Learning Wireless Interference Patterns: Decoupled GNN for Throughput Prediction in Heterogeneous Multi-Hop p-CSMA Networks

Faezeh Dehghan Tarzjani
Dept. of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA
dehghant@usc.edu

Bhaskar Krishnamachari
Dept. of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA
bkrishna@usc.edu

***Abstract*—The p-persistent CSMA protocol is central to random-access MAC analysis, unlocking full network capacity requires actively coordinating nodes by exploring a massive space of topologies and transmission probabilities yet we cannot optimize what we cannot compute efficiently. Simplified models that assume a single, shared interference domain can underestimate throughput by 48–62% in sparse topologies. Exact Markov-chain analyses are accurate but scale exponentially in computation time, making them impractical for large networks. These computational barriers motivate structural machine learning approaches like GNNs for scalable throughput prediction in general network topologies. Yet off-the-shelf GNNs struggle here: a standard GCN yields 63.94% normalized mean absolute error (NMAE) on heterogeneous networks because symmetric normalization conflates a node's direct interference with higher-order, cascading effects that pertain to how interference propagates over the network graph.**

Building on these insights, we propose the Decoupled Graph Convolutional Network (D-GCN), a novel architecture that explicitly separates processing of a node's own transmission probability from neighbor interference effects. D-GCN replaces mean aggregation with learnable attention, yielding interpretable, per-neighbor contribution weights while capturing complex multihop interference patterns. D-GCN attains 3.3% NMAE, outperforms strong baselines, remains tractable even when exact analytical methods become computationally infeasible, and enables gradient-based network optimization that achieves within 5% of theoretical optima.

***Index Terms*—Graph Neural Network, p-CSMA, saturation throughput, network utility maximization, heterogeneous wireless networks, throughput prediction.**

I. Introduction

The p-persistent CSMA (p-CSMA) protocol serves as an analytical model for practical random-access MAC protocols [1] in WLAN and IoT networks, closely replicating the behavior of the basic 802.11 Distributed Coordination Function (DCF) [2]. In particular, p-CSMA underlies several IEEE 802.11 WLAN design studies that optimize contention windows, frame aggregation, and airtime fairness by treating the channel-access attempt as a Bernoulli trial with probability p per time slot [3].

A natural extension of the classical (homogeneous) p-persistent CSMA protocol is its heterogeneous variant, where each node i contends for the channel with an individual attempt probability p_i [4]. Allowing per-node probabilities reflects the reality of modern WLANs and IoT deployments, in which devices differ in traffic load, latency requirements, or power constraints [5]. In heterogeneous networks, nodes experience unequal channel access opportunities based on their local interference environment. Before transmitting, each node must sense whether the channel is idle. When node i transmits, all nodes within its carrier-sense range detect the busy channel and must defer, creating localized contention zones rather than network-wide competition.

Within this framework, a node can only transmit when all neighbors within its sensing range are silent, either not attempting transmission or themselves blocked by their own neighbors. This leads to each node's throughput depending non-linearly on all transmission probabilities:

$$\Theta_i \approx p_i \cdot f(\{\tilde{p}_j : j \in \mathcal{N}(i)\}) \tag{1}$$

where $\tilde{p}_j$ represents the effective transmission probability of neighbor j, not just its attempt probability p_j, but its actual chance to transmit given potential suppression from its own neighbors. Real-world wireless networks exhibit complex multihop interference dependencies. The function f captures both direct interference from immediate neighbors and indirect effects, when a neighbor j is silenced by nodes further away, it cannot interfere with i, creating cascading dependencies through the network topology. These multihop interference patterns make f analytically intractable, as it depends recursively on the entire network state.

This locality of contention is fundamental to practical WLAN/IoT networks. Finite carrier-sense and interference ranges (determined by path loss and shadowing at standard CCA thresholds), physical obstructions, and channelization mean that many node pairs never interact, therefore contention occurs in small, topology dependent zones [6]. These systems are accurately captured by an undirected conflict graph $G = (V, E)$: vertices denote transmitters, and an edge $(v_i, v_j) \in E$ encodes mutual interference, meaning nodes i and j cannot transmit simultaneously [7]. In this representation, $\mathcal{N}(i)$ corresponds to node i's neighbors in G (see Figure 1).

Operators and controllers need models that accurately predict per-node throughput for arbitrary conflict graphs and support optimization of heterogeneous $\{p_i\}$ to meet objectives like proportional fairness, minimum-rate guarantees, or energy/latency trade-offs [8]. Traditional renewal theory approaches [1], which assume complete interference graphs, can underestimate

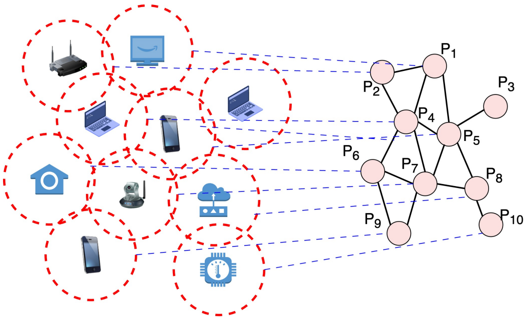

Fig. 1: Example of heterogeneous p-persistent CSMA in a 10-node wireless network

throughput by 48-62% in sparse topologies. Recently, an exact Markov chain method to compute the throughputs for arbitrary topologies correctly has been introduced [9]. However, this approach suffers from state-space explosion as the underlying chain has T^n states (where T is the transmission duration and n the network size). For instance, a modest network with $n = 10$ nodes and $T = 5$ yields 10^7 states, rendering exact analysis computationally intractable for optimization tasks that require repeated throughput evaluations [10], [11].

These computational barriers and the inherent graph structure of interference patterns naturally motivate a learning-based approach. Graph Neural Networks offer compelling advantages for this domain: they operate directly on graph-structured conflict topologies while learning complex non-linear relationships between transmission probabilities and throughputs. Through iterative message passing, GNNs capture multihop interference dependencies, and critically, they provide differentiable throughput predictions that enable gradient-based optimization of network parameters.

Despite this natural alignment, existing applications of machine learning to CSMA protocol optimization have primarily focused on simple scenarios or relied on architectures that fail to exploit the structural properties of interference graphs. Most prior work either assumes simplified conflict models or employs generic neural architectures without incorporating domain-specific inductive biases crucial for wireless network modeling.

This paper addresses these limitations by introducing a new Graph Neural Network architecture tailored to heterogeneous p-CSMA networks that we refer to as Decoupled Graph Convolutional Network (D-GCN). Our aim is to deliver accurate per-node throughput prediction and enable efficient optimization on arbitrary conflict-graph topologies across heterogeneous access probabilities $\{p_i\}$ and packet durations T.

Our D-GCN approach introduces three key innovations that distinguish it from standard GNN architectures:

First, D-GCN decouples self-transmission from neighbor interference processing, mirroring the multiplicative structure of p-CSMA throughput seen in simpler topologies such as complete graphs: $\Theta_i \approx p_i \cdot f(\{\tilde{p}_j : j \in \mathcal{N}(i)\})$. Standard GNNs mix these fundamentally different signals before projection, obscuring the distinction between a node's transmission capability and the suppression it experiences.

Second, D-GCN eliminates degree normalization prevalent in GCN and GraphSAGE, which incorrectly dilutes cumulative interference by averaging neighbor contributions. In wireless networks, interference is additive, more active neighbors mean more contention, not averaged impact. Our architecture preserves this additive nature through unnormalized summation with learnable attention weights.

Third, D-GCN's multi-layer architecture captures k-hop interference cascades, where each layer extends the interference horizon by one hop. This provides a computationally tractable alternative to the exponentially complex analytical methods required for modeling these spatial-temporal dependencies.

We demonstrate that our approach achieves high prediction accuracy (normalized mean absolute error below 8%) across diverse network configurations and scales gracefully with network size and transmission duration. Our learned model serves as an effective surrogate for exact analytical methods in optimization applications, achieving utility values within 1% of theoretical optima. The methodology provides a general framework for applying Graph Neural Networks to wireless protocol optimization, demonstrating how domain-specific architectural modifications can significantly improve performance on structured prediction tasks.

II. Related Work

The analysis of CSMA networks has evolved from single-domain approximations to complex topological models. Bianchi's seminal work [2] established the Markov chain model for saturation throughput in homogeneous, single-collision domains. Subsequent renewal-theoretic models extended this to 802.15.4 and p-persistent CSMA [1], [12]. However, these approaches assume all nodes interfere with each other, failing to capture the spatial reuse inherent in sparse, multi-hop networks.

To address general topologies, Jiang and Walrand [13] connected CSMA scheduling to maximum weight independent set problems, though their exact computation requires exponential complexity $O(T^n 2^n)$. Recently, Tarzjani and Krishnamachari [9] demonstrated that traditional renewal approximations underestimate throughput by 48–62% in sparse graphs. While they proposed an exact Markov chain formulation to correct this, it suffers from state-space explosion ($O(T^n)$), rendering it intractable for large-scale optimization. Other works have focused on standard-specific hybrid access models (e.g., Wi-Fi 6) rather than general heterogeneous conflicts [14].

In the domain of Network Utility Maximization (NUM), machine learning has become essential for handling real-world complexity [15], [16]. Graph Neural Networks (GNNs) specifically have achieved substantial speedups over iterative methods in wireless tasks like power control [17] and link management [18]. Despite this success, GNN application to CSMA protocol optimization remains limited. A notable exception is Neuro-DCF [19], which combines GNNs with multi-agent reinforcement learning. However, Neuro-DCF utilizes the GNN primarily as a feature extractor within a complex RL framework, rather than for direct, differentiable throughput prediction on arbitrary conflict graphs.

To our knowledge, no prior work has investigated GNN architectures for heterogeneous p-CSMA throughput prediction and optimization. While the baseline architectures (GCN, GraphSAGE, GIN, GINE) are well-established in the GNN literature, we are the first to implement and evaluate them for this wireless networking problem, along with our newly proposed D-GCN architecture.

III. Problem Definition

Consider a wireless network that employs the *heterogeneous p-persistent CSMA* (p-CSMA) medium–access protocol. Let $V = \{1, \dots, n\}$ index saturated transmitters. Time is divided into slots of unit length (we set $\sigma = 1$ without loss of generality.)

If a node senses that the channel is idle at the beginning of a time slot, it attempts transmission with its own Bernoulli probability $p_i \in [0, 1]$; otherwise, it defers for one slot. Once a node begins transmission, it occupies the channel for T consecutive time slots. If two neighboring nodes simultaneously sense the channel as idle and initiate transmission, a collision occurs for T consecutive time slots. Figure 2 illustrates a representative scenario for the wireless network and conflict graph shown in Figure 1.

The *saturation throughput* of node i is the long-run fraction of time slots it holds the channel:

$$S_i(t) \;:=\; \mathbf{1}\{i \text{ begins a } \textit{collision-free} \text{ transmission at slot } t\}. \tag{2}$$

$$\Theta_i \;=\; T \cdot \lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} S_i(t). \tag{3}$$

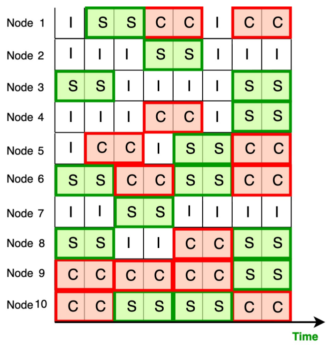

Fig. 2: Example time slot assignment for Figure 1 p-CSMA network with transmission duration T=2. Each row represents a node's channel access pattern over 8 time slots. Green boxes (S) indicate successful transmission, red boxes (C) denote conflicts where transmission attempts fail due to neighbor interference, and white boxes (I) represent idle slots.

The computational challenge of heterogeneous p-CSMA networks manifests at two levels:

State space and scalability. We represent the system at time slot t by the residual-timer vector $s(t) = (a_1(t), \dots, a_n(t)) \in \{0, \dots, T-1\}^n$, where $a_i(t) = 0$ means node i is idle/eligible and $a_i(t) > 0$ are the remaining time slots that it must stay busy. Only idle nodes attempt; a node succeeds if and only if it attempts while none of its graph neighbors do, so winners form an independent set of the undirected conflict graph $G = (V, E)$. Timers update synchronously, winners reset to $T - 1$, while the others decrement to $\max\{a_i(t) - 1, 0\}$. This yields a finite, time-homogeneous Markov chain on $\mathcal{S} = \{0, \dots, T-1\}^n$ with size $|\mathcal{S}| = T^n$.

From any state s, one-step transitions require summing over the $2^{|I(s)|}$ attempt patterns of the idle set $I(s) = \{i : a_i = 0\}$ (worst case $2^{|I(s)|} \le 2^n$). Constructing the kernel P and solving $P^\top \pi = \pi$ therefore scales on the order of $O(T^n 2^n)$ time and $O(T^n)$ memory, which explodes rapidly with T and n (e.g., $T{=}5, n{=}12 \Rightarrow |\mathcal{S}| = 5^{12} \approx 2.44 \times 10^8$; $T{=}7, n{=}12 \Rightarrow 7^{12} \approx 1.38 \times 10^{10}$), making exact analysis impractical beyond small networks or packet durations.

Optimization complexity. The state-space explosion makes evaluating $\Theta_i(\mathbf{p})$ computationally expensive—problematic for optimization, which requires repeated throughput evaluations. The network utility optimization problem is:

$$\max_{\mathbf{p}} \sum_{i=1}^{n} \alpha_i U(\Theta_i(\mathbf{p})) \quad \text{s.t.} \quad p_i \in [0,1],\; g_k(\mathbf{p}) \le 0 \tag{4}$$

where $\Theta_i(\mathbf{p})$ is node i's saturation throughput (a non-linear function of all probabilities, topology, and T), $U(\cdot)$ is a utility function (e.g., log for proportional fairness), $\alpha_i \ge 0$ are priority weights, and $g_k(\cdot)$ captures fairness, QoS, and operational constraints.

Selecting the access probabilities $\mathbf{p}$ that maximize a utility of the resulting throughputs reduces to a *weighted* Maximum Independent Set on the conflict graph, an NP-hard problem. These dual hurdles—state-space explosion (worsened by larger T and network scale) and combinatorial optimization—explain why exact saturation-throughput evaluation and tuning become computationally intractable once the network departs from the single-collision-domain idealization.

IV. Proposed Methodology

Exact methods scale as $\mathcal{O}(T^n 2^n)$ and analytical approximations fail on non-complete conflict graphs, motivating a learning-based approach. GNNs are naturally suited for this task, as wireless interference patterns align well with message passing over graph structures.

As illustrated in Figure 3, in our proposed GNN architecture for this problem, information propagates through the network via iterative message passing layers. A generic message passing layer at depth ℓ performs the following computation:

$$h_v^{(\ell+1)} = \mathrm{UPD}^{(\ell)}\Big(h_v^{(\ell)},\; \mathrm{AGG}^{(\ell)}\{\, \mathrm{MSG}^{(\ell)}(h_v^{(\ell)}, h_u^{(\ell)}, e_{uv})\Big). \tag{5}$$

Here, $h_v^{(\ell)}$ represents the hidden state of node v at layer ℓ (initialized as $h_v^{(0)} = x_v$), $\mathcal{N}(v)$ denotes the neighbors of node v in the conflict graph, and e_{uv} represents edge features. The Message function computes pairwise interactions, Aggregate combines messages from all neighbors, and Update integrates this aggregated information with the node's current state.

After L message passing layers that progressively capture multihop interference patterns, we apply a Multi-Layer Perceptron (MLP) head to each node's final representation:

$$\hat{\Theta}_v = \sigma\Big(\mathrm{MLP}(h_v^{(L)})\Big).$$

The MLP head consists of two fully-connected layers and ReLU activations between layers. The final sigmoid activation $\sigma(\cdot)$ ensures the predicted throughput $\hat{\Theta}_v \in [0, 1]$, respecting the physical constraint that throughput cannot exceed channel capacity.

This architecture serves dual purposes: First, it transforms the graph-aware embeddings into task-specific throughput predictions; Second, it increases model capacity without requiring additional GNN layers, which would risk oversmoothing due to excessive neighborhood aggregation.

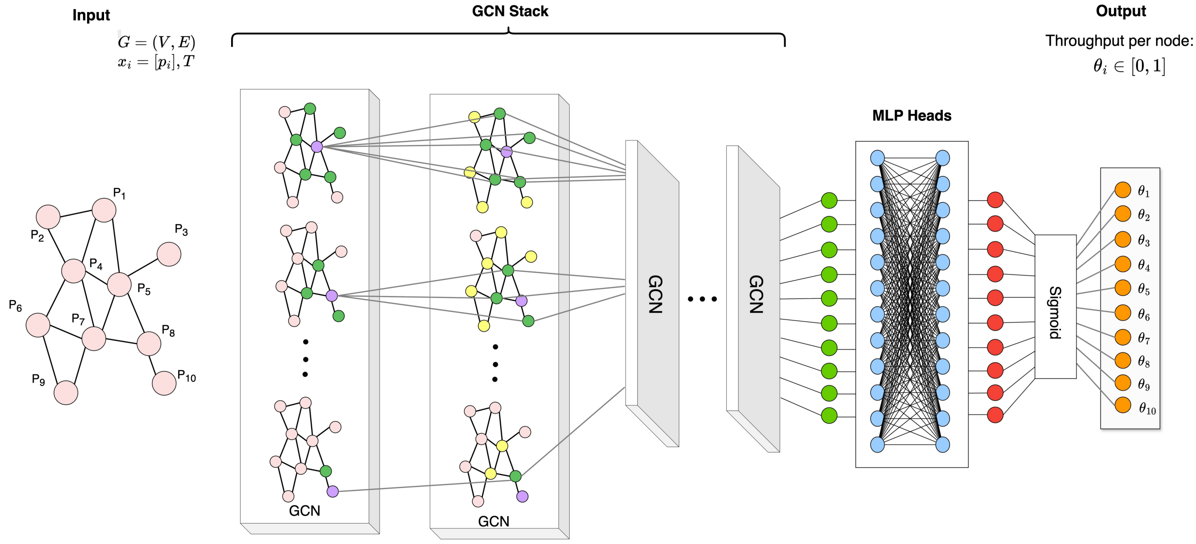

Fig. 3: Proposed Graph Neural Network architecture for heterogeneous p-CSMA throughput

A. *Baseline GNN Architectures for Heterogeneous Interference Modeling*

To understand the architectural requirements for modeling wireless interference, We compare five GNN variants: GCN [20] (symmetric normalization, mixed transformations), GraphSAGE [21] (separate self/neighbor weights, mean aggregation), GIN [22] (sum aggregation, learnable self-weighting), GINE [23] (edge-aware message passing), and our proposed D-GCN (attention-weighted neighbor suppression).

Each architecture instantiates the general message passing framework from Equation (5) differently, leading to distinct inductive biases that we now examine in detail.

1) GCN (Kipf & Welling): The Graph Convolutional Network [20] applies symmetric degree normalization:

$$\tilde{A} = A + I, \qquad \tilde{D}_{vv} = \sum_{u} \tilde{A}_{vu}, \tag{6}$$

$$h^{(\ell+1)} = \sigma\left(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}h^{(\ell)}W^{(\ell)}\right), \tag{7}$$

GCN *averages* neighbor signals (scaled by degree) and applies a single linear transform; self and neighbor information are mixed before projection. In contention graphs, this tends to *over-smooth* high-contrast local load signals, contributing to underfitting.

2) GraphSAGE (Hamilton et al.): The Graph Sample and Aggregate (GraphSAGE) architecture [21] introduces explicit separation between self and neighbor transformations:

$$h_v^{(\ell+1)} = \sigma\left(W_{\text{self}}^{(\ell)} h_v^{(\ell)} + W_{\text{neigh}}^{(\ell)} \cdot \text{mean}_{u\in\mathcal{N}(v)} h_u^{(\ell)}\right), \tag{8}$$

where W_{self} and W_{neigh} are separate weight matrices for self and neighbor transformations. While this decoupling improves upon GCN's mixed transformations, the mean aggregation normalizes neighbor contributions by degree, which our experiments show is detrimental for heterogeneous interference graphs.

3) GIN (Xu et al.): The Graph Isomorphism Network (GIN) [22] matches the discriminative power of the 1-Weisfeiler-Lehman graph isomorphism test and outperforms GCN and GraphSAGE on several benchmarks. The Graph Isomorphism Network replaces degree-normalized averaging with degree-agnostic summation followed by an MLP, and introduces a learnable self-weight $\epsilon^{(\ell)}$:

$$h_v^{(\ell+1)} = \text{MLP}^{(\ell)}\Big((1+\epsilon^{(\ell)})\, h_v^{(\ell)} \;+\; \sum_{u\in\mathcal{N}(v)} h_u^{(\ell)}\Big). \tag{9}$$

Summation treats each neighbor equally while preserving multiset counts; the MLP can learn highly non-linear functions of the aggregated local load ($\sum p_u$, etc.), which is critical for approximating collision probability.

4) GINE (Hu et al.): The Graph Isomorphism Network with Edge features [23] incorporates edge attributes additively inside a ReLU before summation:

$$h_v^{(\ell+1)} = \text{MLP}^{(\ell)}\Big((1+\epsilon^{(\ell)})\, h_v^{(\ell)} \;+\; \sum_{u\in\mathcal{N}(v)} \text{ReLU}\big(h_u^{(\ell)} + \phi(e_{uv})\big)\Big). \tag{10}$$

$$\phi(e_{uv}) = W_e\, e_{uv} + b_e,$$

When e_{uv} encodes link strength or interference severity, GINE can learn to modulate neighbor impact. With binary edges (our data), this reduces to a learnable shift/gating that nonetheless improves stability versus vanilla GIN.

Standard GNN architectures fail to capture wireless interference dynamics. Combining self and neighbor information before projection obscures the distinction between a node's transmission capability and interference suppression. Mean aggregation incorrectly dilutes cumulative interference, while mixed transformations lack interpretability for protocol optimization.

Also, lack of interpretability is particularly problematic for protocol optimization, where understanding which neighbors cause the most contention is essential for parameter tuning.

B. *Decoupled Graph Convolutional Network (D-GCN)*

To address these limitations, we propose the Decoupled Graph Convolutional Network (D-GCN), which incorporates four key architectural innovations: (i) explicit separation of self-transmission and neighbor interference processing channels, (ii) elimination of degree normalization and mean aggregation that incorrectly dilute cumulative interference effects, (iii) learnable attention weights to capture heterogeneous neighbor impacts, and (iv) unnormalized summation that preserves the additive nature of wireless interference—where more active neighbors create more contention, not averaged impact.

$$
\begin{aligned}
z_u^{(\ell)} &= h_u^{(\ell)} W_{\text{nbr}}^{(\ell)} \quad \text{(neighbor transformation)} \\
\alpha_{uv}^{(\ell)} &= \sigma\big(\mathbf{a}^{(\ell)\top} z_u^{(\ell)}\big) \quad \text{(learned importance)} \\
h_v^{(\ell+1)} &= \sigma\Bigg(\underbrace{h_v^{(\ell)} W_{\text{self}}^{(\ell)}}_{\text{self channel}} + \underbrace{\sum_{u \in \mathcal{N}(v)} \alpha_{uv}^{(\ell)} \cdot \text{ReLU}(z_u^{(\ell)})}_{\text{neighbor suppression}} + b^{(\ell)} \Bigg)
\end{aligned}
\tag{11}
$$

Through L stacked layers, this architecture captures *multi-hop interference cascades* up to L hops away, directly mirroring $\Theta_i \approx p_i \cdot f(\{\tilde{p}_j : j \in \mathcal{N}(i)\})$. The learned attention weights α_{uv} act as a soft interference mask capturing heterogeneous neighbor impacts, while the ReLU before aggregation provides essential non-linearity for modeling collision dynamics [23].

Figure 4 visualizes this computation flow. The node's embedding $h_v^{(\ell)}$ follows two parallel paths: the self-channel (left) directly transforms the node's features via W_{self}, while the neighbor path (right) processes each neighbor through transformation (W_{nbr}), attention weighting (α_{uv}), and ReLU non-linearity before aggregation. These pathways are summed with bias $b^{(\ell)}$ and activated to produce the next-layer embedding $h_v^{(\ell+1)}$.

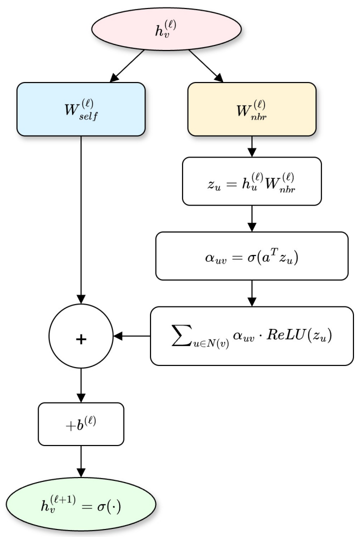

Fig. 4: Architecture of the Decoupled Graph Convolutional Network (D-GCN) for p-CSMA throughput prediction.

C. Summary of Architectural Differences

Table I summarizes the key architectural distinctions, highlighting each model's ability to capture wireless interference dynamics, heterophily robustness, and over-smoothing resistance.

TABLE I: Architectural feature comparison highlighting distinctions between baseline GNNs and the proposed D-GCN

Feature	GCN	SAGE	GIN	GINE	D-GCN
Normalized aggregation	✓	✓	✗	✗	✗
Per-neighbor weighting	✗	✗	✗	✓	✓
Edge-aware messages	✗	✗	✗	✓	✗
Nonlinearity *before* aggregation	✗	✗	✗	✓	✓
Per-layer MLPs	✗	✗	✓	✓	✗
Decoupled self vs. neighbor paths	✗	✓	✗	✗	✓
Heterophily robustness	Low	Medium	Medium	High	High
Over-smoothing resistance	Low	Medium	Medium	Med–High	High

D-GCN achieves 3.3% NMAE compared to 63.94% for standard GCN, demonstrating that domain-specific architectural design, decoupled self/neighbor processing with learnable attention and unnormalized aggregation—is essential for capturing wireless interference dynamics that generic GNN architectures cannot model effectively.

V. Dataset Generation and Evaluation Metrics

To train, validate, and benchmark the proposed GNN, we require *per–topology, per–node* ground-truth throughput labels. We generate data by running a p-CSMA network under two models:

1) Event-Driven Simulation (Approximate): We generate random Erdős–Rényi conflict graphs ($n \in \{3, ..., 20\}$, $p_{\text{edge}} = 0.5$) with independent access probabilities $p_i \sim \mathcal{U}(0, 1)$. D-GCN operates on conflict graph topology rather than physical coordinates [7], [24]. Ground-truth throughput is obtained via event-driven p-CSMA simulation (10^6 slots): successful transmissions occupy T slots, collisions incur $\sigma = 1$ idle slot, yielding $\Theta_i^{\text{Sim}} = (\#\text{succ}_i \times T)/10^6$.

Simulation precision: Simulations use $L = 10^6$ time slots with 95% confidence half-widths $h_i = 1.96\sqrt{\hat{\Theta}_i T / L}$. For typical throughputs (0.01–0.1), this yields 1–5% relative uncertainty at $T = 2$, increasing to 2–10% at $T = 8$. This Monte Carlo variance contributes measurably to prediction error, particularly for low-throughput nodes and larger T.

2) Markov-Chain Solver (Exact): The same network can be analysed exactly by modelling it as a global discrete-time Markov chain whose state at slot t is the vector of *remaining busy times* $\mathbf{a}(t) = (a_1, \ldots, a_n) \in \{0, \ldots, T-1\}^n$. There are T^n such states; for each state we enumerate all 2^n feasible transmission decisions, build the transition matrix P, and solve $P^\top \boldsymbol{\pi} = \boldsymbol{\pi}$ for the stationary distribution $\boldsymbol{\pi}$. Using the reward decomposition in [9] we obtain the exact per-node saturation throughput vector $\boldsymbol{\Theta}^{\text{MC}}$.

We repeat the above procedure for many independently generated random topologies; each iteration run writes one row to a CSV file containing the graph's adjacency matrix, the per-node access-probability vector, and the resulting saturation-throughput vector, providing a reusable dataset for subsequent analysis.

A. Graph-Neural Network Configuration

For each topology, we construct a conflict graph $G = (V, E)$ where vertices represent transmitters and edges encode interference. Each node i has feature $\mathbf{x}_i = [p_i]$ (transmission probability); augmenting with T yielded only marginal improvements.

As shown in Figure 4, our architecture uses 8 D-GCN layers (64 hidden units each) to capture interference up to 8 hops, followed by a 2-layer MLP [64→32→1]. We use ReLU activations (sigmoid for output), AdamW optimizer (lr=0.001, weight decay 10^{-4}), ReduceLROnPlateau scheduling, MSE loss, and gradient clipping (max norm 1.0). Models converge within 150–200 epochs across all configurations.

B. Evaluation Metrics

We assess fidelity using **Mean Squared Error (MSE)** as the training loss. For interpretation, we report **Mean Absolute Error (MAE = $\frac{1}{N}\sum|\hat{\Theta}_i - \Theta_i|$)** and **Normalized MAE (NMAE = MAE/$\overline{\Theta}$)**, where $\overline{\Theta}$ is the mean ground-truth throughput. NMAE allows fair comparison across datasets with varying throughput scales.

VI. Experimental Results & Performance Evaluation

This section evaluates D-GCN's predictive accuracy, generalization, and computational efficiency, comparing it against GNN baselines and assessing its effectiveness in gradient-based utility optimization and fairness.

A. Performance comparison with other GNN architectures

We compared D-GCN against GCN, GraphSAGE, GIN, and GINE on throughput prediction (Table II). All models were trained on identical data ($T = 5$) using only transmission probability p_i as node features, isolating the impact of architectural differences on learning the nonlinear mapping to throughput.

Our D-GCN achieves the lowest normalized mean absolute error (NMAE) of 3.3%, significantly outperforming GCN (63.9%), GraphSAGE (23.7%), GIN (21.4%), and GINE (4.7%). These results highlight that D-GCN's decoupled self/neighbor design and unnormalized attention aggregation enable it to capture the nonlinear interference relationships in wireless networks far more effectively than standard GNN architectures.

TABLE II: Test-set error of evaluated GNN architectures (T=5 dataset, single node feature p_i).

Architecture	MAE	NMAE
GCN	0.0495	0.6394
SAGE	0.0183	0.2372
GIN	0.0165	0.2135
GINE	0.0037	0.0470
D-GCN (ours)	**0.0026**	**0.0330**

Figure 5 demonstrates the consistent superiority of our proposed D-GCN architecture over GINE across all tested configurations, with simpler, more interpretable operations that align with wireless physics.

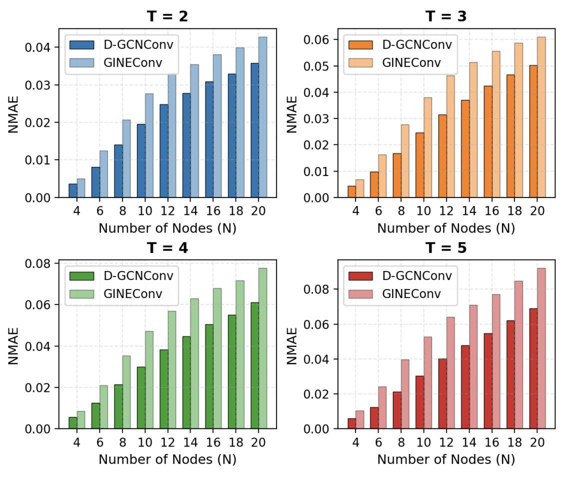

Fig. 5: Comparison of Normalized Mean Absolute Error (NMAE %) between D-GCN (Decoupled GCN) and GINE architectures across different transmission durations (T = 2, 3, 4, 5) and network sizes ($N \in \{4, 6, 8, 10, 12, 14, 16, 18, 20\}$)

B. Performance with Different Number of Training Samples

To evaluate data efficiency, we trained D-GCN on varying fractions (10%–100%) of 45,000 samples across network sizes $N \in \{4, ..., 20\}$ at $T = 5$. Table III shows the model achieves NMAE $\leq$ 8% with just 4,500 samples (10%), improving substantially to 4.06% at 50%, with marginal gains beyond. This rapid convergence is valuable when ground-truth generation via Markov analysis is expensive, and the consistent train-test gap indicates good generalization.

TABLE III: Model performance with varying training dataset sizes ($T = 5$, $N \in \{4, 6, \ldots, 20\}$)

Training Data	Train NMAE (%)	Test NMAE (%)
10% (4500)	6.81	7.30
25%	3.99	4.81
50% (22,500)	3.37	4.06
75%	2.63	3.24
100% (45,000)	2.86	3.32

C. Performance with Different Network Settings

Figure 6 illustrates D-GCN's performance across varying transmission durations ($T \in \{2, 3, 4, 5, 6\}$) and network sizes ($N \in \{4, 6, 8, 10, 12, 14, 16, 18, 20\}$). Prediction accuracy decreases with network size, with NMAE rising from 0.36%–0.61% for 4-node networks to 3.58%–7.46% for 20-node networks, due to increasingly complex multihop interference patterns. Model errors consistently dominate simulation uncertainty, with error-to-uncertainty ratios of approximately 6:1 to 8:1, confirming that prediction errors stem from model approximation rather than simulation noise.

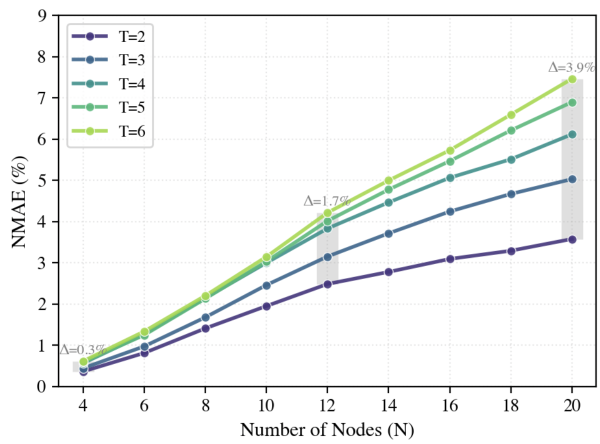

Fig. 6: Performance of the D-GCN across different network configurations, varying in the number of nodes and transmission durations.

The architecture scales via local message passing with complexity $O(|E| \cdot d \cdot L)$, and real wireless deployments exhibit localized interference neighborhoods of 10–15 nodes [6], [25], meaning our experimental scale captures the relevant dynamics. Performance curves converge for $T \geq 5$, reflecting that longer channel occupancy dominates throughput calculations. Notably, the performance curves converge for larger T values ($T \geq 5$)—the NMAE difference between $T = 5$ and $T = 6$ is minimal compared to $T = 2$ versus $T = 3$. This reflects the underlying p-CSMA dynamics, as transmission duration increases, longer channel occupancy periods dominate the throughput calculation, making the relative impact of T less significant. The model accurately captures this inherent property of the protocol.

D. Generalizability to Different Network Settings

We evaluated D-GCN generalization across network size and transmission duration. When trained on small graphs ($N \in \{4, ..., 12\}$) and tested on larger networks, NMAE increases from 6.90% to 15.38% at $N = 20$ (Table IV); training on the full range ($N \in [4, 20]$) keeps NMAE below 7%. The model shows stronger temporal generalization (Table V), achieving NMAE of 4.58%–6.29% on unseen $T \in \{7, 8\}$ when trained only on $T \in \{2, ..., 6\}$. This asymmetric generalization, stronger for temporal parameters than spatial configurations, aligns with the fundamental nature of p-CSMA networks, where temporal dynamics follow predictable protocol behavior while spatial interference patterns grow exponentially with network size. These results confirm that our D-GCN architecture effectively captures the underlying throughput dynamics, making it suitable for practical deployment in heterogeneous wireless networks.

TABLE IV: Generalization across network size at T=5. Left column: model trained on small graphs ($N \in \{4, 6, 8, 10, 12\}$) and tested on larger graphs ($N \in \{14, 16, 18, 20\}$). Right column: model trained on all sizes ($N \in [4, 8, 10, 12, 14, 16, 18, 20]$) and tested on ($N \in \{14, 16, 18, 20\}$). We report normalized MAE (NMAE).

	Train $N = 4, 6$–12	**Train** $N = 4, 6$–20
N	**NMAE (%)**	**NMAE (%)**
14	6.25	4.77
16	8.59	5.47
18	12.09	6.21
20	15.38	6.90

TABLE V: Generalization across transmission duration T (NMAE). Left: trained on $T \in \{2, ..., 6\}$, tested on unseen $T \in \{7, 8\}$. Right: trained on $T \in \{2, ..., 8\}$. All models use mixed sizes $N \in [4, 20]$.

	Train T=2–6	**Train** T=2–8 (**all-**T)
T	**NMAE (%)**	**NMAE (%)**
7	4.58	3.31
8	6.29	3.59

E. Computational Efficiency Analysis

Table VI quantifies D-GCN's computational advantage over exact Markov chain analysis (tested on Apple M2 Pro). While MC scales exponentially as $O(T^n \cdot 2^n)$, D-GCN maintains sub millisecond inference across all configurations, achieving speedups from 3× (small networks) to over 195,000× (10 nodes) enabling real time network optimization and fairness infeasible with exact methods.

TABLE VI: Computation time comparison between exact Markov chain analysis and D-GCN inference across different network configurations.

Num Of Nodes	**T**	**State Space**	**MC Time (s)**	**D-GCN Time (s)**	**Speedup**
5	2	32	1.82×10^{-3}	6.30×10^{-4}	2.9×
5	3	243	2.96×10^{-2}	6.47×10^{-4}	45.7×
6	2	64	1.44×10^{-2}	6.36×10^{-4}	22.7×
6	3	729	1.43×10^{-1}	6.62×10^{-4}	216.8×
7	2	128	4.58×10^{-2}	6.89×10^{-4}	66.5×
7	3	2,187	1.01	7.13×10^{-4}	1,412×
8	2	256	1.76×10^{-1}	7.21×10^{-4}	244.8×
8	3	6,561	8.72	7.30×10^{-4}	11,943×
9	2	512	7.29×10^{-1}	7.34×10^{-4}	993.5×
9	3	19,683	145.31	7.42×10^{-4}	195,770×
10	2	1,024	2.55	7.53×10^{-4}	3,387×
10	3	59,049	*intractable*†	7.80×10^{-4}	–

†Process terminated after exceeding 1000s runtime threshold.

The results reveal two critical insights. First, D-GCN's inference time remains remarkably stable (6.3–7.8 × 10^{-4} seconds) regardless of network size or packet duration, demonstrating $O(1)$ complexity with respect to the state space. Second, the speedup factor increases exponentially with network complexity, making D-GCN particularly valuable for large-scale network optimization where hundreds of throughput evaluations are required.

F. Application to Network Utility Maximization and Fairness

To evaluate the practical utility of D-GCN, we embed it within gradient-based optimization loops for two objectives: (1) weighted log-utility maximization $J(\mathbf{p}) = \sum_i \alpha_i \log(\Theta_i(\mathbf{p}) + \varepsilon)$, which achieves proportional fairness by penalizing throughput disparities, and (2) Jain's fairness index maximization $\mathcal{J}(\mathbf{\Theta}) = \frac{(\sum_i \Theta_i)^2}{n \sum_i \Theta_i^2}$, which drives all throughputs toward equality. We compare D-GCN against the exact Markov chain model [9] using identical initialization, learning rates (0.01), and SGD optimization over 250 iterations.

Figure 7 demonstrates D-GCN's performance on a 10-node heterogeneous IoT network. The conflict graph (Figure 7(a)) illustrates interference relationships among diverse wireless devices. As shown in Figure 7(b), D-GCN closely tracks the exact Markov chain optimization trajectory, with both methods converging to $J \approx -17.0$ (difference $< 0.5\%$). Figures 7(c) and (d) present fairness optimization results: D-GCN learns heterogeneous transmission probabilities adapted to each device's interference neighborhood, driving initially unequal throughputs to converge within 50 iterations and achieving near-perfect Jain's index of 1.0.

VII. Conclusions

This paper presents the first Graph Neural Network application for predicting per-node saturation throughput in heterogeneous p-CSMA networks, addressing the computational intractability of exact Markov methods. Our Decoupled Graph Convolutional Network (D-GCN) introduces an interpretable architecture that separates self-transmission from neighbor interference without degree normalization, using learnable attention weights to capture heterogeneous neighbor impacts. D-GCN achieves 3.3% NMAE versus 63.94% for standard GCN while maintaining interpretability about interference sources.

By providing differentiable throughput estimates, D-GCN enables gradient-based network optimization that achieves utility within 5% of theoretical optima while offering computational speedups of 100-1000× compared to exact Markov chain methods.

Several limitations merit acknowledgment. First, D-GCN's scalability beyond 20 nodes remains unexplored. Second, the model assumes saturated traffic; real-world networks with bursty or time-varying traffic would require incorporating temporal dynamics and queue states into node features. Finally, training relies on ground-truth labels from simulations or analytical methods, though reasonable performance is achieved with just 4,500 samples. Future research directions include: (i) extending the architecture to handle non-saturated traffic patterns and variable packet lengths, (ii) incorporating physical layer parameters such as signal-to-interference ratios and channel conditions, (iii) developing online learning mechanisms that adapt to dynamic network conditions, (iv) investigating the application of our decoupled architecture to other MAC protocols beyond p-CSMA and other use cases.

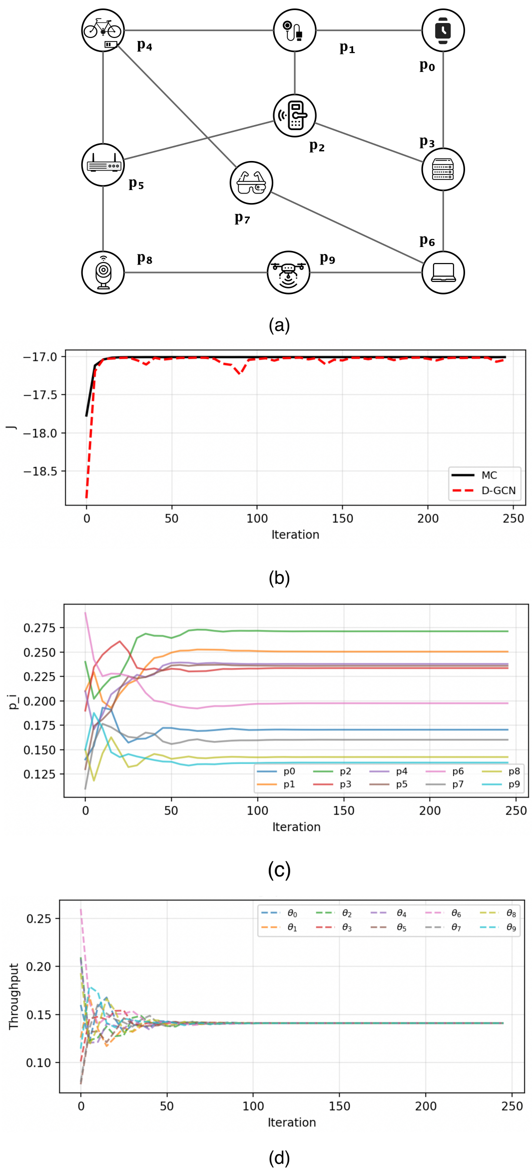

Fig. 7: Real-world application of the proposed D-GCN framework for WiFi coexistence in smart environments. (a) Representative IoT deployment topology with heterogeneous wireless devices operating under p-CSMA Edges represent carrier sensing relationships where simultaneous transmissions cause collisions. (b) Convergence of the weighted log-utility objective, validating D-GCN accuracy against exact Markov Chain computation. (c) Optimized transmission probabilities ensuring fair channel access. (d) Resulting throughput distribution demonstrating equitable bandwidth allocation critical for QoS-sensitive IoT applications.

In conclusion, this work demonstrates that carefully designed GNN architectures can serve as accurate, efficient surrogate models for complex wireless protocol analysis. Source code, datasets, and configurations are available at https://github.com/ANRGUSC/predictCSMA.

References

[1] Y. Gai, S. Ganesan, and B. Krishnamachari, "The saturation throughput region of p-persistent csma," in *2011 Information Theory and Applications Workshop*. IEEE, 2011, pp. 1–4.

[2] G. Bianchi, "Performance analysis of the ieee 802.11 distributed coordination function," *IEEE Journal on selected areas in communications*, vol. 18, no. 3, pp. 535–547, 2000.

[3] L. Bononi, M. Conti, and E. Gregori, "Runtime optimization of ieee 802.11 wireless lans performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 1, pp. 66–80, 2004.

[4] Y. Yu, S. C. Liew, and T. Wang, "Non-uniform time-step deep q-network for carrier-sense multiple access in heterogeneous wireless networks," *IEEE Transactions on Mobile Computing*, vol. 20, no. 9, pp. 2848–2861, 2020.

[5] F. Cali, M. Conti, and E. Gregori, "Dynamic tuning of the ieee 802.11 protocol to achieve a theoretical throughput limit," *IEEE/ACM Transactions on networking*, vol. 8, no. 6, pp. 785–799, 2000.

[6] B. Abdallah, S. Khriji, R. Chéour, C. Lahoud, K. Moessner, and O. Kanoun, "Improving the reliability of long-range communication against interference for non-line-of-sight conditions in industrial internet of things applications," *Applied Sciences*, vol. 14, no. 2, p. 868, 2024.

[7] Y. Lu, Y. Li, R. Zhang, W. Chen, B. Ai, and D. Niyato, "Graph neural networks for wireless networks: Graph representation, architecture and evaluation," *IEEE Wireless Communications*, 2024.

[8] T. Yang, J. Sun, and A. Mohajer, "Queue stability and dynamic throughput maximization in multi-agent heterogeneous wireless networks," *Wireless Networks*, vol. 30, no. 5, pp. 3229–3255, 2024.

[9] F. D. Tarzjani and B. Krishnamachari, "Computing the saturation throughput for heterogeneous p-csma in a general wireless network," in *2025 34th International Conference on Computer Communications and Networks (ICCCN)*, Aug 2025, pp. 1–7.

[10] H. Afifi, S. Pochaba, A. Boltres, D. Laniewski, J. Haberer, L. Paeleke, R. Poorzare, D. Stolpmann, N. Wehner, A. Redder *et al.*, "Machine learning with computer networks: techniques, datasets, and models," *IEEE access*, vol. 12, pp. 54 673–54 720, 2024.

[11] R. Sun, N. Cheng, C. Li, W. Quan, H. Zhou, Y. Wang, W. Zhang, and X. Shen, "A comprehensive survey of knowledge-driven deep learning for intelligent wireless network optimization in 6g," *IEEE Communications Surveys & Tutorials*, 2025.

[12] X. Ling, Y. Cheng, J. W. Mark, and X. Shen, "A renewal theory based analytical model for the contention access period of ieee 802.15. 4 mac," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2340–2349, 2008.

[13] L. Jiang and J. Walrand, "A distributed csma algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2009.

[14] S. Arthi and N. B. Mehta, "Saturation throughput analysis of hybrid access mac protocol in ieee 802.11 ax wlans," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 3901–3906.

[15] Y. Cao, B. Sun, and D. H. Tsang, "Online network utility maximization: Algorithm, competitive analysis, and applications," *IEEE Transactions on Control of Network Systems*, vol. 10, no. 1, pp. 274–284, 2022.

[16] K. Ji and L. Ying, "Network utility maximization with unknown utility functions: A distributed, data-driven bilevel optimization approach," in *Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2023, pp. 131–140.

[17] Y. Wang, Y. Li, Q. Shi, and Y.-C. Wu, "Engnn: A general edge-update empowered gnn architecture for radio resource management in wireless networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 6, pp. 5330–5344, 2023.

[18] Y. Shen, J. Zhang, S. Song, and K. B. Letaief, "Graph neural networks for wireless communications: From theory to practice," *IEEE Transactions on Wireless Communications*, vol. 22, no. 5, pp. 3554–3569, 2022.

[19] S. Moon, S. Ahn, K. Son, J. Park, and Y. Yi, "Neuro-dcf: Design of wireless mac via multi-agent reinforcement learning approach," in *Proceedings of the Twenty-second International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, pp. 141–150.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[21] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[22] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" *arXiv preprint arXiv:1810.00826*, 2018.

[23] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint arXiv:1905.12265*, 2019.

[24] W. Li, J. Zhang, and Y. Zhao, "Conflict graph embedding for wireless network optimization," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[25] A. Cilfone, L. Davoli, L. Belli, and G. Ferrari, "Wireless mesh networking: An iot-oriented perspective survey on relevant technologies," *Future internet*, vol. 11, no. 4, p. 99, 2019.