

Limit Analysis of Graph Neural Networks with Wireless Conflict Graphs

Romina Garcia Camargo Zhiyang Wang Alejandro Ribeiro

Abstract—Graph Neural Networks (GNNs) have emerged as a powerful tool for wireless resource allocation that leverages the underlying graph structure of communication networks. Their transferability property enables models trained on small-scale graphs to generalize to large-scale deployments with little performance deterioration, a desirable property for currently growing networks. Wireless networks are sparse regimes, where a single node is connected to a small number of other users. This work establishes theoretical results for transferability of GNNs over graphs derived from sparse Random Geometric Graphs (RGGs). In particular, we focus on conflict graphs of RGGs used to model interference among links. Our approach considers the closeness between RGGs and Deterministic Grid Graphs (DGG) to establish bounds in the performance loss when a model is transferred across scales. We validate our theoretical findings through the problem of link scheduling, demonstrating that our learned policies consistently outperform existing benchmarks at scale. Finally, we examine the impact of our theoretical assumptions on empirical performance.

Index Terms—transferability, graph neural networks, random geometric graphs, conflict graphs

I. INTRODUCTION

Learning-based solutions are promising tools for addressing complex resource allocation challenges in next-generation networks. Graph Neural Networks (GNNs) are well-suited for these tasks, as the architecture exploits the underlying graph structure of wireless systems [1]–[3]. In particular, it allows for transferability: models trained on small graphs generalize to larger networks with minimal performance loss.

While empirical studies show that GNNs successfully transfer in wireless settings [4], these observations lack a formal foundation. Multiple works have explored the transferability of Graph Neural Networks [5]–[9]. However, most existing frameworks typically rely on limit objects such as graphons [5] or manifolds [8], which define transferability as connectivity increases toward a dense limit. Physical constraints in wireless channels limit the number of connections per user, resulting in sparse graphs that traditional dense-limit theories fail to account for. To bridge this gap, recent work has begun analyzing transferability on sparse graphs via graphops [9] or using Random Geometric Graphs (RGGs), leveraging the geometric structure intrinsic to communication systems [10].

Resource allocation tasks such as link scheduling depend on the relationships between communication links rather than the users themselves. This is often tackled via conflict

graphs, where links are modeled as nodes and edges represent interference or shared resources [11], [12] (Section II). However useful, it creates a specific sparse regime distinct from the underlying physical network.

In this work we establish the theoretical analysis for transferability of GNNs on conflict graphs (Sections III and IV). Extending our framework [10], we model wireless connectivity using Random Geometric Graphs (RGGs) and prove that transferability holds via Deterministic Grid Graphs (DGGs). By leveraging their spatial periodicity, we overcome the analytical challenges posed by the topological irregularities of random node placements. Our results demonstrate that as the number of nodes increases, transferability is guaranteed for conflict graphs of RGGs that approximate conflict graphs of DGGs.

We validate our theoretical framework with numerical experiments on the wireless link scheduling problem, incorporating a minimum transmission requirement per link (Section V). The GNN is trained on conflict graphs derived from RGGs to select which links should be scheduled at each time step. Empirical results demonstrate that our transferred model consistently outperforms benchmarks at larger scales. Furthermore, we characterize the robustness of our RGG-DGG proximity assumption by analyzing model performance as the graph topology deviates from a grid-like structure.

II. RESOURCE ALLOCATION ON CONFLICT GRAPHS

Let $\mathbf{G} = (\mathcal{V}, \mathcal{E})$ be a graph with $|\mathcal{V}| = v$ nodes and $|\mathcal{E}| = e$ edges. Its conflict graph $\mathbf{G}_C = (\mathcal{V}_C, \mathcal{E}_C)$ will have $|\mathcal{V}_C| = e$ nodes, one for each edge in \mathbf{G} . An edge $(i, j) \in \mathcal{E}_C$ will exist if edges i and j are incident on the same node in \mathbf{G} . Figure 1 shows an illustration of graphs and their conflict graphs.

In wireless communication settings, conflict graphs are widely used to model resource allocation tasks (Section II-A, [4]), as they represent dependencies or forbidden combinations between the edges of the original graphs [11], [12]. In particular, these models are used to analyze the links of the network, such as via the primary interference model for wireless link scheduling [13] (Section II-B).

A. Optimal Resource Allocation in Wireless Settings

Let there be a network with m users, where each user is associated with a state at time t given by the vector $\mathbf{x}_m(t) \in \mathbb{R}^m$. Given the channel gains in $\mathbf{S}_m(t) \in \mathbb{R}^{m \times m}$, where $[\mathbf{S}_m(t)]_{ij}$ is the gain from user i to user j , we are interested in allocating resources across the network. Let $\mathbf{p}(\mathbf{x}_m(t), \mathbf{S}_m(t))$ be a mapping from network states to the instantaneous resource allocation policy across the network.

RGC and AR are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA (emails: {rominag, aribeiro}@seas.upenn.edu). ZW is with Halicioğlu Data Science Institute, UCSD, La Jolla, CA (email: zhw135@ucsd.edu).

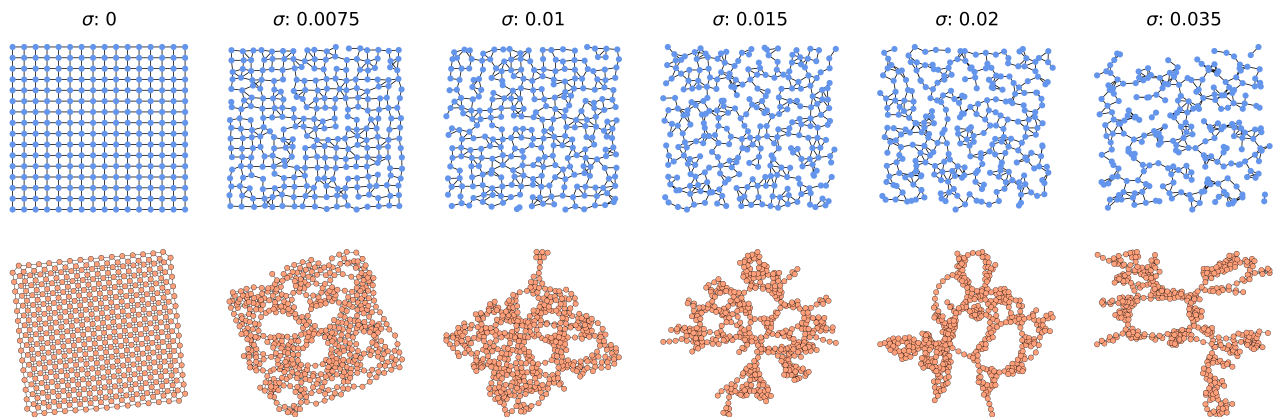


Figure 1: Illustration of Random Geometric Graphs (blue) and their corresponding conflict graphs (pink) as the noise in positions increases. Observe that the RGG deviates from a grid-like structure as the positional noise increases.

The system will then return a reward given the allocation, $\mathbf{f}(\mathbf{p}(\mathbf{x}_m(t), \mathbf{S}_m(t)); \mathbf{x}_m(t), \mathbf{S}_m(t))$. Considering the expectation over the stationary joint distribution of $(\mathbf{x}_m, \mathbf{S}_m)$, we focus on the long-term average performance as

$$\mathbf{r}_m = \mathbb{E}[\mathbf{f}(\mathbf{p}; \mathbf{x}_m, \mathbf{S}_m)]. \quad (1)$$

Consider a utility function $u_0(\mathbf{r}_m)$ and long-term system constraints $\mathbf{u}(\cdot)$. Our goal is to maximize the expected reward (1) by designing our allocation policy $\mathbf{p}(\mathbf{x}_m, \mathbf{S}_m)$:

$$\mathbf{p}^*(\mathbf{x}_m, \mathbf{S}_m) = \arg \max_{\mathbf{p}(\mathbf{x}_m, \mathbf{S}_m) \in \mathcal{P}(\mathbf{x}_m, \mathbf{S}_m)} u_0(\mathbf{r}_m), \quad (2)$$

$$\text{s.t. } \mathbf{r}_m = \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{x}_m, \mathbf{S}_m); \mathbf{x}_m, \mathbf{S}_m)],$$

$$\mathbf{u}(\mathbf{r}_m) \geq \mathbf{0}.$$

The objective in Equation (2) is often nonconvex in \mathbf{p} , which makes finding a solution challenging. The common approach introduces a parameterized policy $\Phi(\mathbf{x}_m, \mathbf{S}_m; \mathbf{H}_m)$, with parameters $\mathbf{H}_m \in \mathbb{R}^s$ and learns a solution (Section III).

B. Example: Wireless Link Scheduling

Let \mathbf{G} be wireless network with m links, where $p_i(t) \in \{0, 1\}$ denotes the status of link $i \in \{1, 2, \dots, m\}$. Entry $p_i(t) = 1$ indicates that the link i is scheduled to transmit at time step t . Two links interfere each other if they are scheduled at the same time and share a common user [13]. The vector $\mathbf{p}(t) = [p_i(t)]_{1 \leq i \leq m} \in \{0, 1\}^m$ describes the scheduling status of all links and is therefore supported on the edges of \mathbf{G} .

Let us define \mathbf{G}_C the conflict graph of \mathbf{G} , then $\mathbf{p}(t)$ is supported on the nodes of \mathbf{G}_C . We denote the adjacency matrix of \mathbf{G}_C as $\mathbf{A}_m \in \{0, 1\}^{m \times m}$. The successful transmissions $\mathbf{r}_m(\mathbf{p})$ resulting from a schedule $\mathbf{p}(t)$ can be computed via the Hadamard product between the schedules and nonnegative projection $[\mathbf{1} - \mathbf{A}_m \mathbf{p}(t)]_+$,

$$\mathbf{r}_m(\mathbf{p}) = \mathbf{p}(t) \odot [\mathbf{1} - \mathbf{A}_m \mathbf{p}(t)]_+. \quad (3)$$

We are interested in the average rate over a time horizon. Wireless link scheduling seeks a policy that maximizes the long-term average of the sum rate $\mathbf{r}_m(\mathbf{p})$ over T time steps. The policy must also satisfy the interference constraints intrinsic to the topology of the network. We incorporate a minimum transmission requirement for each link, $\Delta \in \mathbb{R}_+^m$, such that all links must transmit at least a fraction of time:

$$\begin{aligned} \mathbf{p}^*(1:T) = \arg \max_{\mathbf{s}(t) \in \{0,1\}^K} & \frac{1}{T} \sum_{t=1}^T \mathbf{1}^\top \mathbf{p}(t) \odot [\mathbf{1} - \mathbf{A}_m \mathbf{p}(t)]_+, \\ \text{s.t.} & \frac{1}{T} \sum_{t=1}^T \mathbf{p}(t) \odot [\mathbf{1} - \mathbf{A}_m \mathbf{p}(t)]_+ \geq \Delta. \end{aligned} \quad (4)$$

To solve this constrained optimization task we turn to the Lagrangian dual domain, defining a new learning objective that combines our objective function and a penalty term for constraint violations. Nonetheless, Lagrangian maximizers will not result in workable policies as they will not vary over time. By leveraging the dual descent dynamics, we can incorporate the evolution of the dual variable λ into our learning via state augmentation [14]. More details about the problem formulation and the algorithm can be found in [3].

III. TRAINING BY TRANSFERENCE

Finding optimal solutions to resource allocation problems such as the one in Equation (2) is often computationally prohibitive. Common approaches approximate these solutions by using a parameterized policy $\Phi(\mathbf{x}_m, \mathbf{A}_m; \mathbf{H}_m)$. The objective is to identify the optimal parameters $\mathbf{H}_m^* \in \mathbb{R}^s$ that maximize the network utility while satisfying the expected constraints:

$$\mathbf{H}_m^* = \arg \max_{\mathbf{H}_m \in \mathbb{R}^s} u_0(\mathbf{r}_m), \quad (5)$$

$$\text{s.t. } \mathbf{r}_m = \mathbb{E}[\mathbf{f}(\Phi(\mathbf{x}_m, \mathbf{A}_m; \mathbf{H}_m); \mathbf{x}_m, \mathbf{A}_m)],$$

$$\mathbf{u}(\mathbf{r}_m) \geq \mathbf{0}.$$

In wireless applications, Graph Neural Networks (GNNs) are the preferred architecture for Φ . By training over diverse channel realizations, GNNs learn efficient allocation policies that maintain awareness of the network topology.

Let the graphs used for training have m nodes, and assume we want to execute on graphs with n nodes, such that $m < n$. In this case, the evaluation consists of finding policies $\Phi(\mathbf{x}_n, \mathbf{A}_n; \mathbf{H}_m^*)$. Note that while the descriptions of the network state $(\mathbf{x}_n, \mathbf{A}_n)$ now have n dimensions, the parameter matrix \mathbf{H}_m^* is independent of the number of nodes in the graph. This implies we *can* execute a GNN trained on a small graph on a larger graph, i.e. *transfer* the GNN. Nonetheless, it is necessary to assess whether these policies remain optimal as the network scales. To analyze this, we first present the GNN architecture in Section III-A. Furthermore, we introduce Random Geometric Graphs and their deterministic grid counterparts (Section III-B) to later present our transferability results (Section IV).

A. Graph Neural Networks

A Graph Neural Network is composed of a cascade of layers, each containing a graph convolutional filter and a pointwise nonlinearity γ , with $\gamma: \mathbb{R} \rightarrow \mathbb{R}$. Given a graph signal $\mathbf{x} \in \mathbb{R}^m$, a graph convolutional filter is defined as follows:

$$\mathbf{y} = \sum_{k=0}^{K-1} h_k \mathbf{A}^k \mathbf{x}, \quad (6)$$

where K is the order of the filter and $\{h_k\}_{k=0}^{K-1}$ are the coefficients. The filter is a polynomial on a matrix representation of the graph, $\mathbf{A} \in \mathbb{R}^{m \times m}$.

A single-layer GNN applies the nonlinearity γ to the output of the graph filter in Equation (6). For a GNN with L layers, the l -th layer takes \mathbf{x}_{l-1} , the output of the previous layer, as input signal. This is later processed via γ to obtain \mathbf{x}_l :

$$\mathbf{x}_l = \gamma \left(\sum_{k=0}^{K-1} h_{lk} \mathbf{A}^k \mathbf{x}_{l-1} \right). \quad (7)$$

We can summarize the weights in a matrix $\mathbf{H} \in \mathcal{H}$ as the trainable parameters of the network.

B. Deterministic and Random Geometric Graphs

Define a metric space of size $W \times W$ and place v nodes randomly following a uniform distribution in the space, $\mathcal{V} \sim \mathcal{U}^2(0, W)$. We create an edge between any two nodes that meet that their Euclidean distance is at most a fixed radius r , $\mathcal{E} = \{(i, j) : d(i, j) \leq r\}$. These sets define a Random Geometric Graph (RGG) [15], $\mathbf{G} = (\mathcal{V}, \mathcal{E})$, which appear naturally in wireless settings due to the relation between the fixed radius r and connectivity determined by signal strength. Let $\mathbf{A} \in \mathbb{R}^{v \times v}$ denote its adjacency matrix.

Define a regular lattice in a Euclidean space with v intersections. By placing a node in each intersection and creating links between neighboring nodes following the grid, we can obtain a deterministic grid graph (DGG), $\tilde{\mathbf{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$. Let $\tilde{\mathbf{A}} \in \mathbb{R}^{v \times v}$ denote its adjacency matrix.

RGGs can be seen as a perturbation of DGGs, resulting from adding Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2)$ to the node positions. If the perturbation is small, that is, if the difference between their adjacency matrices meets $\|\mathbf{A} - \tilde{\mathbf{A}}\| < \varepsilon$, an analysis can be done that seamlessly relates RGGs and DGGs [10]. As observed in Figure 1 (see $\sigma = 0$), the conflict graph of a DGG will also be highly regular and can be easily related to the conflict graph of a RGG. This forms the basis of the theoretical analysis we present in Section IV.

IV. TRANSFERABILITY IN CONFLICT RANDOM GEOMETRIC GRAPHS

Consider a DGG $\tilde{\mathbf{G}}$ with m_1 edges. Construct a RGG \mathbf{G} by adding positional noise to $\tilde{\mathbf{G}}$, connecting the nodes given a fixed radius such that it has m_2 edges with $m_1 \simeq m_2$. In order to study the transferability of GNNs from conflict Random Geometric Graphs to conflict Deterministic Grid Graphs, we consider the normalized adjacency matrices of conflict graphs $\tilde{\mathbf{A}}_{m_1} \in \mathbb{R}^{m_1 \times m_1}$, $\mathbf{A}_{m_2} \in \mathbb{R}^{m_2 \times m_2}$.

To assess the similarity between a DGG and a RGG, we are interested in comparing their adjacency matrices. To enable the comparison we zero-pad the normalized adjacency matrices to obtain $\mathbf{A}, \tilde{\mathbf{A}} \in \mathbb{R}^{(m_1+m_2) \times (m_1+m_2)}$. By embedding both graphs into a spatially-aligned vector space, the perturbation captures only the mismatched edges between the two topologies.

We assume that GNN consists of layers cascading the graph filter, which is assumed to be integral Lipschitz continuous and the normalized Lipschitz point-wise nonlinearities [16]. We note that these are reasonable assumptions and have been implemented widely in theory and in practice.

With these definitions, we present in Theorem 1 the result for transferability of GNNs between RGG and DGG.

Theorem 1 *Let $\tilde{\mathbf{G}}$ be a DGG with m_1 edges and \mathbf{G} be a RGG with m_2 edges. Define $\mathbf{A}, \tilde{\mathbf{A}} \in \{0, 1\}^{(m_1+m_2) \times (m_1+m_2)}$ the zero padded adjacency matrices of their conflict graphs, such that $\|\mathbf{A} - \tilde{\mathbf{A}}\| \leq \varepsilon$. Let $\Phi(\mathbf{x}, \mathbf{A}; \mathbf{H})$ be a L -layer GNN with a graph filter \mathbf{H} that is integral Lipschitz with constant C and a normalized Lipschitz nonlinearity γ . The GNN can be transferred between $\tilde{\mathbf{A}}$ and \mathbf{A} with minimal performance loss:*

$$\|\Phi(\mathbf{x}, \tilde{\mathbf{A}}; \mathbf{H}) - \Phi(\mathbf{x}, \mathbf{A}; \mathbf{H})\| \leq 2L\sqrt{\varepsilon C} \|\mathbf{x}\|. \quad (8)$$

Proof. See [17]. ■

Theorem 1 shows that the error between running a GNN on a DGG and a RGG is small, provided the graphs are close.

We then analyze the transferability of GNN trained on a DGG of size m to a DGG of size n , with $m < n$. This is an extension of the transferability studies of CNNs [18]. Let these two graphs be described via their normalized adjacency matrices $\tilde{\mathbf{A}}_m \in \mathbb{R}^{m \times m}$, $\tilde{\mathbf{A}}_n \in \mathbb{R}^{n \times n}$.

Let $\mathcal{L}(\tilde{\mathbf{A}}_m, \mathbf{H})$ be the performance measure associated with running a GNN on $\tilde{\mathbf{A}}_m$ [18]. We assume an input \mathbf{x} and compare the GNN to the optimal policy \mathbf{p}^* .

$$\mathcal{L}(\tilde{\mathbf{A}}_m, \mathbf{H}) = \frac{1}{m} \|\Phi(\mathbf{x}, \tilde{\mathbf{A}}_m; \mathbf{H}) - \mathbf{p}^*\|^2 \quad (9)$$

Similarly, let $\mathcal{L}_n(\mathbf{H})$ be the performance when applied to $\tilde{\mathbf{A}}_n$. We compare \mathcal{L}_m and \mathcal{L}_n by considering a window of the large graph, $\tilde{\mathbf{A}}_n$, and comparing it to the small graph, $\tilde{\mathbf{A}}_m$.

Theorem 2 Let $\Phi(\mathbf{x}, \tilde{\mathbf{A}}_n; \mathbf{H})$ be the parameterized policy that achieves a performance loss \mathcal{L}_n when applied on a grid graph with size n and achieves a loss of \mathcal{L}_m when applied on another grid graph with size m . Suppose $n > m$, the difference of these two losses can be bounded as

$$\mathcal{L}(\tilde{\mathbf{A}}_n, \mathbf{H}) \leq \mathcal{L}(\tilde{\mathbf{A}}_m, \mathbf{H}) + C\mathbb{E}[\mathbf{x}^2] + 2\sqrt{\mathcal{L}(\tilde{\mathbf{A}}_m, \mathbf{H})C\mathbb{E}[\mathbf{x}^2]}, \quad (10)$$

where $C = \frac{H_K^2}{m}(2\sqrt{m}K + K^2)$ and $H_K = \sum_{k=0}^{K-1} |h_k| \|\tilde{\mathbf{A}}_m\|_2^k$.

Proof. See [17]. ■

In Theorem 2, the performance difference between the evaluation on small and large DGGs is bounded by a small constant under mild assumptions. As n increases, the interior nodes of both graphs have the same neighborhoods and the boundary effects go to zero.

A GNN trained on small RGGs can be transferred to small DGGs (Theorem 1). Moreover, a GNN that performs close to optimality on small DGGs can maintain this performance for large DGGs (Theorem 2). Finally, because performance in large DGGs will be similar to performance in large RGGs, it follows that a GNN can be trained in small RGGs and transferred to large RGGs with minimal performance loss as long as the RGGs can be seen as a shift of its corresponding DGGs with $\|\mathbf{A} - \tilde{\mathbf{A}}\| \leq \varepsilon$. We summarize this result in the following theorem:

Theorem 3 Under assumptions analogous to Theorem 1, consider an L -layer GNN $\Phi(\mathbf{x}, \mathbf{A}; \mathbf{H})$ trained to minimize the loss \mathcal{L}_m for RGGs of size m , such that $\mathcal{L}_m \leq \zeta$ and suppose the deviation of RGG to its corresponding DGG is bounded by ε . The GNN can be transferred to graphs of size n with minimal performance deterioration:

$$|\mathcal{L}_n - \mathcal{L}_m| = \mathcal{O}\left(\sqrt{\zeta}(\sqrt{\varepsilon}\|\mathbf{x}_n\| + \sqrt{\varepsilon}\|\mathbf{x}_m\|) + \varepsilon\|\mathbf{x}_n\|^2 + \varepsilon\|\mathbf{x}_m\|^2\right) \quad (11)$$

Proof. [17] ■

V. NUMERICAL EXPERIMENTS

We conduct simulations on the wireless link scheduling problem as per the formulation in Equation (4).¹ We observe the transferability of Graph Neural Networks over conflict graphs in practice, evaluating the scalability of the learned policies via comparisons with baselines at scale (Section V-A). Furthermore, we analyze how robust the policies are to different levels of perturbation between a DGG and a RGG (Section V-B).

¹The code used to run the experiments and the implementation details are available for reproducibility https://github.com/romm32/rgg_transferability

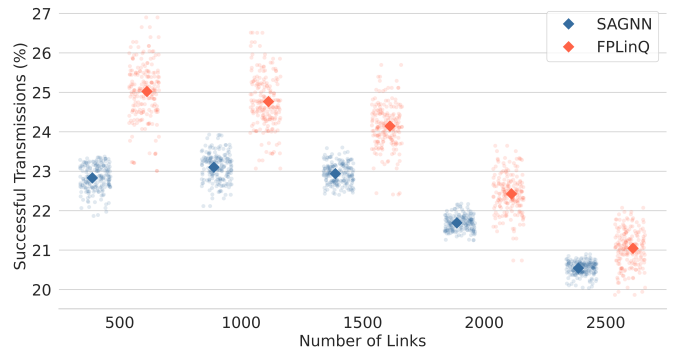


Figure 2: Comparison between the percentage of successfully scheduled links between SAGNN (ours) and FPLinQ. Our model is trained for graphs with $K \simeq 500$ links and transferred to larger scales without significant performance loss.

We consider a GNN with $L = 3$ layers, with leaky ReLU pointwise nonlinearities. The output signal of the last layer is passed through a sigmoid nonlinearity in order to obtain entries between 0 and 1. The performance of the algorithm is measured by binarizing the continuous output of the GNN with a threshold of 0.5 to decide which links to turn on.

We consider $T = 200$ time steps. The percentage of links transmitting a fraction lower than $\Delta = 0.1$ of the time steps (i.e. violating the constraint), along with the long-term average percentage of links transmitting without interference (i.e. the objective function) are the two main performance metrics. All evaluations presented are on datasets unseen during training.

A. Performance at Scale

The policy was trained using 100 graphs with $K \simeq 500$. The graphs are RGGs obtained via DGGs, adding positional noise with $\sigma = 0.01$ (see Figure 1). Figure 2 shows how our State Augmented GNN (SAGNN) transfers across scales. Performance does not degrade as we evaluate on five datasets with different scales, remaining close to 20-25% of active links. Note that we present the average over 100 graphs on each dataset. The values achieved for the total average rates are close to the size of the maximum independent sets of the graphs, which represents the optimal solution to this problem.

We compare our results with FPLinQ [19], a fractional programming approach to solve link scheduling that focuses on instantaneous rate maximization. We perform close to this state-of-the-art baseline and achieve runtimes that are orders of magnitude more efficient (a speed up of 4x for $K \simeq 500$ and an improvement of 30x for $K \simeq 2500$). Furthermore, our GNN-based policy can be implemented in a distributed manner, allowing practical use.

In Figure 3 we show a histogram of the average rates achieved by the different links, comparing both algorithms for scales $K = \{500, 2500\}$. SAGNN diversifies the schedules across all links, with the vast majority achieving rates higher than the 10% transmission requirement. Nonetheless, the FPLinQ baseline repeatedly schedules the same links, not allowing for fair sharing of the resources.

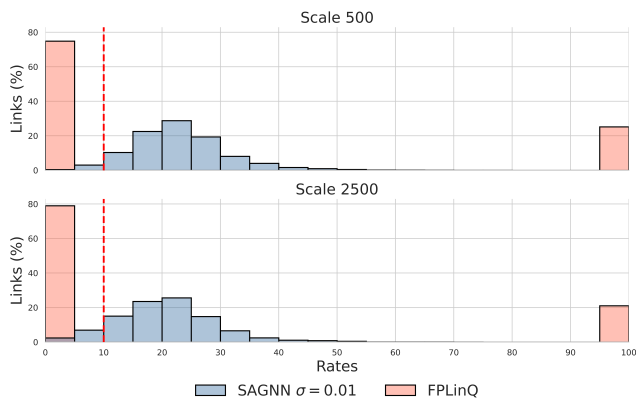


Figure 3: Average rates achieved by different links, presented as the percentage of time T each link is scheduled. The minimum transmission requirement is set to $\Delta = 10\%$ (red).

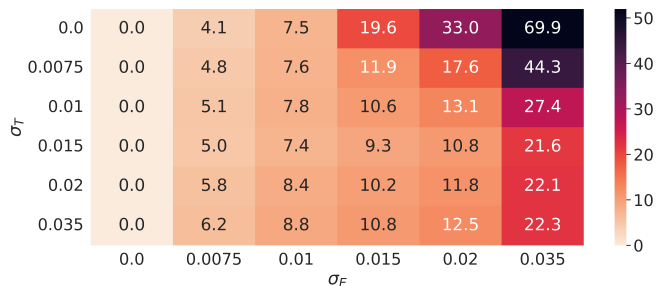


Figure 4: Study of the robustness of policies trained for a given level of position noise σ_T and evaluated on σ_E . We present the percentage of links violating the constraint.

B. Robustness to Different Perturbation Levels

Our theoretical results in Theorem 3 assume that the RGG must be close enough to a DGG, i.e. the perturbation level must be small. This can be translated in practice to adding positional noise with low variance σ .

In Figure 4 we show the constraint violation results of training six models, one for each of the noise levels illustrated in Figure 1. Each of these models is evaluated on six datasets containing 100 graphs each, generated by adding different levels of positional noise to a DGG. As expected, we see that models trained with lower noise levels ($\sigma_T = \{0, 0.0075, 0.01\}$) generalize worse to higher levels noise levels ($\sigma_E = \{0.015, 0.02, 0.035\}$). Models trained for larger noise levels achieve good performance across all noise levels.

VI. CONCLUSIONS

Random Geometric Graphs are a simple, yet accurate model of wireless networks. This work provides the theoretical study of transferability of Graph Neural Networks on conflict graphs derived from sparse RGG. This ensures wireless policies trained to satisfy requirements on the links of a network can be transferred to larger graphs with minimal loss in

performance. We verified our theoretical results hold with numerical experiments on the wireless link scheduling task. It remains crucial to better assess the relevance of our assumptions with respect to practical development, which we leave for future work.

REFERENCES

- [1] M. Randall, S. Paternain, P. Casas, F. Larroca, and P. Belzarena, "User association in wireless networks with distributed gnn-based reinforcement learning," in *2025 12th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 2025, pp. 352–360.
- [2] Z. Wang, M. Eisen, and A. Ribeiro, "Learning decentralized wireless resource allocations with graph neural networks," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1850–1863, 2022.
- [3] R. G. Camargo, Z. Wang, N. NaderiAlizadeh, and A. Ribeiro, "Wireless link scheduling with state-augmented graph neural networks," 2025. [Online]. Available: <https://arxiv.org/abs/2505.07598>
- [4] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE transactions on signal processing*, vol. 68, pp. 2977–2991, June 2020.
- [5] L. Ruiz, L. F. O. Chamon, and A. Ribeiro, "Transferability properties of graph neural networks," *IEEE Transactions on Signal Processing*, vol. 71, pp. 3474–3489, 2023.
- [6] S. Maskey, R. Levie, and G. Kutyniok, "Transferability of graph neural networks: An extended graphon approach," *Applied and Computational Harmonic Analysis*, vol. 63, pp. 48–83, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1063520322000987>
- [7] N. Keriven, A. Bietti, and S. Vaiter, "Convergence and stability of graph convolutional networks on large random graphs," *Advances in Neural Information Processing Systems*, vol. 33, pp. 21 512–21 523, 2020.
- [8] Z. Wang, L. Ruiz, and A. Ribeiro, "Geometric graph filters and neural networks: Limit properties and discriminability trade-offs," *IEEE Transactions on Signal Processing*, 2024.
- [9] T. Le and S. Jegelka, "Limits, approximation and size transferability for gnns on sparse graphs via graphops," in *Advances in Neural Information Processing Systems*, 2023, pp. 41 305–41 342.
- [10] R. G. Camargo, Z. Wang, and A. Ribeiro, "Graph neural networks in large scale wireless communication networks: Scalability across random geometric graphs," 2025. [Online]. Available: <https://arxiv.org/abs/2510.00896>
- [11] Z. Zhao, G. Verma, C. Rao, A. Swami, and S. Segarra, "Link scheduling using graph neural networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 6, pp. 3997–4012, 2023.
- [12] J. Cao, T. Peng, X. Liu, W. Dong, R. Duan, Y. Yuan, W. Wang, and S. Cui, "Resource allocation for ultradense networks with machine-learning-based interference graph construction," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2137–2151, 2020.
- [13] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.
- [14] M. Calvo-Fullana, S. Paternain, L. F. O. Chamon, and A. Ribeiro, "State augmented constrained reinforcement learning: Overcoming the limitations of learning with rewards," *IEEE Transactions on Automatic Control*, vol. 69, pp. 4275–4290, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:232035640>
- [15] M. Penrose, *Random geometric graphs*. OUP Oxford, 2003, vol. 5.
- [16] L. Ruiz, F. Gama, and A. Ribeiro, "Graph neural networks: Architectures, stability, and transferability," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021.
- [17] R. G. Camargo, Z. Wang, and A. Ribeiro, "Limit analysis of graph neural networks with wireless conflict graphs," 2026, <https://zhiyangwang.com/Papers/SPAWC2026.pdf>.
- [18] D. Owerko, C. I. Kanatsoulis, J. Bondarchuk, D. J. B. Jr, and A. Ribeiro, "Transferability of convolutional neural networks in stationary learning tasks," 2023.
- [19] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *2017 IEEE international symposium on information theory (ISIT)*. IEEE, 2017, pp. 2323–2327.