GRAPH NEURAL NETWORKS IN LARGE SCALE WIRELESS COMMUNICATION NETWORKS: SCALABILITY ACROSS RANDOM GEOMETRIC GRAPHS

Romina Garcia Camargo[*], *Zhiyang Wang*[†], *Alejandro Ribeiro*[*]

[*]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, USA
[†] Halıcıoğlu Data Science Institute, University of California San Diego, La Jolla, USA

ABSTRACT

The growing complexity of wireless systems has accelerated the move from traditional methods to learning-based solutions. Graph Neural Networks (GNNs) are especially well-suited here, since wireless networks can be naturally represented as graphs. A key property of GNNs is transferability: models trained on one graph often generalize to much larger graphs with little performance loss. While empirical studies have shown that GNN-based wireless policies transfer effectively, existing theoretical guarantees do not capture this phenomenon. Most works focus on dense graphs where node degrees scale with network size—an assumption that fails in wireless systems. In this work, we provide a formal theoretical foundation for transferability on Random Geometric Graphs (RGGs), a sparse and widely used model of wireless networks. We further validate our results through numerical experiments on power allocation, a fundamental resource management task.

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

1. INTRODUCTION

The use of machine learning to construct efficient wireless resource allocations has become popular. Graph Neural Networks (GNNs) have emerged as a particularly effective architecture. Communication networks can often be modeled as graphs, with devices represented as nodes and their interactions as edges. This natural correspondence makes GNNs a strong candidate for tackling complex tasks in wireless systems [1–4].

GNNs consist of stacked layers, each combining a graph convolutional filter with a point-wise nonlinearity [5–8]. Their adoption in wireless communication is motivated by properties such as permutation equivariance and stability [9–11]. Permutation equivariance enables learning independently of node labeling, improving data efficiency. Stability ensures that perturbations in the input lead to controlled perturbations in the output. These properties support learning policies that generalize across diverse and dynamic wireless network configurations.

A phenomenon of particular importance in wireless communication is transferability: policies trained on networks of one size often generalize to much larger networks with minimal performance loss. This effect has been observed empirically [1, 2, 12, 13] and is in fact not unique to communications, but a widely studied property of GNNs [14–18]. Existing theoretical works consider graphs in the limit, commonly via graphons [14] or manifolds [17], where node degrees grow with network size, yielding dense or relatively sparse graphs. Remarkably, these analyses do not extend to wireless systems, where node degrees remain bounded by physical constraints. In addition, most approaches rely on abstract random models and neglect the geometric structure intrinsic to communication networks. Addressing the gap forms the core contribution of our work.

We develop a theoretical framework for analyzing the transferability of GNN-based resource-allocation policies in wireless networks. The key idea is to relate Random Geometric Graphs (RGGs), which naturally model wireless topologies via random node placements with radius-based connectivity [19, 20], to Deterministic Grid Graphs (DGGs), whose regular structure admits a clear connection to the transferability exploited by convolutional neural networks [21]. Using DGGs as a surrogate "bridge," we quantify the difference between an RGG and a DGG with same size and density through a simple measure of matrix difference. This pair lets us derive transfer guarantees across scales, that is, if the RGG–DGG difference is sufficiently small, then a policy learned at one scale transfers to other scales of RGGs with a provably bounded loss (Theorem 3). In particular, the transfer loss grows at most linearly with the RGG–DGG discrepancy. We validate the theory on the classical power-allocation task, training a GNN policy at one network scale and evaluating it across varying sizes. This contributes to both the theory of GNNs over sparse graphs and the practical applications of GNNs.

2. RESOURCE ALLOCATION WITH GRAPH NEURAL NETWORKS

We consider the problem of resource allocation in a wireless network with n users. At each time slot t, user i is associated with a state $[\mathbf{x}(t)]_i$ (e.g., queue length or priority), summarized in the vector $\mathbf{x}(t) \in \mathbb{R}^n$. The channel gain from user i to user j is denoted $s_{ij}(t)$, with all channel gains collected in $\mathbf{S}(t) \in \mathbb{R}^{n\times n}$. The resource allocation policy is represented by $\mathbf{p} \in \mathbb{R}^n$. At each time step, the controller observes the network states $(\mathbf{x}(t), \mathbf{S}(t))$, selects the allocation strategy $\mathbf{p}(t)$, and receives a reward determined by the system. We define the expected reward $\mathbf{f}(\mathbf{p}(t); \mathbf{x}(t), \mathbf{S}(t))$ as the system reward and focus on the long-term average performance:

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

where the expectation is taken over the stationary joint distribution of $(\mathbf{x}, \mathbf{S})$. This captures user experience under fast time-varying channels and states. The goal is to design a policy $\mathbf{p}(\mathbf{x}, \mathbf{S})$ that maximizes expected reward (1). We introduce a utility function $u_0(\mathbf{r})$ to

Thanks to XYZ agency for funding. [RG: what here?]

formulate the optimization problem:

$$\begin{aligned} \mathbf{p}^\star(\mathbf{S},\mathbf{x}) = & \operatorname*{argmax}_{\mathbf{p}(\mathbf{x},\mathbf{S})\in\mathcal{P}(\mathbf{x},\mathbf{S})} \; u_0(\mathbf{r}), \\ \text{s.t. } \mathbf{r} = & \mathbb{E}[\mathbf{f}(\mathbf{p}(\mathbf{x},\mathbf{S});\mathbf{x},\mathbf{S})], \\ & \mathbf{u}(\mathbf{r}) \geq \mathbf{0}, \end{aligned} \tag{2}$$

where $\mathbf{u}(\cdot)$ captures long-term system constraints (e.g., power budgets). This formulation is challenging as the objective is often nonconvex in $\mathbf{p}$. To address this, we introduce a parameterized policy $\mathbf{\Phi}(\mathbf{x},\mathbf{S};\mathbf{H})$, with parameters $\mathbf{H}\in\mathbb{R}^s$. The problem becomes

$$\begin{aligned} \mathbf{H}^\star = & \operatorname*{argmax}_{\mathbf{H}\in\mathbb{R}^s} \; u_0(\mathbf{r}), \\ \text{s.t. } \mathbf{r} = & \mathbb{E}[\mathbf{\Phi}(\mathbf{x},\mathbf{S};\mathbf{H});\mathbf{x},\mathbf{S})], \\ & \mathbf{u}(\mathbf{r}) \geq \mathbf{0}, \end{aligned} \tag{3}$$

The focus thus shifts from computing allocations directly to learning the parameters $\mathbf{H}$ of a policy class $\mathbf{\Phi}$. In this work, $\mathbf{\Phi}$ will be instantiated as a Graph Neural Network, which we describe next. For more details on the problem formulation see [1].

2.1. Graph Neural Networks

A graph convolutional filter is a polynomial on a matrix representation of the graph. Considering a graph signal $\mathbf{x}\in\mathbb{R}^n$ (i.e. a vector supported on the nodes of a graph), we define a graph filter of order K as follows [5, 22–24]:

$$\mathbf{y} = \sum_{k=0}^{K-1} h_k \mathbf{S}^k \mathbf{x}, \tag{4}$$

where $\{h_k\}_{k=0}^{K-1}$ are the filter coefficients and $\mathbf{S}\in\mathbb{R}^{n\times n}$ is the matrix representation of the graph, commonly referred to as the graph shift operator (GSO) [25]. As the GSO is symmetric, it is possible to diagonalize it as $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$, with $\mathbf{V}$ a matrix with the eigenvectors and $\mathbf{\Lambda}$ a diagonal matrix with the eigenvalues, namely $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \cdots, \lambda_K]$. Through a change of basis it is possible to obtain the spectral representation of the graph convolution, also denoted the graph frequency response of the filter:

$$\hat{h}(\lambda) = \sum_{k=0}^{K-1} h_k \lambda^k. \tag{5}$$

Graph Neural Networks (GNNs) are composed of multiple layers, each combining a graph convolutional filter with a point-wise nonlinearity σ, with $\sigma:\mathbb{R}\to\mathbb{R}$. At the l-th layer, the filter takes as input graph signal $\mathbf{x}_{l-1}\in\mathbb{R}^d_{l-1}$, the output of the previous layer. This signal is then passed through σ:

$$\mathbf{x}_l = \sigma\left(\sum_{k=0}^{K-1} h_{lk}\mathbf{S}^k\mathbf{x}_l\right). \tag{6}$$

This process is repeated across L layers. The full set of trainable parameters is denoted $\mathbf{H}\in\mathcal{H}$, comprising all h_{lk} for $l\in 1,\dots,L$ and $k\in 0,\dots,K-1$. Importantly, the dimensionality of $\mathbf{H}$ does not depend on the size of the graph.

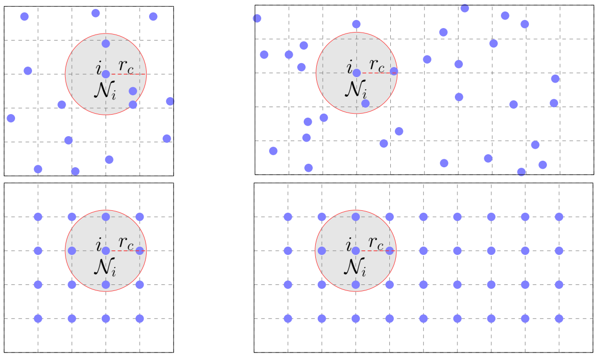

Fig. 1: Visualization of random geometric graphs as perturbations of deterministic grid graphs. **Top:** Illustrations of a small RGG (left) and large RGG (right). **Bottom:** Illustrations of a small DGG (left) and a large DGG (right).

2.2. Random Geometric Graphs

A deterministic grid graph (DGG) $\mathbf{G}_n = (\mathcal{D}_n, \mathcal{E})$ is a graph defined on a regular lattice in a Euclidean space. Each of the n nodes corresponds to a lattice point, and edges connect nodes that are direct neighbors on the grid. A Random Geometric Graph (RGG) [19] is an undirected graph $G_n^r = (\mathcal{V}_n, \mathcal{E}_r)$ constructed by placing n nodes uniformly at random in a metric space of size $L_e \times L_e$:

$$\mathcal{V}_n \sim \mathcal{U}^2(0, L_e). \tag{7}$$

This uniform placement captures the *random* aspect. An edge $(i,j)\in\mathcal{E}_r$ is included whenever the Euclidean distance between nodes i and j is at most a fixed connection radius r_c:

$$\mathcal{E}_r = \{(i,j) : d(i,j) \leq r_c\}, \tag{8}$$

which explains the *geometric* aspect. Suppose the density of the agents over the space is fixed as ρ, the expected number of neighbors of each agent is $\pi r_c^2/\rho$, which is also the average vertex degree. RGGs naturally arise in wireless communication settings, where nodes represent users or devices and connectivity within a fixed radius approximates feasible links determined by signal strength and interference.

Let $\mathbf{S}_{\mathcal{D}_n}, \mathbf{S}_n \in \mathbb{R}^{n\times n}$ denote the adjacency matrices of $\mathbf{G}_n$, $\mathbf{G}_n^r$ respectively. If the norm difference between these matrices is sufficiently small, a RGG can be viewed as a perturbation of a DGG, obtained by adding Gaussian noise $\eta\sim\mathcal{N}(0,\sigma^2)$ to the node positions. This comparison, illustrated in Figure 1 for a radius r_c, forms the basis of our theoretical framework.

We first show that GNNs can transfer across scales on grid graphs from $\mathbf{G}_n$ to $\mathbf{G}_m$ (Theorem 1). Next, we prove that a GNN trained on an RGG $\mathbf{G}_n^r$ close enough to a DGG $\mathbf{G}$ transfers with little performance loss (Theorem 2). Reversing the perturbation then yields transferability from $\mathbf{G}_m$ to a larger RGG $\mathbf{G}_m^r$ (Theorem 3).

3. TRANSFERABILITY OF GRAPH NEURAL NETWORKS IN WIRELESS COMMUNICATION NETWORKS

We begin by establishing transferability across scales for deterministic grid graphs. Consider two DGGs, $\mathbf{G}_n$ and $\mathbf{G}_m$, with normalized adjacency matrices $\mathbf{S}_{\mathcal{D}_n}$ and $\mathbf{S}_{\mathcal{D}_m}$, where $n < m$. The adjacency matrix of a regular grid graph is circulant, with identical non-zero entries. This structure makes it possible to reinterpret the graph convolutional operation on a grid graph as a standard 2-D convolution, provided the nodes are indexed by their 2-D coordinates. Suppose that $n = B \times B$ with $B \in \mathbb{N}^+$ [1], we reform the state matrix $\mathbf{x} \in \mathbb{R}^n$ as a 2-d discrete function as

$$x_B(n_1, n_2) = [\mathbf{x}]_{n_1 + n_2 \times B} \in \mathbb{R}, \tag{9}$$

with $n_1, n_2 = 0, 1, \cdots, B-1$ representing the 2-d coordinates of each grid node. The graph convolution operation as a process of aggregating information from neighbors, is actually the same operation with the entries of the mask matrix equal to the non-zero entries of the adjacency matrix $\mathbf{S}_{\mathcal{D}_n}$. The size of this mask matrix is related to the degree of each node, i.e. decided by r_c. We denote this mask matrix as $\mathbf{L} \in \mathbb{R}^{M \times M}$, with $M = \lceil \sqrt{\pi r_c^2/\rho + 1} \rceil$, which can be defined based on $\mathbf{S}_{\mathcal{D}_n}$. When we see the graph filter operating on the grid graphs, we can see it as a 2-D convolution operation. With the one-step aggregation rewritten as

$$\begin{aligned} x_{B,1}(n_1, n_2) &= \mathbf{L} \otimes x_B(n_1, n_2) && (10)\\ &= \sum_{k_1=0}^{M-1} \sum_{k_2=0}^{M-1} \mathbf{L}(k_1, k_2) x_B(n_1 - k_1, n_2 - k_2), && (11) \end{aligned}$$

which also collects signals over the neighboring nodes. Analogously, the k-step aggregation

$$x_{B,k}(n_1, n_2) = \mathbf{L} \otimes x_{B,k-1}(n_1, n_2) = (\mathbf{L}\otimes)^k x_B(n_1, n_2). \tag{12}$$

The graph convolution operation can be recovered by scaling $x_{B,k}$ with h_k and summing up all the aggregated information, which is written as

$$y_{\mathbf{L},B} = \sum_{k=0}^{K} h_k (\mathbf{L}\otimes)^k x_B := \mathbf{h}_D(\mathbf{L}, x_B). \tag{13}$$

Followed by a point-wise nonlinearity, this could recover the parameterization of $\mathbf{\Phi}(\mathbf{x}, \mathbf{S}_{\mathcal{D}_n}; \mathbf{H})$ with L layers. We assume that the input signal $\mathbf{x}$ and the output performance $\mathbf{r}$ in (1) is jointly stationary over the 2-D space. We use the evaluation metric as the performance difference between the performance achieved by the learned policy $\mathbf{r}_n$ and the optimal policy $\mathbf{r}_n^*$, defined as

$$\mathcal{L}_n = \frac{1}{n} \| \mathbf{r}_n(\mathbf{\Phi}(\mathbf{x}, \mathbf{S}_{\mathcal{D}_n}; \mathbf{H})) - \mathbf{r}_n^* \|^2. \tag{14}$$

Theorem 1. Let 2-D convolutional neural network (i.e. GNN over a grid graph) 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

$$\mathcal{L}_m \leq \mathcal{L}_n + C_M \mathbb{E}[\|\mathbf{x}\|^2] + \sqrt{\mathcal{L}_n C_M \mathbb{E}[\|\mathbf{x}\|^2]}, \tag{15}$$

with the input and output signals are jointly stationary and bounded. $C_M = \frac{H_K^2}{n}[2\sqrt{n}KM + K^2M^2]$ with $H_K = \sum_{l=0}^K \sum_{k=0}^K |h_{lk}| \|\mathbf{L}\|_1^k$. When the neural network is trained on the grid graph with size n, i.e. $\mathcal{L}_n \leq \epsilon$, the loss achieved by implementing the trained neural network on the grid graph with size m.

Proof. See Appendix 1 in [26].

Under this interpretation, the transferability of CNNs extends naturally to grid graphs with fixed density as the number of nodes increases. This observation provides the bridge to GNNs on RGGs if we establish the connection between GNNs on grid graphs and on RGGs with the same scale and density.

To study transferability for a GNN trained on an RGG $\mathbf{G}_n^r$ with normalized adjacency matrix $\mathbf{S}_n$ to the grid graph $\mathbf{G}_n$, we introduce the following definition and assumption.

Definition 3.1. (Integral Lipschitz continuous filter) A filter $\hat{h}$ is integral Lipschitz continuous with constant C if its frequency response satisfies

$$|\hat{h}(a) - \hat{h}(b)| \leq \frac{C|a-b|}{(a+b)/2} \text{ for all } a, b \in (0, \infty). \tag{16}$$

Assumption 3.1. (Normalized Lipschitz nonlinearity) The nonlinearity σ is normalized Lipschitz continuous, i.e., $|\sigma(a) - \sigma(b)| \leq |a - b|$, with $\sigma(0) = 0$.

We note that this assumption is reasonable, since most common nonlinearity functions are normalized Lipschitz. We assume that the graph filters used in the GNN are integral Lipschitz continuous as defined in Definition 3.1. Furthermore, we assume that the difference between the RGG and DGG matrices $\mathbf{S}_n - \mathbf{S}_{\mathcal{D}_n}$ is small.

We define the performance metric on RGGs similar to (14) as

$$\mathcal{L}_n^r = \frac{1}{n} \| \mathbf{r}(\mathbf{\Phi}(\mathbf{x}_n, \mathbf{S}_n; \mathbf{H})) - \mathbf{r}_n^{r*} \|^2, \tag{17}$$

which is the comparison between the performance on the learned policy $\mathbf{\Phi}(\mathbf{x}_n, \mathbf{S}_n; \mathbf{H})$ and the optimal performance. We can now conclude the transferability of GNNs from RGG to DGG in the form of Theorem 2.

Theorem 2. Let $\mathbf{\Phi}(\mathbf{x}, \mathbf{S}; \mathbf{H})$ be an 1-layer GNN applied on a random geometric graph $\mathbf{G}_n^r$ and a grid graph $\mathbf{G}_n$. We define $\mathcal{W}_n = \mathbf{S}_n - \mathbf{S}_{\mathcal{D}_n}$ such that $\mathbb{E}[\|\mathcal{W}_n^2\|] = O(n^{-\alpha})$ with $\alpha > 0$. Suppose that the GNN is trained on $\mathbf{G}_n^r$ with $\mathcal{L}_n^r \leq \epsilon$, The difference of the outputs of GNN with input graph signal $\mathbf{x} \in \mathbb{R}^n$ can be bounded as

$$|\mathcal{L}_n - \mathcal{L}_n^r| \leq C^2 n^{1-\alpha} \|\mathbf{x}\|^2 + 2\sqrt{\epsilon} C n^{\frac{1-\alpha}{2}} \|\mathbf{x}\|. \tag{18}$$

Proof. See Appendix 2 in [26].

This proves that the difference of the performances of a GNN on a DGG $\mathbf{G}_n$ and on a RGG $\mathbf{G}_n^r$ can be bounded.

We have shown that GNNs on RGGs can transfer to DGGs with the same number of nodes when the adjacency matrices are close enough in Theorem 2. Theorem 1 further proves that GNNs transfer on DGGs with different number of nodes. The transference of GNNs on RGGs with different number of nodes can therefore be derived based on the triangle inequality.

Theorem 3. Let $\mathbf{\Phi}(\mathbf{x}, \mathbf{S}; \mathbf{H})$ be a L-layer GNN applied on a graph with GSO $\mathbf{S}$ and input $\mathbf{x}$. Suppose there are two random geometric graphs $\mathbf{G}_n^r$ with adjacency matrix $\mathbf{S}_n$ and $\mathbf{G}_m^r$ with adjacency matrix $\mathbf{S}_m$, such that $n < m$. The network $\mathbf{\Phi}$ has been trained to minimize $\mathcal{L}_n^r \leq \epsilon$. We take $\alpha = 2$ and omit the terms that have order smaller than $\sqrt{\epsilon}$,

$$|\mathcal{L}_n^r - \mathcal{L}_m^r| = \\ \mathcal{O}\left(\sqrt{\epsilon} \left(n^{-1/2} \|\mathbf{x}_n\| + m^{-1/2} \|\mathbf{x}_m\| \right) + n^{-1} \|\mathbf{x}_n\|^2 + m^{-1} \|\mathbf{x}_m\|^2 \right). \tag{19}$$

[1] Here n could be decomposed in a general form $n = P \times B$ with zero-padding, we use this squared form for the ease of presentation.

Proof. See Appendix 3 in [26].

We can see from the theorem that a GNN trained on a small RGG (a small wireless network) can be transferred to a larger RGG (a larger wireless network) with the trained policy approximating the optimal policy well enough. The difference between these two performances decreases with the size of these two networks and depends on the spectral continuity of the filter functions in the GNN. This attests that the trained policy over a wireless network modeled as a random geometric graph, i.e. graph with limited degree, can be transferred across scales without retraining. This fills the theoretical gap of analyzing the transferability of GNNs across sparse random geometric graphs. We verify this conclusion in a real-world power allocation scenario in the following.

4. NUMERICAL EXPERIMENTS

We present numerical simulations for the power allocation problem that support our theoretical results. The utility function u_0 is defined as the sum rate, where the rates r are as follows:

$$r_i := \log\left(1 + \frac{|h_{ij}|^2 \mathbf{p}_i(\mathbf{x}, \mathbf{H})}{\eta^2 + \sum_{k \neq i} |h_{kj}|^2 \mathbf{p}_k(\mathbf{x}, \mathbf{H})}\right). \tag{20}$$

We consider the capacity that each transmitter experiences over the noise η^2 introduced in the AWGN channel and the interference caused by other users. We seek to maximize the expectation of the sum capacity over channel realizations. Assuming a power budget P_{max}, we formulate a simplified version of (2).

$$\begin{aligned} \mathbf{p}^*(\mathbf{x}, \mathbf{H}) = \underset{\mathbf{p}(\mathbf{x},\mathbf{H}) \in \{0, p_0\}^n}{\operatorname{argmax}} \quad & \sum_{i=1}^{n} r_i \qquad (21) \\ \text{s.t.} \quad & \mathbb{E}[\mathbf{1}^\mathsf{T} \mathbf{p}(\mathbf{x}, \mathbf{H})] \leq P_{max} \\ & \mathbf{p}(\mathbf{x}, \mathbf{H}) \in \{0, p_0\}^n. \end{aligned}$$

The solution to (21) can be obtained by defining a learning parameterization and operating in the Lagrangian dual domain to obtain the optimal policy. The approach is analogous to the one seen in [1]. The key difference lies in the construction of the channel matrix, which is aligned with the structure of a weighted adjacency matrix.

We construct RGGs for training datasets as perturbations of DGGs, as described in Section 2.2. Two-dimensional Gaussian noise is added to perturb the grid graphs, obtaining RGGs representative of communication networks. Isolated nodes were removed from the network, which resulted in uneven average number of nodes in each dataset. The channel matrix was created considering a path loss coefficient and a fading channel gain. Different scales were considered to evaluate transferability. Each dataset consists of 100 graphs with an average link count of $n_k \simeq 500 + 100k$, for $k = \{0, 1, \ldots, 7\}$. More details from the implementation, such as the architecture and hyperparameters, can be seen in the repository of the project.[2]

We sample power assignments interpreting the output of the GNN as probability of assignment and sampling Bernoulli variables for the binary allocation. We consider a policy variation of the heuristic baseline *WMMSE* [27], using its outputs as probabilities to sample Bernoulli trials. The results for the evaluation of our algorithm on unseen data can be seen in Table 1. For the power constraint, we compute $\mathbf{1}^\mathsf{T}\mathbf{p}(\mathbf{x}, \mathbf{H}) - P_{max,k}$ for each dataset, dividing over n_k to get average per-node power budget violation. It can be seen that our model outperforms the baseline, successfully finding optimal allocation policies.

	Sum rate	Power constraint
WMMSE	276.70 ± 9.62	$(-1.26 \pm 1.94) \times 10^{-2}$
GNN	771.91 ± 6.63	$(-3.69 \pm 1.29) \times 10^{-2}$

Table 1: Performance against WMMSE. We present mean and standard deviation across 10 experiments over 100 unseen graphs.

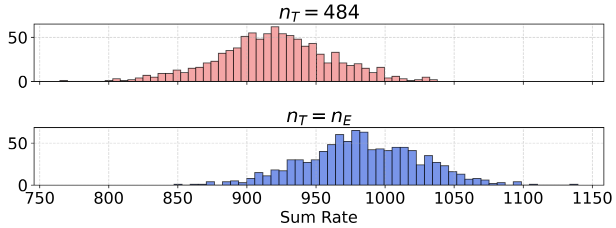

Fig. 2: Empirical distribution of sum rate achieved for 10 experiments. Comparison of a GNN trained for $n \simeq 500$ and a GNN trained for $n \simeq 600$, both evaluated on a test dataset with $n \simeq 600$.

Figures 2 and 3 show transferability results for a model trained with $n \simeq 500$. For comparison, we trained separate models on each scale to evaluate in-distribution performance. Results are reported on graphs unseen during training. The transferred model performs on par with scale-specific models while maintaining low constraint violations. Overall, we observe a favorable trade-off between achieving high rates and avoiding both over- and under-allocation.

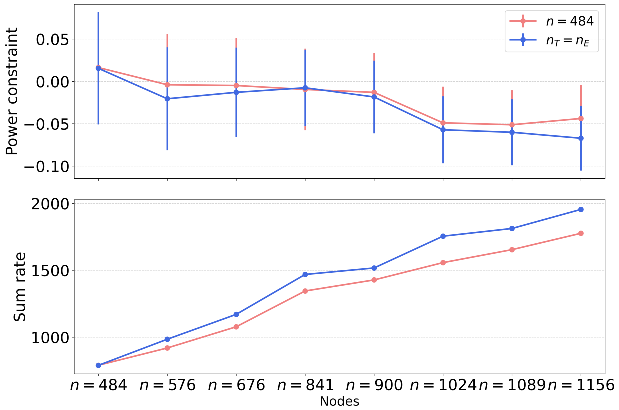

Fig. 3: Sum rate and power constraint values for networks of different scales with models trained in-distribution ($n_T = n_E$) and with a transferred model trained for $n = 484$.

5. CONCLUDING REMARKS

We presented a theoretical analysis of the transferability of wireless resource allocation policies using graph neural networks on random geometric graphs, and supported the findings with numerical experiments. Future work includes developing a more rigorous theoretical framework. It would also be valuable to investigate how transfer-

[2] The code implementation can be found in `https://github.com/romm32/rgg_transferability`

ability degrades when the underlying assumptions are not satisfied.

A. APPENDIX

A.1. Proof of Theorem 1

A.1.1. Transferability of Filters between Grid Graphs

Suppose there is a discrete random stationary signal over the 2-d space $f : \mathbb{N} \to \mathbb{R}$, the input signal is a narrow window of this signal as $x_B = \sqcap_B f$, where $\sqcap_B(n_1, n_2) = \mathbb{1}(0 \leq n_1 \leq B, 0 \leq n_2 \leq B)$. Next, we propose the transference of CNNs over different scales of grid graphs.

Proposition A.1. Let $\mathbf{h}_D(\cdot)$ be a 2-d convolutional filter as defined in (13). The output difference of the filter on a grid graph with size $n = B_1 \times B_1$ and another grid graph with $m = B_2 \times B_2$ can be bounded as

$$\begin{aligned}\mathbb{E}[\| \sqcap_{B_1} (\mathbf{h}_D(\mathbf{L}, x_{B_1}) - \mathbf{h}_D(\mathbf{L}, x_{B_2}))\|_2^2] \\ \leq C_K^2 (B_2^2 - B_1^2) \mathbb{E}[f(0,0)^2],\end{aligned} \tag{22}$$

with $C_K = \sum\limits_{k=1}^{K} \|\mathbf{L}\|_1^k |h_k|$.

Proof. According to the definition of 2-d convolutional filter in (13), the difference can be written as

$$\begin{aligned}&\| \sqcap_{B_1} (\mathbf{h}_D(\mathbf{L}, x_{B_1}) - \mathbf{h}_D(\mathbf{L}, x_{B_2}))\| \\ &= \| \sqcap_{B_1} (\mathbf{h}_D(\mathbf{L}, \sqcap_{B_1} f) - \mathbf{h}_D(\mathbf{L}, \sqcap_{B_2} f))\| && (23)\\ &= \Bigg\| \sqcap_{B_1} \left(\sum_{k=0}^{K} h_k (\mathbf{L}\otimes)^k (\sqcap_{B_1} f) \right) - && (24)\\ &\qquad \sqcap_{B_1} \left(\sum_{k=0}^{K} h_k (\mathbf{L}\otimes)^k (\sqcap_{B_2} f) \right) \Bigg\| && (25)\end{aligned}$$

With triangle inequality, we have

$$\begin{aligned}&\| \sqcap_{B_1} (\mathbf{h}_D(\mathbf{L}, x_{B_1}) - \mathbf{h}_D(\mathbf{L}, x_{B_2}))\| \\ &\leq \sum_{k=0}^{K} \| \sqcap_{B_1} h_k (\mathbf{L}\otimes)^k (\sqcap_{B_1} f) - \sqcap_{B_1} h_k (\mathbf{L}\otimes)^k (\sqcap_{B_2} f)\| && (26)\\ &\leq \sum_{k=0}^{K} \|h_k (\mathbf{L}\otimes)^k (\sqcap_{B_1} f) - h_k (\mathbf{L}\otimes)^k (\sqcap_{B_2} f)\|. && (27)\end{aligned}$$

With Young's convolution inequality

$$\|h_k \mathbf{L} \otimes (\sqcap_{B_1} f)\| \leq \|\mathbf{L}\|_1 \|h_k (\sqcap_{B_1} f)\|, \tag{28}$$

we have

$$\begin{aligned}&\| \sqcap_{B_1} (\mathbf{h}_D(\mathbf{L}, x_{B_1}) - \mathbf{h}_D(\mathbf{L}, x_{B_2}))\| \\ &\leq \sum_{k=0}^{K} \|\mathbf{L}\|_1^k |h_k| \|(\sqcap_{B_1} f) - (\sqcap_{B_2} f)\| && (29)\\ &= C_K \|(\sqcap_{B_1} f) - (\sqcap_{B_2} f)\|, && (30)\end{aligned}$$

if $B_1 + MK \geq B_2$. With expectation, we have

$$\begin{aligned}&\mathbb{E}[\| \sqcap_{B_1} (\mathbf{h}_D(\mathbf{L}, x_{B_1}) - \mathbf{h}_D(\mathbf{L}, x_{B_2}))\|^2] \\ &\leq C_K^2 \mathbb{E}[\|(\sqcap_{B_1} f) - (\sqcap_{B_2} f)\|^2] && (31)\\ &\leq C_K^2 (B_2^2 - B_1^2) \mathbb{E}[f(0,0)^2]. && (32)\end{aligned}$$

A.1.2. Transferability of GNNs between Grid Graphs

We assume that the input signal f and output g are jointly stationary over the 2d space. The evaluation metric is a loss function of a supervised learning similar to the definitions in [21]. To simplify the notation, we assume that $n = B_1^2$ and $m = B_2^2$. To simplify the notation, we denote $\mathcal{L}_n = \mathcal{L}_{B_1}$ and rewrite the output $\mathbf{\Phi}(\mathbf{x}, \mathbf{S}_{\mathcal{D}_n}; \mathbf{H})$ as a 2-D signal $y_{\mathbf{L},B_1}$ similar to (9), specifically $y_{\mathbf{L},B_1} = \sum_{k=0}^{K} h_k (\mathbf{L}\otimes)^k (\sqcap_{B_1} f)$ with a windowed input of f which only inputs the values $f(i,j)$ if $i, j = 0, 1, \cdots, B_1 - 1$.

$$\mathcal{L}_{B_1}(\mathbf{L}) = \frac{1}{B_1^2} \mathbb{E}\left[\sum_{i,j=0}^{B_1-1} |y_{\mathbf{L},B_1}(i,j) - g(i,j)|^2 \right]. \tag{33}$$

We can conclude that the neural networks trained on a small-size grid graph can transfer to a larger grid graph with a bounded loss function.

Proof. We denote the difference between the predicted outputs as $\epsilon(i,j) = y_{\mathbf{L}}(i,j) - g(i,j)$ with $y_{\mathbf{L}}$ as the output of the 2d-CNN when inputting f. For any time length T, let $N = \lceil B_2/B_1 \rceil$, then we have

$$\mathcal{L}_{B_2}(\mathbf{L}) \leq \mathbb{E}\left[\frac{1}{(NB_1)^2} \sum_{n_1=0}^{NB_1-1} \sum_{n_2=0}^{NB_1-1} |\epsilon(i,j)|^2 \right]. \tag{34}$$

Recenter the summations and denote $\mathcal{T} = \{mB_1 - \frac{(N-1)B_1}{2} | m \in \mathbb{Z}, 1 \leq m < N\}^2$ as the center points, the summation can be decomposed as

$$\begin{aligned}\mathcal{L}_{B_2}(\mathbf{L}) &\leq \mathbb{E}\left[\frac{1}{(NB_1)^2} \sum_{\tau \in \mathcal{T}} \left[\sum_{i,j=0}^{B_1-1} |\epsilon(i - \tau_1, j - \tau_2)|^2 \right] \right] && (35)\\ &\leq \frac{1}{(NB_1)^2} \sum_{\tau \in \mathcal{T}} \mathbb{E}\left[\sum_{i,j=0}^{B_1-1} |\epsilon(i - \tau_1, j - \tau_2)|^2 \right]. && (36)\end{aligned}$$

With the inputs and outputs both stationary, we have $\mathbb{E}[|\epsilon(i-\tau_1, j-\tau_2)|^2] = \mathbb{E}[|\epsilon(i,j)|^2]$, which leads to

$$\begin{aligned}\mathcal{L}_{B_2}(\mathbf{L}) &\leq \frac{1}{(NB_1)^2} \sum_{\tau \in \mathcal{T}} \mathbb{E}\left[\sum_{i,j=0}^{B_1-1} |\epsilon(i,j)|^2 \right] && (37)\\ &\leq \frac{1}{B_1^2} \mathbb{E}\left[\sum_{i,j=0}^{B_1-1} |\epsilon(i,j)|^2 \right]. && (38)\end{aligned}$$

Next we replace ϵ with the indicator function and have an intermediate term $y_{\mathbf{L},B_1}$ as the output of 2D-CNN when inputting $\sqcap_{B_1} f$.

$$\begin{aligned}\mathcal{L}_{B_2}(\mathbf{L}) &\leq \frac{1}{B_1^2} \mathbb{E}[\| \sqcap_{B_1} y_{\mathbf{L}} - g\|^2] && (39)\\ &= \frac{1}{B_1^2} \mathbb{E}[\| \sqcap_{B_1} y_{\mathbf{L}} - \sqcap_{B_1} y_{\mathbf{L},B_1} + \sqcap_{B_1} y_{\mathbf{L},B_1} - g\|^2] && (40)\\ &\leq \frac{1}{B_1^2} \mathbb{E}[\| \sqcap_{B_1} y_{\mathbf{L},B_1} - g\|^2] + \frac{1}{B_1^2} \mathbb{E}[\| \sqcap_{B_1} (y_{\mathbf{L},B_1} - y_{\mathbf{L}})\|^2] \\ &\quad + \frac{2}{{B_1}^2} \mathbb{E}[\| \sqcap_{B_1} (y_{\mathbf{L},B_1} - g)\| \| \sqcap_{B_1} (y_{\mathbf{L},B_1} - y_{\mathbf{L}})\|] && (41)\end{aligned}$$

The first term in (41) is $\mathcal{L}_{B_1}(\mathbf{L})$. The second term can be bounded as follows.

$$
\begin{aligned}
&\| \sqcap_{B_1} (y_{\mathbf{L}} - y_{\mathbf{L},B_1})\| \\
&= \left\| \sqcap_{B_1} \left(\sum_{k=0}^{K-1} h_k(\mathbf{L}\otimes)^k f - \sum_{k=0}^{K-1} h_k(L\otimes)^k f_{B_1} \right) \right\| && (42)\\
&\le \|\sqcap_{B_1}(f - f_{B_1})\| + \|\sqcap_{B_1} h_1(\mathbf{L} \otimes f - \mathbf{L} \otimes f_{B_1})\| + \cdots \\
&\qquad + \left\| \sqcap_{B_1} h_{K-1}((\mathbf{L}\otimes)^{K-1} f - (\mathbf{L}\otimes)^{K-1} f_{B_1}) \right\| && (43)\\
&\le \|\sqcap_{B_1}(f - \sqcap_A f)\| + |h_1|\|\mathbf{L}\|_1 \| \sqcap_{B_1+M} (f - \sqcap_{B_1} f)\|_2 \\
&\qquad + |h_2|\|\mathbf{L}\|_1^2 \| \sqcap_{B_1+2M} (f - \sqcap_{B_1} f)\|_2 \cdots && (44)\\
&\le \sum_{k=0}^{K-1} |h_k|\|\mathbf{L}\|_1^k \| \sqcap_{B_1+kM} (f - \sqcap_{B_1} f)\|_2 && (45)\\
&\le \sum_{k=0}^{K-1} |h_k|\|\mathbf{L}\|_1^k \| \sqcap_{B_1+(K-1)M} (f - \sqcap_{B_1} f)\|_2 && (46)\\
&= H_K \| \sqcap_{B_1+(K-1)M} (f - \sqcap_{B_1} f)\|_2, && (47)
\end{aligned}
$$

with $H_K = \sum_{k=0}^{K-1} |h_k|\|\mathbf{L}\|_1^k$. Therefore, we have

$$
\begin{aligned}
\mathcal{L}_{B_2}(\mathbf{L}) \le \mathcal{L}_{B_1}(\mathbf{L}) &+ \frac{H_K^2}{B_1^2} \mathbb{E}\left[\| \sqcap_{B_1+(K-1)M} (f - \sqcap_{B_1} f)\|^2\right] \\
&+ \sqrt{\mathcal{L}_{B_1}(\mathbf{L}) \frac{H_K^2}{{B_1}^2} \mathbb{E}\left[\| \sqcap_{B_1+(K-1)M} (f - \sqcap_{B_1} f)\|^2\right]} \qquad (48)
\end{aligned}
$$

Since f is stationary, the second term can be seen as the variance of f with a volume $(B_1 + (K-1)M)^2 - {B_1}^2$. Finally, we can derive

$$
\mathcal{L}_{B_2}(\mathbf{L}) \le \mathcal{L}_{B_1}(\mathbf{L}) + C_M \mathbb{E}[f^2] + \sqrt{\mathcal{L}_{B_1}(\mathbf{L}) C_M \mathbb{E}[f^2]}, \qquad (49)
$$

where $C_M = \frac{H_K^2}{B_1^2}[2B_1KM + K^2M^2]$ with $H_K = \sum_{k=0}^{K} |h_k|\|\mathbf{L}\|_1^k$. As we have normalized nonlinearities and multiple layers can be seen as a recurrent operation, the conclusion in Theorem 1 can be recovered.

A.2. Proof of Theorem 2

A.2.1. Transferability of Graph Filters across RGGs

Proposition A.2. Let $\mathbf{h}(\cdot)$ be a graph convolutional filter with integral Lipschitz continuous frequency responses with $|\lambda h'(\lambda)| \le C$. Let $\mathbf{S}_n$ and $\mathbf{S}_{\mathcal{D}_n}$ denote the adjancecy matrices of a random geometric graph and a deterministic geometric graph over a unit space respectively with $\mathcal{W}_n = \mathbf{S}_n - \mathbf{S}_{\mathcal{D}_n}$. If it satisfies that $\mathbb{E}[\left\|\mathcal{W}_n^2\right\|] = O(1/n^\alpha)$ with $\alpha > 0$. We have the difference of the outputs of graph convolutional filters with a input graph signal $\mathbf{x} \in \mathbb{R}^n$ bounded as

$$
\mathbb{E}\left[\|\mathbf{h}(\mathbf{S}_n, \mathbf{x}) - \mathbf{h}(\mathbf{S}_{\mathcal{D}_n}, \mathbf{x})\|^2\right] \le C^2 n^{1-\alpha} \|\mathbf{x}\|^2. \qquad (50)
$$

Proof. We denote $\mathcal{A}_n = \mathbf{S}_n$ and $\mathcal{C}_n = \mathbf{S}_{\mathcal{D}_n}$ with $\mathcal{A}_n = \mathcal{C}_n + \mathcal{W}_n$. With $\mathbf{y}_A = \sum_{k=0}^{K} h_k \mathcal{A}_n^k \mathbf{x}$ and $\mathbf{y}_C = \sum_{k=0}^{K} h_k \mathcal{C}_n^k \mathbf{x}$, we have

$$
\begin{aligned}
&\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] \\
&= \mathbb{E}[tr(\mathbf{y}_A \mathbf{y}_A^\intercal - \mathbf{y}_C \mathbf{y}_C^\intercal)] + 2\mathbb{E}[tr(\mathbf{y}_C \mathbf{y}_C^\intercal - \mathbf{y}_A \mathbf{y}_C^\intercal)] && (51)\\
&= \sum_{k=0}^{K} \sum_{l=0}^{K} h_k h_l \left(\mathbb{E}[tr(\mathcal{A}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{A}_n^l)] - tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) \right) \\
&+ 2 \sum_{k=0}^{K} \sum_{l=0}^{K} h_k h_l (tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) - \mathbb{E}[tr(\mathcal{A}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l)]). && (52)
\end{aligned}
$$

We start with the first term in (52) which can be written as

$$
\begin{aligned}
&\sum_{k,l=0}^{K} h_k h_l \left(\mathbb{E}[tr(\mathcal{A}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{A}_n^l)] - tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) \right) \\
&= \sum_{k,l=0}^{K} h_k h_l (\mathbb{E}[tr(\mathcal{C}_n + \mathcal{W}_n)^k \mathbf{x}\mathbf{x}^\intercal (\mathcal{C}_n + \mathcal{W}_n)^l] - tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l)) && (53)\\
&= \sum_{k,l=0}^{K} h_k h_l \Bigg(\mathbb{E}\Bigg[tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l + ((\mathcal{C}_n + \mathcal{W}_n)^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l - \mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) + (\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal (\mathcal{C}_n + \mathcal{W}_n)^l - \mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l)) \\
&\qquad + tr\left(\left(\sum_{r=1}^{k} \mathcal{C}_n^{k-r} \mathcal{W}_n \mathcal{C}_n^{r-1} \right) \mathbf{x}\mathbf{x}^\intercal \left(\sum_{s=1}^{l} \mathcal{C}_n^{s-1} \mathcal{W}_n \mathcal{C}_n^{l-s} \right) \right) \Bigg] - tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l)] \Bigg) && (54)\\
&= \sum_{k,l=0}^{K} h_k h_l \Bigg(\mathbb{E}[tr((\mathcal{C}_n + \mathcal{W}_n)^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l + \mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal (\mathcal{C}_n + \mathcal{W}_n)^l)] - 2tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) + \\
&\qquad \mathbb{E}\left[tr\left(\left(\sum_{r=1}^{k} \mathcal{C}_n^{k-r} \mathcal{W}_n \mathcal{C}_n^{r-1} \right) \mathbf{x}\mathbf{x}^\intercal \left(\sum_{s=1}^{l} \mathcal{C}_n^{s-1} \mathcal{W}_n \mathcal{C}_n^{l-s} \right) \right) \right] \Bigg) + \sum_{k,l=0}^{K} h_k h_l \mathbb{E}[tr(\mathcal{R}_{kl})], && (55)
\end{aligned}
$$

where $\mathcal{R}_{kl}$ represents the sum of the remaining terms that include terms with higher order than $\mathcal{W}_n^2$.

The second term in (52) can be written as

$$
\begin{aligned}
&2 \sum_{k,l=0}^{K} h_k h_l tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) - \mathbb{E}[tr(\mathcal{A}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l)] \\
&= \sum_{k,l=0}^{K} h_k h_l \left(2tr(\mathcal{C}_n^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l) - 2\mathbb{E}[tr((\mathcal{C}_n + \mathcal{W}_n)^k \mathbf{x}\mathbf{x}^\intercal \mathcal{C}_n^l)] \right).
\end{aligned}
\qquad (56)
$$

We notice that putting (55) and (56) together leads to the expression of $\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2]$ as

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] = \sum_{k,l=0}^{K} h_k h_l$$

$$\mathbb{E}\left[tr \left(\left(\sum_{r=1}^{k} \mathcal{C}_n^{k-r} \mathcal{W}_n \mathcal{C}_n^{r-1} \right) \mathbf{x}\mathbf{x}^\mathsf{T} \left(\sum_{s=1}^{k} \mathcal{C}_n^{s-1} \mathcal{W}_n \mathcal{C}_n^{l-s} \right) \right) \right]. \tag{57}$$

By moving all the summations outside, we have

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] = \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l$$

$$\mathbb{E}\left[tr \left(\left(\mathcal{C}_n^{k-r} \mathcal{W}_n \mathcal{C}_n^{r-1} \right) \mathbf{x}\mathbf{x}^\mathsf{T} \left(\mathcal{C}_n^{s-1} \mathcal{W}_n \mathcal{C}_n^{l-s} \right) \right) \right]. \tag{58}$$

With the trace cyclic property $tr(\mathcal{ABC}) = tr(\mathcal{CAB}) = tr(\mathcal{BCA})$, we have

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] = \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l$$

$$\mathbb{E}\left[tr \left(\mathcal{W}_n \mathcal{C}_n^{l+k-r-s} \mathcal{W}_n \mathcal{C}_n^{r-1} \mathbf{x}\mathbf{x}^\mathsf{T} \mathcal{C}_n^{s-1} \right) \right]. \tag{59}$$

For positive semidefinite matrices, the trace is submultiplicative as $tr(\mathcal{AB}) \leq tr(\mathcal{A}) tr(\mathcal{B})$. Therefore, we have

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2]$$

$$= \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l \mathbb{E}\left[tr \left(\mathcal{W}_n \mathcal{C}_n^{l+k-r-s} \mathcal{W}_n \right) tr \left(\mathcal{C}_n^{r-1} \mathbf{x}\mathbf{x}^\mathsf{T} \mathcal{C}_n^{s-1} \right) \right] \tag{60}$$

$$\leq \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l \mathbb{E}\left[tr \left(\mathcal{W}_n \mathcal{C}_n^{l+k-r-s} \mathcal{W}_n \right) \right] tr \left(\mathcal{C}_n^{r-1} \mathbf{x}\mathbf{x}^\mathsf{T} \mathcal{C}_n^{s-1} \right). \tag{61}$$

In (61), the deterministic term $tr(\mathcal{C}_n^{r-1} \mathbf{x}\mathbf{x}^\mathsf{T} \mathcal{C}_n^{s-1})$ can be decomposed with the spectral representation of $\mathbf{x}$. With $\mathbf{x}$ decomposed with respect to the eigenbasis $\{\mathbf{e}_i\}_{i=1}^n$ of $\mathcal{C}_n$, $\mathbf{x} = \sum_{i=1}^n \hat{x}_i \mathbf{e}_{in}$, with $\hat{x}_i = \langle \mathbf{x}, \mathbf{e}_i \rangle$, we have

$$tr(\mathcal{C}_n^{r-1} \mathbf{x}\mathbf{x}^\mathsf{T} \mathcal{C}_n^{s-1})$$

$$= tr \left(\mathcal{C}_n^{r-1} \left(\sum_{i=1}^{n} \hat{x}_i \mathbf{e}_i \right) \left(\sum_{i=1}^{n} \hat{x}_i \mathbf{e}_i \right)^\mathsf{T} \mathcal{C}_n^{s-1} \right) \tag{62}$$

$$= tr \left(\left(\sum_{i=1}^{n} \hat{x}_i \mathcal{C}_n^{r-1} \mathbf{e}_i \right) \left(\sum_{i=1}^{n} \hat{x}_i \mathcal{C}_n^{s-1} \mathbf{e}_i \right)^\mathsf{T} \right) \tag{63}$$

$$= tr \left(\left(\sum_{i=1}^{n} \hat{x}_i \lambda_i^{r-1} \mathbf{e}_i \right) \left(\sum_{i=1}^{n} \hat{x}_i \lambda_i^{s-1} \mathbf{e}_i \right)^\mathsf{T} \right), \tag{64}$$

as the eigenbasis are orthonormal, i.e. $tr(\mathbf{e}_i \mathbf{e}_i^\mathsf{T}) = 1$ and $\mathbf{e}_i \mathbf{e}_j^\mathsf{T} = 0$ for $i \neq j$, we have

$$tr(\mathcal{C}_n^{r-1} \mathbf{x}\mathbf{x}^\mathsf{T} \mathcal{C}_n^{s-1}) = tr \left(\sum_{i=1}^{n} \hat{x}_i^2 \lambda_i^{r+s-2} \mathbf{e}_i \mathbf{e}_i^\mathsf{T} \right) \tag{65}$$

$$= \sum_{i=1}^{n} \hat{x}_i^2 \lambda_i^{r+s-2}. \tag{66}$$

Insert this back to (61), we have

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] \tag{67}$$

$$\leq \sum_{i=1}^{n} \hat{x}_i^2 \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l \mathbb{E}\left[tr \left(\mathcal{W}_n \mathcal{C}_n^{l+k-r-s} \mathcal{W}_n \right) \right] \lambda_i^{r+s-2}. \tag{68}$$

With the trace cyclic property and the inequality that

$$tr(\mathcal{AB}) \leq \|\mathcal{A}\|_2 tr(\mathcal{B}), \tag{69}$$

for any square matrix $\mathcal{A}$ and positive semidefinite matrix $\mathcal{B}$ [28]. Therefore, the inequality can be derived further as

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] \tag{70}$$

$$\leq \sum_{i=1}^{n} \hat{x}_i^2 \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l \mathbb{E}\left[\|\mathcal{W}_n^2\|_2 \right] tr \left(\mathcal{C}_n^{l+k-r-s} \right) \lambda_i^{r+s-2} \tag{71}$$

$$\leq \sum_{i=1}^{n} \hat{x}_i^2 \sum_{k,l=0}^{K} \sum_{r=1}^{k} \sum_{s=1}^{l} h_k h_l \mathbb{E}\left[\|\mathcal{W}_n^2\|_2 \right] \sum_{j=1}^{n} \lambda_j^{l+k-r-s} \lambda_i^{r+s-2}. \tag{72}$$

By changing the summation order, we have

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2]$$

$$\leq \sum_{i,j=0}^{n} \sum_{r,s=1}^{K} \left(\sum_{k=r}^{K} h_k \lambda_i^{r-1} \lambda_j^{k-r} \right) \left(\sum_{l=s}^{K} h_l \lambda_i^{s-1} \lambda_j^{l-s} \right)$$

$$\hat{x}_i^2 \mathbb{E}\left[\|\mathcal{W}_n^2\| \right] \tag{73}$$

$$\leq \sum_{i,j=0}^{n} \sum_{r=1}^{K} \left(\sum_{k=r}^{K} h_k \lambda_i^{r-1} \lambda_j^{k-r} \right)^2 \hat{x}_i^2 \mathbb{E}\left[\|\mathcal{W}_n^2\| \right]. \tag{74}$$

We import the definition of generalized Lipschitz gradient (Definition 4 in [29]) of generalized graph filter frequency response (Definition 3 in [29]). More specifically, let $\boldsymbol{\lambda}^{(r)} = [\lambda_i, \lambda_i \cdots, \lambda_j, \cdots, \lambda_j]^\mathsf{T}$ with r λ_i followed by $K - r$ λ_j and $\boldsymbol{\lambda}^{(r)} \in \mathbb{R}_+^K$. The partial derivative of generalized $h(\boldsymbol{\lambda})$ with respect to the r-th entry λ_r as

$$\frac{\partial h(\boldsymbol{\lambda}^{(r)})}{\partial \lambda_r} = \sum_{k=r}^{K} h_k \lambda_i^{r-1} \lambda_j^{k-r}, \text{for all } r = 1, 2 \cdots, K. \tag{75}$$

The generalized Lipshitz gradient between λ_i and λ_j is defined as

$$\nabla_L h(\lambda_i, \lambda_j) = \left[\frac{\partial h(\boldsymbol{\lambda}^{(1)})}{\partial \lambda_1}, \cdots \frac{\partial h(\boldsymbol{\lambda}^{(K)})}{\partial \lambda_K} \right]^\mathsf{T}. \tag{76}$$

Therefore, the inequality (74) can be further derived combined with the Lipschitz gradient assumption $\|\nabla_L h(\lambda_i, \lambda_j)\| \leq C_L \leq C$ and $\|\mathbb{E}[\mathcal{W}_n^2]\| = \mathcal{O}(1/n^\alpha)$.

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] \leq \sum_{i,j=0}^{n} \|\nabla_L h(\lambda_i, \lambda_j)\|^2 \hat{x}_i^2 \mathbb{E}\left[\|\mathcal{W}_n^2\| \right] \tag{77}$$

$$\leq n C^2 \mathbb{E}\left[\|\mathcal{W}_n^2\| \right] \|\mathbf{x}\|^2 \leq C^2 n^{1-\alpha} \|\mathbf{x}\|^2. \tag{78}$$

This concludes the proof.

A.2.2. Transferability of GNNs across RGGs

Proposition A.3. Let $\Phi(\mathbf{x}, \mathbf{S}_n; \mathbf{H})$ be an 1-layer GNN applied on a random geometric graph $\mathbf{G}_n^r$ and a grid graph $\mathbf{G}_n$. Under the same setting with Theorem A.2, the difference of the outputs of GNN with input graph signal $\mathbf{x} \in \mathbb{R}^n$ can be bounded as

$$\mathbb{E}\left[\|\mathbf{\Phi}(\mathbf{x}, \mathbf{S}_n; \mathbf{H}) - \mathbf{\Phi}(\mathbf{x}, \mathbf{S}_{\mathcal{D}_n}; \mathbf{H})\|^2\right] \leq F^L C^2 n^{1-\alpha} \|\mathbf{x}\|^2. \tag{79}$$

Proof. To bound the output difference of GNNs on RGG and grid graph, we need to write in the form of features of the final layer

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{\Phi}(\mathbf{X}; \mathbf{S}_n, \mathcal{H}) - \mathbf{\Phi}(\mathbf{X}; \mathbf{S}_{\mathcal{D}_n}, \mathcal{H})\|^2\right] \\
&= \sum_{q=1}^{F} \mathbb{E}\left[\left\|\sigma(\mathbf{y}_{A,L}^q) - \sigma(\mathbf{y}_{C,L}^q)\right\|^2\right] && (80) \\
&\leq \sum_{q=1}^{F} \mathbb{E}\left[\left\|\mathbf{y}_{A,L}^q - \mathbf{y}_{C,L}^q\right\|^2\right], && (81) \\
&\leq F\mathbb{E}\left[\left\|\mathbf{y}_{A,L-1}^q - \mathbf{y}_{C,L-1}^q\right\|^2\right] && (82)
\end{aligned}$$

where the inequality comes from the normalized Lipschitz of nonlinearities.

A.2.3. Proof of Theorem 2

Proof. We replace $\mathbf{y}_A = \mathbf{\Phi}(\mathbf{x}, \mathbf{S}_n, \mathbf{H})$ and $\mathbf{y}_C = \mathbf{\Phi}(\mathbf{x}, \mathbf{S}_{\mathcal{D}_n}, \mathbf{H})$ and $\mathbf{r}_n^* = \mathbf{g}$ for the ease of presentation. The MSE loss can be written as

$$\begin{aligned}
&\left|\mathbb{E}\left[\|\mathbf{y}_A - \mathbf{g}\|^2 - \|\mathbf{y}_C - \mathbf{g}\|^2\right]\right| \\
&\leq \mathbb{E}\left[\left|\|\mathbf{y}_A - \mathbf{g}\|^2 - \|\mathbf{y}_C - \mathbf{g}\|^2\right|\right] && (83) \\
&= \mathbb{E}\left[\|\mathbf{y}_A - \mathbf{g} - \mathbf{y}_C + \mathbf{g}\|\|\mathbf{y}_A - \mathbf{g} + \mathbf{y}_C - \mathbf{g}\|\right] && (84) \\
&\leq \mathbb{E}\left[\|\mathbf{y}_A - \mathbf{y}_C\|(\|\mathbf{y}_A - \mathbf{g}\| + \|\mathbf{y}_C - \mathbf{g}\|)\right] && (85) \\
&\leq \mathbb{E}\left[\|\mathbf{y}_A - \mathbf{y}_C\|(\|\mathbf{y}_C - \mathbf{g}\| + \sqrt{\epsilon})\right] && (86)
\end{aligned}$$

By subtracting and adding $\mathbf{y}_A$ in the term $\|\mathbf{y}_C - \mathbf{g}\|$, we have

$$\|\mathbf{y}_C - \mathbf{g}\| \leq \|\mathbf{y}_A - \mathbf{y}_C\| + \|\mathbf{y}_A - \mathbf{g}\|, \tag{87}$$

which depends on the triangle inequality. Inserting this into (86), we have

$$\begin{aligned}
&\left|\mathbb{E}\left[\|\mathbf{y}_A - \mathbf{g}\|^2 - \|\mathbf{y}_C - \mathbf{g}\|^2\right]\right| \\
&\leq \mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2] + 2\sqrt{\epsilon}\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|]. && (88)
\end{aligned}$$

With Jensen inequality, we have

$$\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|] \leq \sqrt{\mathbb{E}[\|\mathbf{y}_A - \mathbf{y}_C\|^2]}. \tag{89}$$

With the conclusion in Proposition 2, we have

$$\mathbb{E}\left[\|\mathbf{y}_A - \mathbf{y}_C\|^2\right] \leq C_L^2 n^{1-\alpha} \|\mathbf{x}\|^2. \tag{90}$$

Bring this to (88), we have

$$|\mathcal{L}_n^r - \mathcal{L}_n)| \leq C^2 n^{1-\alpha} \|\mathbf{x}\|^2 + 2\sqrt{\epsilon} C n^{\frac{1-\alpha}{2}} \|\mathbf{x}\|, \tag{91}$$

which concludes the proof.

A.3. Proof of Theorem 3

Proof. We first decompose the loss difference between GNN on $\mathbf{S}_n$ and $\mathbf{S}_m$ by inserting intermediate terms of loss of GNNs on $\mathbf{S}_{\mathcal{D}_n}$ and $\mathbf{S}_{\mathcal{D}_m}$.

$$|\mathcal{L}_n^r - \mathcal{L}_m^r| \leq |\mathcal{L}_n^r - \mathcal{L}_n| + |\mathcal{L}_n - \mathcal{L}_m| + |\mathcal{L}_m - \mathcal{L}_m^r| \tag{92}$$

The first and the third terms in (92) can be bounded with Theorem 1. The second term can be bounded with Theorem 2.

B. REFERENCES

[1] Mark Eisen and Alejandro Ribeiro, “Optimal wireless resource allocation with random edge graph neural networks,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 2977–2991, 2020.

[2] Zhiyang Wang, Mark Eisen, and Alejandro Ribeiro, “Learning decentralized wireless resource allocations with graph neural networks,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 1850–1863, 2022.

[3] Yang Lu, Yuhang Li, Ruichen Zhang, Wei Chen, Bo Ai, and Dusit Niyato, “Graph neural networks for wireless networks: Graph representation, architecture and evaluation,” *IEEE Wireless Communications*, 2024.

[4] Mengyuan Lee, Guanding Yu, Huaiyu Dai, and Geoffrey Ye Li, “Graph neural networks meet wireless communications: Motivation, applications, and future directions,” *IEEE Wireless Communications*, vol. 29, no. 5, pp. 12–19, 2022.

[5] Luana Ruiz, Fernando Gama, and Alejandro Ribeiro, “Graph neural networks: Architectures, stability, and transferability,” *Proceedings of the IEEE*, vol. 109, no. 5, pp. 660–682, 2021.

[6] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun, “Spectral networks and locally connected networks on graphs,” 2014.

[7] Fernando Gama, Antonio G. Marques, Geert Leus, and Alejandro Ribeiro, “Convolutional neural network architectures for signals supported on graphs,” *IEEE Transactions on Signal Processing*, vol. 67, no. 4, pp. 1034–1049, Feb. 2019.

[8] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds. 2016, vol. 29, Curran Associates, Inc.

[9] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling, “E(n) equivariant graph neural networks,” in *Proceedings of the 38th International Conference on Machine Learning*, Marina Meila and Tong Zhang, Eds. 18–24 Jul 2021, vol. 139 of *Proceedings of Machine Learning Research*, pp. 9323–9332, PMLR.

[10] Fernando Gama, Alejandro Ribeiro, and Joan Bruna, “Stability of graph scattering transforms,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, Curran Associates, Inc.

[11] Fernando Gama, Joan Bruna, and Alejandro Ribeiro, “Stability properties of graph neural networks,” *IEEE Transactions on Signal Processing*, vol. 68, pp. 5680–5695, 2020.

[12] Mark Eisen and Alejandro Ribeiro, “Transferable policies for large scale wireless networks with graph neural networks,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 5040–5044.

[13] Santiago Fernández, Romina García Camargo, Mark Eisen, Alejandro Ribeiro, and Federico Larroca, “On the transferability of graph neural networks for resource allocation in wireless networks,” in *2024 IEEE URUCON*, 2024, pp. 1–5.

[14] Luana Ruiz, Luiz F. O. Chamon, and Alejandro Ribeiro, “Transferability properties of graph neural networks,” *IEEE Transactions on Signal Processing*, vol. 71, pp. 3474–3489, 2023.

[15] Sohir Maskey, Ron Levie, and Gitta Kutyniok, “Transferability of graph neural networks: an extended graphon approach,” 2022.

[16] Nicolas Keriven, Alberto Bietti, and Samuel Vaiter, “Convergence and stability of graph convolutional networks on large random graphs,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, Eds. 2020, vol. 33, pp. 21512–21523, Curran Associates, Inc.

[17] Zhiyang Wang, Luana Ruiz, and Alejandro Ribeiro, “Geometric graph filters and neural networks: Limit properties and discriminability trade-offs,” 2023.

[18] Eitan Levin, Yuxin Ma, Mateo Díaz, and Soledad Villar, “On transferring transferability: Towards a theory for size generalization,” 2025.

[19] Mathew Penrose, *Random geometric graphs*, vol. 5, OUP Oxford, 2003.

[20] Martin Haenggi, Jeffrey G. Andrews, Francois Baccelli, Olivier Dousse, and Massimo Franceschetti, “Stochastic geometry and random graphs for the analysis and design of wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 27, no. 7, pp. 1029–1046, 2009.

[21] Damian Owerko, Charilaos I Kanatsoulis, Jennifer Bondarchuk, Donald J Bucci Jr, and Alejandro Ribeiro, “Transferability of convolutional neural networks in stationary learning tasks,” *arXiv preprint arXiv:2307.11588*, 2023.

[22] Santiago Segarra, Antonio G. Marques, and Alejandro Ribeiro, “Optimal graph-filter design and applications to distributed linear network operators,” *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 4117–4131, 2017.

[23] Aliaksei Sandryhaila and José M. F. Moura, “Discrete signal processing on graphs,” *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[24] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[25] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[26] Romina Garcia Camargo, Zhiyang Wang, and Alejandro Ribeiro, “Graph neural networks in large scale wireless communication networks: Scalability across random geometric graphs,” 2025, https://zhiyangwang.com/Papers/ICASSP2026.pdf.

[27] Søren Skovgaard Christensen, Rajiv Agarwal, Elisabeth De Carvalho, and John M. Cioffi, “Weighted sum-rate maximization using weighted mmse for mimo-bc beamforming design,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4792–4799, 2008.

[28] Sheng-De Wang, Te-Son Kuo, and Chen-Fa Hsu, “Trace bounds on the solution of the algebraic matrix riccati and lyapunov equation,” *IEEE Transactions on Automatic Control*, vol. 31, no. 7, pp. 654–656, 1986.

[29] Zhan Gao, Elvin Isufi, and Alejandro Ribeiro, “Stability of graph convolutional neural networks to stochastic perturbations,” *Signal Processing*, vol. 188, pp. 108216, 2021.