Contrastive Graph Neural Network Explanation

Lukas Faber [*1] **Amin K. Moghaddam** [*1] **Roger Wattenhofer** [*1]

Abstract

Graph Neural Networks achieve remarkable results on problems with structured data but come as black-box predictors. Transferring existing explanation techniques, such as occlusion, fails as even removing a single node or edge can lead to drastic changes in the graph. The resulting graphs can differ from all training examples, causing model confusion and wrong explanations. Thus, we argue that explicability must use graphs compliant with the distribution underlying the training data. We coin this property *Distribution Compliant Explanation* (DCE) and present a novel Contrastive GNN Explanation (CoGE) technique following this paradigm. An experimental study supports the efficacy of CoGE.

1. Introduction

While neural networks have shown many breakthroughs, they come as black-box predictors. Thus, research has developed explicability techniques to shed light on how a neural network makes decisions. Such explanations help to understand and trust the model. One approach is to identify those parts of the input, which are most influential in generating the output. With Graph Neural Networks (GNNs) being generalizations of Convolutional Neural Networks (CNNs), one might hope that the techniques for image explicability of CNNs transfer to GNNs.

For example, we could apply occlusion (Zeiler & Fergus, 2014; Ancona et al., 2017) to GNNs. In computer vision, occlusion measures the importance of a pixel by removing the pixel from the image, e.g., by setting it black. Similarly, we could measure the importance of a node or edge by removing this node or edge. However, such removals are drastic. Especially in sparse graphs node or edge removals may change the topology of the graph completely.

[*]Alphabetical Order [1]ETH Zurich, Switzerland. Correspondence to: Lukas Faber <lfaber@ethz.ch>, Amin K. Moghaddam <khashkhm@ethz.ch>.

Proceedings of the 37^{th} Graph Representation Learning and Beyond Workshop at ICML 2020, Copyright 2020 by the author(s).

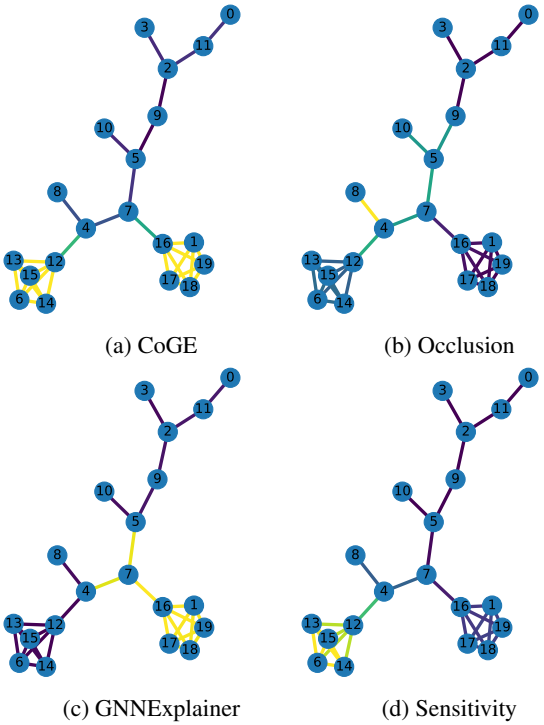
In Figure 1b, we can see the explanation of occlusion, asking whether the graph contains a clique. Occlusion considers the yellow edge most indicative of the existence of a clique. This edge is not part of a clique. Instead, removing the edge creates a disconnected graph. This graph probably confuses the GNN, which was trained with only connected graphs. This phenomenon can happen whenever we use graphs for explanations that substantially differ from the training data. The explanation of the target class might blend with an adversarial attack against it. Thus we find a misleading (in this case even wrong) explanation. Therefore, we argue to only ever use data consistent with the training distribution for making model explanations. We call this requirement doing *Distribution Compliant Explanation* (DCE).

We propose a novel method CoGE (Contrastive Gnn Explanation) based on contrastive explanation (Dhurandhar et al., 2018; 2019). CoGE aims to find similarities to graphs with the same label and differences to graphs with a different label. Using only existing graphs for explanation, CoGE clearly fulfills DCE. In an experimental evaluation, we show the efficacy of CoGE. Our contributions are:

- We motivated the necessity of DCE for explaining GNN predictions.
- We present CoGE, a novel method for explaining GNN predictions for graph classification. CoGE uses a contrastive approach and adheres to DCE.
- We show the efficacy of our method empirically on existing and synthetic datasets.

2. Related work

Graph Neural Networks Starting with Scarselli et al. (2008), graph neural networks have achieved remarkable results for graph-structured predictions (Zhou et al., 2018; Wu et al., 2020). Since then, one family of graph neural networks was developed as generalized convolutional networks (Duvenaud et al., 2015; Kipf & Welling, 2017) where nodes update their embedding via aggregating their neighborhood (Gilmer et al., 2017; Hamilton et al., 2017; Battaglia et al., 2018; Xu et al., 2019b). Other propagation methods such as attention or skip-connections have also been proposed (Veličković et al., 2018; Xu et al., 2018).



(a) CoGE (b) Occlusion

(c) GNNExplainer (d) Sensitivity

Figure 1. Edge explanations methods for explaining a clique predictor. Edges in cliques should be bright for correct explanations.

Explainability Methods for Graphs GNNExplainer explains predictions by finding an edge mask and feature mask that maximize the mutual information between the prediction and the masked substructure (Ying et al., 2019). Image attribution methods (Ancona et al., 2017) such as saliency maps and Layer-wise Relevance Propagation have been adapted to graph data structures (Baldassarre & Azizpour, 2019; Pope et al., 2019; Huang et al., 2020). These methods do not consider the data distribution. For example, an edge mask can "remove" edges from the graph and thus, similar to occlusion, lead to misleading explanations.

Adversarial Graph Attacks Adversarial attacks on graphs exploit that graph neural networks behave in an undefined way outside of the kind of data they saw during training. Several attack angles have been applied successfully(Dai et al., 2018; Zügner et al., 2018; Zügner & Günnemann, 2019; Xu et al., 2019a). At that, Zügner et al. (2018) noted that already small perturbations lead to large changes in the prediction. In this, we see support in our claim that DCE is important for explanations as DCE work contrary to adversarial attacks. Adversarial attacks find and exploit out-of-distribution anomalies in the model, whereas a DCE conforming method aims to explicitly disregard these anomalies and instead searches for genuine patterns.

3. Method

Preliminaries We consider GNNs operating on undirected graphs $G = (V, E)$ labeled by $y(G)$, with node set V and edge set E. Nodes can have attributes in a feature matrix $\mathbf{X}$. We assume a learning model that transforms these initial node features into final embeddings that are aggregated for a graph-level representation. One example of such networks is Message Passing Graph Neural Networks (Gilmer et al., 2017) that iteratively update node embeddings based on their immediate neighbors.

Explanations for graph classification We follow a contrastive approach (Dhurandhar et al., 2018; 2019). Such an approach bases the explanation on other graphs from training, and thus fulfills DCE. In particular, we find the parts of the graph that make this graph distant to graphs with a different label and close to graphs with the same label.

To this end, we need a measure to compare the distance between graphs. In our case, it suffices to define how graphs differ for the classification problem. There exist many graph distance and similarity measures (Sanfeliu & Fu, 1983; Heimann et al., 2018; Zhang & Lee, 2019; Wang et al., 2019) in the literature but they compare principally and offer more than we need. For our explanation purpose, we opt to compare the similarity in the model embedding space. Thus, we can leverage the model information which structures and features are relevant or semantically equivalent for the classification problem. For computing the similarity of two graphs (with respect to the classification problem), we measure a set to set distance of the final node embeddings.

In particular, we measure using the Optimal Transport (OT) distance (Nikolentzos et al., 2017; Fey et al., 2020). Figure 2 shows an example of computing OT between the left and middle graphs. All nodes have a weight (such that the weights of all nodes per graph sum to 1. Now, every node from the source needs to transport its weight to one or more target nodes, whose weight denotes their maximum capacity. The cost of one transport is the transport weight times the distance between the node embeddings — we use L_2 distance for this. Optimal transport finds the globally optimal soft assignment, even if this involves suboptimal choices for some nodes. For example in Figure 2, node 2 does not move all its weight to node 4, even though they are the same and the cost would be 0. Thus, optimal transport allows us to compare embeddings of two graphs on node granularity. We compare this to a graph-level metric in the experiments where we measure the L_2 distance between the average node embeddings.

But we can also change the initial node weights to something different than uniform. If we want to change the source weights as to *minimize* optimal transport, the nodes that do not have counterparts in the target graph receive a low

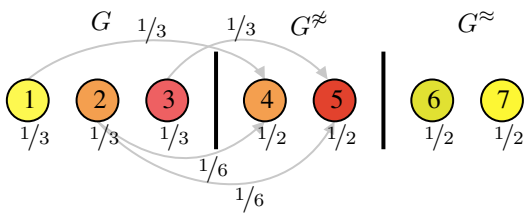


Figure 2. Example for optimal transport (OT).

weight. If we want to change source weights as to *maximize* optimal transport, the nodes with counterparts receive a low weight. These two observations give us a framework to find explaining graph parts. Jointly maximizing OT to graphs with the same label and minimizing OT to graphs with a different label finds us the explanation nodes. These are the nodes with a low weight. Formally, we want to find weights w_opt for the nodes of graph G such that

$$w_{opt}(G) = \arg\min_{w} \mathcal{L}_w^{\not\approx}(G) - \mathcal{L}_w^{\approx}(G) + \mathcal{L}_w^{=}(G) \quad (1)$$

The first loss term captures the average distance to the k most similar graphs with a different label $\mathbb{G}_k^{\not\approx}$. The second loss term captures the average distance to the k most similar graphs with the same label $\mathbb{G}_k^{\approx}$. We compute the similarity to choose the graphs with uniformly weighted optimal transport. The hyperparameter k determines how many graphs to compare against. The third term compares the distance of the w-weighted graph G to its uniformly-weighted version. This term acts as regularization. It penalizes any deviation from uniform weights, thus biasing w to only make few modifications with substantial benefits. Let $d(Z_G, w, Z_H)$ be the optimal transport distance between two sets of embeddings from graphs G and H, where we weight Z_G according to w and Z_H with uniform weights. Formally the losses are then defined as:

$$\mathcal{L}_W^{\not\approx}(G) = \frac{1}{k} \sum_{H \in \mathbb{G}_k^{\not\approx}} d_W(Z_G, Z_H)$$

$$\mathcal{L}_W^{\approx}(G) = \frac{1}{k} \sum_{H \in \mathbb{G}_k^{\approx}} d_W(Z_G, Z_H)$$

$$\mathcal{L}_W^{=}(G) = d_W(Z_G, Z_G)$$

4. Experiments

4.1. CoGE Implementation

For CoGE, we set the number k of graphs to contrast again to 10. Our framework uses the OT as provided by the GeomLoss library (Feydy et al., 2019). For optimization of Equation (1), we use gradient descent using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.1, except for REDDIT, where we use 0.01[1].

4.2. Qualitative Analysis

We analyze our explanations on two well known real-world datasets for graph classification: MUTAG (Debnath et al., 1991) and REDDIT-BINARY (Yanardag & Vishwanathan, 2015). MUTAG labels 4337 chemical molecules for their mutagenic effect. REDDIT-BINARY classifies 2000 Reddit threads whether they are of type Q&A or discussion. We train Graph Isomorphism networks (Xu et al., 2019b), which achieve state-of-the-art performance on both datasets and analyze the trained models.

In MUTAG (see Figure 3), our method identifies the NO_2 structure (red circle) as being primarily important, which is a known mutagenic part (Debnath et al., 1991). However, it is also present in a few non-mutagenic graphs. As a second explanation, our method identifies the C next to an O close to this component (green circle). The combination of structures is only present in the mutagenic examples.

In REDDIT-BINARY, our method considers the central and adjacent nodes to be important for classifying this graph as Q&A (see Figure 4). They are indeed important since a Q&A threads consist of few experts (high degree nodes) and most users ask them questions and getting replies. In contrast, discussions typically have only one central node and the graph has a tree-like structure with higher depth.

4.3. Quantitative Analysis

Dataset We present *CYCLIQ* (cycles and cliques), a new dataset for explaining graph classifications similar to the node classification dataset in Ying et al. (2019). The CYCLIQ dataset is a binary classification problem. CYCLIQ's primary building blocks are random trees, to which we append either cycles or cliques. The target labels store if the graph has cycles or cliques. Nodes have an initial feature vector of size 10, initialized to all ones. The correct explanations for this dataset are the edges in a clique or cycle structure. For evaluation, we count the number x edges being part of a cycle or a clique. The explanation accuracy of a method is the ratio of how many edges in cycle or cliques are in the method's x most important edges.

Experiment Setup We run all explanations on a GCN (Kipf & Welling, 2017) with 5 layers. We use a constant embedding size of 20 across all layers. There are no edge features. We create a total of 2000 graphs, using an 80/20 train-test split. The model understands the task well, reaching 99% test accuracy. We compare our

[1]Code available at https://github.com/lukasjf/contrastive-gnn-explanation

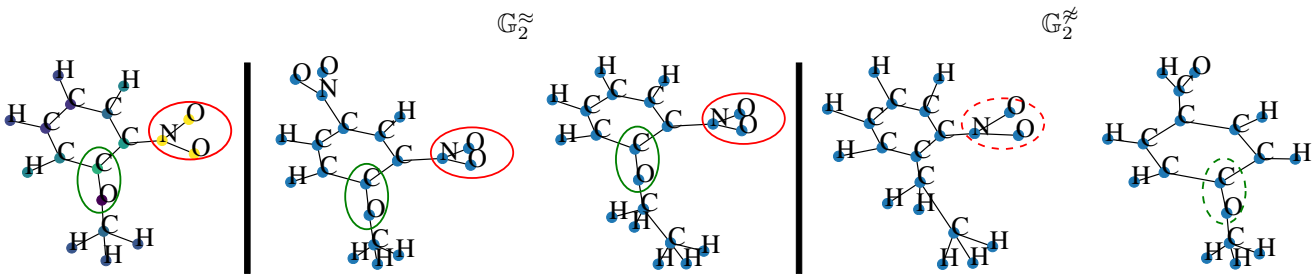


Figure 3. MUTAG Explanation showing important substructures in molecules. Left: Original graph, Middle: Similar graphs with the same label, Right: Similar graphs with a different label. Brighter nodes are more important in the explanation.

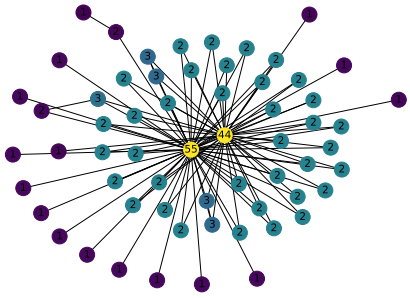


Figure 4. Example Q&A explanation. Numbers depict the node degree. Nodes connected to both central nodes are more important.

method against random guessing, a node-based occlusion method (removing adjacent edges), sensitivity analysis, and GNNExplainer (Ying et al., 2019) using the provided implementation[2]. To allow comparison with GNNExplainer we focus on edge importances by summing the importance of adjacent nodes for node-based explanation methods.

Results We report the explanation accuracies across the 1600 training graphs in table 1. Generally, it seems easier to explain cliques than it is to explain cycles. We note that CoGE produces the best results. For both classes it outperforms the other methods by at least 10%, reaching almost perfect accuracy for cliques. Visual example explanations for all methods (but random) are in Figure 1.

Ablation Study Next, we study the importance of each loss term from Equation (1), reusing above CYCLIQ setting. Table 2 shows that $\mathcal{L}_W^{\not\approx}$ captures most explanation, but $\mathcal{L}_W^{\approx}$ and $\mathcal{L}_W^{=}$ help improve further. Additionally, we try replacing OT distance with the euclidean distance on the weighted average on the node embeddings ($\mathcal{L}$ and Average). This leads to worse accuracy while still outperforming baselines.

[2] https://github.com/RexYing/gnn-model-explainer

Table 1. Explanation accuracies on the CYCLIQ dataset.

Method	Cycle Acc.	Clique Acc.	Avg. Acc.
Random	0.41 ± 0.17	0.58 ± 0.13	0.49
Occlusion	0.39 ± 0.23	0.86 ± 0.16	0.62
Sensitivity	0.36 ± 0.2	0.87 ± 0.12	0.61
GNNExplainer	0.43 ± 0.18	0.73 ± 0.14	0.58
CoGE	$\mathbf{0.78 \pm 0.18}$	$\mathbf{0.99 \pm 0.02}$	**0.88**

Table 2. Explanation accuracies on the CYCLIQ dataset for ablations of the loss and the distance measure.

Loss	Cycle Acc.	Clique Acc.	Avg. Acc.
$-\mathcal{L}_W^{\approx}$	0.63 ± 0.25	0.6 ± 0.35	0.62
$-\mathcal{L}_W^{\approx} + \mathcal{L}_W^{=}$	0.62 ± 0.23	0.61 ± 0.27	0.61
$\mathcal{L}_W^{\not\approx}$	0.65 ± 0.23	0.99 ± 0.02	0.81
$\mathcal{L}_W^{\not\approx} + \mathcal{L}_W^{=}$	0.66 ± 0.24	0.99 ± 0.02	0.82
$\mathcal{L}_W^{\not\approx} - \mathcal{L}_W^{\approx}$	0.7 ± 0.2	0.99 ± 0.02	0.84
$\mathcal{L}_W$ and Average	0.45 ± 0.24	0.99 ± 0.02	0.71
$\mathcal{L}_W$ and OT	0.78 ± 0.18	0.99 ± 0.02	0.88

5. Conclusion

In this work, we discuss the particularities of explaining GNN predictions. In graphs, structure is important, as only slight modifications can lead to graphs out of the known data distribution. Explanations start to blend with adversarial attacks. Therefore, we argue that explanation methods should stay with the training data distribution and produce *Distribution Compliant Explanation* (DCE). We propose a novel explanation method, CoGE, for graph classification that adheres to DCE. Experimental results verify its effectiveness and its robustness to parameter choices. Fur future work, we aim at extending our findings to node classification and better understanding the connection between explanation and adversarial attacks.

References

Ancona, M., Ceolini, E., Öztireli, C., and Gross, M. A unified view of gradient-based attribution methods for deep neural networks. In *NIPS Workshop on Interpreting, Explaining and Visualizing Deep Learning, Long Beach, USA*, December 2017.

Baldassarre, F. and Azizpour, H. Explainability techniques for graph convolutional networks. *arXiv preprint arXiv:1905.13686*, 2019.

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Dai, H., Li, H., Tian, T., Huang, X., Wang, L., Zhu, J., and Song, L. Adversarial attack on graph structured data. In *Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden*, July 2018.

Debnath, A. K., Lopez de Compadre, R. L., Debnath, G., Shusterman, A. J., and Hansch, C. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 1991.

Dhurandhar, A., Chen, P.-Y., Luss, R., Tu, C.-C., Ting, P., Shanmugam, K., and Das, P. Explanations based on the missing: Towards contrastive explanations with pertinent negatives. In *Advances in Neural Information Processing Systems*, 2018.

Dhurandhar, A., Pedapati, T., Balakrishnan, A., Chen, P.-Y., Shanmugam, K., and Puri, R. Model agnostic contrastive explanations for structured data. *arXiv preprint arXiv:1906.00117*, 2019.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in Neural Information Processing Systems*, 2015.

Fey, M., Lenssen, J. E., Morris, C., Masci, J., and Kriege, N. M. Deep graph matching consensus. *arXiv preprint arXiv:2001.09621*, 2020.

Feydy, J., Séjourné, T., Vialard, F.-X., Amari, S.-i., Trouve, A., and Peyré, G. Interpolating between optimal transport and mmd using sinkhorn divergences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), Naha, Japan*, April 2019.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *Proceedings of the International Conference on Machine Learning (ICML), Sydney, Australia*, August 2017.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, 2017.

Heimann, M., Shen, H., Safavi, T., and Koutra, D. Regal: Representation learning-based graph alignment. In *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM), Turin, Italy*, October 2018.

Huang, Q., Yamada, M., Tian, Y., Singh, D., Yin, D., and Chang, Y. Graphlime: Local interpretable model explanations for graph neural networks. *arXiv preprint arXiv:2001.06216*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France Toulon, France, April 24-26, 2017, Conference Track Proceedings*, April 2017.

Nikolentzos, G., Meladianos, P., and Vazirgiannis, M. Matching node embeddings for graph similarity. In *Proceedings of the Conference on Artificial Intelligence (AAAI), San Francisco, USA*, February 2017.

Pope, P. E., Kolouri, S., Rostami, M., Martin, C. E., and Hoffmann, H. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, USA*, June 2019.

Sanfeliu, A. and Fu, K.-S. A distance measure between attributed relational graphs for pattern recognition. *IEEE transactions on systems, man, and cybernetics*, 1983.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 2008.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph Attention Networks. *Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, Canada*, May 2018.

Wang, R., Yan, J., and Yang, X. Learning combinatorial embedding networks for deep graph matching. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR), Long Beach, USA*, June 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Yu, P. S. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2020.

Xu, K., Li, C., Tian, Y., Sonobe, T., Kawarabayashi, K., and Jegelka, S. Representation learning on graphs with jumping knowledge networks. In *Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden*, July 2018.

Xu, K., Chen, H., Liu, S., Chen, P.-Y., Weng, T.-W., Hong, M., and Lin, X. Topology attack and defense for graph neural networks: An optimization perspective. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Macao, China*, July 2019a.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *Proceedings of the International Conference on Learning Representations (ICLR) New Orleans, USA*, May 2019b.

Yanardag, P. and Vishwanathan, S. Deep graph kernels. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), Sydney, Australia*, August 2015.

Ying, Z., Bourgeois, D., You, J., Zitnik, M., and Leskovec, J. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in Neural Information Processing Systems*, 2019.

Zeiler, M. D. and Fergus, R. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision (ECCV), Zurich, Switzerland*, September 2014.

Zhang, Z. and Lee, W. S. Deep graphical feature learning for the feature matching problem. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR), Long Beach, USA*, June 2019.

Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

Zügner, D. and Günnemann, S. Adversarial attacks on graph neural networks via meta learning. In *Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, USA*, May 2019.

Zügner, D., Akbarnejad, A., and Günnemann, S. Adversarial attacks on neural networks for graph data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), London, UK*, August 2018.