A Review of Uncertainty in Graph Neural Networks

Marco Hövekenmeier [1]

Abstract

Recent advances in machine learning research for graph data consider uncertainties inherent in the graph topology and node features to increase the accuracy in the predictions. Other methods have been developed that further quantify the uncertainties in the model's predictions. This paper gives a conceptual introduction to uncertainties in neural networks and to graph neural networks. Moreover, it presents and discusses the state-of-the-art methods dealing with uncertainty in the graph domain and potential future research directions in this field are shown.

1. Introduction

For the use of machine learning methods for safety-critical, high-risk applications it is indispensable to use reliable, i.e. robust and explainable models. For this purpose, an uncertainty quantification (UQ) of the model's predictions is a very important tool [30]. This is mainly due to the fact that traditional models are often *uncalibrated*, i.e. they are very confident in their outputs even if the predictions are wrong, which can have fatal consequences [11]. For this reason, more sophisticated, Bayesian extensions of neural networks which allow for UQ have been developed in the past. However, these models for euclidean i.i.d. data are not directly transferable to graph neural networks (GNNs) for the domain of interdependent graph data. In spite that this data is used in critical application fields such as medicine or finance, UQ is urgently needed here as well [8]. Therefore, in this paper first the theoretical background of techniques for predictive UQ in general and the basics of GNNs are introduced in Section 2. The subsequent section summarizes the current state of the art of uncertainty in GNNs. In Section 4, the mentioned techniques are discussed as well as criticized and an outlook for further research in UQ for GNNs is given in Section 5.

[1]Department of Computer Science, Technical University Munich, Munich, Germany. Correspondence to: Marco Hövekenmeier <hoeveken@in.tum.de>.

2. Background

This section presents the theoretical background of UQ in NNs and introduces the basic concepts of GNNs.

2.1. Uncertainty in Neural Networks

Types of Uncertainty. In Machine Learning research two main types of uncertainty are distinguished: *aleatoric* and *epistemic* (see Figure 1)[16]. The former refers to uncertainties resulting from inherent randomness in the data, e.g. noise. Here, we can further distinguish between homoscedastic - i.e. uncertainties that exist equally over the entire dataset - and heteroscedastic - i.e. uncertainties that exist only for specific input data. Epistemic uncertainty, on the other hand, describes systematic uncertainties that arise from a lack of knowledge (here data) and are thus expressed in the learned model. In general, epistemic uncertainties can be reduced by collecting more data, but aleatoric uncertainties cannot [18].

Bayesian Techniques. While standard training of neural networks with regularization can be viewed as maximum-a-posteriori point estimates of the model parameters, these models can not provide information about the reliability of their predictions [1]. To alleviate this issue, Bayesian extensions aim to model the weights θ with a probability distribution and marginalize over these weights to get the so called posterior predictive distribution [1]

$$p\left(y^* \mid x^*, X, Y\right) = \int p\left(y^* \mid x^*, \theta\right) p(\theta \mid X, Y) d\theta. \quad (1)$$

Here, X and Y denote the training data and ground-truth and x^* and y^* are a new data sample and its predicted value at inference. Since computing the true predictive distribution is intractable, approximations are used. One commonly applied technique is Monte Carlo sampling or dropout, where over multiple forward passes either the model weights are sampled according to their learned distribution or hidden units are randomly dropped. Afterwards, first and second order statistics are computed over the results of each of the S inference passes [1, 9, 35]:

$$p\left(y^* \mid x^*, X, Y\right) \approx \frac{1}{S} \sum_{i=1}^{S} p\left(y^* \mid x^*, \theta_i\right). \quad (2)$$

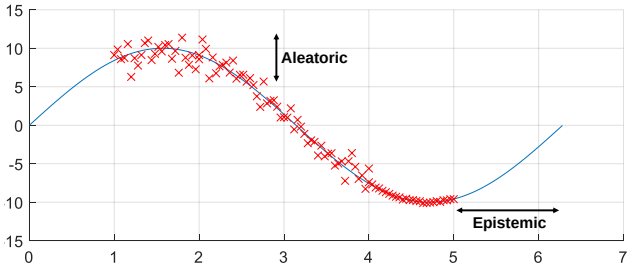

Figure 1. Schematic illustration of (heteroscedastic) aleatoric and epistemic uncertainty. Red crosses indicate data samples from the ground truth in blue. The aleatoric uncertainty decreases with increasing values on the x-axis, whereas epistemic uncertainty arises in the regions without data samples. Adapted from [1]

Another option is to use Variational Inference (VI) where the true predictive distribution is approximated with a variational distribution and the KL divergence between the true and approximated distribution is minimized [1, 2, 19].

Deep Ensembles. A method that currently outperforms Bayesian methods in UQ is to train an ensemble of multiple models with different initializations and to average the predicted likelihoods by multiple forward passes through these models [24, 31, 1, 20]. The resulting predictive distribution follows the form of Eq. 2 and corresponds to averaging the predicted probabilities for classification and a mixture of Gaussians for regression [20].

Gaussian Processes. The previous methods rely on averaging results over multiple forward passes. A method for deterministic UQ with a single forward pass is Gaussian Processes (GPs). These are non-parametric Bayesian models that serve as a prior distribution over the latent function $f(x)$ that is responsible for generating the training data. In GPs the similarity between samples x and x' is calculated using a kernel or covariance function $k_\theta\left(x, x'\right)$ and a mean function $m(x)$ [27, 1]:

$$f(x) \sim \mathcal{GP}\left(m(x), k_\theta\left(x, x'\right)\right) \tag{3}$$

Computational limitations and a lack of expressiveness exist in the basic algorithm. This is due to the inversion of the kernel matrix, whose complexity scales cubically with the number of training samples, and no possibility for hierarchical feature extraction. This led to the introduction of Deep Gaussian Processes (DGPs) or Deep Kernel Learning (DKL) and its approximations [15, 7]. In DKL, the expressiveness of deep neural networks is combined with the probabilistic prediction ability of GPs by using the output of a deep feature extractor $g_\omega(x)$ parameterized by ω as input for the kernel $k_\theta\left(g_\omega(x), g_\omega(x')\right)$ [32]. Moreover, in sparse GPs only a subset of the training data, so-called *inducing points*, is used for the kernel function to reduce complexity.

Metrics for Uncertainty Quantification. In order to compare the quality of UQ for different models, different metrics have been developed. The easiest metric to use for a classification task is the predictive entropy measured on the output layer of a network. A more advanced metric often used is the Expected Calibration Error (ECE) [23]. ECE is a scalar summary statistic of calibration. It measures the difference in expectation between a model's confidence and accuracy of its n outputs divided into M equally sized bins B_m [11]:

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\mathrm{acc}\left(B_m\right) - \mathrm{conf}\left(B_m\right)\right|. \tag{4}$$

Newer extensions of this metric are e.g. the adversarial group calibration, where the ECE is computed for different groups of the input data to get worst-case results [14, 36]. Furthermore, it is possible to use Proper Scoring Rules as a metric. These have the additional advantage that they can be used directly as a loss for training the model [10].

2.2. Graph Neural Networks

We consider data in form of a homogeneous, directed and attributed graph consisting of a set of nodes and edges $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with adjacency matrix $A \in \{0,1\}^{N\times N}$ and node attribute matrix $X \in \mathbb{R}^{N\times D}$ with $N = |\mathcal{V}|$ and D the dimension of the node feature space [30, 34, 38]. Tasks in the graph domain can be divided in node-level (node classification, regression, clustering), edge-level (edge classification) or graph-level (graph classification, regression, matching). The most common use-case is the semi-supervised node classification setting, where only a small amount of labeled nodes is given as ground truth and the labels of the missing nodes need to be predicted.

One commonly used baseline to tackle these tasks is the Graph Convolutional Network (GCN) [17]. The message passing step on graph-level can be described for each layer l as

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right). \tag{5}$$

Here $\tilde{A} = A + I_N$ denotes the self-connection-added adjacency matrix and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$. $\sigma(\cdot)$ stands for the activation function, $W^{(l)}$ is the trainable weight matrix and $H^{(l)} \in \mathbb{R}^{N\times D}$ is the activation of the l-th layer.

Common problems in GNNs include overfitting and oversmoothing, computational challenges for large, dense graphs [5] and only the possibility of transductive learning [12]. These issues have been tackled in a series of extensions to the vanilla GCN [5, 28, 12, 39, 33].

Aleatoric uncertainties in GNNs arise from uncertain graph edges and node features in the training and test data. Epistemic uncertainties in GNNs arise, when the input data is out of distribution (OOD) compared to the training data.

This can e.g. be due to adversarial attacks where edges and features are permuted.

3. Methods for Uncertainty Quantification in Graph Neural Networks

A variety of methods exist that model uncertainties in the node features and edges [35, 26]. Furthermore, some methods make use of Graph Gaussian Processes (GGPs) to model these inherent uncertainties [21, 3, 25]. Other methods use stochastic sampling or aggregation, for example by injecting a learnable noise into the aggregation process [13, 34]. However, these models do not aim to quantify the predictions with an uncertainty value, but only increase the accuracy itself by tackling the oversmoothing and overfitting issues of GNNs mentioned in Section 2.2 [30]. Only few works by now have tackled the task of predictive uncertainty estimation on semi-supervised node classification for graphs [30, 37]. Therefore, this section is divided into methods that account for uncertainty in the graph topology, the features and aggregation process as well as methods that tackle the predictive UQ task.

3.1. Uncertainty in the Graph Structure and Aggregation Process

Bayesian Graph Convolutional Network

One of the first approaches to account for uncertainty in GNNs is presented in [35]. Here a Bayesian framework is introduced, where the observed graph is viewed as a realization from a parametric family of random graphs to consider the uncertainty in the graph topology, i.e. the edges. Inference of the posterior of the random graph parameters and the node or graph labels are then targeted jointly. An extension of this idea is developed in [26] where through node copying, additional information is provided by the node features and training labels at inference.

Graph Gaussian Processes

Another way of accounting for uncertainty in the graph representation is the adaption of Gaussian Processes to the graph domain: Graph Gaussian Processes as first developed by [25]. This concept introduces another latent function with a GP prior and the labeled nodes are used as inducing points in the semi-supervised classification setting. In [3] the nodes are embedded as Gaussian distributions, which makes it possible to capture the uncertainty in their features. In [21] the GGP concept is extended further by considering the graph structure uncertainty to boost the accuracy of the classification.

Graph DropConnect (GDC)

Another option to account for uncertainty in the graph structure is shown in [13]. Here, GDC is introduced as method to further generalize DropEdge [28]. The operation of a GNN layer with GDC is defined as:

$$\mathbf{H}^{(l+1)}[:,j] = \sigma\left(\sum_{i=1}^{f_l} \mathfrak{N}\left(\mathbf{A} \odot \mathbf{Z}_{i,j}^{(l)}\right) \mathbf{H}^{(l)}[:,i]\mathbf{W}^{(l)}[i,j]\right) \quad \text{for } j = 1,\dots,f_{l+1} \tag{6}$$

Here, the normalizing operator $\mathfrak{N}(\mathbf{A}) = \mathbf{I}_N + \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ is introduced for abbreviation and f_l accounts for the number of features in the l-th layer. By introducing the additional free parameters $\left\{\mathbf{Z}_{i,j}^{(l)} \in \{0,1\}^{n\times n}\right\}_{i=1}^{f_l}$ that adjust the binary mask for edges, nodes and channels, respectively, a learnable, adaptive connection sampling is achieved. The joint training of the random masks and corresponding drop rates together with the GNN weights is done with Variational Inference. $\mathbf{Z}_{i,j}^{(l)}$ can also be set as hyperparameter. Therefore GDC generalizes Dropout [29], DropEdge [28] and FastGCN [5].

Stochastic Aggregation (STAG)

A further improvement of GDC is called STAG [34]. Here, the parameter $\mathbf{Z}_{i,j}^{(l)}$ in Eq. 6 is not restricted to be a binary mask on edges, nodes and channels, respectively, but each element can be drawn from an arbitrary distribution, e.g. a Gaussian with mean μ_Z and variance σ_Z:

$$\mathbf{Z}_{i,j}^{(l)} \sim \mathcal{N}\left(\mu_Z, \sigma_Z\right)^{n\times n}. \tag{7}$$

This process can be viewed as stochastic noise injection to the aggregation process, which increases the expressive power of the network. Similar to GDC, those parameters can also be trained jointly together with the model weights using Variational Inference.

Uncertainty-aware Attention GNN (UAG)

In UAG, data and model uncertainty are explicitly separated from each other [8]. To evaluate the model uncertainty, a similar approach as in GDC is used. The variance in predictions of different nodes is calculated via Monte Carlo integration of multiple forward passes with randomly dropped GNN layers, channels, nodes, and edges. The data uncertainty is measured in terms of node diversity $Div_{node}^{(k)}$, which is the number of different labels in the node's k-hop neighbourhood. For node classification, an uncertainty aware attention mechanism is introduced, that adaptively adjusts the edge attention during inference depending on the model and data uncertainty assigned to each node.

3.2. Predictive Uncertainty Quantification in GNNs

Graph-based Kernel Dirichlet distribution Estimation

As mentioned earlier, only few works tackle the predictive UQ task directly. In [37], a Dirichlet conjugate prior is

parameterized and node-level Dirichlet distributions are directly predicted. The method also includes dropout as well as a teacher network for boosting the performance. Moreover, further uncertainty estimates from the evidence/belief theory domain are included.

Graph Posterior Network (GPN)

In [30], the concept of a posterior network introduced in [4] is brought to the graph domain. In this work, a distinction is made between predictions with and without network effects. The latter recovers the intuition about uncertainty for i.i.d. data by only giving weight to the node features. The prediction with network effects also includes the neighbourhood aggregation. Similar to [37] the Dirichlet posterior distribution of node v

$$\boldsymbol{p}^{(v)} \sim \text{Dir}\left(\alpha^{\text{prior}} + \beta^{(v)}\right) \tag{8}$$

is formed. The parameters α^{prior} and $\beta^{(v)}$ are the prior and input-dependent, learned class pseudo-counts. For prediction with network effects, Personalized Page Rank message passing is used to form the aggregated class pseudo counts $\beta_c^{\text{agg},(v)}$ from the feature class pseudo-counts $\beta_c^{\text{ft},(v)}$. Moreover it is shown that GPN fulfills theoretical guarantees, whose axioms are also proposed in the paper.

4. Discussion

When comparing models for UQ, there are two main points that need to be investigated. One is the quality of the predictions measured by the target metric, such as the accuracy for node classification and also measured by the calibration such as the ECE introduced in Section 2.1. Often the UQ is also used to detect OOD data. On the other hand, the computational complexity at inference time is crucial for an adaption for real-life use cases.

Gaussian Processes in general are suitable models that can inherently account for uncertainty. Therefore, compared to ensemble or other approximate Bayesian methods, only one forward pass is needed. The implementation of inducing points for approximate GPs further reduces the computational complexity compared to vanilla GPs. However, as stated in [30], scaling to large graphs is still difficult. Moreover, the presented methods do not capture the uncertainty in their predictions, and therefore do not unfold the full potential of GGPs.

The methods GDC, STAG and UAG show promising results in terms of accuracy on node-classification. Therefore, introducing new, learnable parameters that act on the adjacency matrix and can be jointly optimized together with the model weights seem like a valuable extension. However, in this way an overhead in terms of computation time and needed memory is brought to the optimization. In GDC, this is circumvented by drawing a mask for a block of features, which show that taking advantage of the full flexibility of the proposed approaches is not yet feasible. Furthermore, these papers except for GDC do not quantify the uncertainty on the predictions, even though it would be possible with the Bayesian interpretation of dropout mentioned in Section 2.1.

The methods shown in Section 3.2 show promising results in terms of predictive UQ. GPN especially motivates the development of the model with derived axioms on the expected behaviour in terms of UQ on graph data. This is due to the fact that no general expectation on the models behaviour has been defined in other works before. Furthermore, GPN outperforms other baselines under node and edge shifts as well as in OOD detection. Compared to [37] an OOD data training set is not needed for GPN which is a big advantage in terms of applicability for real-life use cases. Furthermore, due to the single forward pass uncertainty method, it outperforms ensemble and dropout based methods in terms of inference and training time.

5. Future Research

As already mentioned by Abdar et al. [1], the area of UQ for GNNs is still underexplored. In this section, therefore, I present my thoughts for future research in this area.

First, it is very helpful for a research community to have standardized baselines and benchmarks to quantify the added value of new models and approaches and make meaningful comparisons possible. In [22] and [6] this attempt was made for classification and regression, respectively. However, these baselines are limited to image and text data, and accordingly are not practicable for the use case of GNNs. An extension so that UQ can also be evaluated for graph data seems reasonable.

Moreover, many methods solely capture the uncertainty to increase the accuracy of predictions without quantifying the predictive uncertainty and optimizing for calibration as discussed in Section 4. However, all of those methods are actually capable of doing so. More specifically, Gaussian Processes show good results for UQ for i.i.d. data but are not yet used for this task in the graph domain [32].

Furthermore, currently only very few models aim at rigorously separating epistemic and aleatoric uncertainty in the predictions. This could be particularly exciting in view of the fact that aleatoric uncertainty is immanent in the data and thus represents a kind of supremum for the certainty of a model, which cannot be improved any further. A quantifiable value for epistemic uncertainty, on the other hand, gives insights about the models reliability and can guide research in certain directions, as this type of uncertainty can be further reduced by more data and better models.

References

[1] Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P., Cao, X., Khosravi, A., Acharya, U. R., and et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, Dec 2021. ISSN 1566-2535. doi: 10.1016/j.inffus.2021.05.008.

[2] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518): 859–877, Apr 2017. ISSN 1537-274X. doi: 10.1080/01621459.2017.1285773.

[3] Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *International Conference on Learning Representations*, 2018.

[4] Charpentier, B., Zügner, D., and Günnemann, S. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. In *NeurIPS*, 2020.

[5] Chen, J., Ma, T., and Xiao, C. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*, 2018.

[6] Chung, Y., Char, I., Guo, H., Schneider, J., and Neiswanger, W. Uncertainty toolbox: an open-source library for assessing, visualizing, and improving uncertainty quantification, 2021.

[7] Damianou, A. and Lawrence, N. D. Deep Gaussian processes. In Carvalho, C. M. and Ravikumar, P. (eds.), *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, volume 31 of *Proceedings of Machine Learning Research*, pp. 207–215, Scottsdale, Arizona, USA, 29 Apr–01 May 2013. PMLR.

[8] Feng, B., Wang, Y., and Ding, Y. Uncertainty-aware attention graph neural network for defending adversarial attacks. *ArXiv*, 2021.

[9] Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Balcan, M. F. and Weinberger, K. Q. (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1050–1059, New York, New York, USA, 20–22 Jun 2016. PMLR.

[10] Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

[11] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pp. 1321–1330. JMLR.org, 2017.

[12] Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[13] Hasanzadeh, A., Hajiramezanali, E., Boluki, S., Zhou, M., Duffield, N., Narayanan, K., and Qian, X. Bayesian graph neural networks with adaptive connection sampling. In *ICML 2020: International Conference on Machine Learning*, 2020.

[14] Hebert-Johnson, U., Kim, M., Reingold, O., and Rothblum, G. Multicalibration: Calibration for the (Computationally-identifiable) masses. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1939–1948. PMLR, 10–15 Jul 2018.

[15] Jakkala, K. Deep gaussian processes: A survey. *ArXiv*, 2021.

[16] Kendall, A. and Gal, Y. What uncertainties do we need in bayesian deep learning for computer vision? In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 5580–5590, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[17] Kipf, T. N. and Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[18] Kiureghian, A. D. and Ditlevsen, O. Aleatory or epistemic? does it matter? *Structural Safety*, 31 (2):105–112, 2009. ISSN 0167-4730. doi: https://doi.org/10.1016/j.strusafe.2008.06.020. Risk Acceptance and Risk Communication.

[19] Kullback, S. and Leibler, R. A. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. ISSN 0003-4851. doi: 10.1214/aoms/1177729694.

[20] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[21] Liu, Z.-Y., Li, S.-Y., Chen, S., Hu, Y., and Huang, S.-J. Uncertainty aware graph gaussian process for semi-supervised learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4957–4964, apr 2020. doi: 10.1609/aaai.v34i04.5934.

[22] Nado, Z., Band, N., Collier, M., Djolonga, J., Dusenberry, M. W., Farquhar, S., Filos, A., Havasi, M., Jenatton, R., Jerfel, G., Liu, J., Mariet, Z. E., Nixon, J., Padhy, S., Ren, J., Rudner, T. G. J., Wen, Y., Wenzel, F., Murphy, K., Sculley, D., Lakshminarayanan, B., Snoek, J., Gal, Y., and Tran, D. Uncertainty baselines: Benchmarks for uncertainty & robustness in deep learning. *ArXiv*, 2021.

[23] Naeini, M. P., Cooper, G. F., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2901–2907. AAAI Press, 2015. ISBN 0262511290.

[24] Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[25] Ng, Y. C., Colombo, N., and Silva, R. Bayesian semi-supervised learning with graph gaussian processes. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[26] Pal, S., Regol, F., and Coates, M. Bayesian graph convolutional neural networks using node copying. *ArXiv*, 2019.

[27] Rasmussen, C. E. Gaussian processes in machine learning. In *Advanced Lectures on Machine Learning*, pp. 63–71. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-28650-9_4.

[28] Rong, Y., Huang, W., Xu, T., and Huang, J. The truly deep graph convolutional networks for node classification. *ArXiv*, 2019.

[29] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[30] Stadler, M., Charpentier, B., Geisler, S., Zügner, D., and Günnemann, S. Graph posterior network: Bayesian predictive uncertainty for node classification. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021.

[31] van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. *ArXiv*, 2020.

[32] van Amersfoort, J., Smith, L., Jesson, A., Key, O., and Gal, Y. On feature collapse and deep kernel learning for single forward pass uncertainty. *ArXiv*, 2021.

[33] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[34] Wang, Y. and Karaletsos, T. Stochastic aggregation in graph neural networks. *ArXiv*, 2021.

[35] Zhang, Y., Pal, S., Coates, M., and Üstebay, D. Bayesian graph convolutional neural networks for semi-supervised classification. *ArXiv*, 2018.

[36] Zhao, S., Ma, T., and Ermon, S. Individual calibration with randomized forecasting. *ArXiv*, 2020.

[37] Zhao, X., Chen, F., Hu, S., and Cho, J.-H. Uncertainty aware semi-supervised learning on graph data. *ArXiv*, 2020.

[38] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1: 57–81, 2020. ISSN 2666-6510. doi: https://doi.org/10.1016/j.aiopen.2021.01.001.

[39] Zhu, D., Zhang, Z., Cui, P., and Zhu, W. Robust Graph Convolutional Networks Against Adversarial Attacks. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., and Karypis, G. (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1399–1407, New York, NY, USA, 07252019. ACM. ISBN 9781450362016. doi: 10.1145/3292500.3330851.