

# Meta-GCN: A Dynamically Weighted Loss Minimization Method for Dealing with the Data Imbalance in Graph Neural Networks

Mahdi Mohammadzadeh<sup>†</sup>, Arash Mozhdehi<sup>†</sup>, Yani Ioannou<sup>‡</sup>, Xin Wang<sup>†,\*</sup>

<sup>†</sup> Department of Geomatics Engineering, University of Calgary

<sup>‡</sup> Department of Electrical and Software Engineering, University of Calgary

## Abstract

Although many real-world applications, such as disease prediction, and fault detection suffer from class imbalance, most existing graph-based classification methods ignore the skewness of the distribution of classes; therefore, tend to be biased towards the majority class(es). Conventional methods typically tackle this problem through the assignment of weights to each one of the class samples based on a function of their loss, which can lead to over-fitting on outliers. In this paper, we propose a meta-learning algorithm, named Meta-GCN, for adaptively learning the example weights by simultaneously minimizing the unbiased meta-data set loss and optimizing the model weights through the use of a small unbiased meta-data set. Through experiments, we have shown that Meta-GCN outperforms state-of-the-art frameworks and other baselines in terms of accuracy, the area under the receiver operating characteristic (AUC-ROC) curve, and macro F1-Score for classification tasks on two different datasets.

**Keywords:** Graph neural network, Graph convolutional networks, Node classification, Imbalanced class label, Meta-learning

## 1. Introduction

Graph-based data structures are ubiquitously used for modeling the pair-wise relation between the entities in a variety of real-world applications, including social networks [1], citation networks [2], and protein-protein interactions [3]. Graph structures are effectively capable of describing the complex relationship between the objects, i.e. nodes, through edges. Besides, Graph-based representation is an effective method for feature dimensionality reduction [4, 5]. GNNs, as powerful tools for representational learning on graph-structured data, have attracted increasing attention in recent years. GNNs are used for effective deep representational learning to perform graph analysis for tasks such as node classification, link prediction, and clustering in Euclidean and non-Euclidean domains [6]. Among the proposed methods for learning representations on graphs, GCNs proposed by Kipf et al. [7] proved to be a simple and effective GNN model. This model is able to learn hidden representations comprising both node features and local graph structure while scaling linearly relative to the number of edges in the given graph. Most classification algorithms, in GNNs, tend to minimize the average loss over all training examples which produces reasonable outcomes for class-balanced datasets. However, various real-world classification tasks, such as disease prediction [8], fraud detection [9], and fault detection [10] manifest highly-skewed class distribution. In settings that exhibit such class imbalance, these methods favor the majority classes while neglecting the minority ones. In addition, over-smoothing, which is a general issue in the GNN, can be aggravated in the case of data imbalance, as minority class nodes' representations become similar to that of majority ones [11]. Hence, these classifiers cannot well-differentiate the boundary between the minority and majority classes. However, in applications like disease prediction, fault detection, and fraud detection, minorities are important, and classifying them correctly is crucial. Alternative to the existing methods, aiming to alleviate class imbalance for node classification on graph-structured data,

we proposed an algorithm-based method, called Meta-GCN, that uses a meta-learning algorithm that adaptively assigns weights to the training examples in a way that minimizes the aggregated loss of an unbiased example set sampled from a meta-data set.

Our contributions are threefold:

- (1) Proposing a general-purpose online re-weighting algorithm for semi-supervised node classification in graphs that learns weighted loss function which is parameterized by weights that are learned in a meta-learning manner through the use of a small unbiased meta-data set.
- (2) Proposing a novel graph-based sampling method from the designated portion of the dataset to construct the meta-data set.
- (3) Through our experiment, we demonstrated that Meta-GCN outperforms state-of-the-art frameworks and other baselines in terms of accuracy, the area under the receiver operating characteristic (AUC-ROC) curve, and macro F1-Score for classification tasks on two different datasets.

## 2. Related Work

Proposed methods for addressing class imbalance issues on both graph and non-graph data fall into data-level, algorithm-level, or hybrid methods. Over-sampling and under-sampling are among the data-level solutions. While over-sampling methods aim to balance the ratio of classes by having more examples from minority classes in training, under-sampling solutions remedy the disproportion by removing the majority class(es) instances. Over-sampling with replicating the examples is known to tend to overfit; hence, a synthetic minority over-sampling technique (SMOTE) [12] has been proposed to overcome this issue by generating synthetic minority examples through interpolating neighboring minority examples. To improve SMOTE, several variants of it have been proposed. However, over-sampling methods are error-prone by synthesizing examples close to the boundary. Therefore, under-sampling methods are often preferred [13], however, removing examples can lead to losing valuable instances required for discrimination, and as a result poor generalization. On the other hand, re-weighting methods, as algorithm-level solutions, aim to minimize a weighted loss on the training samples by assigning weights in a manner that pays more attention to minority examples. AdaBoost [14] is a re-weighting-based approach that creates an ensemble of classifiers and assigns higher weights to misclassified instances. In [15], authors proposed improvement to AdaBoost by a hybrid method that first combines it with over-sampling and then uses an optimization algorithm that further tunes the class-specific weights. [16] proposed an algorithm that re-weights the training examples online based on gradient direction using a small clean validation set. Alternatively, in [17], authors proposed a generic method for evaluating the semantic completeness of datasets. Since these methods assume that the data is i.i.d they are not applicable to graph-based representations. As GNN-based classifiers are newly emerged, few works have instigated solving class imbalance issues in this area. GraphSMOTE [18], a data-level approach for graph-structured data, has been proposed that generates the synthetic data in the embedding space. These methods proved to be error-prone. DR-GCN [19] is an algorithm-level approach that uses class-conditioned adversarial regularizers to overcome the imbalance in graphs. This approach, as a hybrid method with adversarial training, is shown to suffer from instability and is unable to scale to large graphs. Alternatively, Meta-GCN unlike most of the proposed approaches that deal with data imbalance is for semi-supervised node classification in graphs and we do not assume that the data is i.i.d. Unlike GraphSMOTE, our proposed method is an algorithm-level approach and does not generate synthetic data.

### 3. Method

In this section, we introduce the definitions, state the problem, and present our solution. Assume a feature matrix  $\mathcal{X} \in \mathbb{R}^{N \times F}$ , with an imbalanced distribution of labels, where  $N$  and  $F$  are respectively the numbers of examples in the dataset and the number of features assigned to them. Let  $A \in \{0, 1\}^{N \times N}$  be a binary adjacency matrix, showing the connectivity between the examples in the dataset. For given examples  $i$  and  $j$  the entry  $A_{i,j}$  is 1 iff the examples are connected and 0 otherwise. Given the feature matrix  $\mathcal{X}$  and the adjacency matrix  $A$ , we construct the undirected and unweighted graph  $\mathcal{G} = (V, E, \mathcal{X})$ , where  $V = \{1, \dots, N\}$  stands for the set of vertices while  $E$  is the edge set of the graph. Let the matrix  $\mathcal{Y} \in \mathbb{R}^N$  be the ground-truth class labels corresponding to each vertex of the graph  $\mathcal{G}$ .

Let's assume a graph  $\mathcal{G}^{meta} = (V^{meta}, E^{meta}, \mathcal{X}^{meta})$  be meta-graph constructed by an unbiased meta-data set  $\mathcal{X}^{meta}$  of size  $M$  with the adjacency matrix  $A^{meta}$ . The feature set  $\mathcal{X}^{meta}$  is constructed through unbiased sampling from a separate portion of the whole dataset apart from the training and validation set.  $V^{meta}$  is a set of vertices where each of its items corresponds to each entry of the set  $\mathcal{X}^{meta}$ . There is an edge  $e_{i,j}^{meta}$  in the edge set  $E^{meta}$  if and only if  $i, j \in V^{meta}$ . We represent the ground-truth label set for the meta-data set by  $\mathcal{Y}^{meta} \in \mathbb{R}^M$ . Given the adjacency matrix  $A$  and the feature matrix  $\mathcal{X}$ , we define a GCN model  $f_\theta(\mathcal{X}, A)$ , where  $\theta$  is the set of learnable model parameters. Let  $\hat{y} = f_\theta(\mathcal{X}, A)$  and  $\hat{y}^{meta} = f_\theta(\mathcal{X}^{meta}, A^{meta})$  be the predicted labels for the training set and meta-data set, respectively. Assume that  $\mathcal{L}(\hat{\mathcal{Y}}, \mathcal{Y})$  is the loss function for training data, and  $\mathcal{L}^{meta}(\hat{\mathcal{Y}}^{meta}, \mathcal{Y}^{meta})$  be that for the meta-data set. For the evaluation of this semi-supervised node classification problem, we use the cross-entropy loss function. We define an individual losses  $l_i(\hat{y}_i, y_i)$  for each training example  $i$  and  $l_j^{meta}(\hat{y}_j^{meta}, y_j^{meta})$  for each meta-data example  $j$ , where  $y_i \in \mathcal{Y}$ ,  $y_j^{meta} \in \mathcal{Y}^{meta}$ ,  $\hat{y}_i \in \hat{\mathcal{Y}}$ , and  $\hat{y}_j^{meta} \in \hat{\mathcal{Y}}^{meta}$ . We define our problem as minimizing a weighted loss parameterized with  $w = \{w_i | 1 \leq i \leq N\}$ , as

$$\theta^*(w) = \arg \min_{\theta} \sum_{i=1}^N w_i l_i(\theta). \quad (3.1)$$

At each step,  $\theta$  is updated as estimated weights are updated through:

$$\theta_{t+1}(w) = \theta_t(w) - \alpha \nabla \sum_{i=1}^N w_{i,t} l_i(f_\theta(\mathcal{X}, A), y_i), \quad (3.2)$$

$$\hat{\theta}_{t+1}(\gamma) = \theta_t(\gamma) - \alpha \nabla \sum_{i=1}^N \gamma_{i,t} l_i(f_\theta(\mathcal{X}, A), y_i), \quad (3.3)$$

$$\tilde{w}_{i,t} = \max(0, -\eta \frac{\partial}{\partial \gamma_{i,t}} \frac{1}{m} \sum_{j=1}^m l_j^{meta}(f_{\hat{\theta}}^{meta}(\mathcal{X}^{meta}, A^{meta}), y_j^{meta})), \quad (3.4)$$

$$w_{i,t} = \frac{\tilde{w}_{i,t}}{(\sum_{j=1}^M \tilde{w}_{j,t}) + \delta(\sum_{j=1}^M \tilde{w}_{j,t})}, \quad (3.5)$$

where  $\delta(z)$  is 1 if  $z = 0$  and 0 otherwise.  $\eta$  is the meta-learning rate.  $\gamma$  is the weight perturbing parameter. For the GCN model, we define the normalized adjacency matrix as  $\tilde{A} = A + I_N$ . A degree matrix  $\tilde{D}_{i,i} = \sum_{j=1}^N \tilde{A}_{i,j}$  can be defined for each node  $i$ . We also calculate the renormalized graph Laplacian  $\hat{A} = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$ . Considering an n-layer graph-based neural network GCN model for node classification, we compute the output for the layer  $l$  as

$$Z^l = \sigma(\hat{A} Z^{l-1} \theta^l), \quad (3.6)$$

where  $\sigma(\cdot)$  and  $\theta^l$  are respectively corresponding to the layer’s activation function and weights. Finally, we compute the neural networks classifier’s results through  $f_{\theta}(\mathcal{X}, A) = \text{Simoid}(Z^n)$ . Figure 1 depicts the overall training process of Meta-GCN.

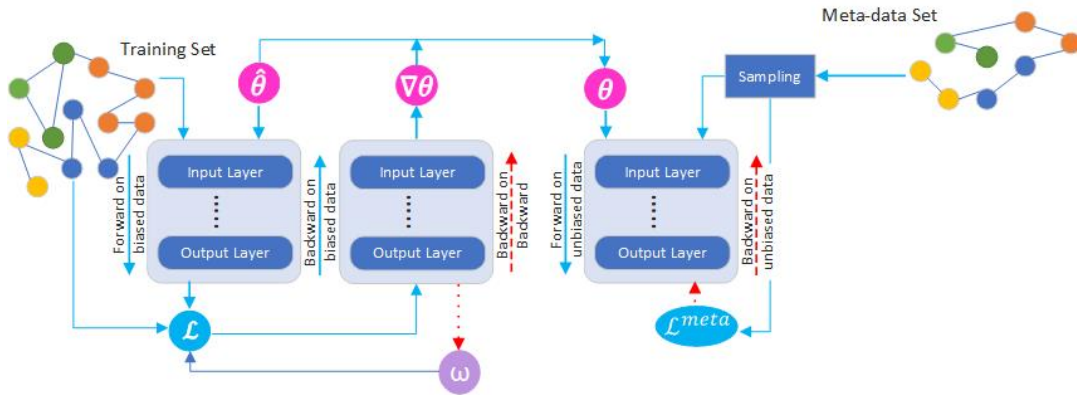


Figure 1. The overall procedure of Meta-GCN’s training in a meta-learning manner using a small unbiased meta-data set.

#### 4. Experiments

To compare the node classification performance of our method with that of the baselines, we ran experiments on two medical datasets, namely Haberman and Diabetes using Meta-GCN and the five other baselines. For the SMOTE method, we used the scikit-learn library in Python [20]. For training, validation, testing, and meta-set generation we respectively used 60%, 10%, 20%, and 10% of the data. For the GraphSMOTE and GCN, we used the same hyperparameters in the publications and the evaluations are done by the source codes shared by the authors. For both oversampling methods, SMOTE and GraphSMOTE, we used an over-sampling scale of 0.8, as it is the best value. We adopted two layers GCN model with 32 hidden units. For the activation function of the layers, we used Rectified Linear Unit (ReLU). We summarized the resources for the three metrics, accuracy, macro F1, and AUC-ROC, in Table 1. It is perceivable that our method, Meta-GCN, outperformed all five baselines on all three metrics. Meta-GCN’s higher macro F1-score on both datasets suggests that our approach has better performance for minorities as well as the majorities. MLP performed worse than other graph-based baselines which are methods on both datasets for all three metrics. Since SMOTE oversampling method simply performs the interpolation on neighboring nodes without considering the graph structure, its performance is worse than the GCN without oversampling.

For the Diabetes dataset, it is evident that utilizing the graph structure led to substantial improvement. Since GraphSMOTE constructs an embedding space for interpolation and considers the graph structure, its performance is significantly better than SMOTE on this dataset, where much information can be gained from the graph information. However, in terms of accuracy, GraphSMOTE does not perform significantly better than GCN-Weighted. It performs even worse compared to the standard GCN in terms of macro F1 and AUC-ROC. For Haberman, nodes have less number of features compared to Diabetes. By comparing the results between GCN and GCN-Weighted on macro F1, it can be concluded that the weighting effect is more considerable in this dataset. Judging by insignificant improvement on three metrics it can be inferred that gaining from the graph information in this dataset is limited. Limited information gain from the graph structure can also explain why the

Methods	Diabetes			Haberman		
	Accuracy	Macro F1	AUC-ROC	Accuracy	Macro F1	AUC-ROC
MLP	0.58 ± 0.07	0.43 ± 0.07	0.55 ± 0.07	0.71 ± 0.03	0.46 ± 0.04	0.45 ± 0.17
GCN	0.70 ± 0.10	0.65 ± 0.09	0.72 ± 0.07	0.74 ± 0.04	0.47 ± 0.05	0.55 ± 0.06
GCN-Weighted	0.71 ± 0.05	0.64 ± 0.24	0.66 ± 0.19	0.71 ± 0.05	0.57 ± 0.05	0.46 ± 0.09
SMOTE	0.65	0.57	0.58	0.71	0.50	0.52
GraphSMOTE	0.72 ± 0.14	0.65 ± 0.09	0.70 ± 0.27	0.74 ± 0.24	0.47 ± 0.05	0.54 ± 0.12
Meta-GCN (ours)	<b>0.74 ± 0.17</b>	<b>0.70 ± 0.12</b>	<b>0.75 ± 0.15</b>	<b>0.76 ± 0.17</b>	<b>0.65 ± 0.07</b>	<b>0.62 ± 0.09</b>

Table 1. Comparison of different methods for imbalanced node classification.

improvement of using GraphSMOTE compared to SMOTE is insignificant. However, in this case, Meta-GCN through the use of the meta-set was able to significantly improve the performance on all three different metrics. Considerable improvement of macro F1 by Meta-GCN demonstrates its superior performance in classifying both minority classes and majority classes. Judging by the improvement of macro F1-score on Diabetes compared to Haberman, which has a higher imbalance ratio, the gain from weighting is more significant.

Based on the results of this experiment, it can be concluded that: 1) For a better comparison and understanding of models none of the metrics are sufficient by themselves. 2) Depending on the dataset the performance can be increased by gaining information from graph structure. 3) Higher performance by the weighting method can be gained with datasets with a higher imbalance ratio. 4) Using the Meta-GCN method significantly improved the classification performance in all three metrics, suggesting that not only improves the overall accuracy of the model in the classification of the examples but also improves the discrimination power of the model for classifying both majorities and minorities.

## 5. Conclusion and Future Works

In this paper, we proposed a meta-learning-based method, name Meta-GCN for dealing with label class imbalance for semi-supervised node classification. The proposed method leverages a small unbiased meta-data set to adaptively learns the example weights through minimization of the meta-data set loss simultaneous to optimizing the model weights. This approach is general purpose and applicable to any graph-structured dataset that suffers from class imbalance. Compared to the other traditional re-weighting methods, Meta-GCN is an end-to-end method and does not require any manual weight setting and extra hyperparameter searching. Our empirical results demonstrate the superiority of Meta-GCN compared to the representative and state-of-the-art approaches in terms of accuracy, macro F1, and AUC-ROC. The comparison of experimental results shows that our model is effective in semi-supervised node classification for graph-structured datasets with the class imbalanced distribution. There are several avenues for further investigation. Firstly, in this paper, we proposed a graph-based sampling method that chooses examples for each class with equal probability without considering the graph structure. Therefore, we intend to improve the sampling method for gaining better performance. Second, the proposed method is for node classification and we intend to extend our approach to other applications such as edge prediction or regression tasks.

## References

- [1] G. Tong. “StratLearner: learning a strategy for misinformation prevention in social networks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 15546–15555.
- [2] M. R. Khan and J. E. Blumenstock. “Multi-gen: Graph convolutional networks for multi-view networks, with applications to global poverty”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 606–613.

- [3] N. Zaki, H. Singh, and E. A. Mohamed. “Identifying protein complexes in protein-protein interaction data using graph convolutional network”. In: *IEEE Access* 9 (2021), pp. 123717–123726.
- [4] H. Zhang and M. Gabbouj. “Feature Dimensionality Reduction with Graph Embedding and Generalized Hamming Distance”. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 1083–1087.
- [5] S. Yan, D. Xu, B. Zhang, H.-j. Zhang, Q. Yang, and S. Lin. “Graph Embedding and Extensions: A General Framework for Dimensionality Reduction”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.1 (2007), pp. 40–51. DOI: [10.1109/TPAMI.2007.250598](https://doi.org/10.1109/TPAMI.2007.250598).
- [6] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. “Graph neural networks: A review of methods and applications”. In: *AI Open* 1 (2020), pp. 57–81.
- [7] M. Welling and T. N. Kipf. “Semi-supervised classification with graph convolutional networks”. In: *J. International Conference on Learning Representations (ICLR 2017)*. 2016.
- [8] A. Kazi, S. Shekarforoush, S Arvind Krishna, H. Burwinkel, G. Vivar, K. Kortüm, S.-A. Ahmadi, S. Albarqouni, and N. Navab. “InceptionGCN: receptive field aware graph convolutional network for disease prediction”. In: *International Conference on Information Processing in Medical Imaging*. Springer. 2019, pp. 73–85.
- [9] J. Jiang, J. Chen, T. Gu, K.-K. R. Choo, C. Liu, M. Yu, W. Huang, and P. Mohapatra. “Anomaly detection with graph convolutional networks for insider threat and fraud detection”. In: *MILCOM 2019-2019 IEEE Military Communications Conference (MILCOM)*. IEEE. 2019, pp. 109–114.
- [10] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He. “Fault location in power distribution systems via deep graph convolutional networks”. In: *IEEE Journal on Selected Areas in Communications* 38.1 (2019), pp. 119–131.
- [11] A. Kammoun, R. Slama, H. Tabia, T. Ouni, and M. Abid. “Generative Adversarial Networks for Face Generation: A Survey”. In: *ACM Comput. Surv.* (2022). Just Accepted. ISSN: 0360-0300. DOI: [10.1145/1122445.1122456](https://doi.org/10.1145/1122445.1122456). URL: <https://doi.org/10.1145/1122445.1122456>.
- [12] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.
- [13] C. Huang, Y. Li, C. C. Loy, and X. Tang. “Deep imbalanced learning for face recognition and attribute prediction”. In: *IEEE transactions on pattern analysis and machine intelligence* 42.11 (2019), pp. 2781–2794.
- [14] Y. Freund, R. E. Schapire, et al. “Experiments with a new boosting algorithm”. In: *icml*. Vol. 96. Citeseer. 1996, pp. 148–156.
- [15] B. Yuan and X. Ma. “Sampling + reweighting: Boosting the performance of AdaBoost on imbalanced datasets”. In: *The 2012 International Joint Conference on Neural Networks (IJCNN)*. 2012, pp. 1–6. DOI: [10.1109/IJCNN.2012.6252738](https://doi.org/10.1109/IJCNN.2012.6252738).
- [16] M. Ren, W. Zeng, B. Yang, and R. Urtasun. “Learning to reweight examples for robust deep learning”. In: *International conference on machine learning*. PMLR. 2018, pp. 4334–4343.
- [17] H. Barzamini and M. Rahimi. “CADE: The Missing Benchmark in Evaluating Dataset Requirements of AI-enabled Software”. In: *2022 IEEE 30th International Requirements Engineering Conference (RE)*. 2022, pp. 64–76. DOI: [10.1109/RE54965.2022.00013](https://doi.org/10.1109/RE54965.2022.00013).
- [18] T. Zhao, X. Zhang, and S. Wang. “Graphsmote: Imbalanced node classification on graphs with graph neural networks”. In: *Proceedings of the 14th ACM international conference on web search and data mining*. 2021, pp. 833–841.
- [19] R. Chen, G. Li, and C. Dai. “DRGCN: Dual Residual Graph Convolutional Network for Hyperspectral Image Classification”. In: *IEEE Geoscience and Remote Sensing Letters* 19 (2022), pp. 1–5. DOI: [10.1109/LGRS.2022.3171536](https://doi.org/10.1109/LGRS.2022.3171536).
- [20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12 (2011), pp. 2825–2830.