Enhancing Graph Neural Networks with Random Graph Ensembles

Jan von Pichowski **Vincenzo Perri** **Ingo Scholtes**
Chair of Machine Learning for Complex Networks
Centre for Artificial Intelligence and Data Science (CAIDAS)
Julius-Maximilians-Universität Würzburg, Germany
{jan.pichowski,vincenzo.perri,ingo.scholtes}@uni-wuerzburg.de

Abstract

Graph Neural Networks (GNNs) have shown remarkable performance in various network analysis tasks. However, their results depend on the reliability of the network structure, making them sensitive to inherent variability in real-world data. This study investigates the use of graph ensembles to improve GNN performance, focusing on node classification tasks. We use random graph ensembles to define edge scores, quantifying the deviation of observed edge frequencies from those expected based on node activity. This approach allows us to distinguish between statistically significant connections and those potentially arising from random fluctuations in the network structure. We use this information to refine the message-passing procedure, aiming to enhance node representations and increase performance in downstream tasks. In our experiments, we propose and evaluate two ensemble-based strategies. Our results show that these strategies lead to better GNN performance in four out of five datasets. Our work lays a foundation for future research, opening new avenues for either applying other random graph ensembles to GNNs, or considering other graph-based tasks.

1 Introduction

In graph-based learning, the observed network structure is often assumed to accurately represent the underlying system. However, this assumption overlooks the inherent variability and uncertainty in network formation processes. Graph ensembles offer a way to address this challenge by providing a statistical baseline against which observed networks can be compared, enabling the distinction between meaningful structural patterns and random fluctuations. This methodology has proven valuable in various network analysis tasks, such as node clustering [1], identifying significant temporal patterns in dynamic networks [2], and various other applications [3]. Despite their demonstrated utility in discerning meaningful structural patterns, graph ensembles have not been used for refining the network topology in message-passing graph neural networks.

Identifying statistically meaningful patterns in networks is crucial for Graph Neural Networks (GNNs) due to their message-passing mechanism. GNNs leverage the structure of graph data to propagate information through nodes and edges, capturing complex relational patterns for predictive modeling. However, this reliance on network structure makes GNNs sensitive to the inherent variability in real-world datasets, which graph ensembles are particularly well-suited to address. While data cleaning approaches partially address similar challenges by removing spurious connections due to measurement errors [4–6], ensemble methods offer deeper insights. Beyond the binary classification of *correct* or *incorrect* edges, the existence and frequency of observations carry valuable statistical information in ensemble frameworks. For instance, the number of connections between two high-degree nodes may still be high in absolute terms, but it acquires a different interpretation when compared to its statistical expectation based on a random model that fixes node degrees. If the observed connection frequency is much higher or lower than expected, it suggests that the interaction is not merely a product of random chance but may instead indicate a meaningful pattern.

von Pichowski et al., Enhancing Graph Neural Networks with Random Graph Ensembles (Extended Abstract). Presented at the Third Learning on Graphs Conference (LoG 2024), Virtual Event, November 26–29, 2024.

In this study, we investigate the role of graph ensembles in improving the performance of GNNs, with a specific focus on node classification tasks. We introduce and evaluate a novel approach based on a *soft configuration model* [7], which we use to generate edge scores that quantify the deviation of observed edge frequencies from those expected under a model of node activity that fixes the nodes' degrees. By identifying edges that are over or under represented compared to the ensemble, this method allows us to distinguish between statistically significant connections and those that may arise from random fluctuations within the network. By leveraging this ensemble-based assessment of network structure, the message-passing process in GNNs can be refined, leading to more accurate node representations and, ultimately, improved performance in downstream tasks. Our findings demonstrate that the proposed ensemble-based strategies can enhance the performance of GNNs. We achieve greater classification performance in four out of five data sets.

This paper highlights how using ensemble models to assign statistically meaningful edge weights refines the message-passing process, leading to higher-quality node representations and, ultimately, better performance on downstream tasks. These improvements underscore the potential of integrating statistical and neural approaches to refine GNN models. The results not only deepen our understanding of the relationship between network structure, statistical significance, and GNN performance but also open new avenues for applying ensemble-based techniques to a wider array of graph-based tasks.

2 Related Work

Our approach is related to works that address structural noise in real-world graph data, where edges may be unreliable due to measurement errors, incomplete information, or data collection variability. Several methods have been proposed to mitigate the impact of such noise in Graph Neural Networks (GNNs). Techniques like DropEdge [8], which randomly removes edges during training, and GraphSAGE [9], which uses sampling and aggregation, help GNNs become more robust to spurious connections [10]. GAT [11] employs attention mechanisms to learn edge weights, potentially filtering noisy edges, while sparsification methods such as NeuralSparse [6] and GSML [5] focus on removing irrelevant edges [12]. Graph rewiring methods [13, 14] also add new edges.

Despite their effectiveness, these approaches often reduce edge significance to a binary classification—either correct or incorrect—which may oversimplify the rich statistical information encoded in edge frequency and patterns. Random graph ensembles offer a more nuanced perspective by analyzing deviations in edge occurrences rather than merely their presence or absence. These ensembles have long been important in network analysis, serving as a baseline to identify relevant structures by comparing observed features to those expected by chance. Integrating such methods into machine learning has yielded promising results. For example, graph classification has benefited from data augmentation techniques based on graph ensembles [15]. DMoN [16] employs modularity [1] to evaluate how clustering patterns deviate from those predicted by random graph ensembles. Finally, HYPA-DBGNN [17] has improved performance in dynamic network tasks by utilizing anomalies in sequential patterns based on HYPA scores [2].

Our work takes an approach close to the one of HYPA-DBGNN. HYPA-DBGNN uses a relaxation [7] of the Molloy-Reed configuration model [18]. The Molloy Reed model provides a principled null baseline by maintaining fixed vertex degrees while shuffling edges, offering insights into whether observed connections deviate from expected random connectivity patterns. In the contract of temporal networks, the soft configuration model was extended to account for over and under represented sequential patterns [2]. In HYPA-DBGNN this information on sequential patterns is used to inform a temporal De Bruijn GNN [19]. Similar to the works above, we consider the Molloy Reed model and its soft configuration extension as our starting point. However, we use this perspective to define edge scores bases on statistical significance and propose strategies to refine the topology for a GNN.

3 Methodology

We consider the problem of node classification on data that are generated by an unknown underlying graphical process. We focus on a scenario where the distribution of edge frequencies is skewed due to the presence of highly active nodes. Already by chance, the high activity of nodes results in larger edge frequencies, and can obscure important but less frequent connections. This hides the underlying relational structure of the graph. To address this issue, in the following paragraphs we define edge scores based on a soft configuration model [7] and propose strategies for refining the graph topology.

Soft configuration model. The *soft* configuration model [7] addresses limitations of the Molloy-Reed model [18], which is both computationally expensive and not analytically tractable. The soft version provides a closed-form expression for the null model that is an urn problem. It relaxes the property of a fixed degree sequence to *expected* vertex degrees. The model defines a matrix $\Xi \in \mathbb{N}^n \times \mathbb{N}^n$ that contains the entries in the urn. The entries are the possible stub combinations defined as the product of the observed in- d_j^{in} and out-degrees d_i^{out} in the observed graph $\mathcal{G}$: $\Xi_{ij} = d_i^{out} \cdot d_j^{in}$ The total number M of stub combinations or edge placements is $M = \sum_{ij} \Xi_{ij}$. Instead of algorithmically merging in- and out-stubs as in the Molloy Reed model, a network is sampled by drawing $m = \sum_i d_i^{out} = \sum_j d_j^{in}$ stub combinations without replacement. This difference results in vertex degrees that are equivalent to the observed graph only in the expected case. However, the model becomes analytically tractable, as the probability of observing an edge A_{ij} between the vertices i and j can be expressed by a hypergeometric distribution: $P(X_{ij} = A_{ij}) = \binom{M}{m}^{-1}\binom{\Xi_{ij}}{A_{ij}}\binom{M-\Xi_{ij}}{m-A_{ij}}$. In our approach, we use $P(X_{ij} = A_{ij})$ for quantifying the relevance of observed edges frequencies; it provides the starting point for our graph augmentation and GNN enhancement approaches.

Defining edge scores. We define edges scores by computing the cumulative of the probability of edge frequencies $P(X_{ij} = A_{ij})$ in the soft configuration model. Accumulating the observation likelihood to $P(X_{ij} \leq A_{ij})$ measures how likely an edge in a random graph instance occurs with a frequency lower than the observed one. A probability of $P(X_{ij} \leq A_{ij}) = 0.5$ means that half of the random realizations have a lower edge frequency than observed and the other half have a higher one. Hence, the observed frequency matches the expected frequency. A probability of $P(X_{ij} \leq A_{ij}) > 0.5$ means that for any random realization the chance is higher that the edge frequency is lower than observed. Hence, the observed edge is *over represented* compared to the null model and thus has a higher relevance. For $P(X_{ij} \leq A_{ij}) < 0.5$ the edge is *under represented*.

While high edge frequencies can occur as a result of high node activities (i.e. degree) only, and not carry information about the relatedness of nodes, these scores help to identify statistically relevant connections in the underlying process. Specifically, these connections are significant because they deviate from the expectation defined by the soft configuration model. In the context of a GNN, we could have over expressed edges with low frequency compared to the average frequencies in the graph. These edges have little impact on standard message passing, but their frequency value, conditioned on the low node activities, is unexpectedly high and should have a strong impact. The opposite holds for under represented edges with high frequency that should have a lower influence on the GNN.

For instance, the high schools, workplaces, and hospitals data have inherent network properties where certain nodes (such as managers in a workplace) are naturally more active due to their roles. Counting their interactions does not reveal how statistically unexpected certain connections are. Two managers might have a high number of connections in absolute numbers, but this number could lower than what one would expect at random given the number of connections each one has. The soft configuration model helps revealing underlying structures by distinguishing statistically significant connections from those expected by chance. Without this, the connection patterns are defined by node activity.

Graph topology refinement. We use this information to refine the graph topology in two ways: (I) We use the edge scores as weights instead of the edge frequencies. This way, over-represented edges have a strong impact while under-represented ones have a lower influence. (II) We filter the graph by removing under representing edges, ensuring that the graph only contains connections that occur more frequently than expected by chance. After applying one of these refinement strategies, we pass the reweighted topology to a GNN. The enhanced topology more accurately reflects the underlying graphical process, and thus has the potential to improve the GNN's performance.

4 Empirical Evaluation

We evaluate the impact of the graph correction methods in a node classification task for social data with frequent interactions. We test our method on face-to-face interaction data collected by the SocioPatterns collaboration in various social settings: two *highschool* [20] datasets, two *workplace* [21] datasets, and one *hospital* [22] dataset. The process of collecting face-to-face interaction is prone to issues that can be addressed with random graph ensembles. Some classes of nodes might have a higher rate of interactions due to their role. For example, two managers might have a number of interactions higher than average, and yet their interaction might be under expressed,

Table 1: Comparison of GCN variants. Models 3-6 use at least one configuration-based augmentation (3 in Edge Scores or Filtered Edges). Best balanced accuracy per dataset is in bold.

	Edge	**Edge**	**Filtered**	**SocioPatterns**				
#	**Freq.**	**Scores**	**Edges**	Highschool11	Highschool12	Hospital	Workplace13	Workplace15
1	✗	✗	✗	50.93 ± 8.71	51.70 ± 5.79	61.58 ± 18.50	86.75 ± 13.39	63.06 ± 6.69
2	✓	✗	✗	**56.72 ± 11.37**	51.47 ± 3.40	59.83 ± 20.78	83.46 ± 14.80	83.88 ± 9.71
3	✗	✗	✓	49.52 ± 8.09	**58.55 ± 5.84**	67.75 ± 13.81	**87.17 ± 10.59**	67.74 ± 4.87
4	✓	✗	✓	54.62 ± 8.05	48.96 ± 8.96	62.00 ± 21.65	84.08 ± 15.40	**84.07 ± 8.19**
5	✗	✓	✗	54.15 ± 8.24	57.10 ± 7.06	**70.58 ± 22.16**	**87.17 ± 10.59**	64.68 ± 8.11
6	✗	✓	✓	52.37 ± 9.35	55.23 ± 5.63	66.50 ± 20.19	**87.17 ± 10.59**	68.25 ± 5.29

possibly indicating a valuable source of information. Additionally, the data collection process is inherently noisy. Interactions in locations like the cafeteria provide numerous interactions that do not reflect relevant patterns like friendships, roles in a hospital or in a work environment. We predict: student gender (two classes) for *highschool*, worker's department for *workplace* (five classes in 13, twelve in 15), and medical personnel roles or patient status (four classes) for *hospital*.

For our experiments we use the standard GCN architecture by Kipf and Welling [23] (where we include edge weights) with two layers with embedding size 32, ReLU activation, dropout and batch normalization. A final linear layer transforms the output for the downstream class. We train and compare six GCN variants, each differing in whether we prune under represented edges (approach (I) in section 3), whether we use edge weights, and whether the edges weights we use are frequencies or the scores we outlined in the section 3 (approach (II)). Models are optimized using Stochastic Gradient Descent (learning rate 0.01) and evaluated using stratified ten-fold cross validation. We use early stopping (patience 100) on a validation set (10% of training data) to select the best epoch.

Table 1 reports the balanced accuracy of our node classification experiments (with numbers in parentheses referring to table rows). We use two baselines: a standard GCN without edge frequencies (1) and a GCN with edge frequencies (2). The remaining four variants incorporate our refinements based on the soft configuration model: GCN with under-expressed edges filtered out (3), GCN with under-expressed edges filtered out and edge frequencies as weights (4), GCN with edge scores as weights (5), and GCN with edge scores and pruned under-expressed edges (6). These variants allow us to evaluate the impact of our edge refinement strategies on classification performance. Our results show that augmenting the graph structure leads an improved performance in 4 out of 5 cases. There is no clear winner among the approaches with enhanced scores, as each one obtains the best score in at least one dataset. This likely indicates that different refinements are appropriate for the different data sets. We compare the best performing refined approach to the best performing baseline. For the *workplace* datasets we get minor improvements of 0.48% and 0.23%, but we note that for these dataset the baselines already achieve a good performance. In contrast, we observe significant gains of 13.25% for *highschool12* and 14.62% for *hospital*. Our straightforward approach yields comparable results to baselines in three scenarios (with slight improvements in two), while significantly outperforming them in two others. This suggests promising avenues for further refinement and enhancement.

5 Conclusion

In this work, we used the soft configuration model to define edge scores that distinguish between statistically significant connections and those arising from random fluctuations. We proposed two approaches to refining the network topology used by GNNs for message passing based on these edge scores, achieving improvements in node classification for four out of five data sets. These improvements indicate the statistical information from graph ensembles can be fruitfully integrated into GNNs, and align with other works highlighting the benefits of incorporating statistical information to enhance neural methods [17, 24, 25]. The results are promising beyond the scope of the datasets we used (social interactions) and the considered task (node classification). Our work provides a basis for further research that examines various tasks (e.g., link prediction and graph classification) across different data sources. One important question for future research is whether general reweighting approaches can be learned from the data in an end-to-end manner. Additionally, exploring other graph ensemble approaches might provide information to enhance the representations of other network structures. In conclusion, this work contributes to the integration of ensemble methods with GNNs and characterizes it as one way to create more reliable models.

References

[1] Mark EJ Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006. 1, 2

[2] Timothy LaRock, Vahan Nanumyan, Ingo Scholtes, Giona Casiraghi, Tina Eliassi-Rad, and Frank Schweitzer. Hypa: Efficient detection of path anomalies in time series data on networks. In *Proceedings of the 2020 SIAM international conference on data mining*, pages 460–468. SIAM, 2020. 1, 2

[3] Mikhail Drobyshevskiy and Denis Turdakov. Random graph modeling: A survey of the concepts. *ACM computing surveys (CSUR)*, 52(6):1–36, 2019. 1

[4] Mingze Dong and Yuval Kluger. Towards understanding and reducing graph structural noise for gnns. In *International Conference on Machine Learning*, pages 8202–8226. PMLR, 2023. 1

[5] Guihong Wan and Harsha Kokel. Graph sparsification via meta-learning. *DLG@ AAAI*, 2021. 2

[6] Cheng Zheng, Bo Zong, Wei Cheng, Dongjin Song, Jingchao Ni, Wenchao Yu, Haifeng Chen, and Wei Wang. Robust graph representation learning via neural sparsification. In *International Conference on Machine Learning*, pages 11458–11468. PMLR, 2020. 1, 2

[7] Giona Casiraghi and Vahan Nanumyan. Configuration models as an urn problem. *Scientific Reports*, 11(1), June 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-92519-y. URL `http://dx.doi.org/10.1038/s41598-021-92519-y`. 2, 3

[8] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019. 2

[9] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017. 2

[10] Zeyu Zhang and Yulong Pei. A comparative study on robust graph neural networks to structural noises. *arXiv preprint arXiv:2112.06070*, 2021. 2

[11] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018. 2

[12] Wei Ju, Siyu Yi, Yifan Wang, Zhiping Xiao, Zhengyang Mao, Hourun Li, Yiyang Gu, Yifang Qin, Nan Yin, Senzhang Wang, et al. A survey of graph neural networks in real world: Imbalance, noise, privacy and ood challenges. *arXiv preprint arXiv:2403.04468*, 2024. 2

[13] Adrian Arnaiz-Rodriguez, Ahmed Begga, Francisco Escolano, and Nuria Oliver. Diffwire: Inductive graph rewiring via the lovász bound, 2022. URL `https://arxiv.org/abs/2206.07369`. 2

[14] Rickard Brüel-Gabrielsson, Mikhail Yurochkin, and Justin Solomon. Rewiring with positional encodings for graph neural networks, 2023. URL `https://arxiv.org/abs/2201.12674`. 2

[15] Zeyu Wang, Jinhuan Wang, Yalu Shan, Shanqing Yu, Xiaoke Xu, Qi Xuan, and Guanrong Chen. Null model-based data augmentation for graph classification. *IEEE Transactions on Network Science and Engineering*, 2023. 2

[16] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *Journal of Machine Learning Research*, 24(127):1–21, 2023. 2

[17] Jan von Pichowski, Vincenzo Perri, Lisi Qarkaxhija, and Ingo Scholtes. Inference of sequential patterns for neural message passing in temporal graphs. *arXiv preprint arXiv:2406.16552*, 2024. 2, 4

[18] Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random Struct. Alg.*, 6(2-3):161–180, March 1995. ISSN 1098-2418. doi: 10.1002/rsa.3240060204. 2, 3

[19] Lisi Qarkaxhija, Vincenzo Perri, and Ingo Scholtes. De bruijn goes neural: Causality-aware graph neural networks for time series data on dynamic graphs. In *Learning on Graphs Conference*, pages 51–1. PMLR, 2022. 2

[20] Julie Fournet and Alain Barrat. Contact patterns among high school students. *PLoS ONE*, 9 (9):e107878, September 2014. ISSN 1932-6203. doi: 10.1371/journal.pone.0107878. URL `http://dx.doi.org/10.1371/journal.pone.0107878`. 3, 6

[21] Mathieu Génois, Christian L. Vestergraad, Julie Fournet, André Panisson, Isabelle Bonmarin, and Alain Barrat. Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science*, 3(3):326–347, March 2015. ISSN 2050-1250. doi: 10.1017/nws.2015.10. URL http://dx.doi.org/10.1017/nws.2015.10. 3, 6

[22] Philippe Vanhems, Alain Barrat, Ciro Cattuto, Jean-François Pinton, Nagham Khanafer, Corinne Régis, Byeul-a Kim, Brigitte Comte, and Nicolas Voirin. Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. *PLoS ONE*, 8(9): e73970, September 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0073970. URL http://dx.doi.org/10.1371/journal.pone.0073970. 3, 6

[23] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017. 4

[24] Soumyasundar Pal, Saber Malekmohammadi, Florence Regol, Yingxue Zhang, Yishi Xu, and Mark Coates. Non parametric graph learning for bayesian graph neural networks. In *Conference on uncertainty in artificial intelligence*, pages 1318–1327. PMLR, 2020. 4

[25] Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5829–5836, 2019. 4

A Complexity

Our method combines GCNs with certain edge scores. The graphs are at most reduced due to the filtering behavior. Other than that we only add the edges scores as edge weight attributes. Hence our method has the same runtime complexity as a vanilla GCN.

Additionally, the edge scores need to be calculated. Therefor we sample the described hyper-geometric distribution. Even though this is a complex operation it is still linear in terms of the edges. During our experiments we observe that the pre-processing requires notably less time than the training of the GCNs.

B Data Sets

Our choice of datasets is driven by the problem we consider, that is, addressing inherent variability of multi-edge observations in real world data. Therefore, our work focuses on the edge frequencies. The used data sets are particularly apt for the task, as they are face-to-face interactions recorded from proximity sensors in a variety of social domains. Thus, they encode frequent (and potentially spurious) interactions between people. Alternative datasets, like popular molecule data sets from Open Graph Benchmark, only contain edges with attributes that describe the bond but they do occur exactly once in the molecule. Hence, they do not fit the problem considered in this work.

Table 2: Overview of data set properties.

	Highschool2011 [20]	Highschool2012 [20]	Hospital [22]	Workplace2013 [21]	Workplace2015 [21]
Vertex Count	126	180	75	92	217
Edge Count	3419	4440	2278	1510	8548