

# A Review of Graph Signal Processing with Neural Networks

Yuzhong Yan, Cajetan M. Akujuobi

Department of Electrical and Computer Engineering  
Prairie View A&M University, Prairie View, TX 77446, USA

Received: July 13, 2021. Revised: January 9, 2022. Accepted: January 29, 2022. Published: February 25, 2022.

**Abstract**—In this paper, we review the development of the traditional graph signal processing methodology, and the recent research areas that are applying graph neural networks on graph data. For the popular topics on processing the graph data with neural networks, the main models/frameworks, dataset and applications are discussed in details. Some challenges and open problems are provided, which serve as the guidance for future research directions.

**Index Terms**—Graph Signal Processing, Graph Neural Networks, Graph Convolutional Networks, Deep Learning

## I. INTRODUCTION

Graph signal processing is not a totally new topic, and it has been developed for years. Traditional graph signal processing focus more on the spectral analysis of graph. However, with growing of the data, such as social network, Internet of Things (IoTs) and E-Commerce, we have entered into the big data era, which bring more challenges on the traditional data analytic methodologies. At the same time, with the great success of deep neural networks applied in image analytics and natural language processing, and the more intuitive thought is applying the deep neural networks on the large graph data. In the following context, we will discuss the traditional graph signal processing and the challenges, and then focus more on the graph neural networks developed in the recent years. The popular applications represented with graph data, Graph Neural Networks (GNN) models and frameworks will be discussed in in Section II and Section III respectively. At last, the new challenges and some open topics will be reviewed in the Section IV.

### A. Graph Signal Processing

Signal processing (SP) is an active area in both research and engineering domains. There are huge amount of practical use cases in the applications, and most of those focus on regular time series data and apply spectral analytic in frequency domain. With the evolution of the communication network, more smart devices and data sensors appear. The topology of those data sources forms a super large graph. Those traditional signal processing cannot handle the correlations between data samples from different data sensors effectively, and those relations between the data samples are irregular graph structure. Graph consists of both vertices and edges, where the data values are defined/sensed in vertices and edges connect these vertices [1]. Graph Signal Processing (GSP) defines the correlation between data samples and are effectively captured

in time and space by inserting the signal structure onto the graph, which leads to scalable and flexible approach to many SP problems [1]. The construction and manipulation of graphs are mainly focused on spectral graph theory as opposed to SP on graphs. The spectra of frequency and its basic expansions are defined by using the tools of spectral graph theory in the area of SP for Graph Fourier Transforms (GFT) [1] [2]. The frequently used operations in signal processing including down sampling, convolution, dilation, filtering, modulation and translation can also be applied on GSP after GFT. More details about GSP processed in graph spectral domains can be found in [3]. As analogues to the classical frequency domain, graph spectral domains highlight the importance of incorporating the irregular structures of graph data domains when processing signals on graphs. Although there are some general frameworks already developed for GSP, there are still some open issues, and in which the computations of transforms for signals require the Normalized Graph Laplacian (NGL) on huge size graph is the largest challenge.

### B. Graph Neural Networks

Traditional signal processing (SP) focus on more data representation and feature extractions, however, most of these hand-craft operators have been replaced by deep neural networks, especially in the domains of computer vision and natural language processing. The similar evolution also happened in graph signal processing domain, and in which GNN plays the key role in this evolution. The deep neural networks have succeeded in several areas in the recent years, in which the most successful case is the convolutional neural network (CNN) in image analytic area. There are some main benefits of CNN: local translational invariance, parameter sharing, locally connection and hierarchical expression. Graph methods have played an important role in traditional machine learning applications for long time, as they provide a natural way to represent the structure of a dataset [4]. Now the question is can the convolutional neural network be applied on graph data? The convolutional operation cannot operated on graph data directly, because local translational invariance does not apply directly on the Non-Euclidean structure in spatial domain as shown in Figure 1. In spectral domain, however, the answer is yes based on the spectral graph theory discussed in I-A. In space domain, before applying the convolutional operations, aggregation of the neighbor vertex is needed as shown in Section III. GNN

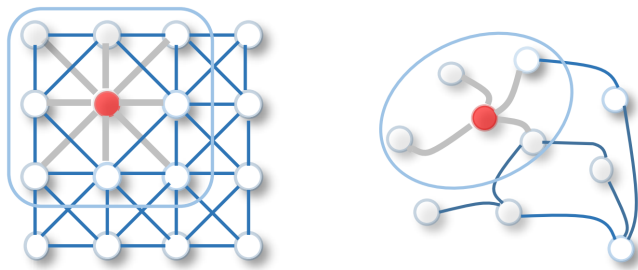


Fig. 1. Euclidean space and Non-Euclidean space [25]

was first proposed in [5], and after that, there are dozens of variants of GNN models proposed in recent research, such as [6] [7] [8] [9] [10] [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23]. We will discuss more details of these GNN models in III. In [24], it proves that graph convolutions with integral Lipschitz filters, in combination with the frequency mixing effect of the corresponding nonlinearities, yields an architecture that is both stable to small changes in the underlying topology and discriminative of information located at high frequencies. These are two properties that cannot simultaneously hold when using only linear graph filters, which are either discriminative or stable, thus explaining the superior performance of GNNs.

## II. APPLICATIONS, DATASETS AND BENCHMARK

As for the applications in real world, there are huge amount of irregular data. Figure 2 shows different kinds of applications building from the graph data. The traditional data analytics can handle the small scale data, however, it will reach the bottleneck on performance and cannot benefit too much from big data. The modern artificial intelligence relies more on big data instead, especially for the deep learning. Most of the graph datasets in public domain are in small scale. Until recently, there were some large datasets published such as [26] [27] [28]. Several benchmarks were also released on measuring the performance of the machine learning models [29] [27] and even for GNN [30].

### A. Applications with Graph Data

Normally, a graph data structure consists of a collection of paths or edges, and a collection of nodes or vertices. There are lots of practical applications of graph data in real life, such as social network graphs from Facebook or WeChat, knowledge graph from Wikipedia or Google search engine, recommendation engines from online Ads or E-Commerce, communication networks, and the path optimization from map/route planning or flight planning etc. To implement the application, we need to build a mathematics model to represent the application with a graph data structure, and then build a processing model to analyze the graph data. The datasets discussed in II-B will be the representations of the graph data.

### B. Datasets

Datasets play the key role in deep learning, such as the ImageNet in image analytics area. It doesn't only provide the

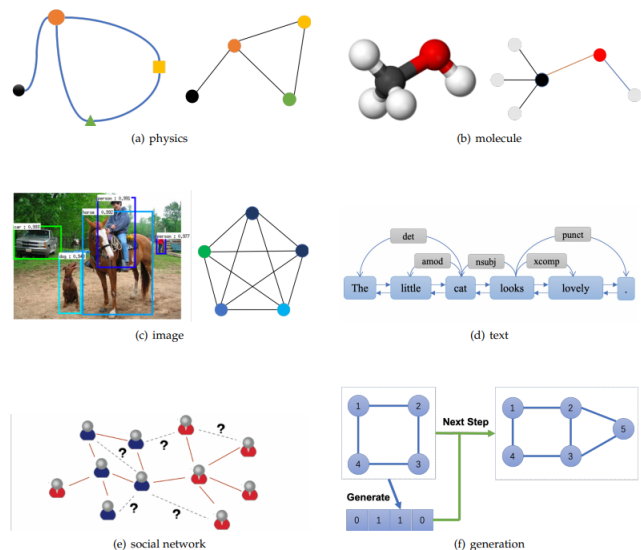


Fig. 2. The Graph Applications [31]

large scale data for training and evaluation, but also build a baseline for comparing the performance of different research models. As for the graph dataset, although there are lots of applications building with graph data, however, it is hard to find some well-defined and labeled large scale datasets in public domain due to the workload and privacy issues. After reviewed dozens of research papers, we list some commonly datasets from the public domain, which were also used as the baseline in most of the papers.

The Cora dataset [32] consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. The CiteSeer dataset [32] consists of 3312 scientific publications classified into one of six classes. The citation network consists of 4732 links. The Pubmed Diabetes dataset [32] consists of 19717 scientific publications from PubMed database pertaining to diabetes classified into one of three classes.

Stanford Large Network Dataset Collection or SNAP [26] collected dozens of large categories of graph datasets, including social networks, networks with ground-truth communities, communication networks, citation networks, collaboration networks, web graph, product co-purchasing networks, Internet peer-to-peer networks, autonomous systems graphs, signed networks, location-based online social networks, Wikipedia networks, temporal networks, user actions, Memetracker and Twitter, online communities, online reviews, face-to-face communication networks and graph classification datasets etc. There are also different network or graph types in SNAP: directed, undirected, bipartite, multigraph, temporal, labeled etc.

TUDatasets [27] is a collection of benchmark datasets for graph classification and regression. It includes some categories such as small molecules, bioinformatics, computer vision, social networks and synthetic.

GraphChallenge [28] encourages community approaches to developing new solutions for analyzing graphs and sparse data derived from social media, sensor feeds, and scientific

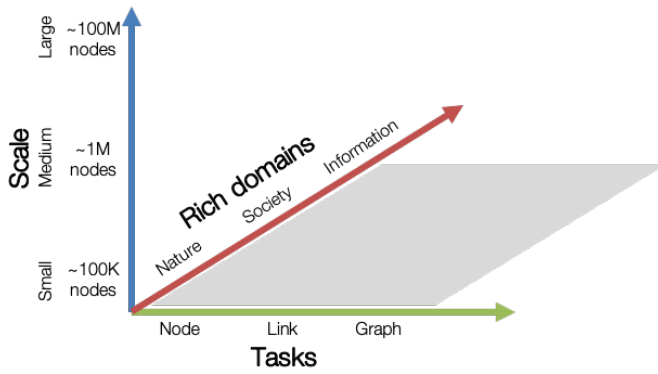


Fig. 3. Open Graph Benchmark [29]

data to enable relationships between events to be discovered as they unfold in the field. In synthetic sparse deep neural network data for the official 2019 challenge of GraphChallenge [28], the datasets were created using RadiX-Net with varying number of neurons and layers, from 1024 neurons to 65536 Neurons.

### C. Benchmark

The Open Graph Benchmark (OGB) [29] is a collection of realistic, large-scale, and diverse benchmark datasets for machine learning on graphs as shown in Figure 3. OGB provide not only realistic benchmark datasets, but also fully automates dataset processing with OGB data loaders that automatically download and process. The most important feature of OGB is, OGB provides standardized dataset splits and evaluators that allow for easy and reliable comparison of different models in a unified manner. OGB uses leaderboards to keep track of the state-of-the-art [29] as shown in Table I.

Not only a collection of benchmark datasets, TUDatasets [27] also offer baseline implementations of standard graph kernels/graph neural network architecture, and standard evaluation procedures. In [30], it introduces a reproducible GNN benchmarking framework, with the facility for researchers to add new models conveniently for arbitrary datasets.

## III. MODELS AND FRAMEWORKS

Since the first release of GNN model [5], there are huge amount of GNN models published in different domains. In [31], it reviewed variants of graph neural networks from different perspectives, such as graph types in Figure 4, graph training methods in Figure 5, or even more details in the propagation step of training as shown in Figure 6. Although there are lots of variants of GNN models released, they still follow the same structure of algorithms, as shown in Figure 7. The only differences among them are aggregation and update functions.

### A. GCN Models

Similar to the CNN models in DNN models, as discussed in I-B, most of GNN models published in recent years are

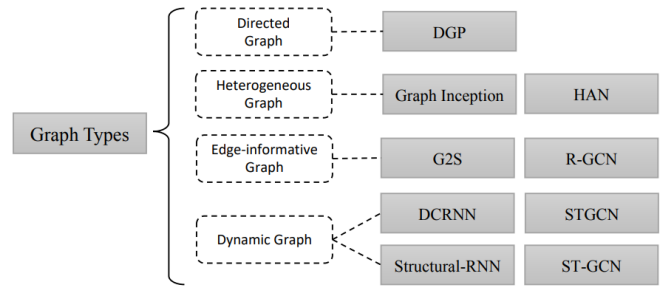


Fig. 4. The Graph Types [31]

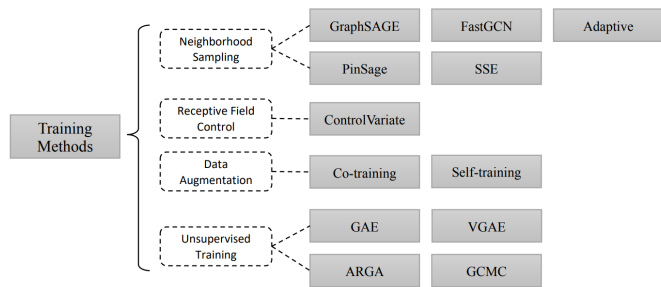


Fig. 5. The Graph Training Methods [31]

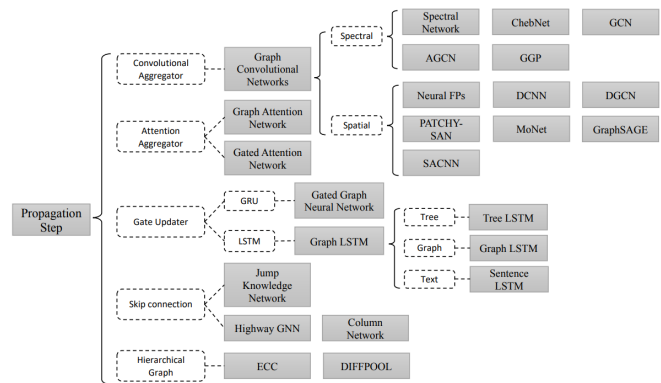


Fig. 6. The Propagation Steps [31]

### Message-Passing GNN

**Input:** a graph  $G = (V, E, X)$ ; GNN functions  $f_{update}$  and  $f_{aggregate}$ .

- 1  $h_v^{(0)} \leftarrow X_v \ (\forall v \in V)$
- 2 for  $i = 1, 2, \dots, L$ 
  - $h_v^{(i)} \leftarrow f_{update}(h_v^{(i-1)}, f_{aggregate}(\{h_w^{(i-1)} \mid w \in N_G(v)\})) \ (\forall v \in V)$
- 3 return  $h_v^{(n)} \ (\forall v \in V)$

Fig. 7. The Common Algorithm of GNN Models

TABLE I  
LEADERBOARD FOR OGBG-MOLHIV CAPTURED AT 04/06/2021 [29]

Rank	Method	Test ROC-AUC	Validation ROC-AUC	Contact	#Params	Date
1	Neural FingerPrints	0.8232 ± 0.0047	0.8331 ± 0.0054	Shanzhuo Zhang (PaddleHelix & PGL)	2,425,102	Mar 15, 2021
2	MorganFP+Rand. Forest	0.8060 ± 0.0010	0.8420 ± 0.0030	Cyrus Maher	230,000	Sep 21, 2020
3	P-WL	0.8039 ± 0.0040	0.8279 ± 0.0059	Daniel Marcos Mendoza	4,600,000	Mar 29, 2021
4	DGN	0.7970 ± 0.0097	0.8470 ± 0.0047	Saro Passaro	114,065	Nov 20, 2020
5	DeeperGCN+FLAG	0.7942 ± 0.0120	0.8425 ± 0.0061	Kezhi Kong	531,976	Oct 20, 2020
6	PNA	0.7905 ± 0.0132	0.8519 ± 0.0099	Gabriele Corso	326,081	Nov 25, 2020
7	GCN+GraphNorm	0.7883 ± 0.0100	0.7904 ± 0.0115	Shengjie Luo	526,201	Sep 16, 2020
8	HIMP	0.7880 ± 0.0082	N/A	Matthias Fey	153,029	Jun 22, 2020
9	DeeperGCN	0.7858 ± 0.0117	0.8427 ± 0.0063	Guohao Li - DeepGCNs.org	531,976	Jun 16, 2020
10	EGC-M (No Edge Features)	0.7818 ± 0.0153	0.8396 ± 0.0097	Shyam Tailor	317,265	Apr 6, 2021

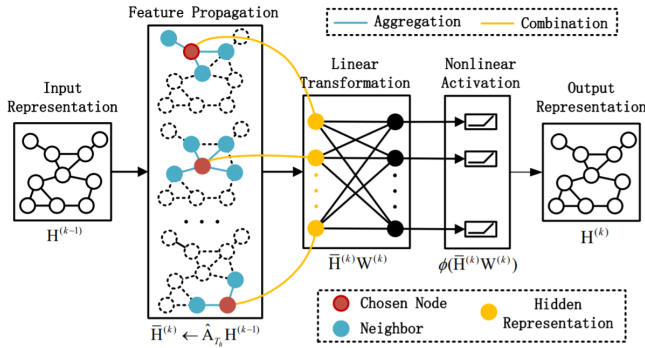


Fig. 8. Diagram of GNN Model [33]

GCN models. Figure 8 shows a simple structure of GCN model, in which Linear Transformation and Nonlinear Activation are same as the counterparts in CNN, and the main difference between GCN and the standard multi-layer perceptron (MLP) lies in the feature propagation [33]. Most of the GCN models use graph Laplacian matrix as the fundamental data structure.

$$y_{output} = \sigma \left( \sum_{j=0}^{K-1} \alpha_j L^j x \right) \quad (1)$$

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(l)} W^{(l)}) \quad (2)$$

$$\begin{aligned} z_i^{(l)} &= W^{(l)} h_i^{(l)} \\ e_{ij}^{(l)} &= \text{LeakyReLU}(\tilde{a}^{(l)T} (z_i^{(l)} || z_j^{(l)})) \\ \alpha_{ij}^{(l)} &= \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})} \\ h_i^{(l+1)} &= \sigma \left( \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} z_j^{(l)} \right) \end{aligned} \quad (3)$$

[14] presents a formulation of CNNs in the context of spectral graph theory, which provides the necessary mathematical background and efficient numerical schemes to design fast localized convolutional filters on graphs, as shown in Equation (1).

Equation (2) shows the formula to update the hidden layer in GCN [10], in which \$H^{(l)}\$ denotes the \$l^{th}\$ layer in the network,

\$\sigma\$ is the non-linearity, and \$W\$ is the weight matrix for this layer. \$D\$ and \$A\$ represent degree matrix and adjacency matrix, respectively.

GAT [34] introduces the attention mechanism as a substitute for the statically normalized convolution operation. Equation (3) shows the procedure to compute the node embedding \$h\_i^{(l+1)}\$ of layer \$l + 1\$ from the embedding of layer \$l\$.

### B. Frameworks Support GNN

TensorFlow [35] is the most widely used neural network framework, and TensorFlow Graphics aims at making useful graphics functions widely accessible to the community by providing a set of differentiable graphics layers (e.g. cameras, reflectance models, mesh convolutions) and 3D viewer functionalities (e.g. 3D TensorBoard) that can be used in your machine learning models of choice.

PyTorch Geometric (PyG) [36] is a geometric deep learning extension library for PyTorch. It consists of various methods for deep learning on graphs and other irregular structures, also known as geometric deep learning, from a variety of published papers.

DGL [37] is an easy-to-use, high performance and scalable Python package for deep learning on graphs. DGL is framework agnostic, meaning if a deep graph model is a component of an end-to-end application, the rest of the logics can be implemented in any major frameworks, such as PyTorch, Apache MXNet or TensorFlow as shown in Figure 9.

There are some other graph deep learning libraries other than these three main popular ones introduced here. StellarGraph [38] is a Python library for machine learning on graphs and networks. The StellarGraph library offers state-of-the-art algorithms for graph machine learning, making it easy to discover patterns and answer questions about graph-structured data. Spektral [39] is a Python library for graph deep learning, based on the Keras API and TensorFlow 2. The main goal of Spektral [39] project is to provide a simple but flexible framework for creating graph neural networks (GNNs). Euler [40] is a distributed graph deep learning framework released by Alibaba Open Source.

## IV. FUTURE RESEARCH TOPICS

Although the researches on GNNs have made great progress in recent years, GNNs still face the more challenges in theoretical research and applications. The limitations of GNNs

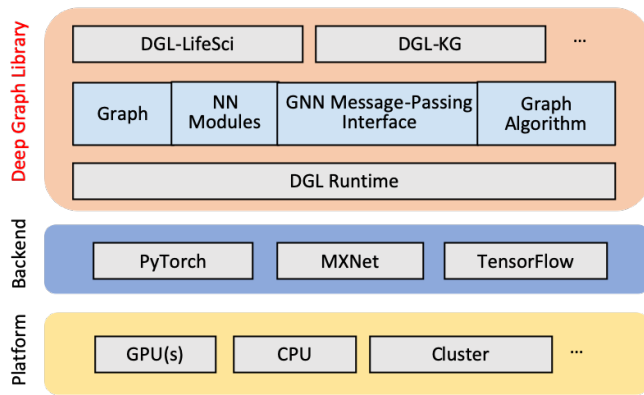


Fig. 9. DGL Overall Architecture [37]

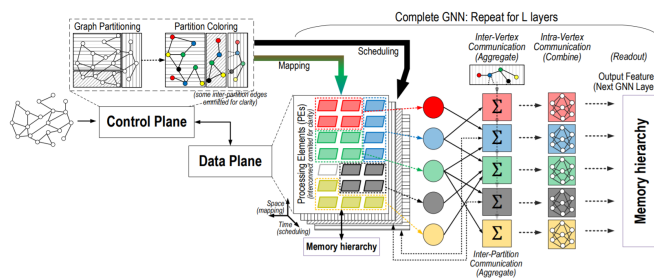


Fig. 10. Architectural vision for GNN accelerators with hardware-software co-design [47]

include the so-called “bottleneck” issue [41], problems with over-smoothing [42], and theoretical limits in terms of representational capacity [43] [44] [45]. With the vanishing gradients issue, most state-of-the-art GCN models are no deeper than 3 or 4 layers, and in [46], it investigated how to bring proven useful concepts (residual connections, dense connections and dilated convolutions) from CNNs to GCNs, and makes GCNs deeper or wider to get better performance, however, investigations on deep architectures in detail are still needed due to the computational constraints. In [25], it suggests four future directions of GNNs: model depth, scalability trade-off, heterogeneity and dynamicity. How to accelerate the performance of GNNs is another hot topic, and there is an architecture of GNN accelerators with hardware-software co-design proposed in [47]. There are more challenges in the heterogeneous network representation learning. [48] provides a unified framework to deeply summarize and evaluate existing research on heterogeneous network embedding (HNE). In this area, however, there are still some limitations of HNE: like when and why do modern HNE algorithms work better compared with traditional network mining approaches is still an open problem. Some other open problems in GNNs, such as robustness, interpretability, graph pre-training and complex graph structures were also discussed in [31] in details.

## V. CONCLUSION

In this paper, we reviewed the main applications modeled with graph data and processed with graph neural networks. The main popular models/frameworks, datasets and benchmarks

are discussed in details. We also provided some challenges and open problems in GNNs, which will guide the future research directions.

## REFERENCES

- [1] D. R. Nagaraju Sonti, “A systematic review of graph signal processing,” *International Journal of Advanced Science and Technology*, vol. 29, no. 05, pp. 862 – 868, Apr. 2020. [Online]. Available: <http://sersc.org/journals/index.php/IJAST/article/view/9622>
- [2] L. Stankovic, D. Mandic, M. Dakovic, B. Scalzo Dees, M. Brajović, E. Sejdic, and A. Constantinides, “Vertex-frequency graph signal processing: A comprehensive review,” *Digital Signal Processing*, vol. 107, p. 102802, 08 2020.
- [3] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE Signal Processing Magazine*, vol. 30, no. 3, p. 83–98, May 2013. [Online]. Available: <http://dx.doi.org/10.1109/MSP.2012.2235192>
- [4] A. Ortega, P. Frossard, J. Kovačević, J. M. F. Moura, and P. Vandergheynst, “Graph signal processing: Overview, challenges, and applications,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.
- [5] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, “The graph neural network model,” *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.
- [6] M. Kampffmeyer, Y. Chen, X. Liang, H. Wang, Y. Zhang, and E. P. Xing, “Rethinking knowledge graph propagation for zero-shot learning,” *CoRR*, vol. abs/1805.11724, 2018. [Online]. Available: <http://arxiv.org/abs/1805.11724>
- [7] Y. Zhang, Y. Xiong, X. Kong, S. Li, J. Mi, and Y. Zhu, “Deep collective classification in heterogeneous information networks,” in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW ’18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 399–408. [Online]. Available: <https://doi.org/10.1145/3178876.3186106>
- [8] D. Beck, G. Haffari, and T. Cohn, “Graph-to-sequence learning using gated graph neural networks,” *CoRR*, vol. abs/1806.09835, 2018. [Online]. Available: <http://arxiv.org/abs/1806.09835>
- [9] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” 2017.
- [10] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: <http://arxiv.org/abs/1609.02907>
- [11] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, “Structural-rnn: Deep learning on spatio-temporal graphs,” *CoRR*, vol. abs/1511.05298, 2015. [Online]. Available: <http://arxiv.org/abs/1511.05298>
- [12] S. Gadgil, Q. Zhao, A. Pfefferbaum, E. V. Sullivan, E. Adeli, and K. M. Pohl, “Spatio-temporal graph convolution for functional mri analysis,” 2020.
- [13] D. Gaiotto, G. W. Moore, and A. Neitzke, “Spectral networks,” *Annales Henri Poincaré*, vol. 14, no. 7, p. 1643–1731, Mar 2013. [Online]. Available: <http://dx.doi.org/10.1007/s00023-013-0239-7>
- [14] M. Defferrard, X. Bresson, and P. Vandergheynst, “Convolutional neural networks on graphs with fast localized spectral filtering,” in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: <https://proceedings.neurips.cc/paper/2016/file/04df4d434d481c5bb723be1b6df1ee65-Paper.pdf>
- [15] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, “Convolutional networks on graphs for learning molecular fingerprints,” in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015. [Online]. Available: <https://proceedings.neurips.cc/paper/2015/file/f9be311e65d81a9ad8150a60844bb94c-Paper.pdf>
- [16] J. Atwood and D. Towsley, “Diffusion-convolutional neural networks,” in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016. [Online]. Available: <https://proceedings.neurips.cc/paper/2016/file/390e982518a50e280d8e2b535462ec1f-Paper.pdf>
- [17] Z. Tong, Y. Liang, C. Sun, D. S. Rosenblum, and A. Lim, “Directed graph convolutional network,” 2020.

- [18] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *CoRR*, vol. abs/1706.02216, 2017. [Online]. Available: <http://arxiv.org/abs/1706.02216>
- [19] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2017.
- [20] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *CoRR*, vol. abs/1503.00075, 2015. [Online]. Available: <http://arxiv.org/abs/1503.00075>
- [21] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph LSTM," *CoRR*, vol. abs/1603.07063, 2016. [Online]. Available: <http://arxiv.org/abs/1603.07063>
- [22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018.
- [23] X. Xin, A. Karatzoglou, I. Arapakis, and J. M. Jose, "Graph highway networks," 2020.
- [24] F. Gama, J. Bruna, and A. Ribeiro, "Stability properties of graph neural networks," *IEEE Transactions on Signal Processing*, vol. 68, p. 5680–5695, 2020. [Online]. Available: <http://dx.doi.org/10.1109/TSP.2020.3026980>
- [25] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *CoRR*, vol. abs/1901.00596, 2019. [Online]. Available: <http://arxiv.org/abs/1901.00596>
- [26] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," <http://snap.stanford.edu/data>, Jun. 2014.
- [27] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, "Tudataset: A collection of benchmark datasets for learning with graphs," in *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. [Online]. Available: [www.graphlearning.io](http://www.graphlearning.io)
- [28] S. Samsi, J. Kepner, V. Gadepally, M. Hurley, M. Jones, E. Kao, S. Mohindra, A. Reuther, S. Smith, W. Song, and et al., "Graphchallenge.org triangle counting performance," *2020 IEEE High Performance Extreme Computing Conference (HPEC)*, Sep 2020. [Online]. Available: <http://dx.doi.org/10.1109/HPEC43674.2020.9286166>
- [29] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec, "Open graph benchmark: Datasets for machine learning on graphs," *arXiv preprint arXiv:2005.00687*, 2020.
- [30] V. P. Dwivedi, C. K. Joshi, T. Laurent, Y. Bengio, and X. Bresson, "Benchmarking graph neural networks," 2020.
- [31] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, and M. Sun, "Graph neural networks: A review of methods and applications," *CoRR*, vol. abs/1812.08434, 2018. [Online]. Available: <http://arxiv.org/abs/1812.08434>
- [32] R. A. Rossi and N. K. Ahmed, "The network data repository with interactive graph analytics and visualization," in *AAAI*, 2015. [Online]. Available: <http://networkrepository.com>
- [33] W. Yan, D. Jin, Z. Lin, and F. Yin, "Graph neural network for large-scale network localization," 2021.
- [34] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018.
- [35] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: <https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf>
- [36] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.
- [37] M. Wang, D. Zheng, Z. Ye, Q. Gan, M. Li, X. Song, J. Zhou, C. Ma, L. Yu, Y. Gai, T. Xiao, T. He, G. Karypis, J. Li, and Z. Zhang, "Deep graph library: A graph-centric, highly-performant package for graph neural networks," *arXiv preprint arXiv:1909.01315*, 2019.
- [38] C. Data61, "Stellargraph machine learning library," <https://github.com/stellargraph/stellargraph>, 2018.
- [39] D. Grattarola and C. Alippi, "Graph neural networks in tensorflow and keras with spektral," 2020.
- [40] A. O. Source, "Euler, a distributed graph deep learning framework," <https://github.com/alibaba/euler>, 2020.
- [41] U. Alon and E. Yahav, "On the bottleneck of graph neural networks and its practical implications," 2021.
- [42] Q. Li, Z. Han, and X. Wu, "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), New Orleans, Louisiana, USA, February 2-7, 2018*, S. A. McIlraith and K. Q. Weinberger, Eds. AAAI Press, 2018, pp. 3538–3545. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16098>
- [43] R. Sato, "A survey on the expressive power of graph neural networks," 2020.
- [44] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" 2019.
- [45] V. K. Garg, S. Jegelka, and T. Jaakkola, "Generalization and representational limits of graph neural networks," 2020.
- [46] G. Li, M. Müller, A. Thabet, and B. Ghanem, "Deepgens: Can gcn as deep as cnns?" 2019.
- [47] S. Abadal, A. Jain, R. Guirado, J. López-Alonso, and E. Alarcón, "Computing graph neural networks: A survey from algorithms to accelerators," 2020.
- [48] C. Yang, Y. Xiao, Y. Zhang, Y. Sun, and J. Han, "Heterogeneous network representation learning: A unified framework with survey and benchmark," *TKDE*, 2020.

## Creative Commons Attribution License 4.0 (Attribution 4.0 International, CC BY 4.0)

This article is published under the terms of the Creative Commons Attribution License 4.0  
[https://creativecommons.org/licenses/by/4.0/deed.en\\_US](https://creativecommons.org/licenses/by/4.0/deed.en_US)