RGCN for Beyond Pairwise Training: Generalizing Monitors Selection in Network Tomography

Zahraa El Attar, Kevin Hoarau, Yassine Hadjadj-Aoul, Géraldine Texier

RGCN for Beyond Pairwise Training: Generalizing Monitors Selection in Network Tomography

Zahraa El Attar*, zahraa.elattar@univ-rennes1.fr
Kevin Hoarau†, kevin.hoarau@univ-reunion.fr
Yassine Hadjadj-Aoul*, yassine.hadjadj-aoul@irisa.fr
Géraldine Texier*‡, geraldine.texier@imt-atlantique.fr

*Univ Rennes, Inria, CNRS, IRISA, Rennes, France
†Université de La Réunion, LIM, France
‡IMT Atlantique, Rennes, France

***Abstract*—In the dynamic field of 5G network monitoring, the ability to generalize monitors' placement across a network is crucial for comprehensive coverage. This study introduces the use of the Relational Graph Convolutional Network (RGCN) model to meet this challenge. We conducted a comparative analysis between the RGCN and a traditional Neural Network (NN) across two network topologies, considering every possible node configuration within these topologies. Our findings indicate that the RGCN model, once trained on a specific node pair, exhibits superior generalization ability and accuracy. It consistently transfers its learning to changes in monitors' placement and accurately estimates links delays beyond its initial training monitors. Unlike the NN, which showed significant limitations in generalizing monitors' placement and high error rates. This paper not only demonstrates the effectiveness of the RGCN model in generalizing the monitors' placement problem but also paves the way for its broader application in dynamic network monitoring contexts.**

***Index Terms*—RGCN, Neural Networks, Monitors, Network Tomography, 5G**

I. Introduction

In the evolving world of telecommunications, the advancements introduced by 5G networks bring to the forefront the limitations of traditional network monitoring methods, especially in terms of scalability, flexibility, and the significant costs associated with the deployment and maintenance of monitoring equipments. Characterized by their heterogeneity, 5G networks integrate diverse technologies, such as small cells, macro cells, and edge computing units. This diversity, while enabling a wide range of services and applications, complicates the task of comprehensive network monitoring. Moreover, 5G's need for accurate, non-intrusive monitoring is critical to support URLLC applications like autonomous vehicles.

Faced with these challenges, network tomography emerges as a viable solution. It provides network metrics estimation based on a limited number of end-to-end measurements that reflect network states, thus avoiding the need for extensive data collection and an intrusive approach. This makes the network monitoring task less costly, more scalable, and more flexible.

A typical network tomography solution involves three phases: selecting nodes for data collection, choosing measurement paths, and applying tomography techniques to infer network metrics.

Traditionally, network tomography solutions have involved three phases: selecting nodes for data collection, choosing measurement paths, and applying tomography techniques to infer network metrics. Previous research has largely focused on optimizing node selection based on specific criteria [1], [2], [3], [4] or on developing tomography techniques that assume a fixed monitor placement [5], [6], [7]. These approaches, while valuable, often lack the flexibility and adaptability required in dynamic 5G environments.

In contrast, our research introduces a new approach to overcome the limitations of traditional node selection and tomography techniques. By proposing a new tomography technique that leverages the Relational Graph Convolutional Network (RGCN) model, we aim to generalize the placement of monitors using just two monitors. Our methodology involves the training of an RGCN model using the data collected between two monitors, with the goal of learning the network's internal dynamics. We then assess the ability of the trained model to apply the knowledge it has gained to all other node pair configurations, thus evaluating its capability for generalization. Our approach, compared to a standard neural network solution across different network topologies, demonstrates the superior efficacy of the RGCN model.

The structure of this paper is as follows: Section II revisits the existing contributions. Section III defines the problem and discusses the conceptual model developed to meet this challenge. Section IV describes our proposed methodology in detail. Section V is devoted to a thorough analysis of the performance of our method. Finally, Section VI offers a concluding summary of our research.

II. Related work

The field of monitor selection in network tomography has seen significant advancements over the past two decades. However, despite these developments, various limitations and challenges persist across different approaches.

Early foundational work in the field, while crucial for establishing theoretical frameworks, often struggled with practical implementations. Horton and Lopez-Ortiz (2003), for instance, provided insights into the theoretical bounds of monitor placement [8]. However, their heuristic approach didn't guarantee optimal solutions, particularly for irregular network topologies. This limitation represent the gap between theoretical optimality and practical efficiency in real-world networks. Building on this foundation, Kumar and Schwabe (2006) made significant step in reducing the number of required monitors [9]. Yet, their approach, didn't address the critical trade-off between the number of monitors and the accuracy of network inference, a key challenge in the field.

As researchers began to explore more sophisticated mathematical approaches, new limitations emerged. Ahuja et al. (2008) focused on Shared Risk Link Group (SRLG) failures in all-optical networks [10], but their method was for high network connectivity. This assumption limits the applicability of their approach. Similarly, Gopalan et al. (2011) introduced an innovative algebraic method using network decomposition into three-edge-connected components [11]. However, this decomposition is computationally expensive for large networks and may not be optimal for networks that don't naturally decompose into such components.

The early 2010s saw a shift towards optimizing monitor placement for maximum link identifiability, with significant contributions from Ma et al. (2013, 2014) [12], [13], [14], [15]. Their Minimum Monitor Placement (MMP) algorithm and subsequent Greedy Maximal Identifiability Monitor Placement (GMMP) algorithm represented important advancements. However, they faced challenges in computational complexity for very large networks.

As network architectures became complex and dynamic, researchers began to address these challenges more directly. Gao et al. (2014) introduced the Scalpel algorithm, focusing on identifying "interesting" links [16]. However, this approach too is computationally intensive for large, complex networks. He and Yao (2016, 2017) made significant progress in addressing dynamic networks, but their approach of considering multiple possible topologies might be computationally expensive. [17], [18].

Recent advancements have seen the introduction of machine learning techniques, as exemplified by Rkhami et al. (2021) [19]. While promising, the approach didn't work on the generalizability of the monitors placement problem.

Our work builds upon these foundations while addressing several key limitations. Inspired by Ma et al. (2013), who demonstrated that up to 97% of links can be identified with just two monitors, we focus on a cost-effective and efficient approach using only two monitors. This strategy significantly reduces the infrastructure required for comprehensive network state knowledge, addressing the persistent challenge of balancing monitor numbers with inference accuracy. Furthermore, we introduce the application of machine learning to the monitor selection problem. While authors in [19] began exploring the use of Graph Neural Networks for predicting the number of monitors needed, our approach takes this further. We employ RGCN not just for prediction, but for determining optimal monitor placement. This represents a more comprehensive application of machine learning to the monitor selection problem, to overcome scalability and generalization issues in traditional methods. We focused on the dual capacity of generalizing monitor placement while accurately estimating link delays beyond its initial training domain. Thus, our research aims to bridge the gap between theoretical optimality and practical efficiency, a challenge that has been present since the early days of the field by leveraging machine learning techniques.

III. Methodology and Problem formulation

To achieve our objective, the problem is formulated to address the generalization of monitor placement as follows:

1) Network Representation: Consider a network represented as an undirected graph $G = (V, L)$, where V denotes the set of nodes and L denotes the set of links. For monitors used, let $M = \{m_1, m_2\} \subseteq V$ a pair of nodes used as monitors. Let P as the set of measurement paths between the monitors m_1 and m_2:

$$P_{m_1 m_2} = \{p_1, p_2, \ldots, p_k\}$$

where each path p_i consists of a sequence of links $(l_1, l_2, \ldots, l_{n_i})$. Let $X = (x_1, x_2, \ldots, x_{|L|})$ be the vector of unknown link delays, where x_i represents the delay of link l_i. For measurement paths, let $Y_{m_1 m_2} = (y_{m_1 m_2,1}, y_{m_1 m_2,2}, \ldots, y_{m_1 m_2,k})$ be the vector of measured path delays between monitors m_1 and m_2:

$$y_{m_1 m_2,i} = \sum_{j \in p_i} x_j \tag{1}$$

2) Links delay estimation for a given node pair: For any pair of nodes $(u, v) \in V$, the links delay X can be estimated using the measured path delays $Y_{m_1 m_2}$:

$$X = f(Y_{m_1 m_2}) \tag{2}$$

where f is an estimation function that calcultes link delays. We will dive deeper into the choice of f and explain it in details later on.

3) Links delay estimation error: To evaluate the accuracy of our link delay estimation, we use the Mean Absolute Percentage Error (MAPE). MAPE is defined as follows:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{X_i - \hat{X}_i}{X_i} \right| \times 100 \tag{3}$$

where X_i represents the actual link delay, $\hat{X}_i$ is the estimated link delay, and n is the total number of link delays being evaluated.

4) Evaluation of monitor placement generalization: The goal is to ensure that the delay estimation accuracy is consistent across all node pairs. If it is consistent then the problem is generalized. This can be evaluated by comparing the estimation errors of different node pairs. Specifically, the difference in estimation errors for different node pairs should be within an acceptable range ϵ:

$$|E_{uv} - E_{xy}| \leq \epsilon \text{ for all } (u, v), (x, y) \in V \quad (4)$$

where E_{uv} and E_{xy} are the estimation errors for node pairs (u, v) and (x, y) respectively.

IV. RGCN BASED SOLUTION

A. Motivation for RGCN

In network tomography, the network is naturally represented as graphs, the relationship between nodes and links is inherently relational, making it suitable for graph-based models. RGCN model is specifically designed to handle relational data, where different types of nodes and edges exist. Moreover, since we are working on generalization, RGCN known for its ability to generalize from a limited set of training data to new, unseen configurations. Once trained on a specific configuration (pair of monitors), it can effectively apply its learned knowledge to other node pairs.

B. Network representation: Relational graphs

RGCN model handles relational graphs. By definition, a relational graph is a type of graph that includes multiple types of nodes and edges to represent different entities and their relationships. What the RGCN model will learn is the relationships present in the relational graph, through various layers.

In our use case, since we are working on monitors generalization and part of the solving the problem is to estimate links delays, we need first to have our data as relational graphs and second we need a function that that can maps the relationships between measurement path delays and links delays. This function is all the functions encapsulated inside the RGCN model. What is then left is to construct the relational graphs from the initial network graph. Having the graph G and the two monitors m_1 and m_1, each measurement path in the network graph G is represented as a node in the relational graph, same thing for links. So each relational graph will have two types of nodes illustrated in figure 1.

- *Type 1 Nodes (Links):* Representing the links within the network having one feature representing delay.
- *Type 2 Nodes (Measurement Paths):* These nodes denote measurement paths in the network having one feature representing delay.

We also define two types of relationships between these node types

- Relation 1: Link $\rightarrow$ Path, indicating that a specific link is a part of a particular path.
- Relation 2: Path $\rightarrow$ Link, signifying that measurement paths are constituted by a set of network links

C. Model process

The functions encapsulated inside the RGCN model will serve us in estimating the links delays from path delays for a given node pair. The model goes through the following process to make predictions:

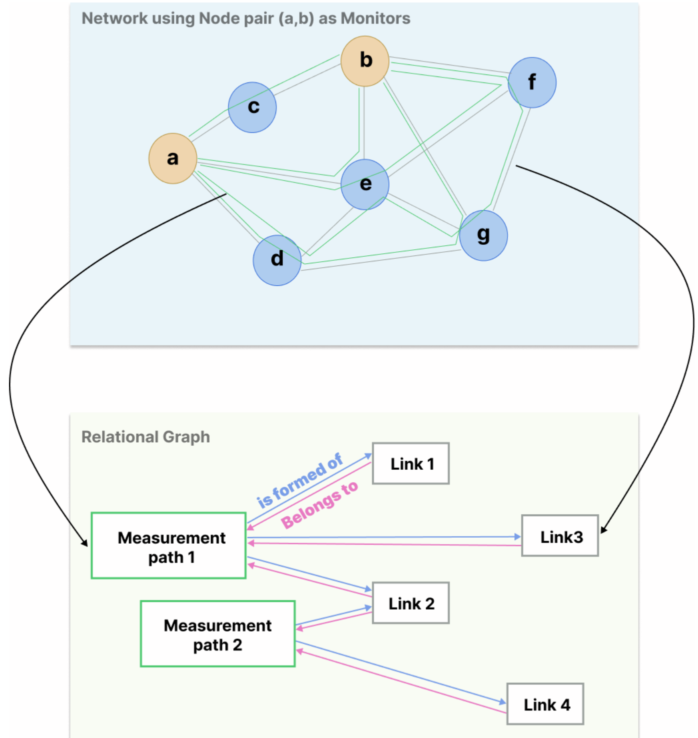

Fig. 1: Modelling Network Topology as Relational Graph

1) Initialization: The model begins by receiving two key inputs:

- **Node features:** we define delay as a feature for each node, denoted by a matrix $\mathbf{X} \in \mathbb{R}^{N \times D}$, where N is the number of nodes and D is the dimensionality of the feature vector for each node (delay), thus D is of dimension 1.
- **Graph structure:** Represented by the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$.

2) Forward Pass: The input data is processed through the network layers to generate the desired output (estimated link delays). The layer iterates through each unique edge type present in the graph. For each edge type, nodes features undergo a linear transformation using a weight matrix specific for each edge type. They are then aggregated and combined with original features. In all layers except the last, a Leaky ReLU activation function is applied to the updated features. This introduces non-linearity into the model, to capture complex relationships between measurement paths and link delays.

D. Training process

The functionalities described so far explain how the RGCN model uses graph information to estimate link delays. However, for the model to truly become capable of computing link delays, it requires training. Thus feeding the model with actual link delay data alongside the path delay data and graph structure.

1) Model training: In each iteration of training, the model performs the forward pass as described earlier, taking the node features and graph structure as input and generating estimated link delays as output. These estimated link delays are then compared with the actual link delays provided in the training data using a loss function. Based on the calculated loss, the model's internal parameters are adjusted. We used the $L1$ loss, defined as the mean of the absolute differences between the predicted values and the actual values. The formula for MAE is given by equation 5 where $X_{\text{estimated}}$ is the estimated link delays, X_{real} is the actual link delays and N is the number of data points in the set.

$$\text{MAE} = \left(\frac{1}{N}\right) \sum_{t=1}^{l} |X_{\text{real}} - X_{\text{estimated}}| \tag{5}$$

2) Model evaluation: To monitor the model's performance during training and prevent overfitting, the model is evaluated on the validation set after each training epoch using MAPE error. The model that have the lowest MAPE on the validation set is selected for deployment.

3) Model testing: The model is also evaluated on a separate testing set, unseen by the model during training. This provides a final assessment of the model's generalizability on unseen data.

E. Generalization assesment:

To assest the generalization monitors placement, we use the RGCN model in testing mode. We conduct exhaustive tests on all the possible node pairs configurations present in the network. We collect the testing MAPE errors and compare them with testing MAPE of the original monitoring nodes.

V. Performance Evaluation

A. Evaluation Setup

Our evaluation employs two distinct network topologies, referred to as Topology A and Topology B. These topologies are illustrated in Fig. 2 and Fig. 3, respectively.

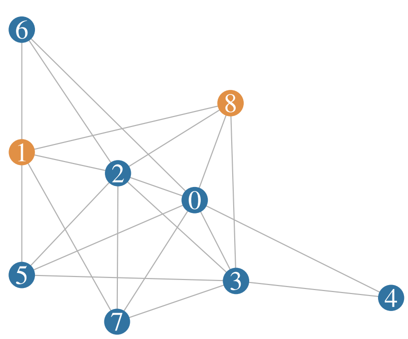

Fig. 2: Topology A

Topology A, consisting of 9 nodes and 22 links. In contrast, Topology B (a mandala topology) was obtained from [20] and has 26 nodes and 48 links. For the training phase: nodes $(1, 8)$ are the monitoring nodes for Topology A, and nodes $(4, 8)$ are the monitoring nodes for Topology B. In order to benchmark our solution, we compare it against the work of the authors in [21] and [22], where they implemented a classic neural network (NN) as their solution.

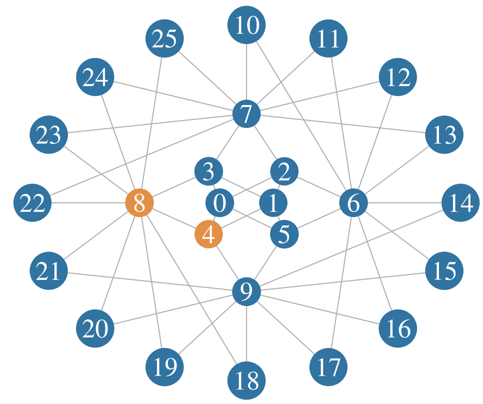

Fig. 3: Topology B

B. Topology A: Simulation Results

We trained the baseline NN model and the RGCN model on 10^5 graphs. The hyperparameters of these models were finely tuned to optimize their performance.

For Topology A, we found the median MAPE over the training dataset to be 12.00% for the NN model and at 14.1%, for the RGCN.

Phase 1 testing results: Upon testing these models on the testing dataset, the NN showed a median MAPE value of 12.1%, while the RGCN model exhibited a value of 13.9% (Fig.4)

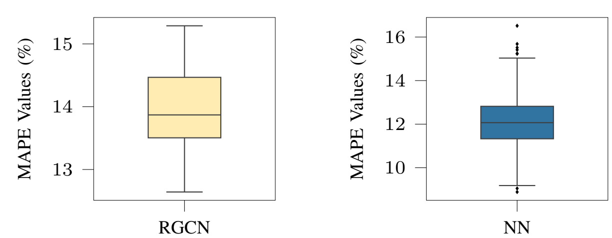

Fig. 4: **Topogoly A**: Testing NN and RGCN Models on Training node pair

Phase 2 testing results: [Generalisation]: The trained model is then tested on all possible node pairs combinations. For topology A, we have 36 node pair combinations, thus 36 test runs for each model. Fig.5 shows the testing results of all node pairs configuration using RGCN model for topology A. It illustrates the RGCN's performance through a consistent yellow palette. This uniformity in color across the 9x9 grid represents stable MAPE values, ranging from 13-15%. Such consistency demonstrates the RGCN's ability to generalize effectively across all possible node pairs, maintaining accuracy levels comparable to its performance on the training pair.

On the other hand, the results for NN performance on the same topology with a blue color scheme is in Fig. 6. The uniform coloration across the grid indicates consistently high MAPE values exceeding 120% for all node pair combinations. This visual representation illustrates the NN's failure to generalize beyond its training scenario.

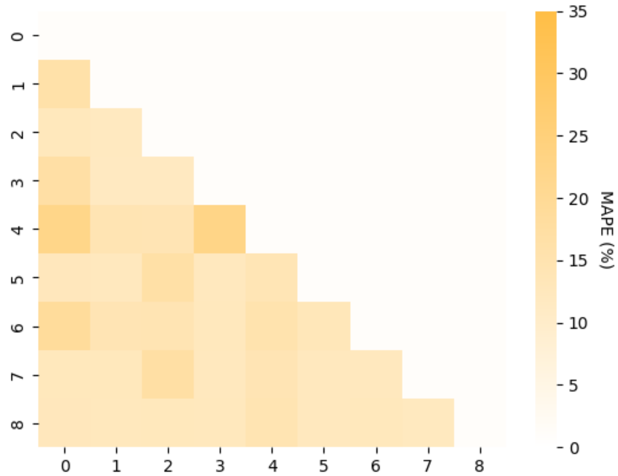

Fig. 5: **Topogoly A**: Testing RGCN Model on all node pairs combinations

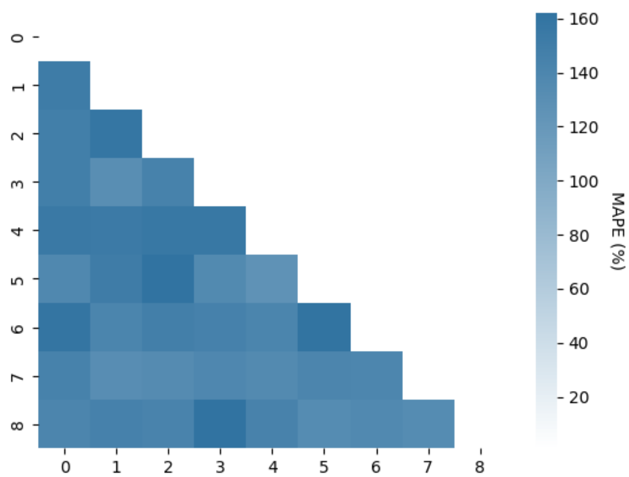

Fig. 6: **Topogoly A**: Testing NN on all node pairs combinations

The conclusion of these simulation results is two fold: In the context of approximating the mapping between measurement path delays and link delays for a fixed node pair, the results of phase 1 demonstrate that the RGCN model achieves competitive performance compared to a simpler, yet easier to train NN model. In terms of generalizing monitor placement, the results from phase 2 emphasize the superiority of the RGCN model. These findings support the idea that the RGCN model has learned patterns and relationships that are not specific to the monitor pair used in training. Instead, the RGCN model demonstrates its ability to generalize its knowledge to new monitor pairs.

C. Topology B: Simulation Results

As we transition to a more complex topology, the results gains additional depth. We replicated the evaluation methodology for Topology B. Training the models on the monitoring node pair 4 and 8 resulted in a training MAPE of 25.9% for the NN and 20.7% for the RGCN.

Phase 1 testing results: The testing results are shown in Fig.7 with the NN showing a median MAPE of 25.5% and the RGCN, 20.6%. So, the results of phase 1 demonstrate that the RGCN model is better approximating at the mapping between measurement path delays and link delays for a fixed node pair than NN model.

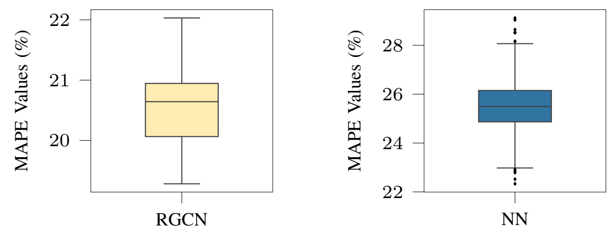

Fig. 7: **Topogoly B**: Testing NN and RGCN Models on Different node pairs configuration

Phase 2 testing results: [Generalisation]: We assess generalization on all node pairs configurations possible. For topology B, we have 326 possible node pair combinations, thus we conducted 326 runs of testing for each model. The results are shown in Fig.8 and Fig.9 for RGCN and NN respectively.

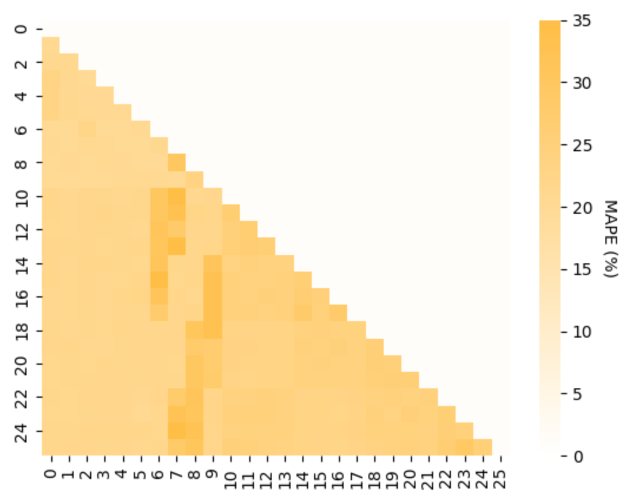

Fig. 8: **Topogoly B**: Testing RGCN Model on all node pair combinations

Fig. 8 show the RGCN's performance across a larger 26x26 grid. The persistent yellow hues, now representing MAPE values in the 20-22% range, demonstrate the model's ability to scale effectively to more complex network structures while maintaining its generalization capabilities. This consistency

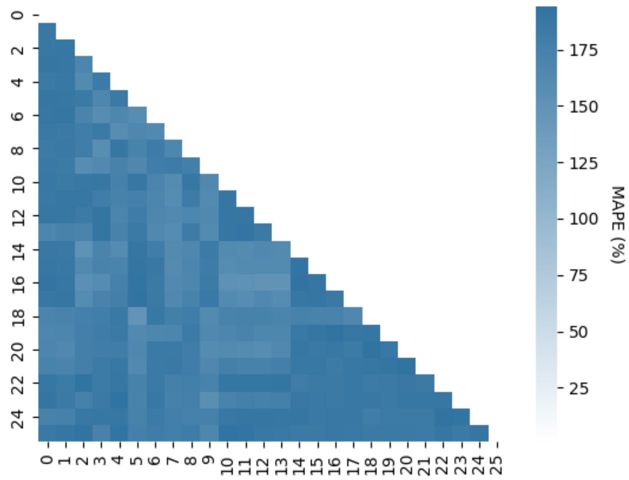

Fig. 9: **Topogoly B**: Testing NN on all node pair combinations

across a larger topology underscores the robustness of the RGCN approach.

Fig. 9, representing the NN's performance on Topology B, presents NN failure to generalize. The uniform coloration indicates MAPE values consistently surpassing 150% across all node pairs. This visual data representation shows the NN's inability to adapt to different node pairs.

VI. Conclusion

In this paper, we investigated the problem of generalizing monitors placement using only two monitors by proposing a new tomography technique based on a machine learning solution. Knowing the relational nature of this problem, we choose the Relational Graph Convolutional Network (RGCN) model to meet this challenge. We performed a comparative evaluation to compare our method against a baseline Neural Network (NN) solution. We applied both models to two distinct network topologies and tested them on all possible node pair configurations. Our findings reveal that RGCN model once trained on a specific node pair, can accurately transfer its knowledge to changes in monitor placement, effectively estimating network conditions beyond the nodes it was initially trained on. In contrast to the NN solution that is incapable to do so. This adaptability of the RGCN model highlights its potential for broader application in network monitoring, offering a versatile solution for dynamic network environments.

References

[1] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 906–916, 2012.

[2] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Identifiability of link metrics based on end-to-end path measurements," in *ACM IMC*, 2013, number of Monitors.

[3] Y. Gao, W. Wu, W. Dong, C. Chen, X. Y. Li, and J. Bu, "Preferential link tomography: Monitor assignment for inferring interesting link metrics," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, October 2014, pp. 167–178.

[4] W. Ren and W. Dong, "Robust network tomography: K-identifiability and monitor assignment," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, April 2016, pp. 1–9.

[5] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Linear algebra*, vol. 2, pp. 134–151, 1971.

[6] Y. Chen, D. Bindel, H. Song, and R. H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004, pp. 55–66.

[7] Y. Chen, D. Bindel, H. H. Song, and R. H. Katz, "Algebra-based scalable overlay network monitoring: algorithms, evaluation, and applications," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1084–1097, 2007.

[8] J. D. Horton and A. López-Ortiz, "On the number of distributed measurement points for network tomography," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, 2003, pp. 204–209.

[9] R. Kumar and J. Kaur, "Practical beacon placement for link monitoring using network tomography," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2196–2209, 2006.

[10] S. S. Ahuja, S. Ramasubramanian, and M. Krunz, "Srlg failure localization in all-optical networks using monitoring cycles and paths," in *IEEE INFOCOM 2008-The 27th Conference on Computer Communications*. IEEE, 2008, pp. 700–708.

[11] A. Gopalan and S. Ramasubramanian, "On identifying additive link metrics using linearly independent cycles and paths," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 906–916, 2011.

[12] L. Ma, T. He, K. K. Leung, A. Swami, and D. Towsley, "Identifiability of link metrics based on end-to-end path measurements," in *Proceedings of the 2013 conference on Internet measurement conference*, 2013, pp. 391–404.

[13] ——, "Link identifiability in communication networks with two monitors," in *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2013, pp. 1513–1518.

[14] ——, "Monitor placement for maximal identifiability in network tomography," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1447–1455.

[15] L. Ma, T. He, A. Swami, D. Towsley, and K. K. Leung, "On optimal monitor placement for localizing node failures via network tomography," *Performance Evaluation*, vol. 91, pp. 16–37, 2015.

[16] Y. Gao, W. Wu, W. Dong, C. Chen, X.-Y. Li, and J. Bu, "Preferential link tomography: Monitor assignment for inferring interesting link metrics," in *2014 IEEE 22nd International Conference on Network Protocols*. IEEE, 2014, pp. 167–178.

[17] T. He, L. Ma, A. Gkelias, K. K. Leung, A. Swami, and D. Towsley, "Robust monitor placement for network tomography in dynamic networks," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[18] T. He, A. Gkelias, L. Ma, K. K. Leung, A. Swami, and D. Towsley, "Robust and efficient monitor placement for network tomography in dynamic networks," *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1732–1745, 2017.

[19] A. Rkhami, Y. Hadjadj-Aoul, G. Rubino, and A. Outtagarts, "Mongnn: A neuroevolutionary-based solution for 5g network slices monitoring," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, 2021, pp. 185–192.

[20] W. da Silva Coelho, A. Benhamiche, N. Perrot, and S. Secci, "Function splitting, isolation, and placement trade-offs in network slicing," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, pp. 1920–1936, 2021.

[21] A. Rkhami, Y. Hadjadj-Aoul, G. Rubino, and A. Outtagarts, "Mongnn: A neuroevolutionary-based solution for 5g network slices monitoring," in *2021 IEEE 46th Conference on Local Computer Networks (LCN)*. IEEE, October 2021, pp. 185–192.

[22] M. Rahali, J.-M. Sanner, and G. Rubino, "Tom: a self-trained tomography solution for overlay networks monitoring," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2020, pp. 1–6.