Centrality Aware Augmentations for Self-Supervised Graph Learning

Aakash Aanegola
Columbia University
aa5506@columbia.edu

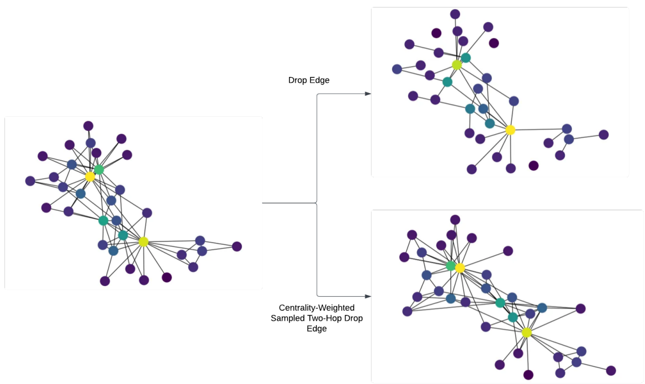

Fig. 1: A qualitative comparison of our technique with DropEdge. Note that the graph structure is maintained in the second image, and fewer nodes are disconnected from the rest of the graph.

***Abstract*—Self-supervised learning on graphs has gained significant traction due to its ability to learn meaningful node representations without labeled data. A critical component of these methods is data augmentation. Existing approaches predominantly rely on uniform edge dropout, ignoring important structural and contextual variations. We propose centrality-aware edge dropout and adaptive edge addition techniques to enhance graph augmentations for self-supervised learning on graphs. Our methods leverage node centrality metrics to dynamically adjust edge dropout probabilities and add edges, improving representation learning, particularly for underrepresented low-centrality nodes. We validate our techniques on two state-of-the-art frameworks, BGRL and GRACE, across diverse datasets, evaluating performance on node classification, node similarity search, and group fairness. Results demonstrate consistent improvements, with eigen-centrality and two-hop sampling emerging as key contributors to the success of our augmentations in addition to accounting for node centrality. This study underscores the importance of structure-aware strategies in advancing graph self-supervised learning and offers a scalable pathway for improving fairness and representation quality with minimal effort.**

***Index Terms*—Self-Supervised Learning, Graph Representation Learning, Fairness, Graph Augmentation**

I. Introduction

Graph neural networks (GNNs) have emerged as powerful tools for learning representations on graph-structured data, with applications ranging from social network analysis to molecular property prediction [1]. A cornerstone of recent advancements in self-supervised GNNs is using data augmentations to create correlated graphs from which representations can be contrasted and learned [2, 3]. However, despite their success, these augmentations generally do not take any structural information into account and could lead to subpar node and graph-level representations. The most popular augmentation used extensively in most self-supervised graph learning methods, edge dropout, treats all edges equally potentially degrading the representation quality of low-centrality nodes that rely on sparse but crucial connections. In real-world graphs, structural information like degree and centrality are often correlated with sensitive attributes as shown in [4] and [5]. Thus it is imperative to study augmentation mechanisms that account for structural information, not only for better but also for fairer representations. FairDrop [6] attempts to create a fair dropout mechanism using node labels (supervised) but also fails to account for structural properties.

We explore novel centrality-aware augmentation techniques tailored for self-supervised learning to address these gaps. Specifically, we aim to answer the following research question: How can data augmentations be improved to enhance representation learning in self-supervised graph neural networks? We propose: (a) Centrality-Aware Edge Dropout, which modulates dropout probabilities based on node centrality to ensure robust representations, and (b) Edge Addition Mechanisms, which dynamically introduce edges based on multi-hop connectivity, improving graph structure while maintaining computational efficiency.

Our methods are designed to remain computationally comparable and focus on augmentations that preserve or enhance graph connectivity based on node-level structural properties. Through extensive evaluations on node classification and clustering tasks, we demonstrate the efficacy of these methods in producing richer, more robust, and fairer graph representations.

II. Prerequisites

A. Centralities

Centrality is a measure used in graph theory and network analysis to quantify the importance or influence of nodes within a graph. It helps identify nodes critical for the structure, connectivity, or graph-level information flow. Different centrality measures focus on different aspects of a graph, such as how well-connected a node is, how pivotal it is for paths between other nodes, or how influential its neighbors are [7]. We focus on two computationally inexpensive measures of centrality.

1) Degree Centrality: The degree centrality of a node is its degree normalized to lie between 0 and 1 by dividing by the number of edges or nodes.

$$C_i^{deg} = \frac{deg_i}{|E|} \tag{1}$$

where C_i^{deg} is the degree centrality of the i^{th} node, deg_i its degree and $|E|$ the number of edges.

2) Eigenvector Centrality: Eigenvector centrality captures the influence of neighboring nodes under the assumption that nodes inherit some importance from their neighbors. Mathematically,

$$C_i^{eig} = \sum_{j \in M(i)} C_j^{eig} \tag{2}$$

Where C_i^{eig} is the eigen centrality of the i^{th} node and $M(i)$ is the neighborhood of the i^{th} node. This is very similar to PageRank [8].

B. Self-Supervised Graph Learning

Self-supervised graph learning enables learning meaningful representations of graph data without relying on labeled information. Instead, models generate supervisory signals by designing special tasks by identifying nodes that should/should not be similar. Two prominent methods that have emerged in this space are GRACE [3], a contrastive graph learning method and BGRL [2, 9], a method inspired by BYOL [10]. Both methods leverage graph augmentations to learn node or graph-level representations (most implementations use DropEdge).

C. Fairness

Most graphs in the real-world follow a power-law distribution [11], [12] and therefore learning better representations for low centrality nodes improves overall performance and also improves fairness. CAFIN [13] demonstrates group fairness in the self-supervised setting, splitting groups based on centrality. We adopt a similar method to evaluate the fairness of our methods in the absence of labels or any sensitive attributes to stay true to the self-supervised/unsupervised paradigm. The groups are divided based on centrality, where nodes with centrality greater than the median form one group and nodes with centrality less than the median form the other.

To evaluate fairness, we use the demographic parity ratio metric which is based on statistical parity and a common metric in the group fairness setting [14].

Dataset	Nodes	Edges	Features	Classes
Cora	2,708	10,556	1,433	7
CiteSeer	3,327	9,104	3,703	6
AMZN-P	7,650	238,162	745	8
AMZN-C	13,752	491,722	767	10

TABLE I: Quantitative description of the datasets used. The datasets are of varying sizes and domains, ensuring a comprehensive evaluation of our methods across diverse structural and contextual settings.

III. Methods

We introduce centrality-aware edge dropout and edge addition mechanisms to enhance graph augmentations for self-supervised learning. Our approach addresses key limitations of traditional uniform edge dropout through:

A. Centrality-Weighted Dropout

Adjusting dropout probabilities based on node centrality ensures underrepresented low-centrality nodes maintain their connectivity, preventing (stochastically) the removal of edges crucial to learning high-quality representations for less central nodes. To augment edge-dropout probabilities using centrality measures, we use two scoring schemes.

1) Low-Low Prioritization (LL): The first scheme prioritizes edges based on the total centrality of participating nodes.

$$s_{uv} = 2 - \frac{c_u + c_v}{2} \tag{3}$$

Note that the centralities are normalized to be between 0 and 1, and therefore $s_{uv} \in [1, 2]$. Here, the lower the centrality of both nodes, the higher the score.

2) Low-High Prioritization (LH): The first scheme prioritizes edges that connect nodes of differing centrality. This stems from our intuition that message-passing favors nodes with larger neighborhoods, and so learning node representations from other nodes with dense neighborhoods may be beneficial.

$$s_{uv} = 1 + |c_u - c_v| \tag{4}$$

Again, since the centralities are normalized, $s_{uv} \in [1, 2]$

Once the scores for all nodes are calculated, edges can be sampled from the multinomial distribution weighted by the normalized scores for both schemes.

B. Edge Addition

To complement centrality-aware edge dropout, we introduce two edge addition schemes designed to maintain graph connectivity. These methods connect two-hop neighbors by strategically adding edges, thereby addressing the potential loss of connectivity during dropout. We hypothesize that better connectivity during training leads to not only better representations but also faster convergence which is empirically validated by our experiments. Both schemes reinforce local structure which is leveraged heavily by message-passing.

1) Global Two-Hop Addition: We randomly sample from the precomputed set of all two-hop edges. During each augmentation iteration, we add $\frac{1}{1-p} * |E|$ edges, where p is the edge dropout probability. This ensures that the total number of edges in the graph remains approximately constant before and after dropout. By leveraging the global set of two-hop edges, this approach enhances overall graph connectivity, while attempting to preserve overall graph structure.

2) Sampling Two-Hop Addition: For each node, we dynamically sample its one-hop and two-hop neighbors, adding an edge to a selected two-hop neighbor. This approach is computationally efficient as it avoids the need for an exhaustive global two-hop edge computation, making it particularly suitable for large-scale graphs. Note however, that this comes

Method	Dataset	Baseline			Best Aug.			
		F1	Sim@5	DPR	Name (Centr.)	F1 (%Δ)	Sim@5 (%Δ)	DPR (%Δ)
BGRL	Cora	80.58	80.77	0.656	$CW + AE_s$ (eig)	81.51 (1.15)	81.34 (0.70)	0.671 (2.28)
	CiteSeer	65.13	64.01	0.694	$CW + AE_s$ (eig)	65.94 (1.24)	64.23 (0.34)	0.711 (2.44)
	Amzn-P	92.03	90.72	0.533	AE (deg)	92.16 (0.14)	90.86 (0.15)	0.530 (-0.56)
	Amzn-C	88.25	87.03	0.414	$CW_d + AE_s$ (eig)	88.67 (0.47)	87.21 (0.20)	0.420 (1.44)
GRACE	Cora	85.03	83.19	0.641	CW (deg)	85.11 (0.09)	83.30 (0.13)	0.641 (0.00)
	CiteSeer							
	Amzn-P	91.71	85.55	0.518	CW (eig)	91.86 (0.16)	84.93 (-0.72)	0.519 (0.19)
	Amzn-C	84.14	70.51	0.375	$CW + AE_s$ (eig)	85.01 (1.03)	71.57 (1.50)	0.399 (6.40)

TABLE II: For each method and dataset, we report the Baseline and best augmentation F1 score, Similarity score and demographic parity ratio (DPR). CW and AE refer to centrality weighted and add-edge respectively - note that we don't differentiate. CW_d indicates the second scoring mechanism (LH prioritization), and AE_s the two-hop neighbor sampling technique (opposed to global precomputation).

with a trade-off during train-time as the sampling step is slower than sampling from the global set of all two-hop edges.

IV. Results

A. Datasets

We evaluate our methods on four prominent datasets in the graph representation learning domain. Cora [15] and CiteSeer [16] are citation graphs, where nodes represent scientific publications, node features are bag-of-words (BoW) representations of the abstracts and edges represent citations. Amazon Photos and Computers [17] are co-purchase graphs where nodes represent product, node features are BoW representations of product reviews and edges represent products bought together.

B. Statistical Analysis

We perform a statistical analysis to understand the augmentations from a structural perspective. All metrics are averaged over 10 runs and the mean percentage change is reported. Numbers closer to 0 are more desirable, and as we can see from table III, WDE and AWDE perform better than their unweighted counterparts on most metrics. The degree assortativity changes significantly (decreases) as expected since weighting by centrality is more likely to preserve edges between heterogenous nodes. This assists message passing which relies on neighborhood information by ensuring information-rich neighbors for less central nodes. As seen in figure1, these results are also reflected qualitatively.

Method	Clustering	Density	Comp.	CC size	Assort.
DropEdge	-35.72	-14.76	25.89	-25.23	-0.84
CW	-30.26	-12.39	19.74	-21.42	-1.69
AE	-13.45	-6.46	14.98	-17.03	-9.73
CW+AE	-11.24	-4.53	12.30	-16.85	-8.21

TABLE III: Statistical analysis of graph structural properties using different augmentation schemes. DropEdge refers to conventional drop-edge, CW centrality-weighted drop-edge and AE drop-edge with added sampled edges.

C. Quantitative Results

We evaluate our proposed augmentation techniques using two prominent unsupervised learning frameworks: BGRL [2] and GRACE [3]. Results are averaged over 10 runs, and we assess performance using three key metrics: F1 score for node classification, node similarity search (Sim@5), and demographic parity ratio (DPR) [18] to measure group fairness. Table II summarizes the results, showcasing the best augmentation's performance and its percentage improvement over the baselines.

Our centrality-weighted edge dropout combined with edge addition (CW + AE) delivers the best results overall, with eigen-centrality emerging as the superior centrality score for weighting edges, indicating its effectiveness in capturing structural properties compared to degree centrality. While statistical significance varies across datasets and metrics, the overall positive impact of the augmentations remains evident.

The sum-based scoring scheme outperforms between the two scoring mechanisms, likely due to its emphasis on low centrality edges, which enrich message-passing algorithms. Additionally, the two-hop sampling-based edge addition proves more effective on average, highlighting its efficiency and contribution to graph augmentation.

All experiments for BGRL were conducted over 200 epochs, using a learning rate tuned between 10^{-3} and 10^{-5} depending on the dataset, and optimized with AdamW [19]. The encoder model consisted of two GCN layers, with embedding sizes ranging from 128 to 512, tailored to the dataset characteristics. For GRACE, experiments were performed with similar hyperparameters, employing the Adam optimizer instead. Specific hyperparameter details are omitted for the sake of brevity. All experiments were executed on an NVIDIA Tesla T4 GPU. All experimental results are averaged over 10 runs and random seeds to ensure statistical significance. We don't report runtime as the augmentations account for less than 10% of total runtime.

V. Conclusion and Future Work

We proposed centrality-aware graph augmentations, combining dropout and edge-addition schemes to enhance self-supervised learning. By leveraging intrinsic structural properties, our methods outperform DropEdge, particularly for low-centrality nodes by improving representation quality and fairness. To the best of our knowledge, this work is the first to explicitly integrate graph structure into self-supervised augmentation strategies, opening avenues for future exploration. We hope to see future work exploring methods that account for inherent graph structural properties explicitly instead of expecting message-passing mechanisms to sufficiently encode this information.

REFERENCES

[1] Jie Zhou et al. "Graph neural networks: A review of methods and applications". In: *AI Open* 1 (2020), pp. 57–81. ISSN: 2666-6510. DOI: https://doi.org/10.1016/j.aiopen.2021.01.001. URL: https://www.sciencedirect.com/science/article/pii/S2666651021000012.

[2] Shantanu Thakoor et al. "Large-Scale Representation Learning on Graphs via Bootstrapping". In: *International Conference on Learning Representations*. 2021. URL: https://api.semanticscholar.org/CorpusID:243756979.

[3] Yanqiao Zhu et al. "Deep Graph Contrastive Representation Learning". In: *ICML Workshop on Graph Representation Learning and Beyond*. 2020. URL: http://arxiv.org/abs/2006.04131.

[4] Zemin Liu, Trung-Kien Nguyen, and Yuan Fang. "On generalized degree fairness in graph neural networks". In: *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN: 978-1-57735-880-0. DOI: 10.1609/aaai.v37i4.25574. URL: https://doi.org/10.1609/aaai.v37i4.25574.

[5] Xianfeng Tang et al. "Investigating and Mitigating Degree-Related Biases in Graph Convoltuional Networks". In: *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. CIKM '20. Virtual Event, Ireland: Association for Computing Machinery, 2020, pp. 1435–1444. ISBN: 9781450368599. DOI: 10.1145/3340531.3411872. URL: https://doi.org/10.1145/3340531.3411872.

[6] Indro Spinelli et al. "FairDrop: Biased Edge Dropout for Enhancing Fairness in Graph Representation Learning". In: *IEEE Transactions on Artificial Intelligence* 3.3 (2022), pp. 344–354. DOI: 10.1109/TAI.2021.3133818.

[7] Linton C. Freeman. "Centrality in social networks conceptual clarification". In: *Social Networks* 1.3 (1978), pp. 215–239. ISSN: 0378-8733. DOI: https://doi.org/10.1016/0378-8733(78)90021-7. URL: https://www.sciencedirect.com/science/article/pii/0378873378900217.

[8] Lawrence Page et al. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report 1999-66. Previous number = SIDL-WP-1999-0120. Stanford InfoLab, Nov. 1999. URL: http://ilpubs.stanford.edu:8090/422/.

[9] Yunhui Liu et al. "Bootstrap Latents of Nodes and Neighbors for Graph Self-Supervised Learning". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer. 2024.

[10] Jean-Bastien Grill et al. "Bootstrap your own latent a new approach to self-supervised learning". In: *Proceedings of the 34th International Conference on Neural Information Processing Systems*. NIPS '20. Vancouver, BC, Canada: Curran Associates Inc., 2020. ISBN: 9781713829546.

[11] Michał Brzeziński. "Power laws in citation distributions: evidence from Scopus". In: *Scientometrics* 103 (2014), pp. 213–228. URL: https://api.semanticscholar.org/CorpusID:419502.

[12] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. "On power-law relationships of the Internet topology". In: *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*. SIGCOMM '99. Cambridge, Massachusetts, USA: Association for Computing Machinery, 1999, pp. 251–262. ISBN: 1581131356. DOI: 10.1145/316188.316229. URL: https://doi.org/10.1145/316188.316229.

[13] Arvindh Arun et al. "CAFIN: Centrality Aware Fairness Inducing IN-Processing for Unsupervised Representation Learning on Graphs". In: Sept. 2023. ISBN: 9781643684369. DOI: 10.3233/FAIA230259.

[14] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. MIT Press, 2023.

[15] Andrew McCallum et al. "Automating the Construction of Internet Portals with Machine Learning". In: *Information Retrieval* 3 (2000), pp. 127–163.

[16] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. "CiteSeer: an automatic citation indexing system". In: *Digital library*. 1998.

[17] Oleksandr Shchur et al. *Pitfalls of Graph Neural Network Evaluation*. 2018. DOI: 10.48550/ARXIV.1811.05868. URL: https://arxiv.org/abs/1811.05868.

[18] Alekh Agarwal et al. "A Reductions Approach to Fair Classification". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, Oct. 2018, pp. 60–69.

[19] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations*. 2019. URL: https://openreview.net/forum?id=Bkg6RiCqY7.