Graph-based Semi-Supervised Learning by Strengthening Local Label Consistency

Chen Li
BDBC, Beihang University
Beijing, China
lichen@act.buaa.edu.cn

Xutan Peng
The University of Sheffield
Sheffield, United Kingdom
x.peng@shef.ac.uk

Hao Peng
Beihang University
Beijing, China
penghao@buaa.edu.cn

Jia Wu
Macquarie University
NSW, Australia
jia.wu@mq.edu.au

Lihong Wang
CNCERT
Beijing, China
wlh@isc.org.cn

Philip S. Yu
University of Illinois at Chicago
Chicago, United States
psyu@uic.edu

Jianxin Li
BDBC, Beihang University
Beijing, China
lijx@act.buaa.edu.cn

Lichao Sun
Lehigh University
Lehigh, United States
james.lichao.sun@gmail.com

ABSTRACT

Graph-based algorithms have drawn much attention thanks to their impressive success in semi-supervised setups. For better model performance, previous studies have learned to transform the topology of the input graph. However, these works only focus on optimizing the original nodes and edges, leaving the direction of augmenting existing data insufficiently explored. In this paper, we propose a novel heuristic pre-processing technique, namely *Local Label Consistency Strengthening* (L^2CS), which automatically expands new nodes and edges to refine the label consistency within a dense subgraph. Our framework can effectively benefit downstream models by substantially enlarging the original training set with high-quality generated labeled data and refining the original graph topology. To justify the generality and practicality of L^2CS, we couple it with the popular graph convolution network and graph attention network to perform extensive evaluations on three standard datasets. In all setups tested, our method boosts the average accuracy by a large margin of 4.7% and consistently outperforms the state-of-the-art.

CCS CONCEPTS

• **Mathematics of computing → Mathematics of computing**; • **Computing methodologies → Supervised learning by classification**.

KEYWORDS

semi-supervised learning, topology enhanced transformation, node classification

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '21, November 1–5, 2021, Virtual Event, Australia.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482114

ACM Reference Format:
Chen Li, Xutan Peng, Hao Peng, Jia Wu, Lihong Wang, Philip S. Yu, Jianxin Li, and Lichao Sun. 2021. Graph-based Semi-Supervised Learning by Strengthening Local Label Consistency. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482114

1 INTRODUCTION

Numerous real-world data can be represented as graphs, *e.g.,* social networks [15, 18], citation networks [12, 19, 29], protein network [7], and knowledge graphs [13, 22]. In many cases, large-scale annotated data is expensive to obtain. The so-called graph-based Semi-Supervised Learning (SSL), which holds promise to bootstrap applications even with limited supervision, has therefore attracted increasing research interest. Different from the traditional sample independent scenarios, graph-based SSL models tend to mine the *structural* relational information to boost effectiveness. Correspondingly, some existing methods have shown that optimizing the input graph structure is beneficial [23, 28]. However, generative ways to refine the topology and the information propagation along the edges are unexplored. In this paper, we propose to mitigate these limitations by a new strategy which strengthens local label consistency, thereby improving the model performance.

Specifically, earlier works develop the classical regularization methods, which achieve SSL by smoothing features or model predictions over local neighborhoods using explicit regularization schemes [2, 17]. Although this direction has been well studied, a later thread of algorithms, namely graph convolution networks, has demonstrated state-of-the-art performance and drawn much attention [3, 11, 25]. By utilizing flexible aggregation strategies, these models selectively fuse the local features of a graph into the hidden representations of its target nodes. To further perform downstream tasks, the hidden layers are coupled with specific task layers [11, 25]. One common characteristic of these two lines of models is that they both adopt the presence of smoothness in the graph structure as a basic assumption. Recently, to better exploit

annotated resources, researchers propose to modify the topology of the input graph [1, 20, 28]. However, to our knowledge, all such methods are limited within handling the *existing* graph topology.

Inspired by the discussion on the relationship between information similarity within the same community and information propagation [14, 28], we explore a novel research direction for the first time, namely *Local Label Consistency Strengthening* ($\mathtt{L}^2\mathtt{CS}$). Through generating some key high-quality nodes and edges, $\mathtt{L}^2\mathtt{CS}$ improves the label consistency of the set of nodes that are structurally closely related to guide more properly feature smoothing on the graph. Specifically, $\mathtt{L}^2\mathtt{CS}$ first identifies dense subgraphs through overlapping clustering algorithms. For each subgraph, by jointly considering node attributes and edge links, it generates a new node with high-quality attributes and a corresponding low-entropy label predicted via the local labeled original nodes in the given data. Lastly, $\mathtt{L}^2\mathtt{CS}$ connects generated nodes with their corresponding original nodes to facilitate information propagation, yielding an updated graph. In addition, we find that this simple pre-processing technique also enhances the class separability of node attributes. Therefore, by producing a high-quality augmented training set, $\mathtt{L}^2\mathtt{CS}$ generally further renders the performance of subsequent SSL models stronger.

To validate the practical usefulness of $\mathtt{L}^2\mathtt{CS}$, we perform extensive evaluations on the SSL benchmark with three standard datasets. Coupled with two popular models (*i.e.,* Graph Convolutional Network (GCN) and Graph Attention Network (GAT)), our method significantly improves the performance of the base algorithms and consistently outperforms baseline models, including the state-of-the-art. In summary, the contribution of this work is three-fold. (1) This is the first study which expands the original graph topology with both new *nodes* and *edges* to improve training. (2) By simulating the generation process of graph signals, theoretically and empirically we show that the data augmented by our method has enhanced quality. (3) In the extensive evaluations of SSL on three standard datasets, our method significantly boosts two popular base algorithms and substantially sets new state-of-the-art scores.

2 OUR APPROACH

By implying the widely-adopted assumption of existing graph-based SSL models, *i.e.,* features exhibit smoothness along the graph edges [14, 24, 26, 27], Yang et al. [28] propose a criterion to assess training samples and given graph, which is highly correlated to the subsequent modeling performance: **Criterion** $\mathscr{C}$: *The higher the label consistency in the dense subgraph, the better the propagation of feature along the edges.* This criterion, which is intuitively evident given the observed presence of graph node communities, has been empirically validated by the experiments of [14, 28]. Therefore, one of the ultimate objectives of our algorithm boils to: compared with the original graph G, the augmented data should satisfy Criterion $\mathscr{C}$ equally well, or even better.

As illustrated in Figure 1, to achieve this goal, in the first step, we learn to partition the original graph into different dense subgraphs (*i.e.,* clusters) based on graph structure. Next, for each cluster, we automatically generate a new node, whose attributes can be regarded as the multiple sampling results on attributes of existing original nodes in the same dense subgraph. Considering the fact that multiple sampling of distribution can stabilize its posterior probability (*i.e., Law of Large Numbers*), compared with original nodes, generated nodes naturally have better class separability in terms of attributes. Consequently, while labels of some generated nodes can be directly confirmed from the original nodes, the labels of remaining generated nodes can also be roughly determined even with a very simple classifier. Lastly, since most edges between generated nodes and their corresponding original nodes can maintain label consistency, the updated graph G' is deemed to have high quality (cf. Criterion $\mathscr{C}$) and has a much larger volume than the original G, thus effectively benefiting subsequent algorithms.

2.1 Substructure-based Overlapping Clustering

In real-world scenarios, it is quite common that a node belongs to multiple communities (dense subgraphs), *e.g.,* an author publishes a highly impacted paper on machine learning theory, which may get cited by different communities such as computer vision and natural language processing. Therefore, we identify subgraphs through overlapping clustering, which is a well-studied topic in community detection. Its workflow of overlapping clustering is two-phased. First, it learns to cluster nodes within local regions. To handle nodes belonging to multiple neighborhoods, it will create *personas* for each cluster. Next, it performs a standard global clustering and re-associates the personas whose sources are the same. We denote the resulting set of dense subgraphs as S'.

2.2 Refinement based on Label Consistency

From the generative perspective of graph [16], original nodes in a cluster $C_m \in S'$ become $|C_m|$ samples from the golden distribution of attributes. Similarly, the attributes of the corresponding generated node $x_i^c \in X^c$ can be generated through multiple samplings, *i.e.,* aggregating the attributes of original nodes as

$$x_i^c = \frac{\sum_{v_j \in C_m} x_{v_j}}{|C_m|}, x_{v_j} \in X, \tag{1}$$

where X^c denotes the attribute matrix of all generated nodes, and Y^c is for the label matrix, likewise. To determine the values of Y^c, we proceed with our discussions case by case.

"Winner takes all". We assume that the nodes in the same dense subgraph share similar features/label. Hence, for each generated node, if its original nodes are from the training set, *i.e.,* have annotated labels, then it can straightforward inherit the dominating (*i.e.,* most numerous) label, such that

$$y_i^c = [0, \cdots, \underbrace{1}_{j}, \cdots, 0], \quad n_j = \max(n_0, \cdots, n_k), \tag{2}$$

where n_p $(p = 0, \cdots, k)$ is the occurrence of the p-th kind of label observed in C_m. Y^c_{obs} denotes the label matrix of this category of generated nodes. In practice, we find applying the additional constraint of $n_j \geq 2$ can guarantee the reliability of assigned labels.

Data Augmentation. However, only a small portion (*e.g.,* roughly 1/3 in the Cora dataset) of generated nodes can be directly labeled using Eq. (2). For other generated nodes, we find that the label propagation paradigm leads to unsatisfactory results, mainly due to two reasons: on the one hand, real-world graphs (*e.g.,* Cora, Citeseer, and Pubmed) are often not fully connected, so many nodes cannot receive the broadcast of training labels; on the other hand, long-range dependencies may lead to under-smoothing. As theoretically explained by [9], the stability of Eq. (1) will get strengthened if

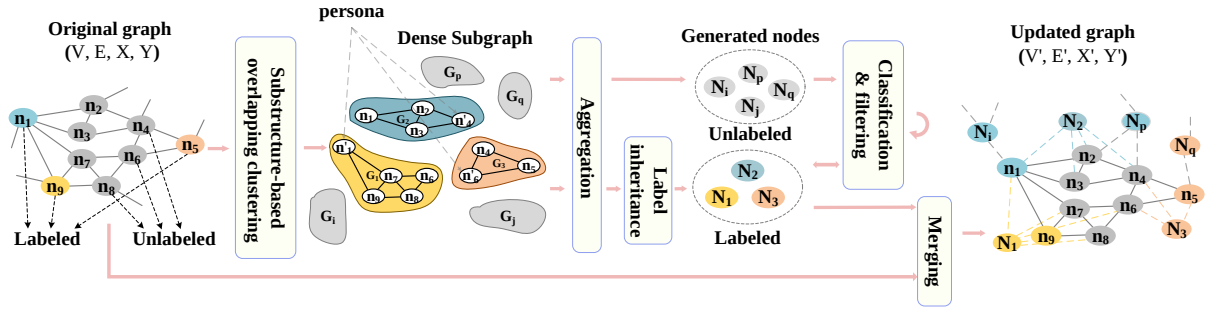

Figure 1: Overview of the L^{2}CS pipeline. First, L^{2}CS learns to cluster nodes within local regions, and aggregates the internal node features of the dense subgraph (*i.e.,* cluster) to generate new nodes. Next, some original nodes in the training set can assign their labels to correspond generated nodes, and then L^{2}CS predicts more unlabeled generated nodes through self-learning. Last, L^{2}CS connects the generated nodes with the original nodes in the subgraph and augments the training set.

the number of samplings increases, and X^c will thereupon exhibit stronger class separability (of attributes). This property inspires us to attempt a novel self-learning scheme to handle unlabeled generated nodes. To begin with, we learn a simple binary classifier based on the already-labeled generated nodes (*i.e.,* the *union* of generated nodes labeled in the last paragraph and those labeled in the previous iterations). Next, after predicting on-the-fly labels using this classifier, we filter out the generated nodes whose labels are assigned with probability lower than a given "labeling threshold". The above two steps are iteratively performed to produce enough generated nodes with high-quality predicted labels.

2.3 Downstream Coupling

By merging high-quality labeled generated nodes into G (*i.e.,* inserting generated nodes and connecting them with original nodes in the corresponding cluster.), we obtain the updated graph $G' = \{V', E', X', Y'\}$ as shown in Figure 1. Apart from the significantly enlarged volume, another outstanding advantage of G' is that, for each dense graph, as generated node becomes the common neighbor of all its original nodes (*i.e.,* they are linked with new edges), the maximum distance between any two nodes in the same dense subgraph becomes 2, i.e., long-range dependencies get generally shortened. Finally, G' can be fed into subsequent graph-based SSL models, with the single aggregation operation for v_i at depth l being represented as $h_i^l = \sigma(\sum_{j \in \mathcal{V}_i \cup \{n_i\}} \alpha_{i,j} W h_j^{l-1})$ (h_i^l is the hidden representation of v_i at the l-th layer, $\mathcal{V}_i$ is the neighbor set of v_i, W is a learnable linear transformation matrix, $\sigma(\cdot)$ is an element-wise nonlinear activation function, and $\alpha_{i,j}$ is the evaluation parameter set in feature aggregation, *e.g.,* the attention function of GAT).

3 EXPERIMENTS

Following previous studies [11, 25, 29], we demonstrate the effectiveness of L^{2}CS on the widely-adopted semi-supervised node classification benchmark.

3.1 Experimental Setup

Datasets. Our evaluation is based on three datasets (*i.e.,* Cora, Citeseer, and Pubmed [29]) which are the *de facto* standards for assessing graph-based SSL algorithms. They are all sampled via citation networks, where nodes are for research publications and edges for the citation. In Cora and Citeseer, node attributes are represented as bag-of-words, while Pubmed uses TF-IDF weights. For a fair comparison, we adopt the same splits as Yang et al. [29], i.e., 20 training samples per class, 500/1000 verification/test samples.

Baselines. To justify the generality of L^{2}CS, we respectively integrate it with GCN [11] and GAT [25] as they are the two most popular graph-based SSL methods in the GNN community. As shown in Table 1, we select three recently-published approaches as our reference baselines [10, 27, 30]. In addition, we include six variants of GCN [3, 4, 6, 16, 27, 28, 32], and three variants of GAT [6, 16, 24, 31].

Parameters. In practice, we find that our proposed L^{2}CS is robust towards configuration variations. Therefore, we exploit the default setting for parameters without much fine-tuning. To be specific, for overlapping clustering we select the Ego-Splitting algorithm [5, 21]), with the resolution set at 1.0. In the high-level information diffusion step, we utilize the simple GBDT [8] as our classifier, with a learning rate at 0.25, max depth at 3, and other parameters selected as default. During all experiments, we terminate the training when the verification accuracy no longer increases for 2K iterations. Test scores based on the models and hyper-parameters with the best verification performance are reported, and all the hyper-parameters used are listed (the number of iterative diffusions and the labeling threshold is 10 and 0.99, respectively).

3.2 Results Analysis

Table 1 reports the results of our baseline methods, base algorithms (GCN and GAT), and L^{2}CS-enhanced models (L^{2}CS-GCN and L^{2}CS-GAT). To reduce randomness, we run each model for 10 independent trials and calculate the average score and standard deviation. On all the three datasets, L^{2}CS-GAT consistently sets new state-of-the-art performance, with margins of 0.6% to 3.3% compared with the best baselines which are not coupled with L^{2}CS. Before being stacked with L^{2}CS, the base GCN is 1.2% inferior to GAT on average; after the data augmentation, L^{2}CS-GCN still falls behind L^{2}CS-GAT. However, compared with other baseline methods, L^{2}CS-GCN ranks second on Citeseer and Pubmed and third on Cora, exhibiting strong competitiveness. Please note that neither GCN nor GAT achieves outstanding accuracies compared with their strong counterparts: more concretely, even the original GAT cannot rank within the top three (with L^{2}CS-enhanced models excluded) on any dataset. This fact emphasizes the substantial effectiveness of L^{2}CS.

Table 1: Accuracy (%) of the classification benchmark. The highest performance per dataset is highlighted in bold.

	Method	Cora	Citeseer	Pubmed
Reference	SIG-VAE [10]	79.7	70.4	79.3
	CurvGN-n [30]	82.7±0.7	72.1±0.6	79.2±0.5
	GIL [27]	86.2	74.1	83.1
GCN-based	Chebyshev [3]	81.2	69.8	74.4
	TAGCN [4]	83.3	72.5	79.0
	TO-GCN [28]	83.1	72.7	79.5
	DGCN [32]	83.5	72.6	80.0
	ConfGCN [27]	82.0±0.3	72.7±0.7	79.5±0.5
	LSM_GCN [16]	82.5±0.2	74.4±0.3	77.9±0.4
	GRAND_GCN [6]	84.5±0.3	74.2±0.3	80.0±0.3
	GCN [11]	81.5	70.3	79.0
	L^2CS-GCN (Ours)	85.6±0.4	75.7±0.3	83.7±0.3
GAT-based	LSM_GAT [16]	82.9±0.3	73.1±0.5	77.6±0.7
	GAT_{128}+GAM [24]*	85.0	73.6	-
	ADSF-RWR [27]	85.4±0.3	74.3±0.4	81.2±0.3
	GRAND_GAT [6]	84.3±0.4	73.2±0.4	79.2±0.6
	GAT [25]	83.0±0.7	72.5±0.7	79.0±0.3
	L^2CS-GAT (Ours)	**87.6±0.5**	**76.7±0.4**	**84.3±0.3**

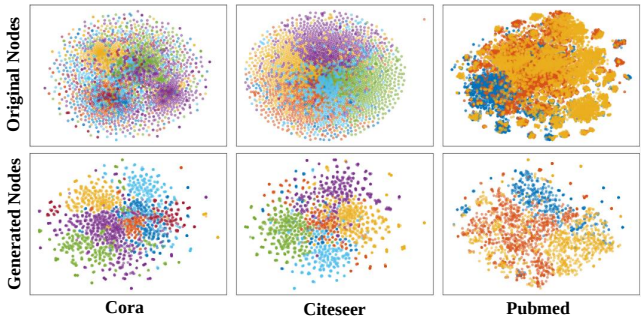

Figure 2: The visualization results of the attributes of the original nodes and the generated nodes after the dimension reduction by using t-sne.

When calculating the specific accuracy enhancement brought by L^2CS, we witness very significant 4.1% to 5.4% and 4.2% to 4.7% increases for GCN and GAT, respectively. Put these increments in the context: among all the other GCN-based approaches, the range of performance gain over GCN is -4.6% to 4.1%; among all the other GAT-based ones, it is -1.4% to 2.4% over GAT.

3.3 Ablation Analysis

In this section, we conduct some ablation analysis on the node attributes of the generated nodes, the label consistency within the updated subgraph, and the performance of each component.

To begin with, in Figure 2, we visualize original nodes and generated nodes with attributes as axes. While the former seem more crisscross, the latter exhibit clearer "community borders", demonstrating that generated nodes have better class separability.

Next, to assess the label consistency within a dense subgraph, we design a straightforward experiment, which counts the percentage of dominating labels in each cluster partitioned by L^2CS. We evaluate the original G and the updated G' together and plot the occurrence densities of dominating label proportion in Figure 3. On all datasets tested, compared with the curves in G, those in G' have

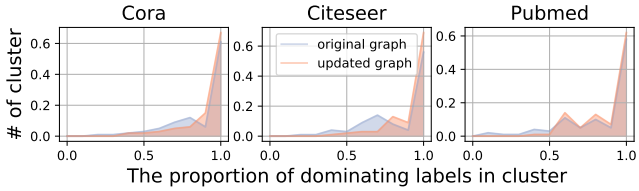

Figure 3: Distribution of per cluster dominating label proportion. X-axes denote the proportion of nodes with the dominating label in each cluster. Areas closer to the right indicates higher label consistency in the cluster.

Table 2: The results of ablation experiments.

Method	Clustering	Augmentation	Cora	Citeseer	Pubmed
L^2CS-GCN	Overlapping	-	83.1±0.4	72.9±0.4	82.5±0.5
	Overlapping	✓	85.6±0.4	75.7±0.3	83.7±0.3
	Non-overlapping	-	82.8±0.4	71.5±0.3	80.9±0.4
	Non-overlapping	✓	84.7±0.5	75.1±0.3	83.3±0.4
L^2CS-GAT	Overlapping	-	84.2±0.5	73.9±0.4	82.9±0.4
	Overlapping	✓	87.6±0.5	76.7±0.4	84.3±0.3
	Non-overlapping	-	82.8±0.4	73.1±0.3	82.3±0.4
	Non-overlapping	✓	85.9±0.5	75.8±0.3	84.1±0.4

higher peaks and are more right-gathered, implying that a larger proportion of nodes with subgraphs share the same labels.

Finally, we remove the components in L^2CS and perform ablation experiments. The results are shown in Table 2. We find that the effect of overlapping clustering components can obviously achieve better results than non-overlapping ones. We think this is mainly due to the existence of mass fuzzy nodes in the graph, whose class separability of their attributes is weak (as shown in Figure 2). If such nodes are forced to belong to only a certain cluster, it is easy to introduce noise in the cluster or cause missing information in other clusters. In addition, as the augmented nodes exhibit higher quality and the added edges improve local label consistency, our data augmentation leads to huger performance gains.

4 CONCLUSION

In this paper, we propose a simple yet effective L^2CS framework, which boosts the performance of graph-based SSL models by augmenting training resources. Aiming to strengthen the label similarity within dense subgraphs, L^2CS generates high-quality nodes, which also exhibit refined class separability of attributes. Results of extensive evaluations indicate that this generic pre-processing technique can dramatically enhance the base algorithms and further outperform state-of-the-art baselines. Followup experiments and analyses present more insights regarding the superiority of L^2CS. In the future, we will test L^2CS in more setups, as well as explore other novel graph augmentation strategies.

ACKNOWLEDGMENTS

This work was supported by the Natural Science Foundation of China program (No. U20B2053, 62002007 and 61772151), S&T Program of Hebei through grant 20310101D, and SKLSDE-2020ZX-12. Philip S. Yu is supported by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

REFERENCES

[1] Sami Abu-El-Haija, Bryan Perozzi, Amol Kapoor, Nazanin Alipourfard, Kristina Lerman, Hrayr Harutyunyan, Greg Ver Steeg, and Aram Galstyan. 2019. MixHop: Higher-Order Graph Convolutional Architectures via Sparsified Neighborhood Mixing. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019 (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 21–29.

[2] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *J. Mach. Learn. Res.* 7 (2006), 2399–2434.

[3] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016*, Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (Eds.). Curran Associates Inc., Barcelona, Spain, 3837–3845.

[4] Jian Du, Shanghang Zhang, Guanhang Wu, José M. F. Moura, and Soummya Kar. 2017. Topology adaptive graph convolutional networks. *CoRR* abs/1710.10370 (2017).

[5] Alessandro Epasto, Silvio Lattanzi, and Renato Paes Leme. 2017. Ego-Splitting Framework: from Non-Overlapping to Overlapping Clusters. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, August 13 - 17, 2017*. ACM, Halifax, NS, Canada, 145–154. https://doi.org/10.1145/3097983.3098054

[6] Wenzheng Feng, Jie Zhang, Yuxiao Dong, Yu Han, Huanbo Luan, Qian Xu, Qiang Yang, Evgeny Kharlamov, and Jie Tang. 2020. Graph Random Neural Networks for Semi-Supervised Learning on Graphs. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., Online, 22092–22103.

[7] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. 2017. Protein Interface Prediction using Graph Convolutional Networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). Curran Associates Inc., Long Beach, CA, USA, 6530–6539.

[8] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29, 5 (2001), 1189 – 1232.

[9] Keinosuke Fukunaga. 1990. *Introduction to Statistical Pattern Recognition (2nd Ed.)*. Academic Press Professional, Inc., USA.

[10] Arman Hasanzadeh, Ehsan Hajiramezanali, Krishna R. Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. 2019. Semi-Implicit Graph Variational Auto-Encoders. In *NeurIPS 2019*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). Curran Associates Inc., Vancouver, BC, Canada, 10711–10722.

[11] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *5th International Conference on Learning Representations, ICLR 2017, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, Toulon, France.

[12] Chen Li, Xutan Peng, Hao Peng, Lihong Wang, and Jianxin Li. 2021. TextGTL: Graph-based Transductive Learning for Semi-Supervised Text Classification via Structure-Sensitive Interpolation. In *IJCAI*. ijcai.org.

[13] Chen Li, Xutan Peng, Shanghang Zhang, Hao Peng, Philip S. Yu, Min He, Linfeng Du, and Lihong Wang. 2020. Modeling relation paths for knowledge base completion via joint adversarial training. *Knowl. Based Syst.* 201-202 (2020), 105865.

[14] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper Insights Into Graph Convolutional Networks for Semi-Supervised Learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), February 2-7, 2018*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, New Orleans, Louisiana, USA, 3538–3545. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16098

[15] Zhiwei Liu, Yingtong Dou, Philip S. Yu, Yutong Deng, and Hao Peng. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. In *SIGIR*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1569–1572.

[16] Jiaqi Ma, Weijing Tang, Ji Zhu, and Qiaozhu Mei. 2019. A Flexible Generative Framework for Graph-based Semi-supervised Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). Curran Associates Inc., Vancouver, BC, Canada, 3276–3285.

[17] Grgoire Montavon, Genevive Orr, and Klaus-Robert Mller. 2012. *Neural Networks: Tricks of the Trade* (2nd ed.). Springer Publishing Company, Incorporated, USA.

[18] Francesco Orsini, Daniele Baracchi, and Paolo Frasconi. 2018. Shift Aggregate Extract Networks. *Front. Robotics and AI* 2018 (2018).

[19] Hao Peng, Jianxin Li, Senzhang Wang, Lihong Wang, Qiran Gong, Renyu Yang, Bo Li, Philip S. Yu, and Lifang He. 2021. Hierarchical Taxonomy-Aware and Attentional Graph Capsule RCNNs for Large-Scale Multi-Label Text Classification. *IEEE Trans. Knowl. Data Eng.* 33, 6 (2021), 2505–2519.

[20] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*. OpenReview.net, Addis Ababa, Ethiopia. https://openreview.net/forum?id=Hkx1qkrKPr

[21] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. 2020. An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. arXiv:2003.04819 [cs.LG]

[22] Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling Relational Data with Graph Convolutional Networks. In *The Semantic Web*, Aldo Gangemi, Roberto Navigli, Maria-Esther Vidal, Pascal Hitzler, Raphaël Troncy, Laura Hollink, Anna Tordai, and Mehwish Alam (Eds.). Springer International Publishing, Cham, 593–607.

[23] Min Shi, Yufei Tang, and Xingquan Zhu. 2020. Topology and Content Co-Alignment Graph Convolutional Learning. *CoRR* abs/2003.12806 (2020).

[24] Otilia Stretcu, Krishnamurthy Viswanathan, Dana Movshovitz-Attias, Emmanouil A. Platanios, Sujith Ravi, and Andrew Tomkins. 2019. Graph Agreement Models for Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.). Curran Associates Inc., 8710–8720.

[25] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *6th International Conference on Learning Representations, ICLR 2018, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, Vancouver, BC, Canada.

[26] Bingbing Xu, Junjie Huang, Liang Hou, Huawei Shen, Jinhua Gao, and Xueqi Cheng. 2020. Label-Consistency based Graph Neural Networks for Semi-supervised Node Classification. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, July 25-30, 2020*, Jimmy Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1897–1900.

[27] Chunyan Xu, Zhen Cui, Xiaobin Hong, Tong Zhang, Jian Yang, and Wei Liu. 2020. Graph inference learning for semi-supervised classification. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*. OpenReview.net, Addis Ababa, Ethiopia.

[28] Liang Yang, Zesheng Kang, Xiaochun Cao, Di Jin, Bo Yang, and Yuanfang Guo. 2019. Topology Optimization based Graph Convolutional Network. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, Sarit Kraus (Ed.). ijcai.org, 4054–4061.

[29] Zhilin Yang, William W. Cohen, and Ruslan Salakhutdinov. 2016. Revisiting Semi-Supervised Learning with Graph Embeddings. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, June 19-24, 2016 (JMLR Workshop and Conference Proceedings, Vol. 48)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.). JMLR.org, New York City, NY, USA, 40–48. http://proceedings.mlr.press/v48/yanga16.html

[30] Ze Ye, Kin Sum Liu, Tengfei Ma, Jie Gao, and Chao Chen. 2020. Curvature Graph Network. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*. OpenReview.net, Addis Ababa, Ethiopia.

[31] Kai Zhang, Yaokang Zhu, Jun Wang, and Jie Zhang. 2020. Adaptive Structural Fingerprints for Graph Attention Networks. In *8th International Conference on Learning Representations, ICLR 2020, April 26-30, 2020*. OpenReview.net, Addis Ababa, Ethiopia.

[32] Chenyi Zhuang and Qiang Ma. 2018. Dual Graph Convolutional Networks for Graph-Based Semi-Supervised Classification. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, April 23-27, 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, Lyon, France, 499–508.