Predicting Wireless RSSI Using Machine Learning on Graphs

Claudina Rattaro, Federico Larroca and Germán Capdehourat
Facultad de Ingeniería, *Universidad de la República*, Montevideo, Uruguay
{crattaro, flarroca, gcapde}@fing.edu.uy

***Abstract*—In wireless communications, optimizing the resource allocation requires the knowledge of the state of the channel. This is even more important in device-to-device communications, one typical use case in 5G/6G networks, where such knowledge is hard to obtain at reasonable signaling costs. In this paper, we study the use of graph-based machine learning methods to address this problem. That is to say, we learn to predict the channel state on a given link through measurements on other links, thus decreasing signaling overhead. In particular, we model the problem as a link-prediction one and we consider two representative approaches: Random Dot Product Graphs and Graph Neural Networks. The key point is that these methods consider the geometric structure underlying the data. They thus enable better generalization and require less training data than classic methods, as we show on our evaluation using a dataset of RSSI measurements of real-world Wi-Fi operating networks.**

***Index Terms*—Embeddings, Graph Representation Learning, Link-prediction**

I. Introduction

Communications play a crucial role in human interactions, in the economy, in education, in accessing and democratizing different services and information. With the arrival of COVID-19, such influence increased. Due to the role of wireless communications in our everyday lives, a plethora of research papers have focused their attention on the quality aspects of wireless links (see for example the recent articles [1]–[4] and the references therein).

For future wireless communications, to enhance coverage and capacity, an accurate channel state information is essential. In this work we focus on the Received Signal Strength Indicator (RSSI), although our methods are readily extended to other channel state information. Moreover, since RSSI is available in mainstream wireless signal measurements, the present results are applicable to practically any wireless network. In any case, in some important scenarios, like device-to-device (D2D) or Internet of Things (IoT) communications, the acquisition of the channel state information can bring significant overheads.

In this sense, an increasing body of work is devoted to methods that minimize the need of measuring during the network's operation. For example, authors of [5] proposed a blind radio tomographic approach that, given a set of attenuation measurements between several sensors, learns the spatial loss fields (SLFs), which quantify absorption of radio frequency waves at each location, together with a weight function (needed to use SLFs which depend on the transmitter-receiver locations). They based their algorithms on a non-parametric kernel regression. However, this type of model requires a lot of processing time and its performance depends highly on the number of sensors.

On the other hand, a recent trend has been the application of machine learning (ML) to overcome this problem [1]–[3]. Basically, given measurements between a subset of the network's nodes, the algorithm is trained to learn a mapping to the RSSI of other links (for which measurements are unavailable). However, these proposals consider the measurements as a vector, meaning that the underlying geometric structure of the problem is discarded (see Fig. 1), and is expected to be learned from the data instead. Including this structural information *a priori* will result in an algorithm that needs less training samples and with better generalization properties.

The main contribution of the present paper is thus to approach the problem through graph-based machine learning techniques. In particular, we cast it as a link-prediction problem and study two complementary graph representation learning based methods [18]: Random Dot Product Graphs (RDPGs) and Graph Neural Networks (GNNs). RDPG [6] is an spectral-based embedding method where each node has an associated latent vector and the inner product between these vectors dictate the edge existence probabilities. In addition to its simplicity, the model offers interpretability and intuition. On the other hand, GNNs [8] may be regarded as the extension to graphs of Convolutional Neural Networks (CNNs). We study the performance of these two methods for predicting the RSSI in wireless links. To this end, we used a dataset corresponding to a month of RSSI periodic measurements between access points in an indoor Wi-Fi network. We also compare the resulting performance with the blind radio tomographic approach of [5].

The remainder of this article is organized as follows. In the next section we briefly present the notation and state the problem. In Section III the dataset collection and characteristics are described. In Section IV we discuss both graph-based machine learning approaches with more detail and present the main results. Finally, we conclude and discuss future work in Section V.

II. Definitions and notation

A. Basic notions of graphs

A graph is a triplet $G = (\mathcal{V}, \mathcal{E}, W)$ of vertices, edges, and weights. Vertices or nodes are a set of N labels ($\mathcal{V} = \{1, \dots N\}$), edges are ordered pairs of labels $(i, j) \in \mathcal{E}$ and weights $w_{ij} \in \mathbb{R}$ are numbers associated to edges (i, j). In undirected graphs $w_{ij} = w_{ji}$, however, in directed graphs

This work has been supported by the Agencia Nacional de Investigación e Innovación (ANII), Uruguay, grants FMV_3_2018_1_148149 and FMV_1_2019_1_155700

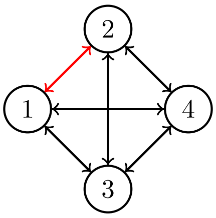

Fig. 1: A toy example with $N = 4$ nodes. RSSI measurements for black edges are available and the objective is to predict the value at the red edge.

edge (i, j) is different from edge (j, i) and then w_{ij} may be different from w_{ji}. The adjacency matrix of a graph is the matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ with $[\mathbf{A}]_{ij} = w_{ij}$ (and thus $\mathbf{A}$ is symmetric for undirected graphs). A graph signal is a column vector $\mathbf{x} \in \mathbb{R}^N$ in which component $\mathbf{x}_i$ is associated with node i. The Graph Shift Operator $\mathbf{S} \in \mathbb{R}^{N\times N}$ (GSO), important in the GNN formulation, is a stand-in for any of the matrix representations of the graph, e.g. $\mathbf{A}$ or its normalized version.

B. Problem statement

Consider a wireless network with N nodes, where power measurements are available for a certain subset of node pairs. For instance, it is typically the case that the channel is known between the base-stations and the devices, but not between devices. As we mentioned before, this is very important in, for instance, D2D communications, although the signaling overhead to obtain this is too significant.

In our particular case then, the problem is to use the available RSSI measurements to predict the value at the rest of the links. Consider Fig. 1 as a simple example to illustrate the problem. Assume nodes 3 and 4 correspond to two base-stations and nodes 1 and 2 correspond to two mobile devices. Measurements between base-stations and devices are available (black arrows in the figure), and we would like to estimate the channel between the devices (red arrow).

The typical ML approach to this problem (e.g. [1]) is to learn to map the vector $(w_{1,3}, w_{1,4}, w_{2,3}, w_{2,4}, w_{3,4}) = \mathbf{w} \in \mathbb{R}^5$ to the missing value $w_{1,2}$ (where we have assumed a symmetric channel to ease the exposition). However, this vector effectively hides the geometry of the problem to the ML algorithm. For instance, attenuations are typically spatially correlated, and these correlations are expected to be learned from the dataset. Providing this structural information of the data to the algorithm will help in increasing its generalization power.

As we mentioned before, we will approach the RSSI prediction problem with techniques of ML on graphs, in particular a link-prediction one [9], [10]. Link prediction has attracted considerable attention from interdisciplinary research communities, due to its ubiquitous applications in many areas. There exists a wide range of link prediction techniques like scoring methods, probabilistic methods, dimensionality reduction approaches, etc. In a nutshell, it consists in estimating the weight of unobserved links (in static networks) or predicting the likelihood of future links (in dynamic networks). After discussing the dataset we used in the evaluation in the following section, Sec. IV presents in certain detail the ML methods we analyzed.

III. Dataset

The dataset was obtained from Plan Ceibal [11], a major education service provider, which runs Uruguay's nation-wide one-to-one computing program. Most of the connectivity solution is administered by two Wireless LAN Controllers (WLCs), which are configured to use 20 MHz channels in 2.4 GHz and 40 MHz in the 5 GHz band. As part of the Radio Resource Management (RRM) algorithm executed by the WLCs [12], each AP (Access Point) in the network periodically sends a so-called NDP (Neighbor Discovery Protocol) packet on every channel and band possible. These broadcast messages are sent at the maximum allowed power for the channel/band, at the lowest supported data rate and using a single radio chain (meaning no beamforming is applied in their transmission). By default, an NDP packet is sent over all channels every 180 seconds. All received NDP packets, along with the corresponding RSSI (expressed in dBm and with a resolution of 1 dBm), are forwarded to the WLC, where an average of the last five values is stored. The dataset consists of one of these averages per hour. Further details may be obtained from [13], [14].

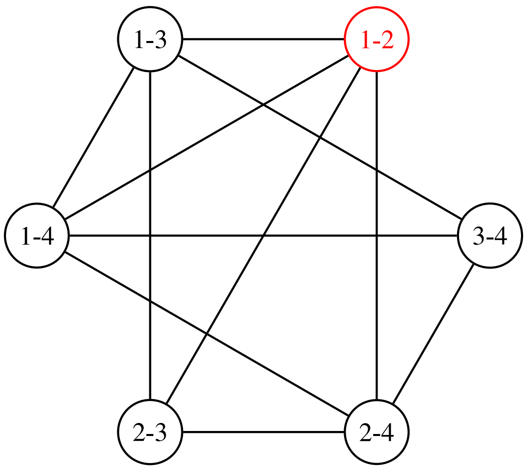

Fig. 2: Line graph of the toy example of Fig. 1. In the same way as the former, the objective is to predict the signal of node $1-2$ knowing the others.

Thus, although the resulting dataset corresponds to RSSI measurements, they may be regarded as a fixed offset of the channel gain, since the power is constant. In particular, we will consider a month worth of the 2.4 GHz band measurements of a particular school network (counting 10 APs) for which we had floor plans (a single-story building spanning $450\,\mathrm{m}^2$ and including several classrooms). Depending on the machine learning method, as explained in the following section, we will work with the directed graph G which vertices are the APs ($\mathcal{V} = \{1, \ldots 10\}$) and the RSSI are the associated weights or we will work with the line graph $L(G)$. In L the nodes are the physical links (edges of G, then in graph L we have $\mathcal{V} = \{1-2, 1-3, 2-3 \ldots 9-10\}$) and an edge in L exists when two physical links have an AP in common (i.e. between nodes $1-2$ and $2-3$ is a link because

AP 2). In L representation RSSI values are considered as a graph signal (see Fig. 2).

IV. Machine learning techniques and results

A. Traditional methods

Before proceeding let us present a baseline and the results we obtained with it. To this end, we will consider the blind radio tomographic (BRT) approach of [5]. This is a method specifically for estimating radio maps, and given a set of sensors, their positions and measurements between them, it provides a function that estimates the attenuation between any pair of points. We have thus tested its performance by removing a single AP from the dataset, estimating the function with the resulting measurements, and evaluating it between the missing AP and the remaining ones. In this method we use the building floor plans to know the location of the APs.

When compared with the actual measurements, working with the month worth of measurements, this results in a root mean squared error (RMSE) of 6.1 dBm averaged over all APs. In wireless networks, the choice of data rate directly impacts coverage and performance. As this choice is related to the RSSI, then another interesting link indicator is when the signal level is above a certain threshold. If we focus on classifying each link as above or below -75 dBm (since below that value the link would work at very low data rates, which is not desirable), we obtain an accuracy of 80%. Note that the choice of the threshold (in this case -75 dBm) is totally arbitrary.

B. Random dot product graphs (RDPGs)

In RDPG, which may be considered a special case of latent position models, each node i has an associated vector $\mathbf{x}_i \in \mathbb{R}^d$, and the probability of an edge existing between nodes i and j is the dot product of the corresponding latent position vectors [6]. In other words, if we stack these vectors in the matrix $\mathbf{X} \in \mathbb{R}^{N\times d}$, the connection probability matrix $\mathbf{P}$ is given by $P = \mathbf{X}\mathbf{X}^T$. Since $\mathbf{P}$ is typically not observable and instead we observe $\mathbf{A}$, which is a noisy version of $\mathbf{P}$ since $\mathbb{E}\{\mathbf{A}\} = \mathbf{P}$, the latent position matrix $\mathbf{X}$ can be estimated by solving $\hat{\mathbf{X}} = \arg\min_{\mathbf{X}} ||\mathbf{A} - \mathbf{X}\mathbf{X}^T||_F^2$. This is readily solved by the spectral decomposition of $\mathbf{A}$ (see [6] for the details and consistency proofs).

In our case, as the graph is directed (the adjacency matrix and $\mathbf{P}$ are not symmetric) then the embeddings can be estimated using the singular value decomposition (SVD) of $\mathbf{A}$ obtaining $\hat{\mathbf{X}}_{in} \in \mathbb{R}^{N\times d}$ and $\hat{\mathbf{X}}_{out} \in \mathbb{R}^{N\times d}$ instead of a unique matrix $\mathbf{X}$ [16]. Moreover, as our graphs are weighted (i.e. the RSSI value of the physical link) the i, j entry of the product $\hat{\mathbf{X}}_{out}\hat{\mathbf{X}}_{in}^T$ will estimate $\mathbb{E}\{w_{ij}\}$ (the expected weight between nodes i and j), which we will still denote as $\mathbf{P}$.

In this sense, we build $\mathbf{A_{avg}}$ as a month's worth average of adjacency matrices. In order to work with positive weights (and that values further away from zero represent stronger links), we apply the embedded method to matrix $\mathbf{A_{avg}} - \min_{ij} [\mathbf{A_{avg}}]_{ij}\mathbf{1_N}$ where $\mathbf{1_N}$ represents all ones matrix. In Fig. 3 we show $\mathbf{A_{avg}}$ and the estimated matrix $\mathbf{P}$ obtained by $\hat{\mathbf{X}}_{out}$ and $\hat{\mathbf{X}}_{in}^T$. In this case, comparing the estimation directly with the average RSSI of each pair of APs, using as the metric the RMSE we obtain a relatively good performance of 7 dBm, similar to BRT. In the classification case, where we predict whether RSSI is above the -75 dBm threshold, results are excellent, obtaining 90% of accuracy. It is important to remark that the values on the diagonal of $\mathbf{A_{avg}}$ and $\mathbf{P}$ are ignored in the RSME and accuracy calculation.

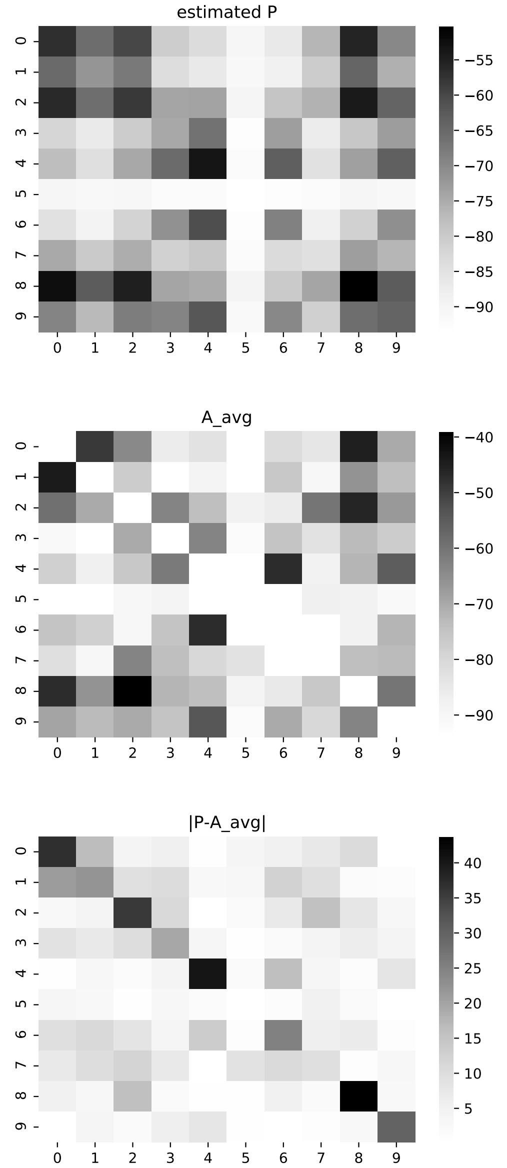

Fig. 3: $\mathbf{A_{avg}}$, the estimated matrix $\mathbf{P}$ using $\hat{\mathbf{X}}_{out}\hat{\mathbf{X}}_{in}^T$ and their difference $|\mathbf{P} - \mathbf{A_{avg}}|$. Note that outside the diagonal the diference $|\mathbf{P} - \mathbf{A_{avg}}|$ is in most cases less than 5 dB.

Let us now consider the link-prediction problem. We proceed to "remove" one edge (r, s) (the one to be estimated) setting $[\mathbf{A_{avg}} - \min_{ij} [\mathbf{A_{avg}}]_{ij}\mathbf{1_N}]_{rs} \leftarrow 0$. Applying the embedding method to the resulting matrix we obtain $\hat{\mathbf{X}}_{in}$ and $\hat{\mathbf{X}}_{out}$ where $[\hat{\mathbf{X}}_{out}\hat{\mathbf{X}}_{in}^T + \min_{ij} [\mathbf{A_{avg}}]_{ij}\mathbf{1_N}]_{rs}$ represents the predicted RSSI of the edge (r, s). We repeat these steps with all the pairs $(r, s) \in \mathcal{V} \times \mathcal{V}$. Comparing the estimation directly with the average RSSI of each pair of APs, we obtain a significantly worse performance (RMSE=17 dB); whereas considering the classification problem, the performance we obtain was 64% of accuracy.

Due to the arbitrary weight choice on the "missing" link, we also tried removing the link setting $[\mathbf{A_{avg}} - \min_{ij}[\mathbf{A_{avg}}]_{ij}\mathbf{1_N}]_{rs}$ as the mean value of the matrix $\mathbf{A_{avg}} - \min_{ij}[\mathbf{A_{avg}}]_{ij}\mathbf{1_N}$. In this case, the results improve significantly, reaching a RMSE of 10 dB and 73% of accuracy.

C. Graph Neural Networks (GNNs)

GNNs are a class of machine learning models that have emerged in recent years for learning on graph-structured data [8], [17]. GNNs may be regarded as an extension of CNNs to graphs. We thus need to define convolution on graphs first, for which the GSO $\mathbf{S} \in \mathbb{R}^{N\times N}$ and the graph signal $\mathbf{x} \in \mathbb{R}^N$ we defined earlier are needed. In our work we have used the normalized adjacency matrix (i.e. divided by its largest eigenvalue) as GSO.

Note that the matrix product $\mathbf{Sx} = \mathbf{y}$ results in another graph signal that aggregates at each node the information of its neighbors. By writing $\mathbf{S}^K\mathbf{x} = \mathbf{S}(\mathbf{S}^{K-1}\mathbf{x})$ we may see that this way we aggregate the information K hops away. Graph convolution is defined simply as a weighted sum of these K signals (i.e. $\sum_k \mathbf{S}^k\mathbf{x}h_k$, where scalars h_k are the taps of the filter).

We may even consider multi-dimensional signals by using $\mathbf{X} \in \mathbb{R}^{N\times F}$, and even change the resulting signal's dimension (i.e. number of features) by considering an $F_{in} \times F_{out}$ matrix $\mathbf{H}_k$ instead of the scalar taps. A single-layer GNN (or graph perceptron) results of applying a pointwise non-linear function $\sigma(\cdot)$ to this convolution:

$$\mathbf{Y} = \sigma\left(\sum_{k=0}^{K-1} \mathbf{S}^k\mathbf{X}\mathbf{H}_k\right), \tag{1}$$

and a deep GNN is constructed by concatenating several perceptrons.

Notice that in our case the RSSI are defined on the edges, so to have signals on the nodes we work with the line graph L. This is a graph where its nodes represent the physical links between APs ($N = 81$), and they are connected when they share an AP (i.e. all physical links of an AP are interconnected on the line graph). Thus, the signal x_i correspond to the RSSI value associated to each node i on graph L (each link between APs).

In order to train the algorithm, we proceed similarly to the RDPG case. Assuming we want to estimate the signal at node i, we first set this entry at zero (i.e. $\mathbf{x}_i \leftarrow 0$). Then, the prediction will be the i-th entry of evaluating the deep GNN on the resulting signal. Of the month worth of measurements (amounting to more than 500 graphs), we took 90% for training and optimized the filter taps so as to minimize the mean squared error of the predicted RSSI on node i. Results for the remaining 10% are reported.

A relatively simple GNN with a single hidden layer using 64 features and $K = 5$ was implemented using a PyTorch-based library for GNNs (https://github.com/alelab-upenn/graph-neural-networks). Considering all the 81 physical links separately, using this method we obtain a RSME of 0.66 dB, whereas considering the threshold -75 dBm in the classification problem we obtain an encouraging result of 99% of accuracy.

TABLE I: Performance comparison. Accuracy (%) and RSME (dB) obtained considering the classification problem with a threshold of -75 dBm and measurements of the dataset (working with the same month worth of measurements in all the methods).

	RDPG-1	RDPG-2	GNN	BRT
Accuracy	64	73	99	80
RSME	17	10	0.66	6.1

V. Conclusions and Future Work

We study the problem of predicting wireless RSSI using machine learning on graphs. In Table I we summarize the results obtained by the considered approaches and the BRT algorithm.

The obtained performance results give an idea of how powerful GNNs are, specially taking into account that BRT is much more intensive computationally. Notice that during a month it is not rare to face changes on the propagation environment (see [13] for a study on this sense using the same dataset), so the method effectively learns to predict the actual RSSI on a given link during different situations. A natural question is whether it is possible to train a single GNN to predict all links. However, by using a GNN trained on a link and applying the model to predict the rest, we obtain a RSME of 26 dB and an accuracy of 51%. This lack of transferability and how to avoid it is our ongoing research.

Regarding RDPGs, we believe the underwhelming performance we obtained was due to the arbitrary weight choice on the "missing" link. By using the matrix's average instead of 0, we obtain a much better performance (RMSE=10 dB and an accuracy of 73%, see RDPG-2 of Table I). Our ongoing research on this sense are the design of more sophisticated methods to consider missing values on RDPGs.

References

[1] M. Najla, Z. Becvar, P. Mach and D. Gesbert, "Predicting Device-to-Device Channels From Cellular Channel Measurements: A Learning Approach," in IEEE Transactions on Wireless Communications, vol. 19, no. 11, pp. 7124-7138, Nov. 2020

[2] N. Raj, "Indoor RSSI Prediction using Machine Learning for Wireless Networks," 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS), 2021

[3] G. Cerar, H. Yetgin, M. Mohorčič and C. Fortuna, "Machine Learning for Wireless Link Quality Estimation: A Survey," in IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 696-728, Second quarter 2021

[4] Miguel Landry Foko Sindjoung, Pascale Minet. "Wireless Link Quality Prediction in IoT Networks". 8th IFIP/IEEE International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks, Nov 2019, Paris, France.

[5] D. Romero, D. Lee and G. B. Giannakis, "Blind Radio Tomography," in IEEE Transactions on Signal Processing, vol. 66, no. 8, pp. 2055-2069, 15 April15, 2018

[6] J. Young and E. R. Scheinerman, "Random dot product graph modelsfor social networks," inAlgorithms and Models for the Web-Graph,Anthony Bonato and Fan R. K. Chung, Eds., Berlin, Heidelberg, 2007,pp. 138–149, Springer Berlin Heidelberg.

[7] Benedek Rozemberczki and Rik Sarkar.Fast Sequence-Based Embedding with Diffusion Graphs. 2020. arXiv:2001.07463 [cs.LG].

[8] L. Ruiz, F. Gama and A. Ribeiro, "Graph Neural Networks: Architectures, Stability, and Transferability," in Proceedings of the IEEE, vol. 109, no. 5, pp. 660-682, May 2021

[9] Ajay Kumar et al. "Link prediction techniques, applications, and performance: A survey".In:Physica A: Statistical Mechanics and its Applications553 (2020), p. 124289.issn: 0378-4371

[10] Muhan Zhang and Yixin Chen. "Link Prediction Based on Graph Neural Networks". In:(Feb. 2018).

[11] Plan Ceibal, "About Plan Ceibal", https://www.ceibal.edu.uy/en/institucional, 2021 [Online; accessed July 21]

[12] Cisco, "Radio Resource Management White Paper," https://www.cisco.com/c/en/us/td/docs/wireless/controller/technotes/8-3/ b RRM White Paper.html, 2016, [Online; accessed July 2021].

[13] Capdehourat, G., Larroca, F., and Morales, G. (2020). A nation-wide Wi-Fi RSSI dataset: Statistical analysis and resulting insights. In: 2020 IFIP Networking Conference, Networking 2020, Paris, France, June 22–26, 2020, IEEE 370–378.

[14] Capdehourat, G., Bermolen, P., Fiori, M. et al. Large-Scale 802.11 Wireless Networks Data Analysis Based on Graph Clustering. Wireless Pers Commun (2021). https://doi.org/10.1007/s11277-021-08535-8

[15] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. "Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs". In:Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM '20). ACM. 2020

[16] C. E. Priebe, Y. Park, M. Tang, A. Athreya, V. Lyzinski, J. T. Vogelstein,Y. Qin, B. Cocanougher, K. Eichler, M. Zlatic, and A. Cardona, "Semiparametric spectral modeling of the drosophila connectome," arXiv:1705.03297 [stat.ML], 2017.

[17] F. Gama, A. G. Marques, G. Leus, and A. Ribeiro, "Convolutional Neural Network Architectures for Signals Supported on Graphs," IEEE Trans. Signal Process., vol. 67, no. 4, pp. 1034–1049, Feb. 2019.

[18] Ines Chami and Sami Abu-El-Haija and Bryan Perozzi and Christopher Ré and Kevin Murphy, "Machine Learning on Graphs: A Model and Comprehensive Taxonomy," arXiv:2005.03675, 2021.