Journal of Machine Learning Research 22 (2021) 1-9 Submitted 3/21; Revised 8/21; Published 9/21

DIG: A Turnkey Library for Diving into Graph Deep Learning Research

Meng Liu* MENGLIU@TAMU.EDU
Youzhi Luo* YZLUO@TAMU.EDU
Limei Wang* LIMEI@TAMU.EDU
Yaochen Xie* ETHANYCX@TAMU.EDU
Hao Yuan* HAO.YUAN@TAMU.EDU
Shurui Gui* SHURUI.GUI@TAMU.EDU
Haiyang Yu* HAIYANG@TAMU.EDU
Zhao Xu ZHAOXU@TAMU.EDU
Jingtun Zhang ZJT6791@TAMU.EDU
Yi Liu YILIU@TAMU.EDU
Keqiang Yan KEQIANGYAN@TAMU.EDU
Haoran Liu LIUHR99@TAMU.EDU
Cong Fu CONGFU@TAMU.EDU
Bora Oztekin BORA@TAMU.EDU
Xuan Zhang XUAN.ZHANG@TAMU.EDU
Shuiwang Ji SJI@TAMU.EDU
Department of Computer Science and Engineering
Texas A&M University
College Station, TX 77843-3112, USA

Editor: Alexandre Gramfort

Abstract

Although there exist several libraries for deep learning on graphs, they are aiming at implementing basic operations for graph deep learning. In the research community, implementing and benchmarking various advanced tasks are still painful and time-consuming with existing libraries. To facilitate graph deep learning research, we introduce *DIG: Dive into Graphs*, a turnkey library that provides a unified testbed for higher level, research-oriented graph deep learning tasks. Currently, we consider graph generation, self-supervised learning on graphs, explainability of graph neural networks, and deep learning on 3D graphs. For each direction, we provide unified implementations of data interfaces, common algorithms, and evaluation metrics. Altogether, *DIG* is an extensible, open-source, and turnkey library for researchers to develop new methods and effortlessly compare with common baselines using widely used datasets and evaluation metrics. Source code is available at `https://github.com/divelab/DIG`.

Keywords: graph deep learning, generation, self-supervised learning, explainability, 3D graphs, Python

1. Introduction

Graph deep learning (Bronstein et al., 2017; Hamilton et al., 2017b; Wu et al., 2020; Zhou et al., 2018; Battaglia et al., 2018; Hamilton, 2020; Ma and Tang, 2020) has been drawing

∗. These authors contributed equally.

increasing attention due to its effectiveness in learning from rich graph data. It has achieved remarkable successes in many domains, such as social networks (Kipf and Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017a), drug discovery (Gilmer et al., 2017; Wu et al., 2018; Stokes et al., 2020; Wang et al., 2020), and physical simulation (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2020). Several libraries, such as PyG (Fey and Lenssen, 2019), DGL (Wang et al., 2019), tf_geometric (Hu et al., 2021), Spektral (Grattarola and Alippi, 2020), GraphNet (Battaglia et al., 2018), StellarGraph (Data61, 2018), GraphGallery (Li et al., 2021), CogDL (Cen et al., 2021), and OGB (Hu et al., 2020), have been developed to facilitate deep learning on graphs. However, most existing libraries focus on providing basic components of graph neural networks and mainly consider elementary tasks, such as node classification and graph classification. With these libraries, it still costs a lot of efforts to implement and benchmark algorithms for advanced tasks, such as graph generation.

To bridge this gap, we present a Python library *DIG: Dive into Graphs*. We currently consider several research directions in graph deep learning. These are graph generation, self-supervised learning on graphs, explainability of graph neural networks, and deep learning on 3D graphs. For each direction, *DIG* provides unified and extensible implementations of data interfaces, common algorithms, and evaluation metrics. *DIG* naturally endows researchers the convenience of developing their algorithms and conducting empirical comparisons with baselines. Altogether, DIG is an extensible, open-source, and turnkey library for researchers to develop new methods and effortlessly compare with common baselines using widely used datasets and evaluation metrics.

2. Library Description

Our *DIG* is based on Python and PyTorch (Paszke et al., 2017). For some implementations, we also use PyG (Fey and Lenssen, 2019) and RDKit (Landrum et al., 2006) for basic operations on graphs and molecules. *DIG* currently considers 4 directions and contains 18 algorithms. Note that more interesting directions and algorithms can be easily incorporated into *DIG* based on the unified and extensible implementations. An overview of the *DIG* library is illustrated in Figure 1. We introduce the main implementations as follows.

Graph generation. Given a set of graphs, graph generation algorithms aim at generating novel graphs (Guo and Zhao, 2020; Faez et al., 2020). Graph generation is potentially useful for molecule discovery. Hence, we mainly consider algorithms that can generate molecular graphs. We include the following advanced algorithms: JT-VAE (Jin et al., 2018), GraphAF (Shi et al., 2019), GraphDF (Luo et al., 2021), and GraphEBM (Liu et al., 2021a). We implement data interfaces for widely used datasets. These are QM9 (Ramakrishnan et al., 2014), ZINC250k (Irwin et al., 2012), and MOSES (Polykovskiy et al., 2020). Metrics for evaluating random generation, property optimization, and constrained property optimization are also implemented as APIs.

Self-supervised learning on graphs. Self-supervised learning can help to obtain expressive representations by leveraging specified pretext tasks and has been recently extended to graph domain (Jin et al., 2020; Xie et al., 2021). We incorporate InfoGraph (Sun et al., 2020), GRACE (Zhu et al., 2020), MVGRL (Hassani and Khasahmadi, 2020), and GraphCL (You et al., 2020) in *DIG*. We provide the data interfaces of TUDataset (*i.e.*, NCI1, PROTEINS, *etc.*) (Morris et al., 2020) for graph-level classification tasks, and citation net-

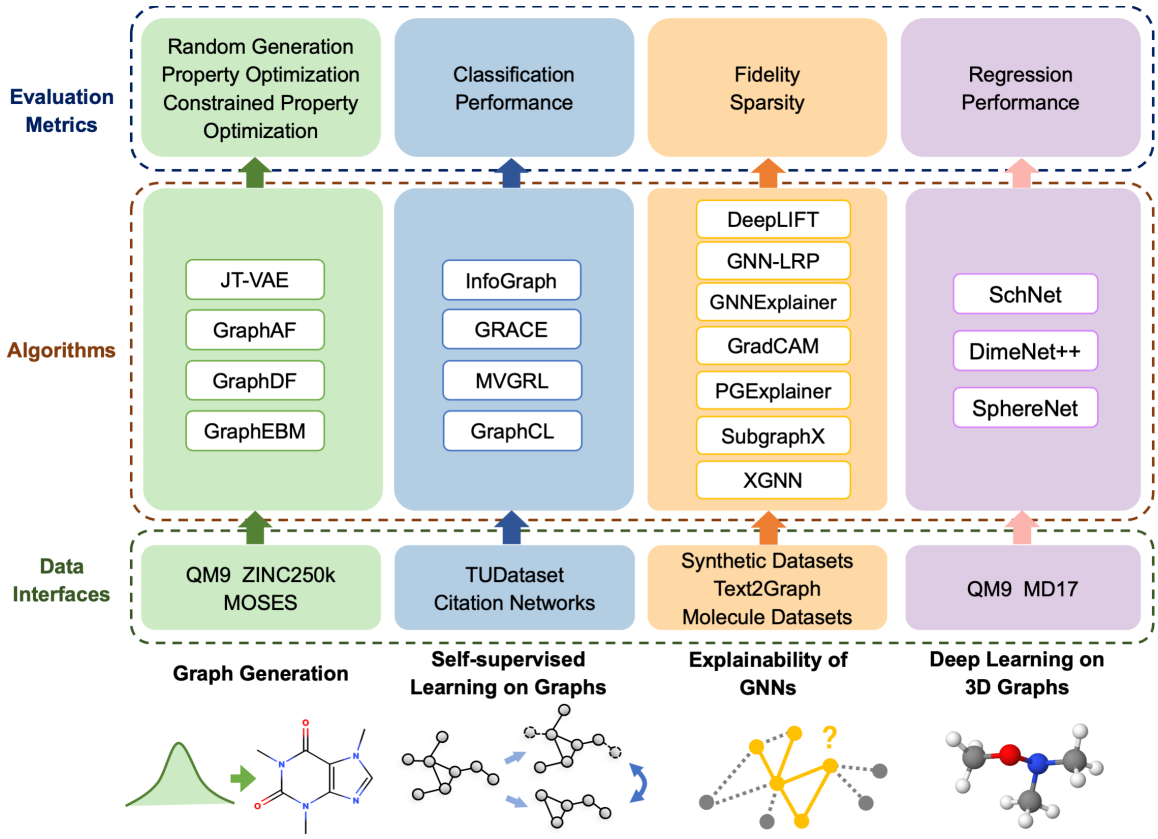

Figure 1: A graphical overview of *DIG: Dive into Graphs*.

works (*i.e.*, Cora, CiteSeer, and PubMed) (Yang et al., 2016) for node-level classification tasks. Standard metrics are also realized to evaluate the classification performance.

Explainability of graph neural networks. Since graph neural networks have been increasingly deployed in our real-world applications, it is critical to develop explanation techniques for better understanding of models (Yuan et al., 2020b). We include the following algorithms: GNNExplainer (Ying et al., 2019), PGExplainer (Luo et al., 2020), DeepLIFT (Shrikumar et al., 2017), GNN-LRP (Schnake et al., 2020), Grad-CAM (Pope et al., 2019), SubgraphX (Yuan et al., 2021), and XGNN (Yuan et al., 2020a). For data interfaces, we consider the widely used synthetic datasets (*i.e.*, BA-shapes, BA-Community, *etc.*) (Ying et al., 2019; Luo et al., 2020) and molecule datasets (*i.e.*, BBBP, Tox21, *etc.*) (Wu et al., 2018). In addition, we also build human-understandable graph datasets from text data and provide the corresponding data interfaces. Details of our proposed datasets (*i.e.*, Graph-SST2, Graph-SST5, *etc.*) are described by Yuan et al. (2020b). Recently proposed metrics for explanation tasks, including Fidelity and Sparsity (Pope et al., 2019), are implemented in our *DIG*.

Deep learning on 3D graphs. 3D Graphs refer to graphs whose nodes are associated with 3D positions. For instance, in molecules, each atom has a relative 3D position. It is significant to investigate how to obtain expressive graph representations with such essential

information. We consider three algorithms in the unified 3DGN framework (Liu et al., 2021b). These are SchNet (Schütt et al., 2017), DimeNet++ (Klicpera et al., 2020b,a), and SphereNet (Liu et al., 2021b). We implement data interfaces for two benchmark datasets: QM9 (Ramakrishnan et al., 2014) and MD17 (Chmiela et al., 2017). We apply mean absolute error (MAE), a standard metric for regression tasks, as the evaluation technique.

3. Key Design Considerations

In this section, we described the key design considerations of *DIG*, including unified implementation, extensibility, and customization.

Unified implementation. As described in Section 2 and illustrated in Figure 1, we provide APIs of data interfaces, common algorithms, and evaluation metrics for each direction. This provides a standardized testbed for various algorithms in each direction. In addition, our implementations are unified for different algorithms if they enjoy non-trivial commonalities. To be specific, implementations of the three algorithms on 3D graphs can be unified using the 3DGN framework (Liu et al., 2021b) with different internal functions. Also, many self-supervised learning algorithms on graphs can be viewed as contrastive models (Xie et al., 2021). Hence, we provide unified objective functions for these algorithms.

Extensibility and customization. As a benefit of our unified implementations, it is easy to incorporate new datasets, algorithms, and evaluation metrics. Additionally, users can customize their own experiments on their new algorithms by flexibly choosing desired data interfaces and evaluation metrics. Therefore, our *DIG* can serve as a platform for implementing and benchmarking algorithms in the covered directions.

4. Quality Standards

In the following, we evaluate our *DIG* according to several quality standards of open source software.

Code reliability and reproducibility. The APIs of data interfaces and evaluation metrics in *DIG* have been extensively tested by Travis CI, a continuous integration tool. In addition, for APIs corresponding to advanced algorithms, we provide the benchmark examples, which can reproduce the experimental results reported in the original papers within reasonable or negligible differences.

Documentation. *DIG* has complete documentations online[1], including the detailed descriptions of APIs and hands-on tutorials.

Openness. Contributions from the community are strongly welcome and encouraged. In our documented contribution guideline, we describe how to provide various types of contributions. The library is distributed under the GNU GPLv3 license.

5. Conclusion and Outlook

In this paper, we present *DIG: Dive into Graphs* that contains unified and extensible implementations of data interfaces, common algorithms, and evaluation metrics for several significant research directions, including graph generation, self-supervised learning on graphs,

1. `https://diveintographs.readthedocs.io`

explainability of graph neural networks, and deep learning on 3D graphs. We hope *DIG* can enable researchers to easily implement and benchmark algorithms. In the future, we are interested in incorporating more emerging directions and advanced algorithms into *DIG*.

Acknowledgments

This work was supported in part by National Science Foundation grants IIS-2006861, IIS-1955189, IIS-1908220, IIS-1908198, DBI-2028361, and DBI-1922969.

References

Peter Battaglia, Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, et al. Interaction networks for learning about objects, relations and physics. In *Advances in neural information processing systems*, pages 4502–4510, 2016.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Yukuo Cen, Zhenyu Hou, Yan Wang, Qibin Chen, Yizhen Luo, Xingcheng Yao, Aohan Zeng, Shiguang Guo, Peng Zhang, Guohao Dai, Yu Wang, Chang Zhou, Hongxia Yang, and Jie Tang. CogDL: An extensive toolkit for deep learning on graphs. *arXiv preprint arXiv:2103.00959*, 2021.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5):e1603015, 2017.

CSIRO's Data61. StellarGraph machine learning library. `https://github.com/stellargraph/stellargraph`, 2018.

Faezeh Faez, Yassaman Ommi, Mahdieh Soleymani Baghshah, and Hamid R Rabiee. Deep graph generators: A survey. *arXiv preprint arXiv:2012.15544*, 2020.

Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th international conference on machine learning*, pages 1263–1272, 2017.

Daniele Grattarola and Cesare Alippi. Graph neural networks in tensorflow and keras with spektral. *arXiv preprint arXiv:2006.12138*, 2020.

Xiaojie Guo and Liang Zhao. A systematic survey on deep generative models for graph generation. *arXiv preprint arXiv:2007.06686*, 2020.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017a.

William L Hamilton. Graph representation learning. *Synthesis Lectures on Artifical Intelligence and Machine Learning*, 14(3):1–159, 2020.

William L. Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40(3):52–74, 2017b.

Kaveh Hassani and Amir Hosein Khasahmadi. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR, 2020.

Jun Hu, Shengsheng Qian, Quan Fang, Youze Wang, Quan Zhao, Huaiwen Zhang, and Changsheng Xu. Efficient graph deep learning in tensorflow with tf_geometric, 2021.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

John J Irwin, Teague Sterling, Michael M Mysinger, Erin S Bolstad, and Ryan G Coleman. ZINC: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

Wei Jin, Tyler Derr, Haochen Liu, Yiqi Wang, Suhang Wang, Zitao Liu, and Jiliang Tang. Self-supervised learning on graphs: Deep insights and new direction. *arXiv preprint arXiv:2006.10141*, 2020.

Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction tree variational autoencoder for molecular graph generation. In *International Conference on Machine Learning*, pages 2323–2332, 2018.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Johannes Klicpera, Shankari Giri, Johannes T. Margraf, and Stephan Günnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. In *NeurIPS-W*, 2020a.

Johannes Klicpera, Janek Groß, and Stephan Günnemann. Directional message passing for molecular graphs. In *International Conference on Learning Representations (ICLR)*, 2020b.

Greg Landrum et al. RDKit: Open-source cheminformatics. 2006.

Jintang Li, Kun Xu, Liang Chen, Zibin Zheng, and Xiao Liu. Graphgallery: A platform for fast benchmarking and easy development of graph neural networks based intelligent software. *arXiv preprint arXiv:2102.07933*, 2021.

Meng Liu, Keqiang Yan, Bora Oztekin, and Shuiwang Ji. GraphEBM: Molecular graph generation with energy-based models. *arXiv preprint arXiv:2102.00546*, 2021a.

Yi Liu, Limei Wang, Meng Liu, Xuan Zhang, Bora Oztekin, and Shuiwang Ji. Spherical message passing for 3d graph networks. *arXiv preprint arXiv:2102.05013*, 2021b.

Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. *Advances in Neural Information Processing Systems*, 33, 2020.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A discrete flow model for molecular graph generation. In *International Conference on Machine Learning*, pages 7192–7203, 2021.

Yao Ma and Jiliang Tang. *Deep Learning on Graphs*. Cambridge University Press, 2020.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. TUDataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL `www.graphlearning.io`.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, Artur Kadurin, Simon Johansson, Hongming Chen, Sergey Nikolenko, Alan Aspuru-Guzik, and Alex Zhavoronkov. Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Frontiers in Pharmacology*, 2020.

Phillip E Pope, Soheil Kolouri, Mohammad Rostami, Charles E Martin, and Heiko Hoffmann. Explainability methods for graph convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10772–10781, 2019.

Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole Von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1): 1–7, 2014.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020.

T Schnake, O Eberle, J Lederer, S Nakajima, KT Schütt, KR Müller, and G Montavon. Higher-order explanations of graph neural networks via relevant walks. *arXiv: 2006.03589*, 2020.

Kristof T Schütt, PJ Kindermans, Huziel E Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus R Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Advances in Neural Information Processing Systems*, pages 1–11, 2017.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. GraphAF: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2019.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR, 2017.

Jonathan M Stokes, Kevin Yang, Kyle Swanson, Wengong Jin, Andres Cubillos-Ruiz, Nina M Donghia, Craig R MacNair, Shawn French, Lindsey A Carfrae, Zohar Bloom-Ackerman, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

Fan-Yun Sun, Jordan Hoffmann, Vikas Verma, and Jian Tang. InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization. In *International Conference on Learning Representations*, 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representation*, 2018.

Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv preprint arXiv:1909.01315*, 2019.

Zhengyang Wang, Meng Liu, Youzhi Luo, Zhao Xu, Yaochen Xie, Limei Wang, Lei Cai, and Shuiwang Ji. Advanced graph and sequence neural networks for molecular property prediction and drug discovery. *arXiv preprint arXiv:2012.01981*, 2020.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 2020.

Yaochen Xie, Zhao Xu, Zhengyang Wang, and Shuiwang Ji. Self-supervised learning of graph neural networks: A unified review. *arXiv preprint arXiv:2102.10757*, 2021.

Zhilin Yang, William Cohen, and Ruslan Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. GNNExplainer: Generating explanations for graph neural networks. *Advances in neural information processing systems*, 32:9240, 2019.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Advances in Neural Information Processing Systems*, volume 33, pages 5812–5823, 2020.

Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. XGNN: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 430–438, 2020a.

Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. Explainability in graph neural networks: A taxonomic survey. *arXiv preprint arXiv:2012.15445*, 2020b.

Hao Yuan, Haiyang Yu, Jie Wang, Kang Li, and Shuiwang Ji. On explainability of graph neural networks via subgraph explorations. In *International Conference on Machine Learning*, pages 12241–12252, 2021.

Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, Lifeng Wang, Changcheng Li, and Maosong Sun. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434*, 2018.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Deep graph contrastive representation learning. *arXiv preprint arXiv:2006.04131*, 2020.