRL framework for energy-efficient task offloading in UAV-assisted edge computing. Collectively, these studies demonstrate the adaptability of DRL to mobility-driven scenarios.

Despite these advances, DNN and CNN architectures often encountered key limitations. They typically required fixed-size inputs and outputs, risking inflexibility in scenarios with variable user populations or channel states. Moreover, these models heavily depended on the quality and representativeness of training data, making them vulnerable to overfitting or instability when network conditions deviated from those observed during training.

5.1.3 Graph Neural Network (GNN)-based Solutions

Graph Neural Networks (GNNs) offer a robust alternative for dynamic wireless networks by handling irregular data structures and capturing complex interactions. Several recent GNN-based models have targeted specific challenges in wireless networks. For example, a reinforcement learning (RL)-based method with graph convolutional networks was shown to enhance channel allocation and system throughput but faced scalability issues Nakashima *et al.* (2020). To address this, Random Edge Graph Neural Networks (REGNNs) were introduced for improved scalability and adaptability, though they struggled in dynamic environments Eisen & Ribeiro (2020). Heterogeneous Graph Neural Networks (HetGNNs) were applied for efficient power control in cellular networks Guo & Yang (2021), while the unfolded weighted minimum mean square error (UWMMSE) was developed for robust and generalizable power allocation Chowdhury *et al.* (2021). Message Passing Graph Neural Networks (MPGNNs) and Wireless Channel Graph Convolution Networks (WCGCNs) were also explored for scalable radio resource management (RRM) Shen *et al.* (2021). Semi-supervised learning with GNNs

was proposed to optimize user association and power control in ultra-dense networks, although these models suffered performance degradation in dynamic conditions Zhang *et al.* (2021a). Additionally, the Heterogeneous Interference Graph Neural Network (HIGNN) demonstrated scalable and adaptable power control and beamforming capabilities Zhang *et al.* (2021b). Moreover, GraphSAGE was used for access point selection in cell-free massive multiple-input multiple-output (MIMO) systems to improve spectrum efficiency Ranasinghe *et al.* (2021), while a similar GNN framework focused on resource allocation in wireless IoT networks, adapting well to different operational settings Chen *et al.* (2022). While GNN-based solutions have shown promise, several challenges persist, particularly in dynamic environments. Specifically, models must better adapt to time-varying topologies, scale to large networks, support real-time processing, generalize across diverse network conditions, operate with limited training data, and remain robust to uncertainty and noise.

Table 5.1 Comparison of TGNN with existing RRM methods

	Temporal modeling	**Dynamic topology**	**Memory aware**	**Scalable design**
Non-learnable	No	No	No	No
DL-based	Partial	Partial	No	No
GNN-based	No	Yes	No	Partial
TGNN (Proposed)	Yes	Yes	Yes	Yes

5.1.4 Motivation and Contribution

Existing resource allocation methods for wireless networks exhibit critical limitations. While often comprehensive, heuristic and iterative solutions tend to be computationally expensive and impractical for real-time decision-making in rapidly evolving environments. DL models provide faster inference but suffer from limited training data, rigid input–output structures, and overfitting risks. Meanwhile, GNNs capture node and edge interactions but treat each time-slot independently, neglecting the temporal evolution of wireless networks. A fundamental shortcoming of these approaches is their memoryless, per-time-slot decision-making process,

which disregards the temporal correlation of both user activities and wireless channels. Consequently, past states do not inform current decisions, compromising performance when user mobility and channel conditions shift over time. Addressing this gap requires an adaptive, time-aware solution that effectively manages complexity and variability in wireless environments. To clearly delineate these gaps, Table 5.1 compares representative non-learnable, DL-based, and GNN-based resource allocation methods across key criteria, including temporal modeling, support for dynamic topologies, memory-awareness, and scalability. To meet this need, we propose a **Continuous-Time Dynamic Graph (CTDG) modeling framework** for wireless systems, wherein network events—such as link activations, updates, and deactivations—occur continuously, rather than at discrete snapshots. Our approach employs a **Temporal GNN (TGNN)** to incorporate these events into an internal memory, enabling the generation of temporal embeddings for power and channel allocation decisions.

Our work introduces a TGNN-based resource allocation framework that explicitly captures temporal dynamics in wireless networks. The main contributions include:

1. **CTDG System Modeling:** We introduce a continuous-time representation of D2D communications, allowing a varying number of links with fluctuating channels to communicate and interfere. We also demonstrate how to convert the network's dynamic behavior into a CTDG structure for long-term performance optimization.
2. **TGNN-Based Solution:** We develop a novel embedding mechanism based on TGNNs that processes network events and maintains node-level and global memory, thus retaining historical knowledge for future decisions. We further present a combined training and deployment process leveraging both supervised and unsupervised learning.
3. **Realistic Wireless Environment:** We validate our approach in a detailed SUMO-based simulation environment, incorporating pedestrian and vehicular mobility patterns. The resulting dynamic topology accurately reflects real-world D2D conditions where links activate/deactivate based on distance constraints and pairing probabilities.
4. **Performance Evaluation:** Extensive evaluations across various network settings underscore the effectiveness of our approach. We conduct ablation studies to reveal each component's

impact, demonstrating near-optimal performance with low computational overhead. Compared to heuristic and other DL-based methods (DNN, CNN, and snapshot-based GNNs), our scheme excels in preserving long-term constraints and maximizing the average network rate in dynamic wireless scenarios.

5.1.5 Organization

The remainder of this paper is organized as follows. Section II introduces the system model and problem formulation. Section III details the proposed resource allocation framework, including the architecture and the training/deployment methodology. Section IV presents the simulation setup and numerical results. Finally, Section V concludes the paper.

5.2 SYSTEM MODEL AND PROBLEM FORMULATION

5.2.1 System model

In the considered setup, an RRM control center, $\mathcal{BS}$, manages a dynamic D2D overlay multi-channel communication network, $\mathcal{D}(t) = \{1, 2, \ldots, D(t)\}$, where $D(t)$ is the number of active D2D links at time t. Each D2D link, with its specific hardware limitations, has a maximum transmit power $p_i^{\max}$ and mobility patterns defined by the velocity $v_i(t)$. In this scenario, K orthogonal resource blocks (RBs) with bandwidth B, denoted as $\mathcal{K} = \{1, 2, \ldots, K\}$, are available. Each link may occupy up to $L_{\max}$ RBs per time-slot (with $1 \leq L_{\max} \leq K$). Accordingly, co-channel interference occurs whenever multiple D2D links use the same RB. The network operates on a time-slotted basis, with $\mathcal{BS}$ performing the following tasks in each time-slot: identifying D2D links and their activity (start, continue, or stop communicating), acquiring instantaneous CSI via pilot feedback, computing power and channel allocation decisions, and broadcasting these decisions to the D2D pairs.

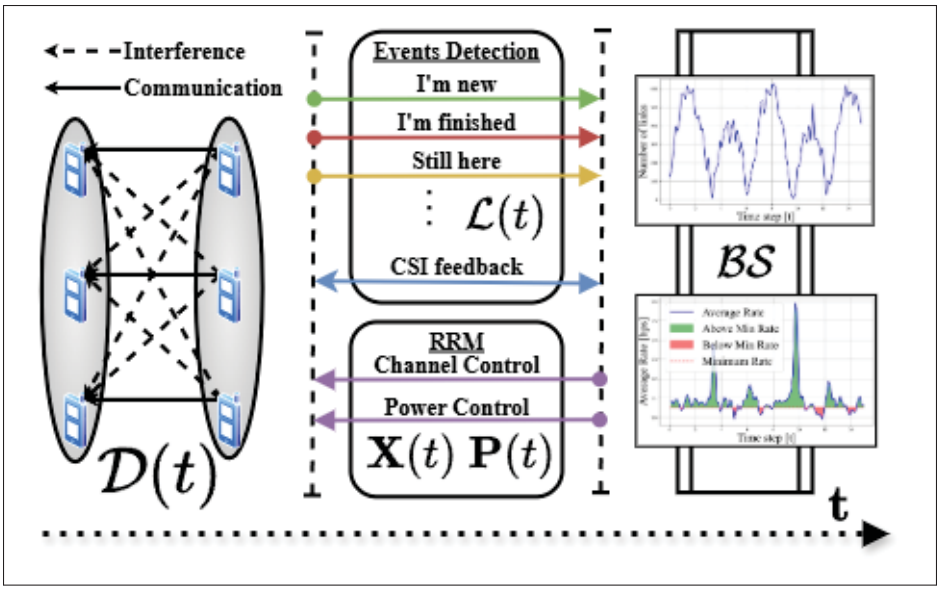


Figure 5.1 Over the period $\mathcal{T}$, D2D links communicate and interfere. At each time t, the control unit $\mathcal{BS}$ estimates CSI via pilot feedback and detects events (deactivations, updates, and activations). Together with the CSI $\mathbf{G}(t)$, these events form the interaction set $\mathcal{L}(t)$, which is used to determine power $\mathbf{P}(t)$ and channel allocation $\mathbf{X}(t)$

The CSI tensor is defined as $\mathbf{G}(t) \in \mathbb{R}^{D(t)\times D(t)\times K}$, representing the channel state information for all D2D links across all RBs at time t. The instantaneous channel gain is given by:

$$g_{ijk}^{(t)} = \xi_{ijk}^{(t)}\, \tau_{ijk}^{(t)} \left| \sqrt{1-\sigma_e^2}\, \kappa_{ijk}^{(t)} + \sigma_e e_{ijk}^{(t)} \right|^2, \tag{5.1}$$

where $\xi_{ijk}^{(t)}$ models large-scale path loss, $\tau_{ijk}^{(t)}$ captures shadowing, and $\kappa_{ijk}^{(t)}$ is the small-scale Rayleigh fading. To capture Doppler effects caused by device mobility, the small-scale fading component $\kappa_{ijk}^{(t)}$ is modeled as $\kappa_{ijk}^{(t)} = \left|\sum_n \alpha_n \exp\left(\mathrm{i}(2\pi f_{D,n} t + \phi_n)\right)\right|$, where α_n are Rayleigh-distributed amplitudes, ϕ_n are uniformly distributed phases, and $\mathrm{i} = \sqrt{-1}$. The Doppler shift is given by $f_{D,n} = \frac{|v_i - v_j| f_c}{c} \cos(\theta_n)$, where v_i and v_j are the velocities of nodes i and j, f_c is the carrier frequency, c is the speed of light, and θ_n is the angle of arrival. To incorporate temporal correlation, both the fading term and estimation error are modeled using first-order Gauss–Markov processes: $\kappa_{ijk}^{(t)} = \rho_c \kappa_{ijk}^{(t-1)} + \sqrt{1-\rho_c^2}\, \xi(t)$ and $e_{ijk}^{(t)} = \rho_e e_{ijk}^{(t-1)} + \sqrt{1-\rho_e^2}\, \epsilon(t)$,

where ρ_c and ρ_e are the correlation coefficients for fading and CSI error, respectively, and $\xi(t), \epsilon(t) \sim \mathcal{N}(0,1)$ are i.i.d. Gaussian processes. In practice, parameters ρ_c and ρ_e can be estimated from measured channel and error traces using temporal autocorrelation as shown below:

$$\rho_c = \frac{\mathbb{E}[\kappa(t)\,\kappa^*(t-\Delta t)]}{\mathbb{E}[|\kappa(t)|^2]}, \quad \rho_e = \frac{\mathbb{E}[e(t)\,e^*(t-\Delta t)]}{\mathbb{E}[|e(t)|^2]}. \tag{5.2}$$

We define the per-RB power allocation matrix $\mathbf{P}(t) \in \mathbb{R}_+^{D(t)\times K}$ with entries $p_{ik}^{(t)}$ and the channel allocation matrix $\mathbf{X}(t) \in \{0,1\}^{D(t)\times K}$ with entries $x_{ik}^{(t)}$, where $x_{ik}^{(t)} = 1$ indicates that RB k is assigned to link i at time t. The throughput of link i at time-slot t is defined as follows:

$$r_i^{(t)} = B \sum_{k=1}^{K} \log_2\left(1 + \frac{x_{ik}^{(t)}\, p_{ik}^{(t)}\, g_{iik}^{(t)}}{\sigma^2 + \sum_{j\neq i}^{D(t)} x_{jk}^{(t)}\, p_{jk}^{(t)}\, g_{ijk}^{(t)}}\right). \tag{5.3}$$

where σ^2 denotes the noise power (per RB), and the summation captures co-channel interference from other active D2D links on RB k.

5.2.2 Optimization problem formulation

Considering the varying number of D2D links, their dynamic interference, and the evolving channel conditions, the objective is to find an optimal resource allocation policy that maximizes the average network rate while satisfying a minimum data-rate requirement for each active link. Without loss of generality, let $\mathcal{T} = \{1, 2, \ldots, T\}$ be the time horizon. The problem is formulated

as:

$$\begin{aligned}
\max_{\{\mathbf{P}(t),\mathbf{X}(t)\}_{t\in\mathcal{T}}} \; & \sum_{t\in\mathcal{T}} \frac{1}{|\mathcal{D}(t)|} \sum_{i\in\mathcal{D}(t)} r_i^{(t)}(\mathbf{P}(t),\mathbf{X}(t),\mathbf{G}(t)) \\
\text{s.t} \quad & r_i^{(t)} \geq r_{\min} \quad \forall i,\ \forall t \ (C.1) \\
& \mathbf{X}(t)\,\mathbf{1}_{\mathcal{K}} \leq L_{\max}\,\mathbf{1}_{\mathcal{D}(t)} \quad \forall t \ (C.2) \\
& \sum_{k=1}^{K} p_{ik}^{(t)} \leq p_i^{\max} \quad \forall i,\ \forall t \ (C.3) \\
& 0 \leq p_{ik}^{(t)} \leq x_{ik}^{(t)}\, p_i^{\max} \quad \forall i,\ \forall k,\ \forall t \ (C.4) \\
& \mathbf{X}(t) \in \{0,1\}^{|\mathcal{D}(t)|\times|\mathcal{K}|} \quad \forall t \ (C.5)
\end{aligned} \tag{5.4}$$

where (C.1) enforces the minimum throughput requirement, (C.2) limits the number of RBs assigned to each link to $L_{\max}$, (C.3) enforces the per-link power budget across active RBs, and (C.4) couples transmit power with RB activation such that $p_{ik}^{(t)} = 0$ whenever $x_{ik}^{(t)} = 0$. Setting $L_{\max} = 1$ yields the single-RB formulation.

5.2.3 Continuous Dynamic Wireless Graph Modelling

The temporal dynamics of the considered D2D network, coupled with relational knowledge over time, naturally translate into a dynamic graph in which the number of nodes/edges and their features (e.g., CSI between D2D links) vary over time. At time-slot t, a snapshot of the network is represented by the graph $\mathcal{G}(\mathcal{V}(t),\mathcal{E}(t))$, where $\mathcal{V}(t)$ is the temporal set of vertices representing $\mathcal{D}(t)$, and $\mathcal{E}(t) = \mathcal{V}(t) \times \mathcal{V}(t)$ is the temporal set of edges capturing interference relationships among D2D links. At each time-slot t, three types of link-level events can occur:

1. **Activation of D2D links**: a set of D2D links becomes active and is added to $\mathcal{D}(t)$. These links start transmitting after $\mathcal{BS}$ allocates power and channels to them.
2. **Update of D2D links**: a set of D2D links remains active (kept in $\mathcal{D}(t)$). They update their link information, such as CSI and allocated resources.
3. **Deactivation of D2D links**: a set of D2D links disconnects and is deleted from $\mathcal{D}(t)$ (i.e., becomes inactive). Consequently, $\mathcal{BS}$ stops their CSI estimation and resource allocation processes.

These network events can be modeled as a continuous-time dynamic graph (CTDG). We define $\mathcal{L}(t)$ as the set of interaction events occurring at time t. An interaction event is defined by the tuple:

$$l(t) = \left(i, j, \mathbf{v}_i, \mathbf{e}_{ij}, \mathbf{v}_j, t, \rho\right), \quad (i, j) \in \mathcal{V}(t) \times \mathcal{V}(t), \tag{5.5}$$

where node i with feature $\mathbf{v}_i$ interacts with node j with feature $\mathbf{v}_j$ at time t, with edge feature $\mathbf{e}_{ij}$, and the event type is $\rho \in \{0 : \textbf{add}, 1 : \textbf{update}, 2 : \textbf{delete}\}$. For simplicity, we focus on features common to most wireless systems, specifically the network's estimated CSI. Note that $\mathbf{v}_i(t)$ denotes a node-feature vector and is distinct from the velocity $v_i(t)$ used in the system model. When a new link i appears in the network at time t, it generates an addition event with $\rho = 0$ (add). The node feature $\mathbf{v}_i(t) = \{G^{(t)}_{iik},\ k \in \mathcal{K}\}$ represents the link's characteristics, and the edge feature $\mathbf{e}_{ij}(t) = \{G^{(t)}_{ijk},\ k \in \mathcal{K}\}$ represents the interference between the new link i and selected existing links j with the strongest interference (highest $G^{(t)}_{ijk}$). For an active link i, an update event with $\rho = 1$ (update) is generated. This updates the link feature $\mathbf{v}_i(t) = \{G^{(t)}_{iik},\ k \in \mathcal{K}\}$ and the interference features $\mathbf{e}_{ij}(t) = \{G^{(t)}_{ijk},\ k \in \mathcal{K}\}$ based on the latest CSI estimates, ensuring that the interaction data accurately reflects the current network state. When a link i disconnects, a deletion event with $\rho = 2$ (delete) is generated, with $\mathbf{v}_i(t) = \{G^{(t-1)}_{iik},\ k \in \mathcal{K}\}$ and $\mathbf{e}_{ij}(t) = \{G^{(t-1)}_{ijk},\ k \in \mathcal{K}\}$, representing the cessation of interference from the disconnected link to the remaining active links.

5.2.4 CTDG optimization problem formulation

Our objective captures the network state's evolution over time. We aim to develop a policy that combines historical data with current observations to optimize performance over a time horizon. Accordingly, the power and channel allocations are defined as

$$\mathbf{P}(t),\ \mathbf{X}(t) = \mathcal{F}_\theta\big(\mathcal{L}(t)\big), \tag{5.6}$$

where $\mathcal{F}_\theta$ is a DL model that processes the current interaction events to determine these allocations. We then seek the optimal model parameters θ that maximize the time-averaged

performance of the active D2D links:

$$\theta^* = \arg\max_{\theta} \sum_{t\in\mathcal{T}} \frac{1}{|\mathcal{D}(t)|} \sum_{i\in\mathcal{D}(t)} r_i^{(t)}\Big(\mathbf{P}(t), \mathbf{X}(t), \mathbf{G}(t)\Big), \tag{5.7}$$

with $(\mathbf{P}(t), \mathbf{X}(t)) = \mathcal{F}_\theta(\mathcal{L}(t))$, subject to constraints (C.1)–(C.4). The next section details the $\mathcal{F}_\theta$ architecture and its adaptation to the dynamically changing network environment.

5.3 Temporal Graph Radio Resource Management

In this section, we detail the key components of our proposed framework, the Temporal Graph Neural Network (TGNN), and provide an overview of its architecture. We begin by describing how CTDG inputs are processed and transformed into temporal embeddings through message computation and memory management. These embeddings are subsequently used to generate the optimization variables. Finally, we outline the training and deployment procedures.

5.3.1 Overview

Following the encoder–decoder framework proposed in Kazemi *et al.* (2019), our model consists of a Temporal Graph Neural Network (TGNN) encoder and a decoder. The encoder processes a sequence of time-stamped graph events, $\mathcal{L}(t)$, as input and outputs node embeddings, $\mathcal{Z}(t) = \big(\mathbf{z}_1(t), \ldots, \mathbf{z}_{|\mathcal{D}(t)|}(t)\big)$, at each time-slot t. The decoder then maps these embeddings to power control $\mathbf{P}(t)$ and channel allocation $\mathbf{X}(t)$. Fig. 5.2 provides an overview of the model architecture. At each time-slot t, the set of events $\mathcal{L}(t)$ is collected and processed through seven distinct modules. For each event, messages denoted by $\mathcal{M}(t)$ are computed using message functions based on the event type and sent to the relevant nodes. If multiple messages are sent to the same node, they are combined into a single message using a message aggregator with a learnable weighting mechanism. The aggregated messages are denoted by $\bar{\mathcal{M}}(t)$. Based on these aggregated messages, the memory states $s_i(t)$ of the involved nodes $i \in \mathcal{V}(t)$ are updated in the memory updater module using the previous memory states $\mathcal{S}(t^-)$ and the aggregated messages. The node memories are then aggregated using a self-attention mechanism to produce

$\bar{s}(t)$, which is used to update the graph memory $s_{\mathcal{G}}(t)$ based on its previous state $s_{\mathcal{G}}(t^-)$. Finally, the embedding module generates the nodes' temporal embeddings $\mathcal{Z}(t)$, which are passed to the decoder for power and channel allocation. The following subsections provide detailed explanations of each module.

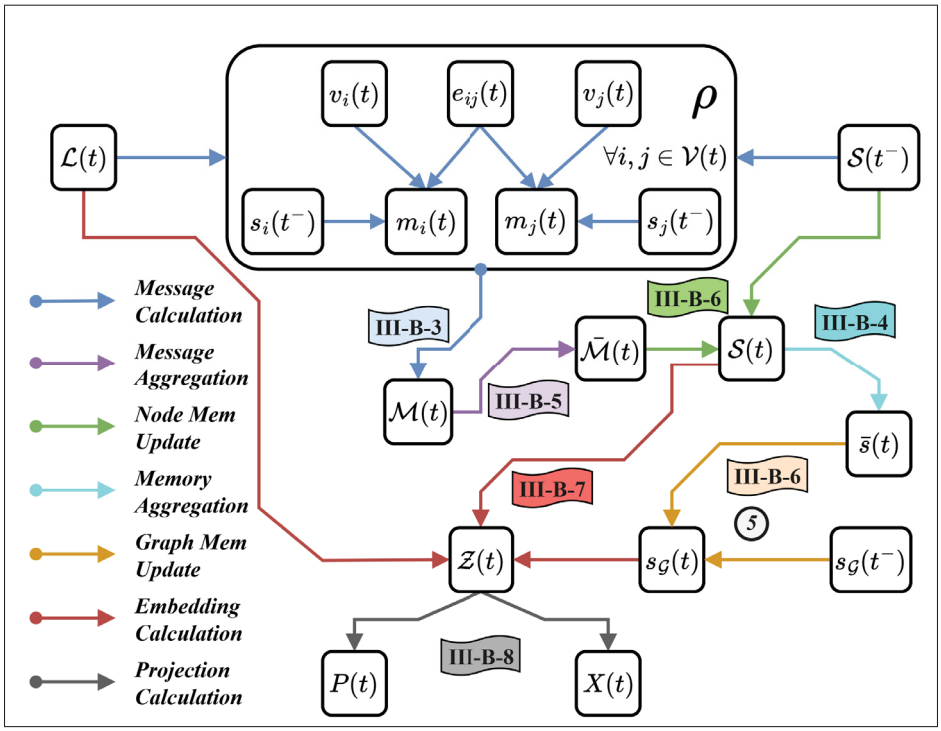


Figure 5.2 TGNN model architecture. The system processes time-stamped graph events $\mathcal{L}(t)$ to generate node embeddings $\mathcal{Z}(t)$, which are then mapped to power $\mathbf{P}(t)$ and channel $\mathbf{X}(t)$ decisions. Each module is labeled according to its corresponding subsection in Section III-B

5.3.2 Core Modules

5.3.2.1 Node-wise memory

At time t, the model maintains a memory state for each node, denoted by $\mathcal{S}(t)$. This memory consists of vectors $s_i(t)$ for each node i that the model has encountered so far (including currently active nodes). After each event, a node's memory is updated to represent its past in a

compressed form, allowing the TGNN to capture long-term dependencies. When a new node i is encountered at time-slot t, its memory $s_i(t)$ is initialized from a normal distribution, i.e., $s_i(t) \sim \mathcal{N}(0, 1)$. Thereafter, the memory state is updated with each event involving the node, including during deployment after training. Due to the network's dynamic nature, new links that have not previously interacted with the model frequently emerge, while older links may remain inactive for long periods. As a result, the memory states of inactive links become less informative and occupy space that could be used to track newly active links. Since the node-wise memory is finite, it is impractical to retain all nodes indefinitely. To manage this, we delete the memory states of inactive links using a First-In, First-Out (FIFO) policy. Consequently, the node-wise memory does not retain information about nodes that have been inactive for a certain period, treating their next appearance as a first encounter. This motivates the use of an additional mechanism to handle rapid link churn in dynamic networks.

5.3.2.2 Graph-wise memory

Unlike node-wise memory, which focuses on individual nodes and their interactions over time, graph-wise memory is represented by a single vector $s_{\mathcal{G}}(t)$ that captures a global view of the network state and its evolution. Similar to node-wise memory, $s_{\mathcal{G}}(t)$ is initialized as a zero vector, i.e., $s_{\mathcal{G}}(t = 0) = \mathbf{0}_{d_{\mathcal{G}} \times 1}$. This graph-wise memory is updated based on its previous state and an attention-based aggregation of the node memory states.

5.3.2.3 Message function

For each interaction event between node i and node j, a message is computed to update their respective memory states based on the nature of the interaction. Specifically, the event type ρ dictates the structure of this message. We define $\mathbf{msg}_s^{\rho}$ and $\mathbf{msg}_d^{\rho}$ as the message functions for the source and destination nodes, respectively, corresponding to the event type ρ. These functions are implemented as trainable MLP blocks. The messages $m_i(t)$ and $m_j(t)$ at time-slot

t are then calculated as follows:

$$\begin{aligned} m_i(t) &= \mathbf{msg}_s^{\rho}\big(s_i(t^-), s_j(t^-), \phi(t, \Delta t_1, \Delta t_2), \mathbf{e}_{ij}(t), \mathbf{v}_i(t), \mathbf{v}_j(t)\big), \\ m_j(t) &= \mathbf{msg}_d^{\rho}\big(s_j(t^-), s_i(t^-), \phi(t, \Delta t_1, \Delta t_2), \mathbf{e}_{ij}(t), \mathbf{v}_j(t), \mathbf{v}_i(t)\big). \end{aligned} \tag{5.8}$$

Here, $s_i(t^-)$ and $s_j(t^-)$ represent the memory states of nodes i and j, respectively, just before their interaction. The temporal encoding function $\phi(t, \Delta t_1, \Delta t_2)$ incorporates multiple temporal aspects: t denotes the absolute time of the current interaction, Δt_1 captures the time elapsed since the last interaction between i and j, and Δt_2 reflects the cumulative interaction duration between i and j up to the current time-slot. Thus, following Vaswani *et al.* (2017), we define the temporal encoding as:

$$\phi(t, \Delta t_1, \Delta t_2) = \cos(\omega \cdot t + \omega' \cdot \Delta t_1 + \omega'' \cdot \Delta t_2), \tag{5.9}$$

where ω, ω', and ω'' are learnable parameters. The parameter ω controls the frequency of the encoding related to the absolute time t, ω' adjusts the frequency based on the time difference Δt_1, while ω'' captures the influence of the cumulative interaction duration Δt_2. This encoding enables the model to capture the absolute temporal position, relative timing between events, and cumulative temporal context of interactions, which is crucial for modeling temporal dependencies in dynamic systems.

5.3.2.4 Memory self-attention aggregator

The process starts with the set of node memory vectors $\mathcal{S}(t)$. Each memory vector is transformed into query, key, and value vectors using linear projections with learned weight matrices $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$. The self-attention mechanism Vaswani *et al.* (2017) then computes attention weights by forming the scaled dot-product between queries and keys and applying the softmax function:

$$A = \text{softmax}\left(\frac{\mathcal{S}(t)\mathbf{W}_Q \left(\mathcal{S}(t)\mathbf{W}_K\right)^T}{\sqrt{d}}\right). \tag{5.10}$$

These attention weights are used to compute a weighted sum of the value vectors. The resulting output is averaged across the sequence and passed through an MLP block, producing the aggregated memory vector $\bar{s}(t)$:

$$\bar{s}(t) = \mathrm{MLP}\left(\frac{1}{|\mathcal{S}(t)|} \sum_{i=1}^{|\mathcal{S}(t)|} (A\,\mathcal{S}(t)\mathbf{W}_V)_i \right). \tag{5.11}$$

5.3.2.5 Message weighted aggregator

In scenarios where nodes receive numerous messages due to interference from multiple nearby links or requirements for batch processing, effectively managing this information is vital. We incorporate a *Context-Aware Softmax Weighted Aggregation Mechanism* that applies a softmax function to MLP-transformed message scores. The mechanism is expressed as:

$$\begin{aligned} \bar{m}_i(t) &= \sum_{f \in \mathcal{M}_i(t)} w_f\, m_f, \quad i \in \mathcal{V}(t), \\ w_f &= \frac{\exp\big(\mathrm{MLP}(m_f)\big)}{\sum_{f' \in \mathcal{M}_i(t)} \exp\big(\mathrm{MLP}(m_{f'})\big)}. \end{aligned} \tag{5.12}$$

In this formulation, $\mathcal{M}_i(t)$ represents the set of messages received by node i at time t, and m_f is an individual message within this set. The function $\mathrm{MLP}(\cdot)$ maps each message to a scalar score, and the softmax normalization yields weights $w_f = \mathrm{softmax}(\mathrm{MLP}(m_f))$. This design weights each message according to its learned relevance, enabling a prioritized integration of the most pertinent information into the node's memory update.

5.3.2.6 Memory updater

The memory of a node $s_i(t)$ is updated upon each event involving the node, and the graph-wise memory $s_{\mathcal{G}}(t)$ is updated at each time-slot. We employ separate Gated Recurrent Unit (GRU) functions, $\mathbf{GRU}_1$ for nodes and $\mathbf{GRU}_2$ for the graph. The generic GRU function $\mathbf{GRU}(u_t, h_{t-1})$

is defined as follows Dey & Salem (2017):

$$\begin{aligned} z_t &= \sigma(\mathbf{W}_z \cdot [h_{t-1}, u_t] + b_z), \\ r_t &= \sigma(\mathbf{W}_r \cdot [h_{t-1}, u_t] + b_r), \\ \tilde{h}_t &= \tanh(\mathbf{W} \cdot [r_t \odot h_{t-1}, u_t] + b), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t. \end{aligned} \tag{5.13}$$

where u_t is the input at time-slot t, h_{t-1} is the previous hidden state, z_t is the update gate, r_t is the reset gate, $\tilde{h}_t$ is the candidate hidden state, $\sigma(\cdot)$ denotes the sigmoid function, and $\odot$ denotes element-wise multiplication. Parameters $\mathbf{W}_z$, $\mathbf{W}_r$, $\mathbf{W}$, b_z, b_r, and b are learnable weights and biases. Using the GRU formulation, the updates for the node and graph memories are computed as follows:

$$\begin{aligned} s_i(t) &= \mathbf{GRU}_1\big(\bar{m}_i(t), s_i(t^-)\big), \quad \forall i \in \mathcal{V}(t), \\ s_{\mathcal{G}}(t) &= \mathbf{GRU}_2\big(\bar{s}(t), s_{\mathcal{G}}(t^-)\big). \end{aligned} \tag{5.14}$$

5.3.2.7 Embedding

The temporal embedding module computes the node embedding $\mathbf{z}_i(t)$ at any time-slot t. Initially, each node's embedding is calculated independently of its neighbors to ensure that even isolated nodes are meaningfully embedded:

$$\mathbf{z}_i^{(0)}(t) = \mathrm{MLP}_1^{(0)}\big(s_i(t), \mathbf{v}_i(t), s_{\mathcal{G}}(t)\big). \tag{5.15}$$

In subsequent iterations, $p = \{1, 2, \ldots, P\}$, the model aggregates messages from node i's neighbors over the time window $\mathcal{T}_h = [\max(0, t - T_h),\ t]$. The embedding message $\Phi_{ij}(t, t')$ from node j at time t' is computed as:

$$\Phi_{ij}(t, t') = \mathrm{MLP}_2^{(p)}\big(s_i(t), s_j(t), \mathbf{e}_{ij}(t'), \mathbf{v}_i(t'), \mathbf{v}_j(t')\big), \tag{5.16}$$

where $\mathbf{e}_{ij}(t')$ represents the edge features between nodes i and j. The set $\mathcal{N}_i(t')$ denotes the neighbors of node i at time t', i.e., all nodes j connected to node i by an edge in the graph at time t'. An attention mechanism then assigns weights to these messages. The attention score $\alpha_{ij}(t,t')$ is computed by the attention network $\mathrm{MLP}_3^{(p)}$ as follows:

$$\alpha_{ij}(t,t') = \frac{\exp\Big(\mathrm{MLP}_3^{(p)}\big(\mathbf{z}_i^{(p-1)}(t), \mathbf{z}_j^{(p-1)}(t')\big)\Big)}{\sum_{k\in\mathcal{N}_i(t')} \exp\Big(\mathrm{MLP}_3^{(p)}\big(\mathbf{z}_i^{(p-1)}(t), \mathbf{z}_k^{(p-1)}(t')\big)\Big)}. \tag{5.17}$$

Node i's embedding is iteratively updated with the weighted sum of these messages:

$$\mathbf{z}_i^{(p)}(t) = \mathbf{z}_i^{(p-1)}(t) + \sum_{t'\in\mathcal{T}_h} \sum_{j\in\mathcal{N}_i(t')} \alpha_{ij}(t,t')\,\Phi_{ij}(t,t'). \tag{5.18}$$

Thus, this embedding mechanism captures both the node's historical state and its recent interactions. It is also possible to incorporate multi-head or normalization-based attention mechanisms to enhance robustness under high feature variability.

5.3.2.8 Projection layer

The embedding set $\mathcal{Z}(t)$ is processed by two MLP blocks to produce power and RB-allocation outputs. For power control, we apply a sigmoid to bound the output and then scale it by the per-link maximum power:

$$\mathbf{P}(t) = \sigma\big(\mathrm{MLP}_P(\mathcal{Z}(t))\big) \odot \mathbf{p}_{\max}(t), \tag{5.19}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$, $\odot$ denotes element-wise multiplication, and $\mathbf{p}_{\max}(t) = [p_1^{\max}, \ldots, p_{|\mathcal{D}(t)|}^{\max}]^T$. For RB allocation, the softmax function is applied to obtain per-link RB-selection probabilities:

$$\hat{\mathbf{X}}(t) = \mathrm{softmax}\big(\mathrm{MLP}_X(\mathcal{Z}(t))\big). \tag{5.20}$$

During inference, the binary allocation matrix $\mathbf{X}(t)$ is obtained from the predicted probability vector $\hat{\mathbf{X}}_i(t)$ for each link i. Specifically, $x_{ik}^{(t)}$ is set to 1 if RB k ranks among the top-$L_{\max}$

probabilities for link i, and 0 otherwise:

$$x_{ik}^{(t)} = \begin{cases} 1, & \text{if } k \in \text{Top-}L_{\max}\left(\hat{\mathbf{X}}_i(t)\right), \\ 0, & \text{otherwise,} \end{cases} \tag{5.21}$$

where Top-$L_{\max}\left(\hat{\mathbf{X}}_i(t)\right)$ denotes the set of indices corresponding to the $L_{\max}$ largest elements of $\hat{\mathbf{X}}_i(t)$. In the single-RB case ($L_{\max} = 1$), this reduces to a winner-takes-all rule: $x_{ik}^{(t)} = 1$ if $k = \arg\max_k \hat{x}_{ik}^{(t)}$, and 0 otherwise.

5.3.3 Training and Deployment

The model operates in two distinct phases: an offline training phase based on a supervised loss using a labeled dataset, and a deployment phase using unsupervised and penalty-based losses.

5.3.3.1 Training dataset generation

The labeled dataset consists of tuples over a time period, noted as $\left\{\left(\mathbf{G}(t), \tilde{\mathbf{X}}(t), \tilde{\mathbf{P}}(t)\right)\right\}_{\mathcal{T}_{\text{train}}}$, where $\mathcal{T}_{\text{train}}$ is the training period, and $\tilde{\mathbf{X}}(t)$ and $\tilde{\mathbf{P}}(t)$ are the optimal solutions for a given CSI $\mathbf{G}(t)$. The problem we aim to solve (Eq. (3)) is a mixed-integer nonlinear programming (MINLP) problem, requiring an appropriate MINLP optimization technique. In this context, an exhaustive search is impractical due to the vast solution space; therefore, we employ PyMOO Blank & Deb (2020), a Python library for multi-objective optimization that utilizes a genetic algorithm-based approach. PyMOO is renowned for its flexibility, proven effectiveness, and parallel computation capabilities, which significantly expedite the dataset construction process.

5.3.3.2 Offline Training

In this phase, we employ supervised learning to guide the model in acquiring an initial strategy derived from the training dataset. The power control task is treated as a supervised continuous prediction problem. We use the mean squared error (MSE) to evaluate the difference between

the predicted and target power levels. The channel allocation task is framed as a multi-label supervised classification problem, and the categorical cross-entropy (CCE) loss is employed to penalize misclassification. The combined supervised loss function during training is defined as follows:

$$\mathbf{L}_{\text{sup}}^{(t)}(\theta) = \sum_{i \in \mathcal{D}(t)} \left(\tilde{\mathbf{P}}_i(t) - \mathbf{P}_i(t)\right)^2 - \sum_{k \in \mathcal{K}} \tilde{\mathbf{X}}_{ik}(t) \log\left(\hat{\mathbf{X}}_{ik}(t)\right). \tag{5.22}$$

During training, the model processes events sequentially within each epoch, with events grouped into batches that preserve chronological order, ensuring that all events in a batch occur before those in subsequent batches. Algorithm 1 outlines the offline training process. Initially, the algorithm sets up the node and graph memory states. At each time-slot, these states are updated by incorporating new nodes, processing and aggregating messages, and refining the embeddings. The model parameters are then iteratively updated based on the computed batch loss until convergence. Of note, PYMOO-generated labels are used offline for supervised pretraining only, while unsupervised fine-tuning during deployment mitigates the effect of approximation noise

5.3.3.3 Deployment

In this phase, the model transitions from supervised learning to unsupervised optimization, aiming to improve overall performance across all links while ensuring QoS by meeting minimum required data rates. The objective during deployment is to minimize a weighted combination of the rate loss, regulation loss, and fairness loss. The primary unsupervised loss function used during deployment is the negative sum of the data rates across all links, defined as:

$$L_{\text{rate}}^{(t)}(\theta) = -\sum_{i \in \mathcal{D}(t)} r_i^{(t)}\left(\hat{\mathbf{P}}(t), \hat{\mathbf{X}}(t)\right). \tag{5.23}$$

In addition, we enforce $r_i^{(t)} \geq r_{\min}$ via an ℓ_1 exact–penalty:

$$L_{\text{reg}}^{(t)}(\theta) = \lambda \sum_{i \in \mathcal{D}(t)} \left[r_{\min} - r_i^{(t)}\big(\hat{\mathbf{P}}(t), \hat{\mathbf{X}}(t)\big)\right]_+. \tag{5.24}$$

Algorithm 5.1 Model Offline Training with Batch Processing

Input: Training set $\{(\mathbf{G}(t), \tilde{\mathbf{X}}(t), \tilde{\mathbf{P}}(t))\}_{t \in \mathcal{T}_{\text{train}}}$
Output: Trained parameters $\boldsymbol{\theta}^*$

Initialize node memory $\mathcal{S} \leftarrow \{\}$ and graph memory $s_{\mathcal{G}} \leftarrow \mathbf{0}_{d_{\mathcal{G}} \times 1}$;
while $\neg$ *Converged* **do**
  for *each batch* $\mathcal{B} \subset \mathcal{T}_{\text{train}}$ **do**
    for *each* $t \in \mathcal{B}$ **do**
      $\mathcal{L}(t) \leftarrow$ **Process**$(\mathbf{G}(t))$;
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{\, s_i(t) \sim \mathcal{N}(0,1) \mid i \in \mathcal{V}(t) \setminus \mathcal{V}(t^-) \,\}$;
      $\mathcal{M}(t) \leftarrow$ **msg**$(\mathcal{S}(t^-), \mathcal{L}(t))$;
      $\bar{\mathcal{M}}(t) \leftarrow$ **msg_agg**$(\mathcal{M}(t))$;
      $\mathcal{S}(t) \leftarrow$ **node_mem_update**$(\mathcal{S}(t^-), \bar{\mathcal{M}}(t))$;
      $\bar{s}(t) \leftarrow$ **node_mem_agg**$(\mathcal{S}(t))$;
      $s_{\mathcal{G}}(t) \leftarrow$ **graph_mem_update**$(s_{\mathcal{G}}(t^-), \bar{s}(t))$;
      $\mathcal{Z}(t) \leftarrow$ **emb**$(\{(\mathcal{L}(t'), \mathcal{Z}(t'))\}_{t' \in \mathcal{T}_h}, \mathcal{S}(t), s_{\mathcal{G}}(t))$;
      $(\mathbf{X}(t), \mathbf{P}(t)) \leftarrow$ **proj**$(\mathcal{Z}(t))$;
    end for
    Compute supervised loss $\mathbf{L}_{\text{sup}}^{(\mathcal{B})}$;
    Update $\boldsymbol{\theta}$ using $\nabla_{\boldsymbol{\theta}} \mathbf{L}_{\text{sup}}^{(\mathcal{B})}$;
  end for
end while
return $\boldsymbol{\theta}^*$;

This penalty activates only on violations and is linear in the amount of infeasibility, yielding a bounded, non-vanishing gradient signal: if $r_i^{(t)} < r_{\min}$, a valid subgradient w.r.t. $r_i^{(t)}$ equals $-\lambda$, so first-order updates push rates upwards until feasibility; on the feasible set, the term is inactive (zero). Under standard constraint qualifications, there exists $\lambda^\star > 0$ such that, for any $\lambda \geq \lambda^\star$, local optima of the penalized objective are KKT-consistent with the original constrained problem.

Moreover, to promote fairness, we maximize the minimum data rate across all links by minimizing the following loss:

$$L_{\min}^{(t)}(\theta) = - \min_{i \in \mathcal{D}(t)} \left\{ r_i^{(t)} \left(\hat{\mathbf{P}}(t), \hat{\mathbf{X}}(t) \right) \right\}. \tag{5.25}$$

Algorithm 2 outlines the deployment process, which mirrors the offline training procedure. At each time-slot, the model begins with the pretrained parameters θ^* and computes the rate, regulation, and minimum-rate losses. We combine the deployment losses with tunable weights $(\lambda, \lambda_1, \lambda_2)$, which are selected empirically Leveraging the supervised offline initialization, the model converges significantly faster during deployment, enabling efficient and adaptive performance refinement.

Algorithm 5.2 Model Deployment

Input: Test CSI sequence $\{\mathbf{G}(t)\}_{t\in\mathcal{T}_{\text{test}}}$ and pre-trained parameters $\boldsymbol{\theta}^*$
Output: Decisions $\{(\mathbf{X}(t), \mathbf{P}(t))\}_{t\in\mathcal{T}_{\text{test}}}$

Initialize node memory $\mathcal{S} \leftarrow \{\}$ and graph memory $s_{\mathcal{G}} \leftarrow \mathbf{0}_{d_{\mathcal{G}}\times 1}$;
for *each* $t \in \mathcal{T}_{\text{test}}$ **do**
  Initialize $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^*$;
  while $\neg$ *Converged* **do**
    $\mathcal{L}(t) \leftarrow$ **Process**$(\mathbf{G}(t))$;
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\, s_i(t) \sim \mathcal{N}(0,1) \mid i \in \mathcal{V}(t) \setminus \mathcal{V}(t^-) \,\}$;
    $\mathcal{M}(t) \leftarrow$ **msg**$(\mathcal{S}(t^-), \mathcal{L}(t))$;
    $\bar{\mathcal{M}}(t) \leftarrow$ **msg_agg**$(\mathcal{M}(t))$;
    $\mathcal{S}(t) \leftarrow$ **node_mem_update**$(\mathcal{S}(t^-), \bar{\mathcal{M}}(t))$;
    $\bar{s}(t) \leftarrow$ **node_mem_agg**$(\mathcal{S}(t))$;
    $s_{\mathcal{G}}(t) \leftarrow$ **graph_mem_update**$(s_{\mathcal{G}}(t^-), \bar{s}(t))$;
    $\mathcal{Z}(t) \leftarrow$ **emb**$(\{(\mathcal{L}(t'), \mathcal{Z}(t'))\}_{t'\in\mathcal{T}_h}, \mathcal{S}(t), s_{\mathcal{G}}(t))$;
    $(\mathbf{X}(t), \mathbf{P}(t)) \leftarrow$ **proj**$(\mathcal{Z}(t))$;
    $(L_{\text{rate}}^{(t)}, L_{\min}^{(t)}, L_{\text{reg}}^{(t)}) \leftarrow$ **Loss**$(\mathbf{G}(t), \mathbf{X}(t), \mathbf{P}(t))$;
    Update $\boldsymbol{\theta}$ using $\nabla_{\boldsymbol{\theta}}\left(\lambda_1 L_{\min}^{(t)} + \lambda_2 L_{\text{rate}}^{(t)} + L_{\text{reg}}^{(t)}\right)$;
  end while
  Output $(\mathbf{X}(t), \mathbf{P}(t))$;
end for

5.3.4 Complexity Analysis

In a single forward iteration, the dominant operations involve processing the set of events $\mathcal{L}(t)$ at a cost of $O\big(|\mathcal{L}(t)|\, d\big)$, updating the node memories for the active links $\mathcal{D}(t)$ at $O\big(|\mathcal{D}(t)|\, d\big)$, and performing self-attention over the aggregated memory sequence at $O\big(m^2 d\big)$,

where $m = |\mathcal{S}(t)|$ and d denotes the hidden/embedding dimension (corresponding to the dimensionality of the node memories, graph memory, and node embeddings). Additionally, the iterative embedding updates over a temporal horizon of T_h time-slots, repeated over P iterations, contribute $O\big(P\,T_h\,(|\mathcal{L}(t)|\,d + P)\big)$. Therefore, the overall complexity for a single forward iteration is:

$$O\Big((|\mathcal{L}(t)| + |\mathcal{D}(t)| + m^2)d + P\,T_h\big(|\mathcal{L}(t)|d + P\big)\Big). \tag{5.26}$$

5.4 Simulation results

5.4.1 Environment setup

To simulate a realistic wireless communication scenario, a 10 km × 5 km map is generated using the Simulation of Urban MObility (SUMO) Lopez *et al.* (2018). Mobility is modeled by randomly injecting pedestrians and vehicles, each assigned random source and destination coordinates, injection times, and realistic velocity profiles based on their location (e.g., roads and sidewalks) and traffic rules. We consider three datasets with different numbers of mobile entities (50, 100, and 150) over 10,000 time-steps, each with a 0.01 s resolution. Using each entity's location and mobility over time, we develop a procedure to randomly generate D2D links. Fig. 5.3 illustrates the geographical map, the temporal variation in the number of D2D links, and the corresponding average communication duration. At each time-step, the pairing process consists of three steps:

1. **Update states of paired entities:** If the distance between paired entities remains below $d_{\max} = 200$ m and the connection duration is still valid, the connection persists; otherwise, it is terminated.
2. **Add new paired entities:** Unpaired entities are filtered by the distance threshold, then a random sample is selected for pairing. Connection durations are assigned based on an exponential distribution, representing around 10% of the total simulation time.

3. **Generate events and save them:** Communication events (e.g., add, update, or delete) are computed based on changes in communication states, including CSI.

For the rest of this section, we denote the average number of links as $N \approx 10$, $N \approx 25$, and $N \approx 45$ for mobility scenarios involving 50, 100, and 150 users, respectively.

The simulation employs the ITU-R P.1411 ITU-R (2009) short-range outdoor channel model at a carrier frequency of 2.4 GHz, with an antenna height of 1.5 m, an antenna gain of 2.5 dBi, and a maximum transmit power of 4 dBm. Path loss is assumed to have an exponent of 2.0–3.0, and shadowing is modeled by a normal distribution (in dB) with $\sigma_{sh} = 4$–12 dB. The background noise spectral density is -169 dBm/Hz. Fast fading follows a Rayleigh distribution with zero mean and unit variance, and we employ a sum-of-sinusoids approach with 6–20 clusters to model Doppler effects. A CSI estimation error standard deviation of $\sigma_e = 0.1$–0.2 is introduced, with correlation coefficients of $\rho_c = 0.9$ for the fading process and $\rho_e = 0.9$ for the error process. The system comprises $K = 5$ RBs with $L_{\max} = 1$, unless stated otherwise, each of 500 Hz bandwidth, yielding roughly -142 dBm of noise per RB. At pedestrian or vehicular speeds (e.g., 1–7 m/s), the channel coherence time is on the order of a few milliseconds, which allows time-slot durations of approximately 1–10 ms to ensure quasi-static fading within each slot. Unless otherwise stated, the model's node memory, graph memory, temporal encoding, message, and embedding dimensions are all set to 128, and the temporal graph embedding uses three layers ($P = 3$). A history window of $T_h = 10$ time-steps is maintained. Every MLP in the network consists of four layers, each with a hidden size of 64, and includes batch normalization with LeakyReLU as the activation function. The implementation is open-sourced and publicly available at https://github.com/mahermarwani/GNN-CTDG-RRM.

5.4.2 Benchmarks

We evaluate our model against four distinct methodologies, each employing a unique approach to the problem. These benchmarks include a Genetic Algorithm (GA) and techniques based on CNN, DNN, and a memoryless GNN. The specifics of these approaches are as follows:

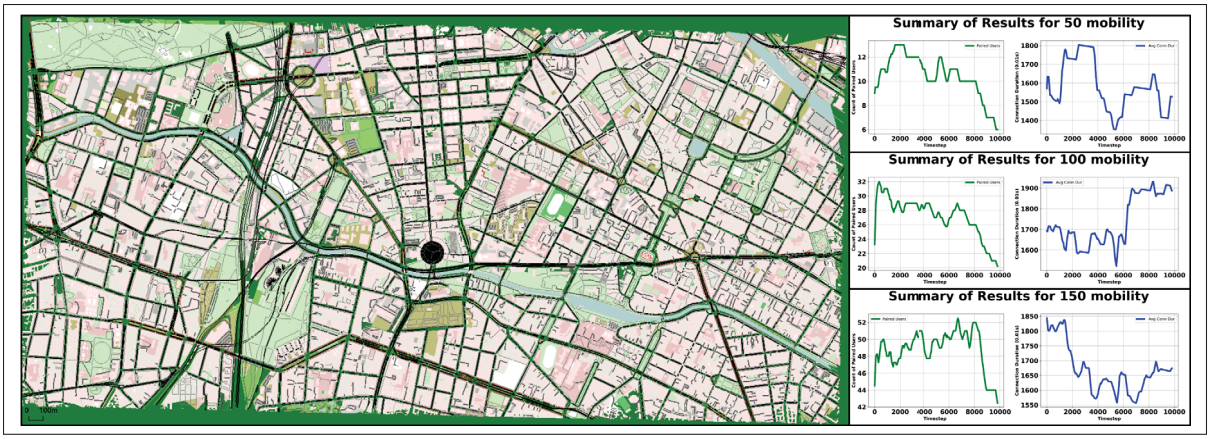


Figure 5.3 Left: geographical map used for the simulation, including roads and pedestrian pathways. Right: results for three mobility scales (50, 100, and 150 entities). Each row depicts the number of paired users and the average connection duration over 10,000 time-steps, illustrating how connectivity evolves over time

- **GA** Sun *et al.* (2021): Employs a *Quantity-Weight Adaptive Salp Swarm Algorithm* for D2D optimization. This evolutionary algorithm explores the solution space through adaptive weight adjustments, enabling efficient convergence toward high-quality resource allocation strategies.
- **CNN** Li *et al.* (2022): Utilizes a CNN architecture to emulate traditional optimization methods. It processes CSI to generate power and RB allocation outputs.
- **DNN** Lee & Schober (2022): Applies two DNNs for power control and RB allocation. The training process relies on brute-force simulated data, with performance closely tied to network size, necessitating size-dependent supervised training for effective generalization.
- **GNN** Marwani & Kaddoum (2024b): Implements a snapshot GNN-based policy for joint RB and power management. The model iteratively maximizes the mean network throughput by capturing spatial dependencies between nodes. It employs a hybrid learning strategy that combines supervised learning for initial policy training with unsupervised learning for iterative performance refinement.

5.4.3 Convergence Analysis

5.4.3.1 Supervised pre-training

Following Section III-C, separate simulations are conducted to generate the training dataset. Fig. 5.4 shows the training and test losses over 300 epochs using a learning rate of 1×10^{-3} with the Adam optimizer. Both losses decrease sharply during the initial epochs, indicating effective early learning, and then stabilize with minor fluctuations. The test loss remains slightly lower than the training loss, indicating good generalization and no significant overfitting. While the loss could be further reduced, the current convergence offers a strong balance and helps prevent overfitting. This training phase primarily establishes a strong initialization. As demonstrated later, retraining for each time-slot significantly boosts performance, making this initial training crucial for efficient fine-tuning.

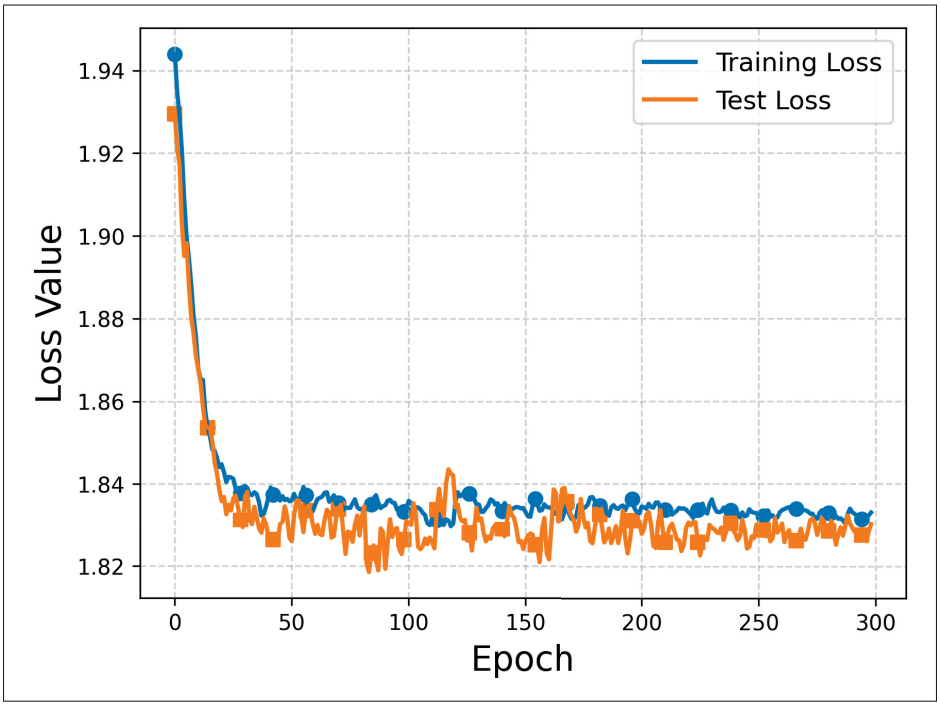
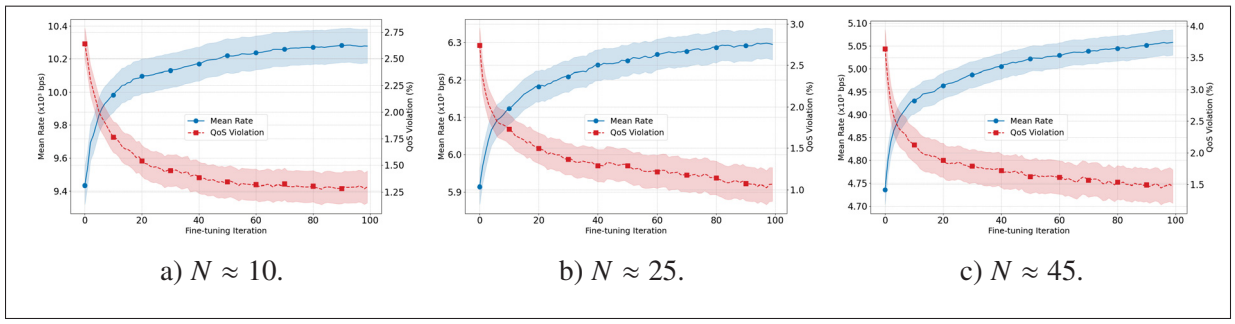


Figure 5.4 Training and test loss variation over epochs

5.4.3.2 Deployment inference

The three plots in Fig. 5.5 show the convergence of the fine-tuning process over 10,000 time-steps, tracking the mean rate ($\times 10^3$ bps) and QoS violation percentage across 100 iterations for network sizes $N \approx 10$, $N \approx 25$, and $N \approx 45$. In all cases, the mean rate increases and stabilizes after 60 iterations, while QoS violations decrease, indicating effective optimization. Confidence intervals highlight variability across iterations. Although the final mean rate decreases for larger networks due to higher complexity, QoS violations remain low. These results confirm that per-time-slot fine-tuning during deployment is crucial for dynamic adaptation, ensuring balance between throughput and QoS.



a) $N \approx 10$. b) $N \approx 25$. c) $N \approx 45$.

Figure 5.5 Mean rate and QoS violation over fine-tuning iterations for different network size scenarios with $r_{\min} = 7 \times 10^3$ bps

In Fig. 5.6, we compare three emphases: high λ (constraint penalty), high λ_1 (max–min), and high λ_2 (throughput), all from the same initialization. High λ reduces QoS violations the fastest, but slows early rate gains; high λ_1 provides a milder trend with steadier worst-link improvement; finally, high λ_2 accelerates early rate gains, but more slowly reduces violations. After the transient, all settings reach similar operating points, supporting the robustness and practicality of uniform summation (Fig. 5.5) as a balanced default.

5.4.3.3 Sensitivity Analysis

We evaluate TGNN's robustness to offline label quality in a small-scale setting ($N \approx 10$). Five PYMOO datasets (5k, 10k, 20k, 25k, and 30k generations) provide labels of progressively higher

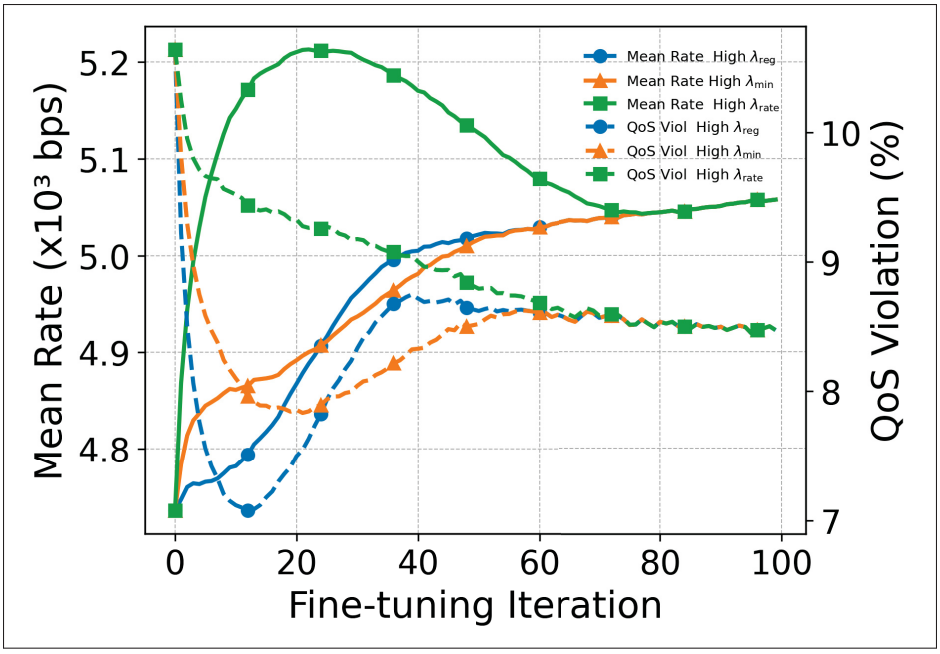


Figure 5.6 Mean rate and QoS violation over fine-tuning iterations for different loss-weighting strategies

Table 5.2 TGNN sensitivity to offline label quality ($N \approx 10$). Values are measured after deployment fine-tuning

PYMOO Iterations	**Mean Rate** ($\times 10^3$ **bps**)	**QoS Violation (%)**	**Convergence (iter)**
5k	13.9 ± 0.35	2.10 ± 0.40	72 ± 3
10k	14.2 ± 0.30	1.80 ± 0.35	66 ± 3
20k	14.4 ± 0.28	1.60 ± 0.30	61 ± 4
25k	14.5 ± 0.30	1.50 ± 0.30	59 ± 3
30k	14.5 ± 0.30	1.50 ± 0.30	59 ± 2
None (no pretrain)	14.3 ± 0.32	1.70 ± 0.35	95 ± 8

quality. After supervised pretraining, we report the time-averaged fine-tuning mean rate, QoS violation, and the number of iterations to converge. The results show that final performance is essentially insensitive to label quality: after fine-tuning, the mean rate remains ≈ 14.3–14.5×10^3 bps with QoS $< 2\%$. The main effect of pretraining is faster convergence (≈ 59–72 iterations vs. ≈ 95 without pretraining). Hence, PYMOO labels serve as an efficient warm start.

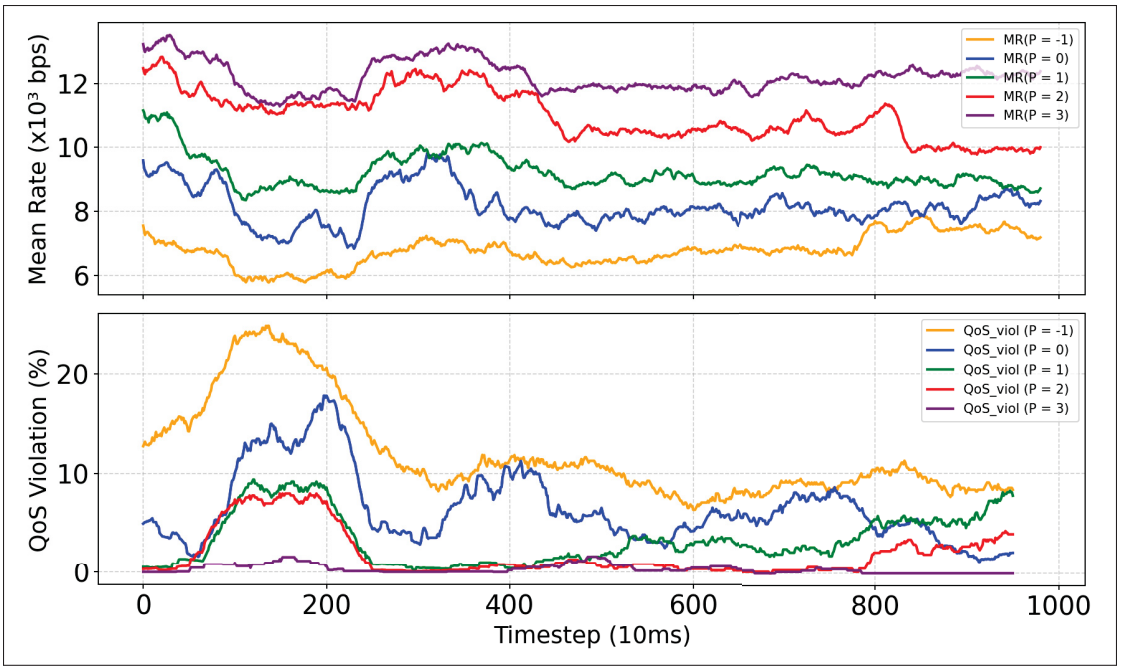


Figure 5.7 Ablation study of mean rate ($\times 10^3$ bps) and QoS violation (%) for different embedding strategies

5.4.4 Ablation study

Fig. 5.7 shows the impact of embedding strategies on the mean rate and QoS violation over 1,000 time-steps for temporal embedding levels $P \in \{-1, 0, 1, 2, 3\}$. $P = -1$ **uses only memory**, yielding the lowest mean rate and highest QoS violations due to the lack of contextual information about temporal patterns. $P = 0$ adds an **initial embedding**, improving performance by providing static context but still resulting in notable QoS violations. $P = 1, 2, 3$ incorporate **temporal embeddings**, enabling the model to capture time-varying interactions, leading to higher throughput and minimal QoS violations, with $P = 3$ performing best. These results highlight the importance of temporal embeddings for modeling temporal dependencies, boosting throughput, and maintaining QoS.

5.4.5 Impact of the Number of Links

Table 5.3 compares five models under different network sizes, including $N \approx 10$, $N \approx 25$, and $N \approx 45$. Performance is evaluated using the mean rate ($\times 10^3$ bps) and QoS violation (%), with a minimum-rate constraint of $r_{\min} = 10 \times 10^3$ bps applied across all scenarios to ensure fairness. For DNN and CNN, separate models are trained for each N, as these architectures require fixed-size inputs. By contrast, GNN and TGNN are trained once and evaluated across all N, leveraging their inherent size invariance TGNN consistently outperforms all benchmarks, achieving the highest mean rate and the lowest QoS violation across all network sizes. For $N \approx 10$, TGNN attains 14.5 ± 0.30 ($\times 10^3$ bps) with 1.50 ± 0.30% QoS violation, indicating efficient resource allocation in less complex environments. When $N \approx 45$, TGNN maintains superior performance at 7.30 ± 0.30 ($\times 10^3$ bps) with 29.0 ± 0.50% QoS violation, outperforming the closest competitor (GNN) by a notable margin. The increasing performance gap with network density highlights TGNN's scalability and adaptability. Unlike memoryless GNNs, TGNN's memory-aware architecture—featuring node-wise and graph-wise memory—effectively captures long-term temporal dependencies, enabling dynamic adjustments in high-interference environments. In contrast, GA and DNN exhibit higher QoS violations due to limited adaptability; the GA's evolutionary search and the DNN's static training are less effective under dynamic conditions. The CNN, while improving over GA and DNN, underperforms TGNN due to its limited ability to capture evolving temporal dynamics. The snapshot GNN captures spatial relationships but lacks temporal memory, resulting in degraded performance as network complexity increases. Overall, these results illustrate TGNN's ability to balance throughput and QoS, ensuring that more links satisfy the QoS constraint even in dense networks. TGNN's temporal attention mechanisms and memory modules allow it to sustain high performance without retraining, demonstrating practical scalability for future wireless systems with higher user densities and stronger dynamic interactions.

Next, Table 5.4 reports the online inference time (ms) for GA, DNN, CNN, GNN, and the proposed TGNN across $N \approx 10$, $N \approx 25$, and $N \approx 45$, evaluated on an Intel® Xeon® W-1270 CPU. Only inference time is considered, as training can be performed offline. The DNN and

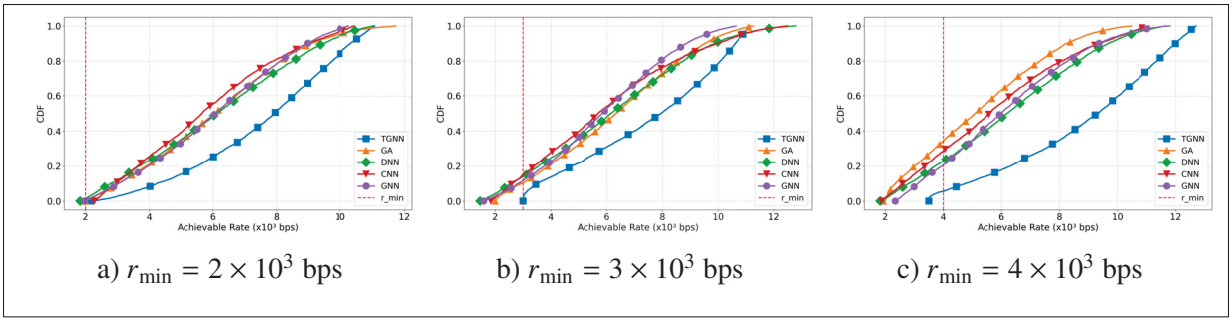
Table 5.3 Comparative analysis across different network sizes with mean rate ($\times 10^3$ bps) and QoS violation (%)

Benchmark Solutions	$N \approx 10$		$N \approx 25$		$N \approx 45$	
	Mean Rate	**QoS**	**Mean Rate**	**QoS**	**Mean Rate**	**QoS**
GA	12.8 ± 0.60	6.50 ± 0.20	7.30 ± 0.30	27.5 ± 2.00	6.00 ± 0.25	40.0 ± 2.50
DNN	13.1 ± 0.15	5.20 ± 0.15	7.60 ± 0.15	24.0 ± 0.60	6.30 ± 0.15	37.5 ± 0.20
CNN	13.5 ± 0.20	4.00 ± 0.20	7.90 ± 0.20	21.5 ± 0.70	6.60 ± 0.20	35.0 ± 0.30
GNN	13.9 ± 0.25	2.80 ± 0.25	8.20 ± 0.25	19.0 ± 0.80	6.90 ± 0.25	32.0 ± 0.40
TGNN	**14.5 ± 0.30**	**1.50 ± 0.30**	**8.60 ± 0.30**	**16.0 ± 0.90**	**7.30 ± 0.30**	**29.0 ± 0.50**

Table 5.4 Execution time (ms) across different mobilities

Benchmarking Solutions	$N \approx 10$	$N \approx 25$	$N \approx 45$
	Time (ms)	**Time (ms)**	**Time (ms)**
GA	242 ± 2.50	401 ± 1.20	678 ± 2.00
CNN	1.07 ± 0.03	1.15 ± 0.02	1.15 ± 0.03
DNN	1.02 ± 0.01	1.02 ± 0.01	1.02 ± 0.01
GNN	43.29 ± 2.73	45.35 ± 2.82	50.13 ± 3.01
TGNN	**17.84 ± 0.53**	**19.34 ± 0.61**	**21.05 ± 0.89**

CNN achieve the fastest inference ($\approx$ 1 ms) due to their non-iterative architectures, though at the expense of adaptability. The GA and GNN, requiring iterative optimization for convergence (stabilized over 20 iterations), exhibit higher execution times, up to 678 ms (GA) and 50.13 ms (GNN) at $N \approx 45$. TGNN achieves a balanced trade-off, with execution times increasing modestly from 17.84 ms to 21.05 ms as N increases, while maintaining superior adaptability. TGNN's memory-aware architecture enables faster convergence than GA and GNN, making it suitable for fast-fading environments where coherence time is low. With improved hardware (e.g., GPUs), TGNN could deliver even faster real-time performance. In summary, TGNN combines low latency and high adaptability, outperforming traditional and DL benchmarks in scalability and real-time applicability for dynamic wireless networks.



a) $r_{\min} = 2 \times 10^3$ bps b) $r_{\min} = 3 \times 10^3$ bps c) $r_{\min} = 4 \times 10^3$ bps

Figure 5.8 CDF of user rates for different minimum-rate constraints $r_{\min}$ with $N \approx 25$

5.4.6 Impact of QoS Constraints on Performance

The CDF plots in Fig. 5.8 compare achievable user rates for TGNN, GA, DNN, CNN, and GNN under minimum-rate constraints $r_{\min} \in \{2, 3, 4\} \times 10^3$ bps with $N \approx 25$. TGNN consistently outperforms all models, with its CDF curve shifted rightward, indicating a higher proportion of users achieving rates above $r_{\min}$. As $r_{\min}$ increases, all models experience a leftward shift, with TGNN exhibiting the smallest shift, demonstrating superior adaptability to stricter QoS requirements. Unlike GA, which struggles as constraints become more stringent, and DNN/CNN, which lack adaptability due to static architectures, TGNN's memory-aware design captures temporal dynamics, yielding a more favorable rate distribution and improved QoS compliance. The snapshot GNN, though spatially aware, falls short due to the absence of temporal memory. Overall, TGNN offers the best trade-off between throughput, fairness, and QoS compliance, supporting its scalability and real-time applicability in dynamic wireless networks.

5.4.7 Impact of Imperfect CSI

Table 5.5 shows that all models experience performance degradation as the channel estimation error (σ_e) increases, with TGNN consistently outperforming the rest. To evaluate robustness, we start from clean CSI and progressively introduce estimation error by varying σ_e (lower σ_e indicates higher CSI accuracy). For each σ_e, power and RB-allocation decisions are computed from the noisy CSI, while the resulting link rates are evaluated using the clean CSI, providing a direct measure of resilience to estimation errors. At $\sigma_e = 1.00$, TGNN maintains

Table 5.5 Performance under imperfect CSI: mean rate ($\times 10^3$ bps) and QoS violation (%) vs. σ_e ($N \approx 25$, $r_{\min} = 10^3$ bps)

σ_e	0.00		0.33	
Metrics	Mean Rate	QoS Viol.	Mean Rate	QoS Viol.
GA	8.10 ± 0.20	17.5 ± 2.50	7.50 ± 0.30	22.0 ± 2.80
DNN	7.90 ± 0.17	23.3 ± 0.66	7.30 ± 0.28	27.5 ± 1.80
CNN	8.90 ± 0.20	11.5 ± 0.10	8.30 ± 0.25	16.0 ± 0.80
GNN	9.10 ± 0.23	12.0 ± 0.60	8.60 ± 0.27	14.5 ± 1.00
TGNN	**9.60 ± 0.21**	**8.5 ± 1.93**	**9.10 ± 0.24**	**10.0 ± 1.20**

σ_e	0.66		1.00	
Metrics	Mean Rate	QoS Viol.	Mean Rate	QoS Viol.
GA	6.80 ± 0.35	30.5 ± 3.00	5.90 ± 0.40	38.0 ± 3.20
DNN	6.60 ± 0.33	34.0 ± 2.50	5.80 ± 0.38	41.5 ± 3.00
CNN	7.50 ± 0.30	22.5 ± 1.20	6.40 ± 0.40	30.0 ± 2.00
GNN	7.80 ± 0.32	20.0 ± 1.50	6.90 ± 0.38	27.0 ± 2.20
TGNN	**8.50 ± 0.28**	**15.5 ± 1.80**	**7.60 ± 0.33**	**22.5 ± 2.50**

the highest mean rate (7.60×10^3 bps) and the lowest QoS violation (22.5%), outperforming GNN (6.90×10^3 bps, 27.0%) and substantially surpassing DNN and CNN, which exhibit higher QoS violations due to their static architectures. GA performs the worst, with QoS violations reaching 38.0%, reflecting its limited effectiveness in dynamic environments. TGNN's memory-aware temporal architecture, featuring temporal embeddings and attention mechanisms, enables dynamic adaptation even under severe CSI uncertainty. By capturing long-term temporal dependencies and retaining historical context, TGNN achieves greater performance stability than the snapshot GNN, which lacks temporal memory. The more gradual performance decline of TGNN, compared to the steeper drops observed in other models, underscores its robustness and scalability.

5.4.8 Extension to UAV and Aerial Mobility Patterns

To evaluate TGNN under diverse 3D dynamics, we extend the evaluation to aerial mobility models commonly used in UAV-network research. Following the taxonomy in Mowla, Rahman & Ahmad

Table 5.6 Comparative performance on four 3D UAV mobility models: mean rate ($\times 10^3$ bps) and QoS violation (%)

Mobility	**UAV–RWP**		**UAV–GM**	
Method	**Mean Rate**	**QoS Viol**	**Mean Rate**	**QoS Viol**
GA	6.20 ± 0.30	39.0 ± 2.5	6.10 ± 0.35	37.0 ± 2.8
DNN	6.40 ± 0.25	36.5 ± 1.8	6.30 ± 0.30	35.0 ± 1.9
CNN	6.60 ± 0.20	34.0 ± 1.5	6.50 ± 0.25	32.5 ± 1.6
GNN	6.70 ± 0.28	35.5 ± 0.7	6.60 ± 0.25	33.0 ± 0.9
TGNN	**7.10 ± 0.25**	**32.0 ± 0.6**	**7.05 ± 0.28**	**30.5 ± 0.7**

Mobility	**UAV–SDC**		**UAV–FS**	
Method	**Mean Rate**	**QoS Viol**	**Mean Rate**	**QoS Viol**
GA	6.00 ± 0.40	36.5 ± 2.4	5.90 ± 0.45	38.0 ± 2.6
DNN	6.20 ± 0.28	34.5 ± 1.7	6.10 ± 0.30	35.5 ± 1.8
CNN	6.40 ± 0.25	32.0 ± 1.4	6.30 ± 0.25	33.0 ± 1.5
GNN	6.50 ± 0.25	32.0 ± 0.8	6.40 ± 0.25	33.5 ± 0.9
TGNN	**6.95 ± 0.25**	**30.0 ± 0.6**	**6.90 ± 0.27**	**31.0 ± 0.7**

(2019), we consider four representative models: Random Waypoint (RWP), Gauss–Markov (GM), Semi-Deterministic Circular (SDC), and Flocking/Swarm (FS). These span uncoordinated stochastic motion, temporally correlated trajectories, deterministic patrolling/orbiting, and cooperative group behaviors. For fairness, we retain the same system parameters, TGNN hyperparameters, and power/RB settings as in the ground-based experiments. Each mobility model is simulated for 10,000 time-steps with ≈ 25 UAVs over a $2\,\text{km} \times 2\,\text{km} \times 0.3\,\text{km}$ region. The results (Table 5.6) show TGNN's resilience to larger Doppler spread and faster topology variation: across all aerial models, TGNN achieves the highest mean rate and the lowest QoS violation. The memory-aware temporal embeddings enable adaptation to unsteady 3D motion without retraining, supporting scalability to heterogeneous aerial networks.

5.4.9 Extension to Multi-Antenna Evaluation

We evaluate TGNN under multi-antenna propagation by extending the channel to multiple Tx/Rx antennas while preserving the overall problem formulation and learning architecture.

We adopt an effective scalar channel view, projecting each MIMO link onto a single-stream SISO-equivalent gain capturing the joint transmit–receive spatial effect Tse & Viswanath (2005). For a transmitter–receiver pair $(j \to i)$, RB k, and time t,

$$g_{ijk}^{\text{eff}}(t) = \left|\mathbf{w}_{i,k}^{H}(t)\, \mathbf{H}_{ij,k}(t)\, \mathbf{v}_{j,k}(t)\right|^2, \tag{5.27}$$

where $\mathbf{H}_{ij,k}(t) \in \mathbb{C}^{N_r \times N_t}$ is the MIMO channel, while $\mathbf{v}_{j,k}(t), \mathbf{w}_{i,k}(t)$ are unit norm precoder and combiner vectors (e.g., MRT/MRC or dominant-eigenmode) Tse & Viswanath (2005). This mapping integrates MIMO into the TGNN pipeline without changing the rate expression, training protocol, or optimization variables (power and RB assignment), which is standard in learning-based RRM where beamforming is handled analytically and learning focuses on scheduling/power Shen *et al.* (2021). For channel generation, we use a temporally correlated Rayleigh–Kronecker model as shown below:

$$\mathbf{H}_{ij,k}(t) = \mathbf{R}_{r,ij}^{1/2}\, \mathbf{W}_{ij,k}(t)\, \mathbf{R}_{t,ij}^{1/2}, \tag{5.28}$$

$$\mathbf{W}_{ij,k}(t+\Delta t) = \rho_c\, \mathbf{W}_{ij,k}(t) + \sqrt{1-\rho_c^2}\, \mathbf{Z}_{ij,k}(t), \tag{5.29}$$

with i.i.d. $\mathcal{CN}(0,1)$ entries and exponential Tx/Rx correlation. We fix the topology at $N \approx 25$ D2D links and vary $(N_t, N_r) \in \{(1,1), (2,2), (4,4), (8,8)\}$ under identical traffic/mobility and QoS settings. For single-stream transmission per link, the transmitter employs a dominant-eigenmode (SVD) precoder and the receiver applies MRC; effective gains are computed shown as in Table 5.7, and all learning components (including TGNN) remain unchanged.

With $N \approx 25$ and analytical SVD/MRC beamforming, increasing (N_t, N_r) yields monotonic gains in mean rate and reductions in QoS violations across all methods, reflecting array/combining benefits captured by $g_{ijk}^{\text{eff}}(t)$. The results reveal that TGNN consistently outperforms GA, DNN, CNN, and snapshot GNN at every antenna size, with a slight widening of the TGNN–GNN gap as (N_t, N_r) grows, suggesting that temporal memory and event-driven updates become more valuable with the strengthening of spatial links and intensification of interference patterns.

Table 5.7 Performance at $N \approx 25$ vs. (N_t, N_r) (SVD+MRC) Mean Rate ($\times 10^3$ bps) and QoS Violation (%)

(N_t, N_r)	(1,1)		(2,2)	
Metrics	Mean Rate	QoS Viol.	Mean Rate	QoS Viol.
GA	7.30 ± 0.30	27.5 ± 2.0	7.60 ± 0.30	25.8 ± 1.9
DNN	7.60 ± 0.15	24.0 ± 0.6	7.85 ± 0.18	22.8 ± 0.6
CNN	7.90 ± 0.20	21.5 ± 0.7	8.15 ± 0.20	20.5 ± 0.7
GNN	8.20 ± 0.25	19.0 ± 0.8	8.50 ± 0.23	18.0 ± 0.7
TGNN	**8.60 ± 0.30**	**16.0 ± 0.9**	**8.90 ± 0.30**	**15.0 ± 0.8**

(N_t, N_r)	(4,4)		(8,8)	
Metrics	Mean Rate	QoS Viol.	Mean Rate	QoS Viol.
GA	7.90 ± 0.28	24.6 ± 1.8	8.05 ± 0.28	24.0 ± 1.7
DNN	8.05 ± 0.18	21.9 ± 0.6	8.15 ± 0.18	21.5 ± 0.6
CNN	8.30 ± 0.20	19.8 ± 0.6	8.40 ± 0.20	19.3 ± 0.6
GNN	8.70 ± 0.22	17.2 ± 0.7	8.80 ± 0.22	16.9 ± 0.7
TGNN	**9.20 ± 0.28**	**14.3 ± 0.7**	**9.40 ± 0.28**	**13.8 ± 0.7**

Table 5.8 Performance vs. RB budget $L_{\max}$. Mean Rate ($\times 10^3$ bps) and QoS Violation (%)

Benchmark Solutions	$L_{\max} = 2$		$L_{\max} = 3$		$L_{\max} = 5$	
	Mean Rate	**QoS Viol**	**Mean Rate**	**QoS Viol**	**Mean Rate**	**QoS Viol**
GA	9.10 ± 0.30	22.0 ± 1.6	9.85 ± 0.30	21.0 ± 1.5	10.50 ± 0.32	20.2 ± 1.5
DNN	9.25 ± 0.20	19.8 ± 0.6	9.95 ± 0.20	18.9 ± 0.6	10.60 ± 0.22	18.3 ± 0.6
CNN	9.55 ± 0.20	17.8 ± 0.6	10.25 ± 0.22	16.9 ± 0.6	10.95 ± 0.22	16.3 ± 0.6
GNN	9.95 ± 0.23	15.0 ± 0.6	10.70 ± 0.24	14.1 ± 0.6	11.45 ± 0.25	13.6 ± 0.6
TGNN	**10.65 ± 0.28**	**11.8 ± 0.6**	**11.45 ± 0.30**	**10.7 ± 0.6**	**12.25 ± 0.30**	**10.2 ± 0.6**

5.4.10 Extension to Multi-RB Selection

We extend the evaluation to multi-RB assignment while keeping the TGNN encoder unchanged. We fix $N \approx 25$ and vary $L_{\max} \in \{2, 3, 5\}$ under identical traffic/mobility and QoS settings. All baselines are extended analogously via top-$L_{\max}$ selection.

5.5 Conclusion

This paper introduced a TGNN framework for dynamic RRM in wireless networks, specifically for power and channel allocation in D2D communications. TGNN leverages CTDGs to model temporal network evolution and employs a memory-aware architecture with node-wise and graph-wise memory modules, enabling efficient adaptation to dynamic topologies and interference patterns. Extensive benchmarking showed that the proposed TGNN outperforms the GA, DNN, CNN, and memory-less GNN in throughput, QoS compliance, and execution time. It consistently achieved the highest mean rates, lowest QoS violations, and faster convergence balance, demonstrating scalability and real-time applicability. The TGNN also proved robust under imperfect CSI, maintaining superior performance with minimal degradation despite channel estimation errors. This resilience stems from its temporal embeddings and attention-based message aggregation, ensuring dynamic adaptability and fairness even in dense, noisy environments. In conclusion, TGNN offers a robust, scalable, and adaptable solution for next-generation wireless networks, excelling in real-time decision-making and dynamic resource management.

CHAPTER 6

CONCLUSION

Today, Wireless networks are entering an era where dense deployments, heterogeneous services, and aggressive spectrum reuse make interference the dominant limiting factor. In these setting, RRM must continuously adapt power and spectrum decisions under stringent control-loop and computational constraints. This thesis addressed this challenge through a unifying perspective: learning architectures should respect the structure of wireless systems interference topology, the spatial and geometric information embedded in multi-RB CSI, and the time-varying nature of practical deployments.

6.1 Unified Progression and Impact

The first contribution laid down a graph-learning foundation for joint power control and RB allocation in interference-coupled networks. By modeling wireless interactions as interference graphs, the proposed framework exploited permutation-equivariant and size-invariant mappings that scale with the number of links while enabling fast inference suitable for online RRM.

Building on this foundation, the second contribution addressed a key representation limitation for multi-RB systems: namely rich spatial and geometric structure contained in tensorized CSI can be partially lost when collapsed into flat graph features. To preserve this information, a scalable spatial-geometric learning framework, combining complementary tensor and graph representations to extract robust embeddings was introduced. This design was found to improve generalization across network sizes and layouts, strengthen QoS-aware performance under CSI uncertainty, and provide a principled bridge between wireless structure and learning-based optimization.

The third contribution moved beyond snapshot-based decision making to dynamic wireless environments. By introducing an event-based temporal graph learning approach, the thesis captured continuous-time evolution driven by fading, mobility, and sporadic link activity. This

temporal modeling enabled memory-aware allocation decisions that remained stable under network evolution, thus improving robustness and responsiveness in time-varying scenarios.

6.2 Alignment with IMT-2030 (6G) Requirements

The results revealed that the proposed progression directly supports the direction of IMT-2030 systems, where tighter latency budgets and higher reliability requirements increase the value of fixed-cost inference and robust policies. In particular, the thesis emphasizes (i) fast online decision making through amortized learning and constant-cost inference, (ii) robust QoS-aware allocation under imperfect CSI, and (iii) adaptive decision making under mobility and link churn through event-based temporal memory. Collectively, these elements align with next-generation objectives that prioritize ultra-low latency and extreme reliability, while maintaining scalability in dense deployments.

6.3 Strategic Recommendations for Deployment (from Research to Practice)

To translate these contributions into practical RAN control, a natural path is integrating the proposed learning policies into industrial control frameworks that will separate optimization across timescales. For example, an O-RAN-style architecture can host (i) model training, policy guidance, and model lifecycle management at longer timescales, and (ii) near-real-time inference and control applications at shorter timescales. Concretely:

- **Non-real-time layer (model lifecycle):** Training and validating models offline, manage model versions, performing periodic domain adaptation when channel statistics drift, and generating high-level policies (e.g., constraint targets, fairness weights, operating points) for online controllers.
- **Near-real-time layer (fast control):** Using the GNN/TGNN inference as an online controller that consumes streaming measurements, produces per-interval power/RB decisions, and (when enabled) performs lightweight fine-tuning within a bounded iteration budget.
- **Event interface for TGNN:** Construction of CTDG events (add/update/delete) from routine RAN telemetry (link activation, CSI updates, mobility-induced topology changes), thus

enabling the temporal model to maintain node-wise and graph-wise memory without requiring a full re-optimization from scratch.

This split reflects the philosophy adopted in this thesis: namely, shifting heavy computation offline and retaining fast, repeatable online inference, while keeping a controlled mechanism for adaptation when the environment evolves.

6.4 Signaling Overhead vs. Computational Gain: Explicit Trade-offs

A practical deployment must balance computational savings of fast inference against the signaling overhead required to maintain sufficiently fresh inputs and temporal state. The main overhead drivers include the following: pilot/measurement resources to estimate CSI, reporting latency and quantization for centralized decision making, and (for TGNN) the bookkeeping needed to maintain node identities, event streams, and memory updates. The main gains are: avoiding repeated solver iterations per decision epoch, improved stability under dynamics via memory, and better generalization across network sizes without retraining.

This motivates system-level design choices that will trade signaling for performance in a controlled way, including: (i) sparse event construction (track only dominant interferers, rather than all cross-links), (ii) compressed or quantized memory states, (iii) hierarchical update rates (update global memory less frequently than local memory), and (iv) hybrid execution where parts of event processing occur locally, while the controller maintains only aggregated state. Making these trade-offs explicit is essential for deployment-oriented RRM, especially under dense networks and mobility.

6.5 Outlook and Research Directions

Several high-impact extensions remain. Moving from single-cell interference-coupled links to multi-cell and cell-free settings will require distributed execution and careful signaling design. Handling partial observability (i.e., Limited pilots, feedback delays, missing measurements) motivates uncertainty-aware learning and robust constraint enforcement beyond point CSI.

Finally, strengthening reliability for safety- and mission-critical services calls for constrained learning with stronger guarantees, for instance via differentiable optimization layers, projection operators enforcing feasibility by construction, and certification mechanisms that bound QoS violations under uncertainty. Together, these directions would complement the contributions of this thesis and help to translate structure-aligned graph learning into practical, trustworthy, and scalable RRM for next-generation wireless networks.

BIBLIOGRAPHY

A, Y., PS, P., P, A. & N, S. (2023). Joint power allocation and channel assignment for device-to-device communication using the Hungarian model and enhanced hybrid Red Fox-Harris Hawks Optimization. *International Journal of Communication Systems*, 36(7), e5425. doi: https://doi.org/10.1002/dac.5425.

Abdallah, A., Mansour, M. M. & Chehab, A. (2018). Power Control and Channel Allocation for D2D Underlaid Cellular Networks. *IEEE Transactions on Communications*, 66(7), 3217-3234. doi: 10.1109/TCOMM.2018.2812731.

Akhtar, T., Tselios, C. & Politis, I. (2021). Radio resource management: approaches and implementations from 4G to 5G and beyond. *Wireless Networks*, 27(1), 693–734.

Alsabah, M., Naser, M. A., Mahmmod, B. M., Abdulhussain, S. H., Eissa, M. R., Al-Baidhani, A., Noordin, N. K., Sait, S. M., Al-Utaibi, K. A. & Hashim, F. (2021). 6G Wireless Communications Networks: A Comprehensive Survey. *IEEE Access*, 9, 148191-148243. doi: 10.1109/ACCESS.2021.3124812.

Alwis, C. D., Kalla, A., Pham, Q.-V., Kumar, P., Dev, K., Hwang, W.-J. & Liyanage, M. (2021). Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research. *IEEE Open Journal of the Communications Society*, 2, 836-886. doi: 10.1109/OJCOMS.2021.3071496.

Blank, J. & Deb, K. (2020). Pymoo: Multi-Objective Optimization in Python. *IEEE Access*, 8, 89497-89509. doi: 10.1109/ACCESS.2020.2990567.

Chen, J. & Chen, H. (2021). Edge-Featured Graph Attention Network.

Chen, L., Zhu, J. & Evans, J. (2024). GNN-Based Joint Channel and Power Allocation in Heterogeneous Wireless Networks. *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 233–238. doi: 10.1109/ICCWorkshops59551.2024.10615380.

Chen, L., She, C., Zhu, J. & Evans, J. (2025). Graph Neural Networks for Resource Allocation in Interference-Limited Multi-Channel Wireless Networks with QoS Constraints. *arXiv preprint arXiv:2509.06395*. doi: 10.48550/arXiv.2509.06395.

Chen, T., Zhang, X., You, M., Zheng, G. & Lambotharan, S. (2022). A GNN-Based Supervised Learning Framework for Resource Allocation in Wireless IoT Networks. *IEEE Internet of Things Journal*, 9(3), 1712-1724. doi: 10.1109/JIOT.2021.3091551.

Cheng, P., Chen, G. & Han, Z. (2023). Graph Neural Networks based Resource Allocation in Heterogeneous Wireless Networks. *Proceedings of the 7th International Conference on Intelligent Information Processing*, (ICIIP '22). doi: 10.1145/3570236.3570293.

Chowdhury, A., Verma, G., Rao, C., Swami, A. & Segarra, S. (2021). Efficient Power Allocation Using Graph Neural Networks and Deep Algorithm Unfolding. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4725-4729. doi: 10.1109/ICASSP39728.2021.9415106.

Cui, W., Shen, K. & Yu, W. (2018). Spatial Deep Learning for Wireless Scheduling. *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-6. doi: 10.1109/GLOCOM.2018.8647453.

Dai, Y., Lyu, L., Cheng, N., Sheng, M., Liu, J., Wang, X., Cui, S., Cai, L. & Shen, X. (2025a). A Survey of Graph-Based Resource Management in Wireless Networks—Part I: Optimization Approaches. *IEEE Transactions on Cognitive Communications and Networking*, 11(4), 2078–2100. doi: 10.1109/TCCN.2024.3508783.

Dai, Y., Lyu, L., Cheng, N., Sheng, M., Liu, J., Wang, X., Cui, S., Cai, L. & Shen, X. (2025b). A Survey of Graph-Based Resource Management in Wireless Networks—Part II: Learning Approaches. *IEEE Transactions on Cognitive Communications and Networking*, 11(4), 2101–2122. doi: 10.1109/TCCN.2024.3508777.

Dey, R. & Salem, F. M. (2017). Gate-variants of Gated Recurrent Unit (GRU) neural networks. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, pp. 1597-1600. doi: 10.1109/MWSCAS.2017.8053243.

Dominic, S. & Jacob, L. (2020). Joint resource block and power allocation through distributed learning for energy efficient underlay D2D communication with rate guarantee. *Computer Communications*, 159, 26-36. doi: 10.1016/j.comcom.2020.05.005.

Duan, C. & Zhang, T. (2020). Two-Stream Convolutional Neural Network Based on Gradient Image for Aluminum Profile Surface Defects Classification and Recognition. *IEEE Access*, 8, 172152-172165. doi: 10.1109/ACCESS.2020.3025165.

Eisen, M. & Ribeiro, A. (2020). Optimal Wireless Resource Allocation With Random Edge Graph Neural Networks. *IEEE Transactions on Signal Processing*, 68, 2977–2991. doi: 10.1109/tsp.2020.2988255.

Garcia Camargo, R., Wang, Z., NaderiAlizadeh, N. & Ribeiro, A. (2025). Wireless Link Scheduling with State-Augmented Graph Neural Networks. *arXiv preprint arXiv:2505.07598*. doi: 10.48550/arXiv.2505.07598.

Ghasemi, A. & Pishro-Nik, H. (2024). Tiny Graph Neural Networks for Radio Resource Management. *arXiv preprint arXiv:2403.19143*. doi: 10.48550/arXiv.2403.19143. Presented at the tinyML Research Symposium 2024.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. (2017, 06–11 Aug). Neural Message Passing for Quantum Chemistry. *Proceedings of the 34th International Conference on Machine Learning*, 70(Proceedings of Machine Learning Research), 1263–1272. Retrieved from: https://proceedings.mlr.press/v70/gilmer17a.html.

Gu, Y., She, C., Quan, Z., Qiu, C. & Xu, X. (2023). Graph Neural Networks for Distributed Power Allocation in Wireless Networks: Aggregation Over-the-Air. *IEEE Transactions on Wireless Communications*, 22(11), 7551-7564. doi: 10.1109/TWC.2023.3253126.

Guo, J. & Yang, C. (2021). Learning Power Control for Cellular Systems with Heterogeneous Graph Neural Network. *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1-6. doi: 10.1109/WCNC49053.2021.9417260.

Gupta, S., Patel, R., Gupta, R., Tanwar, S. & Patel, N. (2022). A Survey on Resource Allocation Schemes in Device-to-Device Communication. *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, pp. 140-145. doi: 10.1109/Confluence52989.2022.9734183.

Hamilton, W. L., Ying, R. & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *CoRR*, abs/1706.02216. Retrieved from: http://arxiv.org/abs/1706.02216.

Han, L., Shi, X. & Lu, T. (2025). Graph Neural Networks for Minimizing Worst-Case Outage Probability in Dense Spectrum-Sharing Networks. *ICT Express*, 11(6), 1226–1231. doi: 10.1016/j.icte.2025.11.014.

He, C., Tian, C., Zhang, C., Feng, D., Pan, C. & Zheng, F.-C. (2020a). Energy Efficiency Optimization for Distributed Antenna Systems With D2D Communications Under Channel Uncertainty. *IEEE Transactions on Green Communications and Networking*, 4(4), 1037-1047. doi: 10.1109/TGCN.2020.2997689.

He, S., Xiong, S., Ou, Y., Zhang, J., Wang, J., Huang, Y. & Zhang, Y. (2021). An Overview on the Application of Graph Neural Networks in Wireless Networks. *IEEE Open Journal of the Communications Society*, 2, 2547-2565. doi: 10.1109/OJCOMS.2021.3128637.

He, S., Xiong, S., Zhang, W., Yang, Y., Ren, J. & Huang, Y. (2022). GBLinks: GNN-Based Beam Selection and Link Activation for Ultra-Dense D2D mmWave Networks. *IEEE Transactions on Communications*, 70(5), 3451-3466. doi: 10.1109/TCOMM.2022.3158646.

He, Z., Wang, L., Ye, H., Li, G. Y. & Juang, B.-H. F. (2020b). Resource Allocation based on Graph Neural Networks in Vehicular Communications. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, pp. 1-5. doi: 10.1109/GLOBECOM42002.2020.9322537.

Huang, J., Huang, S., Xing, C.-C. & Qian, Y. (2018). Game-Theoretic Power Control Mechanisms for Device-to-Device Communications Underlaying Cellular System. *IEEE Transactions on Vehicular Technology*, 67(6), 4890-4900. doi: 10.1109/TVT.2018.2800051.

International Telecommunication Union (ITU). [Accessed 2026-01-21]. (2023). IMT towards 2030 and beyond (IMT-2030). Retrieved from: ITU-RWP5DWebpage.

ITU-R. (2009). *Propagation data and prediction methods for the planning of short-range outdoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz* (Report n°P.1411). Retrieved from: https://www.itu.int/rec/R-REC-P.1411.

ITU-R Working Party 5D. (2023). *Framework and overall objectives of the future development of IMT for 2030 and beyond* (Report n°ITU-R M.2160). Retrieved on 2026-01-27 from: https://www.itu.int/en/ITU-R/study-groups/rsg5/rwp5d/imt-2030/Documents/IMT-2030%20Framework_WP%205D%20Management%20Team_WEB%20POST%20v2_11-2023.pdf.

Jameel, F., Hamid, Z., Jabeen, F., Zeadally, S. & Javed, M. A. (2018). A Survey of Device-to-Device Communications: Research Issues and Challenges. *IEEE Communications Surveys & Tutorials*, 20(3), 2133-2168. doi: 10.1109/COMST.2018.2828120.

Kai, C., Xu, L., Zhang, J. & Peng, M. (2018). Joint Uplink and Downlink Resource Allocation for D2D Communication Underlying Cellular Networks. *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1-6. doi: 10.1109/WCSP.2018.8555896.

Kazemi, S. M., Goel, R., Jain, K., Kobyzev, I., Sethi, A., Forsyth, P. & Poupart, P. (2019). Relational Representation Learning for Dynamic (Knowledge) Graphs: A Survey. *CoRR*, abs/1905.11485. Retrieved from: http://arxiv.org/abs/1905.11485.

Ketkar, N. & Moolayil, J. (2021). Automatic Differentiation in Deep Learning. In *Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch* (pp. 133–145). Berkeley, CA: Apress. doi: 10.1007/978-1-4842-5364-9_4.

Kingma, D. P. & Welling, M. (2022). Auto-Encoding Variational Bayes.

Kipf, T. N. & Welling, M. (2017). Semi-Supervised Classification with Graph Convolutional Networks.

Lee, M., Yu, G. & Li, G. Y. (2020). Learning to Branch: Accelerating Resource Allocation in Wireless Networks. *IEEE Transactions on Vehicular Technology*, 69(1), 958-970. doi: 10.1109/TVT.2019.2953724.

Lee, S., Kim, J. & Cho, S. (2019). Resource Allocation for NOMA based D2D System Using Genetic Algorithm with Continuous Pool. *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 705-707. doi: 10.1109/ICTC46691.2019.8939884.

Lee, W. & Schober, R. (2022). Deep Learning-Based Resource Allocation for Device-to-Device Communication. *IEEE Transactions on Wireless Communications*, 21(7), 5235-5250. doi: 10.1109/TWC.2021.3138733.

Lee, W., Kim, M. & Cho, D.-H. (2018). Deep Power Control: Transmit Power Control Scheme Based on Convolutional Neural Network. *IEEE Communications Letters*, 22(6), 1276-1279. doi: 10.1109/LCOMM.2018.2825444.

Li, J., Shen, C. & Zhang, K. (2022). Convolutional neural network-based channel allocation and resource optimization for D2D communications. *2022 6th International Conference on Wireless Communications and Applications (ICWCAPP)*, pp. 109-113. doi: 10.1109/ICWCAPP57292.2022.00034.

Li, Y. & Aghvami, A. H. (2023). Radio Resource Management for Cellular-Connected UAV: A Learning Approach. *IEEE Transactions on Communications*, 71(5), 2784-2800. doi: 10.1109/TCOMM.2023.3262826.

Li, Y., Sellathurai, M., Chu, Z., Xiao, P. & Aghvami, A. H. (2023). DRL-Aided Joint Resource Block and Beamforming Management for Cellular-Connected UAVs. 3045-3050. doi: 10.1109/GLOBECOM54140.2023.10437176.

Li, Y., Madhukumar, A. S., Zheng Hui Ernest, T., Zheng, G., Saad, W. & Hamid Aghvami, A. (2025). Energy-Efficient UAV-Driven Multi-Access Edge Computing: A Distributed Many-Agent Perspective. *IEEE Transactions on Communications*, 73(9), 8405-8420. doi: 10.1109/TCOMM.2025.3552746.

Liang, F., Shen, C., Yu, W. & Wu, F. (2020). Towards Optimal Power Control via Ensembling Deep Neural Networks. *IEEE Transactions on Communications*, 68(3), 1760-1776. doi: 10.1109/TCOMM.2019.2957482.

Lin, T. & Zhu, Y. (2020). Beamforming Design for Large-Scale Antenna Arrays Using Deep Learning. *IEEE Wireless Communications Letters*, 9(1), 103-107. doi: 10.1109/LWC.2019.2943466.

Lin, Z. & Liu, Y. (2019). Joint Uplink and Downlink Transmissions in User-Centric OFDMA Cloud-RAN. *IEEE Transactions on Vehicular Technology*, 68(8), 7776-7788. doi: 10.1109/TVT.2019.2924437.

Liu, S., Wu, Y., Li, L., Liu, X. & Xu, W. (2019). A Two-Stage Energy-Efficient Approach for Joint Power Control and Channel Allocation in D2D Communication. *IEEE Access*, 7, 16940-16951. doi: 10.1109/ACCESS.2019.2894003.

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P. & Wießner, E. (2018). Microscopic Traffic Simulation using SUMO. *2018 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582. doi: 10.1109/ITSC.2018.8569938.

Maas, A. L. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. Retrieved from: https://api.semanticscholar.org/CorpusID:16489696.

Mach, P., Becvar, Z. & Najla, M. (2019). Resource Allocation for D2D Communication With Multiple D2D Pairs Reusing Multiple Channels. *IEEE Wireless Communications Letters*, 8(4), 1008-1011. doi: 10.1109/LWC.2019.2903798.

Marwani, M. & Kaddoum, G. (2024a). Scalable Spatial and Geometric Learning Approach for Joint Power Control and Channel Allocation. *IEEE Transactions on Wireless Communications*, 23(11), 16976-16991. doi: 10.1109/TWC.2024.3449036.

Marwani, M. & Kaddoum, G. (2024b). Graph Neural Networks Approach for Joint Wireless Power Control and Spectrum Allocation. *IEEE Transactions on Machine Learning in Communications and Networking*, 2, 717-732. doi: 10.1109/TMLCN.2024.3408723.

Marwani, M. & Kaddoum, G. (2026). Event-Based Temporal Graph Neural Network for Radio Resource Management. *IEEE Transactions on Wireless Communications*, 1-1. doi: 10.1109/TWC.2026.3654880.

Moussaoui, M., Bertin, E. & Crespi, N. (2022). 5G shortcomings and Beyond-5G/6G requirements. *2022 1st International Conference on 6G Networking (6GNet)*, pp. 1-8. doi: 10.1109/6GNet54646.2022.9830439.

Mowla, M. M., Rahman, M. A. & Ahmad, I. (2019). Assessment of Mobility Models in Unmanned Aerial Vehicle Networks. *2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2)*, pp. 1-4. doi: 10.1109/IC4ME247184.2019.9036678.

NaderiAlizadeh, N., Eisen, M. & Ribeiro, A. (2022). State-Augmented Learnable Algorithms for Resource Management in Wireless Networks. *IEEE Transactions on Signal Processing*, 70, 5898-5912. doi: 10.1109/TSP.2022.3229948.

NaderiAlizadeh, N., Eisen, M. & Ribeiro, A. (2023). Learning Resilient Radio Resource Management Policies With Graph Neural Networks. *IEEE Transactions on Signal Processing*, 71, 995-1009. doi: 10.1109/TSP.2023.3255547.

Najla, M., Becvar, Z. & Mach, P. (2021). Reuse of Multiple Channels by Multiple D2D Pairs in Dedicated Mode: A Game Theoretic Approach. *IEEE Transactions on Wireless Communications*, 20(7), 4313-4327. doi: 10.1109/TWC.2021.3057825.

Nakashima, K., Kamiya, S., Ohtsu, K., Yamamoto, K., Nishio, T. & Morikura, M. (2020). Deep Reinforcement Learning-Based Channel Allocation for Wireless LANs With Graph Convolutional Networks. *IEEE Access*, 8, 31823-31834. doi: 10.1109/ACCESS.2020.2973140.

NGMN Alliance. (2023). *6G Requirements & Design Considerations*.

Nguyen, D. H. N., Le, L. B. & Han, Z. (2016). Optimal uplink and downlink channel assignment in a full-duplex multiuser system. *2016 IEEE International Conference on Communications (ICC)*, pp. 1-6. doi: 10.1109/ICC.2016.7510742.

Nocedal, J. & Wright, S. J. (1999). *Numerical optimization*. Springer.

O-RAN Alliance. (2018). *O-RAN: Towards an Open and Smart RAN*. Retrieved on 2026-01-27 from: https://mediastorage.o-ran.org/white-papers/O-RAN.White-Paper-2018-10.pdf.

Ozpoyraz, B., Dogukan, A. T., Gevez, Y., Altun, U. & Basar, E. (2022). Deep Learning-Aided 6G Wireless Networks: A Comprehensive Survey of Revolutionary PHY Architectures. *IEEE Open Journal of the Communications Society*, 3, 1749-1809. doi: 10.1109/OJCOMS.2022.3210648.

O'Shea, T. & Hoydis, J. (2017). An Introduction to Deep Learning for the Physical Layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4), 563-575. doi: 10.1109/TCCN.2017.2758370.

Pan, Y., Pan, C., Yang, Z. & Chen, M. (2018). Resource Allocation for D2D Communications Underlaying a NOMA-Based Cellular Network. *IEEE Wireless Communications Letters*, 7(1), 130-133. doi: 10.1109/LWC.2017.2759114.

Peng, Y., Guo, J. & Yang, C. (2023). Learning Resource Allocation Policy: Vertex-GNN or Edge-GNN?

Qamar, F., Siddiqui, M. U. A., Hindia, M. N., Hassan, R. & Nguyen, Q. N. (2020). Issues, Challenges, and Research Trends in Spectrum Management: A Comprehensive Overview and New Vision for Designing 6G Networks. *Electronics*, 9(9), Art. no. 9. doi: 10.3390/electronics9091416.

Rahmani, M., Norouzi, S., Chen, J., Ahmadi, H., Braun, T., Chowdhury, K. & Burr, A. G. (2026). Hybrid GNN-Centric Architectures for AI-Native 6G Wireless Networks: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 28, 5678–5712. doi: 10.1109/COMST.2026.3681198.

Ramezani-Kebrya, A., Dong, M., Liang, B., Boudreau, G. & Seyedmehdi, S. H. (2017). Joint Power Optimization for Device-to-Device Communication in Cellular Networks With Interference Control. *IEEE Transactions on Wireless Communications*, 16(8), 5131-5146. doi: 10.1109/TWC.2017.2706259.

Ranasinghe, V., Rajatheva, N. & Latva-aho, M. (2021). Graph Neural Network Based Access Point Selection for Cell-Free Massive MIMO Systems. *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 01-06. doi: 10.1109/GLOBECOM46510.2021.9685221.

Ruby, R., Zhong, S., Yang, H. & Wu, K. (2018). Enhanced Uplink Resource Allocation in Non-Orthogonal Multiple Access Systems. *IEEE Transactions on Wireless Communications*, 17(3), 1432-1444. doi: 10.1109/TWC.2017.2778105.

Ruiz, L., Gama, F. & Ribeiro, A. (2021). Graph Neural Networks: Architectures, Stability, and Transferability. *Proceedings of the IEEE*, 109(5), 660-682. doi: 10.1109/JPROC.2021.3055400.

Shcherbakova, G., Krylov, V., Abakumov, V., Brovkov, V. & Kozina, I. (2011). Sub gradient iterative method for neural networks training. *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems*, 1, 361-364. doi: 10.1109/IDAACS.2011.6072774.

Shehata, M. K., Gasser, S. M., El-Badawy, H. M. & Khedr, M. E. (2015). Optimized dual uplink and downlink resource allocation for multiple class of service in OFDM network. *2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pp. 597-601. doi: 10.1109/ISSPIT.2015.7394407.

Shen, Y., Shi, Y., Zhang, J. & Letaief, K. B. (2020). LORM: Learning to Optimize for Resource Management in Wireless Networks With Few Training Samples. *IEEE Transactions on Wireless Communications*, 19(1), 665-679. doi: 10.1109/TWC.2019.2947591.

Shen, Y., Shi, Y., Zhang, J. & Letaief, K. B. (2021). Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis. *IEEE Journal on Selected Areas in Communications*, 39(1), 101-115. doi: 10.1109/JSAC.2020.3036965.

Shen, Y., Zhang, J., Song, S. H. & Letaief, K. B. (2023). Graph Neural Networks for Wireless Communications: From Theory to Practice. *IEEE Transactions on Wireless Communications*, 22(5), 3554-3569. doi: 10.1109/TWC.2022.3219840.

Shi, Q., Razaviyayn, M., Luo, Z.-Q. & He, C. (2011). An Iteratively Weighted MMSE Approach to Distributed Sum-Utility Maximization for a MIMO Interfering Broadcast Channel. *IEEE Transactions on Signal Processing*, 59(9), 4331-4340. doi: 10.1109/TSP.2011.2147784.

Su, N. & Zhu, Q. (2019). Power Control and Channel Allocation Algorithm for Energy Harvesting D2D Communications. *Algorithms*, 12(5). doi: 10.3390/a12050093.

Sun, E., Qu, H., Yuan, Y., Li, M., Wang, Z. & Chen, D. (2021). A Joint Channel Allocation and Power Control Scheme for D2D Communication in UAV-Based Networks. *Wireless Communications and Mobile Computing*, 2021, 1-15. doi: 10.1155/2021/7400156.

Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X. & Sidiropoulos, N. D. (2018). Learning to Optimize: Training Deep Neural Networks for Interference Management. *IEEE Transactions on Signal Processing*, 66(20), 5438-5453. doi: 10.1109/TSP.2018.2866382.

Sun, J., Wang, Q., Chen, T., Yang, S. & Sun, W. (2023a). *O-RAN Native AI Architecture Description* (Report n°RR-2023-02). Retrieved on 2026-01-27 from: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2023-02-Native%20AI%20Architecture%20Description-v1.2.pdf.

Sun, M., Mei, E., Wang, S. & Jin, Y. (2023b). Joint DDPG and Unsupervised Learning for Channel Allocation and Power Control in Centralized Wireless Cellular Networks. *IEEE Access*, 11, 42191-42203. doi: 10.1109/ACCESS.2023.3270316.

Sun, Y., Xu, D., Ng, D. W. K., Dai, L. & Schober, R. (2019). Optimal 3D-Trajectory Design and Resource Allocation for Solar-Powered UAV Communication Systems. *IEEE Transactions on Communications*, 67(6), 4281-4298. doi: 10.1109/TCOMM.2019.2900630.

Sun, Z. & Nakhai, M. R. (2021). Channel Selection and Power Control for D2D Communication via Online Reinforcement Learning. *ICC 2021 - IEEE International Conference on Communications*, pp. 1-6. doi: 10.1109/ICC42927.2021.9501055.

Tan, J., Liang, Y.-C., Zhang, L. & Feng, G. (2021). Deep Reinforcement Learning for Joint Channel Selection and Power Control in D2D Networks. *IEEE Transactions on Wireless Communications*, 20(2), 1363-1378. doi: 10.1109/TWC.2020.3032991.

Tse, D. & Viswanath, P. (2005). *Fundamentals of Wireless Communication*. Cambridge University Press.

Uslu, Y. B., Hadou, S., Saeedi Bidokhti, S. & Ribeiro, A. (2026). Graph Signal Diffusion Models for Wireless Resource Allocation. *arXiv preprint arXiv:2604.05175*. doi: 10.48550/arXiv.2604.05175.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. & Polosukhin, I. (2017). Attention Is All You Need. *CoRR*, abs/1706.03762. Retrieved from: http://arxiv.org/abs/1706.03762.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. & Bengio, Y. (2018). Graph Attention Networks.

Wang, D., Qin, H., Song, B., Xu, K., Du, X. & Guizani, M. (2021). Joint resource allocation and power control for D2D communication with deep reinforcement learning in MCC. *Physical Communication*, 45, 101262. doi: 10.1016/j.phycom.2020.101262.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y. et al. (2019). Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*.

Wang, Y., Li, Y., Shi, Q. & Wu, Y.-C. (2023). ENGNN: A General Edge-Update Empowered GNN Architecture for Radio Resource Management in Wireless Networks. *IEEE Transactions on Wireless Communications*, 1-1. doi: 10.1109/TWC.2023.3325735.

Wang, Z., Eisen, M. & Ribeiro, A. (2022). Learning Decentralized Wireless Resource Allocations With Graph Neural Networks. *IEEE Transactions on Signal Processing*, 70, 1850-1863. doi: 10.1109/TSP.2022.3163626.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C. & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(1), 4-24. doi: 10.1109/TNNLS.2020.2978386.

Yang, Z.-Y. & Kuo, Y.-W. (2017). Efficient Resource Allocation Algorithm for Overlay D2D Communication. *Computer Networks*, 124, 61-71. doi: 10.1016/j.comnet.2017.06.002.

Yuan, Y., Yang, T., Feng, H. & Hu, B. (2018). An Iterative Matching-Stackelberg Game Model for Channel-Power Allocation in D2D Underlaid Cellular Networks. *IEEE Transactions on Wireless Communications*, 17(11), 7456-7471. doi: 10.1109/TWC.2018.2867474.

Zeiler, M. D. (2012). ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701. Retrieved from: http://arxiv.org/abs/1212.5701.

Zhang, C., Patras, P. & Haddadi, H. (2019). Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys & Tutorials*, 21(3), 2224–2287. doi: 10.1109/COMST.2019.2904897.

Zhang, X., Zhang, Z. & Yang, L. (2021a). Joint User Association and Power Allocation in Heterogeneous Ultra Dense Network via Semi-Supervised Representation Learning. *CoRR*, abs/2103.15367. Retrieved from: https://arxiv.org/abs/2103.15367.

Zhang, X., Zhao, H., Xiong, J., Liu, X., Zhou, L. & Wei, J. (2021b). Scalable Power Control/Beamforming in Heterogeneous Wireless Networks with Graph Neural Networks. *2021 IEEE Global Communications Conference (GLOBECOM)*, pp. 01-06. doi: 10.1109/GLOBECOM46510.2021.9685457.

Zhang, Z. & Tao, M. (2022). Learning-Based Branch-and-Bound for Non-Convex Complex Modulus Constrained Problems With Applications in Wireless Communications. *IEEE Transactions on Wireless Communications*, 21(6), 3752-3763. doi: 10.1109/TWC.2021.3123620.

Zhang, Z., Cui, P. & Zhu, W. (2022). Deep Learning on Graphs: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 34(1), 249-270. doi: 10.1109/TKDE.2020.2981333.

Zhao, Y., Zhao, J., Zhai, W., Sun, S., Niyato, D. & Lam, K. (2021). A Survey of 6G Wireless Communications: Emerging Technologies. *Advances in Information and Communication*, 1363(Advances in Intelligent Systems and Computing). doi: 10.1007/978-3-030-73100-7_12.

Zheng, Y., Yi, L. & Wei, Z. (2025). A Survey of Dynamic Graph Neural Networks. *Frontiers of Computer Science*, 19(6), 196323. doi: 10.1007/s11704-024-3853-2.

Zheng, Z., Chi, Y., Ding, G. & Yu, G. (2022). Deep-Learning-Based Resource Allocation for Time-Sensitive Device-to-Device Networks. *Sensors*, 22(4), 1551. doi: 10.3390/s22041551.

Zhou, J. (2021). Deep Reinforcement Learning for Channel Selection and Power Allocation in D2D Communications. *Journal of Physics: Conference Series*, 2082(1), 012003. doi: 10.1088/1742-6596/2082/1/012003.