A Graph Is More Than Its Nodes: Towards Structured Uncertainty-Aware Learning on Graphs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Current graph neural networks (GNNs) that tackle node classification on graphs tend to only focus on nodewise scores and are solely evaluated by nodewise metrics. This limits uncertainty estimation on graphs since nodewise marginals do not fully characterize the joint distribution given the graph structure. In this work, we propose novel edgewise metrics, namely the edgewise expected calibration error (ECE) and the agree/disagree ECEs, which provide criteria for uncertainty estimation on graphs beyond the nodewise setting. Our experiments demonstrate that the proposed edgewise metrics can complement the nodewise results and yield additional insights. Moreover, we show that GNN models which consider the structured prediction problem on graphs tend to have better uncertainty estimations, which illustrates the benefit of going beyond the nodewise setting.

1 Introduction

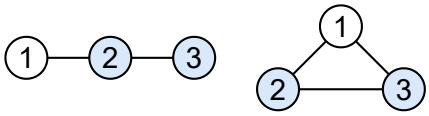
Learning from graph-structured data has gained increasing attention from the deep learning community and numerous graph neural networks (GNNs) have been introduced to cope with this new paradigm [7, 25, 17, 28]. Commonly used GNN models like Graph Convolution Network (GCN) [7] and Graph Attention Network (GAT) [25] produce nodewise scores from which good classification results can be achieved.

On the other hand, learning on graphs poses particular challenges for uncertainty estimation. Given the graph structure, the node predictions are no longer independent samples. Instead, we have a structured prediction problem [15] and should also account for their interdependencies. In other words, high-quality uncertainty estimation for GNNs requires reliable estimation of the joint output distribution. Hence considering only nodewise marginal output distributions is insufficient.

From this structured prediction perspective, we see a considerable limitation of existing works on uncertainty-aware learning for GNNs, namely that they only employ nodewise metrics as evaluation criteria. Given the abundant existing work on uncertainty-aware learning for standard multi-class classification [20, 3, 19], many ideas have been adapted to the GNN case. This includes approaches based on Bayesian formulations [29, 4], evidential learning [30, 22], as well as post-hoc calibration methods [24, 27, 5]. However, for estimating the quality of predictive uncertainty, these approaches directly use the nodewise metrics from the multi-class classification literature, which are intended for i.i.d. test samples and ignore the graph structure.

To tackle this issue, we investigate novel uncertainty estimation metrics that consider the interdependencies in the graph structure. Specifically, we propose the edgewise expected calibration error (ECE) which relies on the graph structure and the edgewise marginal distributions. Furthermore, we formulate agree and disagree ECEs to analyze the agreeing and disagreeing edge marginals separately.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



$(p_1; p_2; p_3)$	Nodewise	Edgewise Chain	Edgewise Cycle
$(0; 1; 1)$	0	0	0
$(2/3; 2/3; 2/3)$	0	$1/18$	$1/9$
$(0.55; 0.8; 0.7)$	0.017	0	0.128

Figure 1: Examples of nodewise and edgewise ECEs for a chain and a cycle. The nodewise ECEs stay the same for both graphs given the same node settings, while the proposed edgewise ECEs can have different values depending on the graph structure. Further details are available in Appendix A.

Our experiments in Section 3 show that the proposed edgewise metrics can complement the nodewise ECE results and offer additional insights.

Equipped with the proposed edgewise metrics, we compare uncertainty estimation of GNN models which simply produce nodewise scores, e.g., GCN and GAT, with models which explicitly tackle the structured prediction problem, like Graph Markov Neural Network (GMNN) [17] and Explicit Pairwise Factorized Graph Neural Network (EPFGNN) [28]. While naive mean field [15] can approximate predictions based on nodewise scores to the joint output distribution, we observe in practice that approaches treating graph learning as structured prediction problems tend to have better uncertainty estimation. This demonstrates the benefit of going beyond the nodewise setting, especially for the uncertainty estimation of GNNs.

2 Uncertainty estimation on graphs beyond nodewise criteria

Given a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ with c output classes for each node, the task of node classification aims at predicting the labels $y_U = \{y_i\}_{i \in U}$ of the test nodes $U \subseteq \mathcal{N}$ given the labels $y_L = \{y_i\}_{i \in L}$ of the training nodes $L = \mathcal{N} \setminus U$ and the input features $x = \{x_i\}_{i \in \mathcal{N}}$. Ideal uncertainty estimation is achieved when the predicted joint distribution $\hat{p}(y_U | x, y_L)$ matches the true joint distribution $p(y_U | x, y_L)$. However, this condition cannot be verified in practice, for the following two reasons:

1. It is impossible to estimate $p(y_U | x, y_L)$ with a single set of ground-truth labels y_U^*;
2. The joint distributions are intractable given their exponential size w.r.t. the node count.

Instead, one can consider weaker but more tractable alternatives based on marginal distributions. Existing works [24, 27, 5] consider the confidence calibration of nodewise marginal distributions and measure the quality of uncertainty estimation using their expected calibration error (ECE) as defined by Guo et al. [3]: Denote $(\hat{y}_i)_{i \in U} = \operatorname{argmax}_{y_U} \hat{p}(y_U | x, y_L)$ the class predictions and $\hat{c}_i = \max_l p(y_i = l | x, y_L)$ the confidence of node i, we partition the test nodes into m bins $(B_1^n, \ldots, B_m^n)$ with $B_k^n = \{j \in U | \frac{k-1}{m} < \hat{c}_j \leq \frac{k}{m}\}$, the nodewise ECE is defined as

$$ECE^n = \sum_{k=1}^{m} \frac{|B_k^n|}{|U|} \left| acc(B_k^n) - conf(B_k^n) \right|, \text{ with} \tag{1}$$

$$acc(B_k^n) = \frac{1}{|B_k^n|} \sum_{i \in B_k^n} \mathbf{1}(\hat{y}_i = y_i^*) \text{ and } conf(B_k^n) = \frac{1}{|B_k^n|} \sum_{i \in B_k^n} c_i. \tag{2}$$

2.1 Edgewise expected calibration error

An important drawback of the nodewise ECE is that it neither reflects the dependency w.r.t graph structure nor the interdependencies between the nodes. To remedy this, we consider the confidence calibration of edgewise marginals, and propose the following edgewise expected calibration error: Denote $\mathcal{E}_U = \{(i, j) \in \mathcal{E} | i, j \in U\}$ the subset of edges constrained to test nodes, and for edge $(i, j) \in \mathcal{E}_U$ denote $\hat{c}_{i,j} = \max_{l,m} p(y_i = l, y_j = m | x, y_L)$ its confidence, we construct a similar edge binning $(B_1^e, \ldots, B_m^e)$ with $B_k^e = \{(i, j) \in \mathcal{E}_U | \frac{k-1}{m} < \hat{c}_{i,j} \leq \frac{k}{m}\}$ and define the edgewise

ECE as

$$ECE^e = \sum_{k=1}^{m} \frac{|B_k^e|}{|\mathcal{E}_U|} \big|acc(B_k^e) - conf(B_k^e)\big|, \text{ with} \tag{3}$$

$$acc(B_k^e) = \frac{1}{|B_k^e|} \sum_{(i,j)\in B_k^e} \mathbf{1}\big(\hat{y}_i = y_i^*\big) \cdot \mathbf{1}\big(\hat{y}_j = y_j^*\big) \text{ and } conf(B_k^e) = \frac{1}{|B_k^e|} \sum_{i \in B_k^e} c_{i,j}. \tag{4}$$

Figure 1 provides a concrete example that shows the dependency of edgewise ECE w.r.t graph structures. Similar issues also exist for other nodewise metrics like negative log-likelihood (NLL) and Brier score. Their edgewise extensions are also defined in Appendix B.

Along with the quantitative results of edgewise ECE, we can also plot its corresponding reliability diagram [3] for edgewise marginal predictions. They provide additional insights by displaying the over/under-confident tendency of the predictions at each confidence level.

It is possible to go beyond the edgewise setting and calibrate marginal distributions of larger subgraphs. We focus on the edgewise metrics in this work because it is a minimal extension that accounts for the graph structure and has manageable complexity. Moreover, edge marginals have a special significance for probabilistic graphical models: they determine the exact joint distribution of a tree and correspond to the Bethe approximation in the general case [26].

2.2 Agree and disagree ECEs

While the edgewise ECE estimates the general quality of the predicted edge marginals, it can be interesting to consider some additional subsets. Especially, since homophily plays an important role in graph learning [31], we find it insightful to separate the homophilous and heterophilous cases and consider the subsets of agreeing $\mathcal{E}_A = \{(i,j) \in \mathcal{E}_U | y_i^* = y_j^*\}$ and disagreeing edges $\mathcal{E}_D = \{(i,j) \in \mathcal{E}_U | y_i^* \neq y_j^*\}$. This allows us to define the agree ECE (ECE^a) and the disagree ECE (ECE^d) similar to Equations (3) and (4) but computed from $\mathcal{E}_A$ and $\mathcal{E}_D$ respectively instead of $\mathcal{E}_U$:

- To compute the agree ECE, we construct the binning $(B_1^a, \ldots, B_m^a)$ for the agreeing edge set $\mathcal{E}_A$ with $B_k^a = \{(i,j) \in \mathcal{E}_A | \frac{k-1}{m} < \hat{c}_{i,j} \leq \frac{k}{m}\}$. The agree ECE is defined as

$$ECE^a = \sum_{k=1}^{m} \frac{|B_k^a|}{|\mathcal{E}_A|} \big|acc(B_k^a) - conf(B_k^a)\big|, \text{ with} \tag{5}$$

$$acc(B_k^a) = \frac{1}{|B_k^a|} \sum_{(i,j)\in B_k^a} \mathbf{1}\big(\hat{y}_i = y_i^*\big) \cdot \mathbf{1}\big(\hat{y}_j = y_j^*\big) \text{ and } conf(B_k^a) = \frac{1}{|B_k^a|} \sum_{i \in B_k^a} c_{i,j}. \tag{6}$$

- Analogously, we construct the binning $(B_1^d, \ldots, B_m^d)$ for the disagreeing edge set $\mathcal{E}_D$ with $B_k^d = \{(i,j) \in \mathcal{E}_D | \frac{k-1}{m} < \hat{c}_{i,j} \leq \frac{k}{m}\}$. The disagree ECE is defined as

$$ECE^d = \sum_{k=1}^{m} \frac{|B_k^d|}{|\mathcal{E}_D|} \big|acc(B_k^d) - conf(B_k^d)\big|, \text{ with} \tag{7}$$

$$acc(B_k^d) = \frac{1}{|B_k^d|} \sum_{(i,j)\in B_k^d} \mathbf{1}\big(\hat{y}_i = y_i^*\big) \cdot \mathbf{1}\big(\hat{y}_j = y_j^*\big) \text{ and } conf(B_k^d) = \frac{1}{|B_k^d|} \sum_{i \in B_k^d} c_{i,j}. \tag{8}$$

3 Experiments

We conduct a series of experiments to empirically assess the proposed edgewise metrics and analyze the uncertainty estimation of GNNs. We focus on solving the node classification problem on commonly used benchmark datasets Cora [18], Citeseer [18], and Pubmed [13]. In terms of GNN learning frameworks, we compare commonly used GCN [7] and GAT [25] which directly produce nodewise output scores against GMNN [17] and EPFGNN [28] which explicitly address the structured prediction problem formulation. We analyze the quality of their predictive uncertainty using the nodewise, edgewise, and agree/disagree ECE.

Table 1: Uncertainty estimation results using various GNN models on Cora, Citeseer, and Pubmed in terms of nodewise, edgewise, agree and disagree ECEs. For all metrics, lower is better.

Dataset	Model	Nodewise	Edgewise	Agree	Disagree
Cora	GCN	12.47±4.37	16.64±5.53	24.18±5.89	17.87±3.23
	GAT	15.27±3.82	25.37±5.34	33.11±5.60	**13.60±2.51**
	GMNN	**4.05±1.30**	**4.07±1.86**	**9.96±2.59**	30.78±3.37
	EPFGNN	7.24±1.96	8.66±3.11	15.21±3.55	25.55±3.99
Citeseer	GCN	11.34±8.28	16.96±10.46	34.45±11.69	29.80±7.88
	GAT	16.40±8.51	27.63±9.93	46.70±10.88	**23.14±7.43**
	GMNN	**7.97±2.04**	**9.21±2.62**	**11.16±4.22**	54.95±6.02
	EPFGNN	8.05±5.33	11.07±7.37	18.91±12.74	43.95±13.16
Pubmed	GCN	7.16±1.40	3.34±1.53	12.64±1.68	38.28±1.78
	GAT	10.58±1.98	18.60±3.01	29.66±3.21	**28.18±2.17**
	GMNN	**4.24±0.65**	10.78±0.77	**2.62±0.53**	62.34±1.36
	EPFGNN	4.84±1.33	**3.12±1.41**	10.23±2.09	44.75±2.07

Calculation of the edgewise metrics requires estimating the edgewise marginal distributions. GMNN [17] and EPFGNN [28] model the joint output distribution from which the edge marginals can be inferred. For GCN [7] and GAT [25], we assume that they represent the joint output distribution as the product of nodewise marginal distributions, i.e., a naive mean field [15] assumption is made. This means that their predicted edgewise marginals are simply products of the neighboring node marginals. Further details about the experimental setup are provided in Appendix C.

We summarize the results in Table 1. Additional results, including the corresponding test accuracies and plots of reliability diagrams, are included in Appendix D.

Nodewise and edgewise expected calibration errors Firstly, we consider the nodewise and edgewise ECE results. Both metrics provide indications for the general quality of uncertainty estimation. From the results in Table 1, we observe two noticeable findings:

- GNNs which address node classification as structured prediction (GMNN and EPFGNN) tend to have better uncertainty estimation, in terms of both nodewise and edgewise ECEs, compared to models like GCN and GAT which simply output nodewise scores.
- There is a significant correlation between nodewise and edgewise ECEs. This is not surprising, as good uncertainty estimations should make both metrics small. That said, the edgewise ECE has a different focus w.r.t. the nodewise ECE, resulting the performance ranking of the Pubmed at odds with each other.

Agree and disagree ECEs To further analyze the edgewise probabilistic predictions, we partition the edge set into agreeing and disagreeing subsets. We evaluate the quality of their predictive uncertainty separately using agree and disagree ECEs.

- Structured prediction approaches achieve better agree ECEs and worse disagree ECEs. Given the fact that Cora, Citeseer, and Pubmed are all assortative graphs where neighbors tend to agree with each other, these methods seem to focus on calibrating the agreeing edges to reach good overall uncertainty estimation.
- GAT consistently yields the best disagree ECEs. This might be an effect of its attention mechanism which allows for distinct treatments of the agreeing and disagreeing edges.
- Reliability diagrams from Appendix D indicate that predictions on agreeing edges tend to be under-confident. On the contrary, the predictions on disagreeing edges tend to be over-confident given that GNNs fail to produce correct high-confidence predictions.

4 Discussions

In this work, we demonstrate the benefits of going beyond the nodewise setting for better uncertainty-aware graph learning. We discuss the need for structured evaluation metrics and propose edgewise metrics that offer additional insights. And our experiments show that tackling the structured prediction on graphs can help the uncertainty estimation.

Our experiments also highlight the well-known challenge of graph learning in heterophilous scenarios. This is currently an active area of research [31] and existing work is less established. We hope our proposed edgewise metrics can help the related research, and in general bolster the development and evaluation of new approaches for uncertainty-aware learning on graphs.

References

[1] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[2] M. Fey and J. E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[3] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *ICML*, 2017.

[4] A. Hasanzadeh, E. Hajiramezanali, S. Boluki, M. Zhou, N. Duffield, K. Narayanan, and X. Qian. Bayesian graph neural networks with adaptive connection sampling. In *ICML*, 2020.

[5] H. H.-H. Hsu, Y. Shen, C. Tomani, and D. Cremers. What makes graph neural networks miscalibrated? In *NeurIPS*, 2022.

[6] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[7] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[8] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[9] M. Kull, M. Perello Nieto, M. Kängsepp, T. Silva Filho, H. Song, and P. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *NeurIPS*, 2019.

[10] A. Kumar, P. S. Liang, and T. Ma. Verified uncertainty calibration. In *NeurIPS*, 2019.

[11] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[12] K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *UAI*, 1999.

[13] G. Namata, B. London, L. Getoor, and B. Huang. Query-driven active surveying for collective classification. In *ICML Workshop*, 2012.

[14] R. M. Neal and G. E. Hinton. *A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants*, pages 355–368. Springer, 1998.

[15] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

[16] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[17] M. Qu, Y. Bengio, and J. Tang. GMNN: Graph Markov neural networks. In *ICML*, 2019.

[18] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008.

[19] M. Sensoy, L. M. Kaplan, and M. Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018.

[20] J. Snoek, Y. Ovadia, E. Fertig, B. Lakshminarayanan, S. Nowozin, D. Sculley, J. V. Dillon, J. Ren, and Z. Nado. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019.

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

[22] M. Stadler, B. Charpentier, S. Geisler, D. Zügner, and S. Günnemann. Graph posterior network: Bayesian predictive uncertainty for node classification. In *NeurIPS*, 2021.

[23] C. Sutton and A. McCallum. Piecewise training for structured prediction. *Machine Learning*, 77(2-3):165, 2009.

[24] L. Teixeira, B. Jalaian, and B. Ribeiro. Are graph neural networks miscalibrated? In *ICML Workshop*, 2019.

[25] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. In *ICLR*, 2018.

[26] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, 2008.

[27] X. Wang, H. Liu, C. Shi, and C. Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. In *NeurIPS*, 2021.

[28] Y. Wang, Y. Shen, and D. Cremers. Explicit pairwise factorized graph neural network for semi-supervised node classification. In *UAI*, 2021.

[29] Y. Zhang, S. Pal, M. Coates, and D. Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *AAAI*, 2019.

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

[31] J. Zhu, Y. Yan, L. Zhao, M. Heimann, L. Akoglu, and D. Koutra. Beyond homophily in graph neural networks: Current limitations and effective designs. In *NeurIPS*, 2020.

A Nodewise and edgewise ECEs for concrete examples

To compare nodewise and edgewise ECE and show the dependency of edgewise ECE w.r.t. the graph structure, we do a case study for two simple graphs shown in Figure 1 Left: a chain and a cycle, both having three binary nodes ($\mathcal{N} = U = \{1, 2, 3\}$) with labels represented by their white or blue color ($\{w, b\}$), and both having the same ground-truth labels $(y_1^*, y_2^*, y_3^*) = (w, b, b)$. We use the naive mean field approximation and factorize the predicted joint distribution $\hat{p}(y_1, y_2, y_3) = \hat{p}(y_1)\hat{p}(y_2)\hat{p}(y_3)$ into product of nodewise predictions, and denote $p_i = \hat{p}(y_i = b)$ the predicted probability of node i having blue label. In the following, we compute the nodewise and edgewise ECEs using a single bin on the two graphs for the predictions listed in Figure 1 Right. As both graphs have the same node settings, their nodewise ECEs are identical for any given prediction.

$(p_1, p_2, p_3) = (0, 1, 1)$: This is the perfect forecast where the true labels are correctly predicted with full certainty, and we reach zero nodewise and edgewise ECEs for both graphs;

$(p_1, p_2, p_3) = (\frac{2}{3}, \frac{2}{3}, \frac{2}{3})$: This is a uniform prediction case that is nodewise calibrated. For both graphs, the label predictions are $(\hat{y}_1, \hat{y}_2, \hat{y}_3) = (b, b, b)$ and the nodewise accuracy and confidence terms are computed as

$$acc(|U|) = \frac{1}{|U|}\sum_{i=1}^{3} \mathbf{1}(y_i^* = \hat{y}_i) = \frac{2}{3}; \quad conf(|U|) = \frac{1}{|U|}\sum_{i=1}^{3} \hat{p}(\hat{y}_i) = \frac{1}{3}\sum_{i=1}^{3} p_i = \frac{2}{3}. \tag{9}$$

Thus the nodewise ECE for both graphs is

$$ECE^n = \frac{|U|}{|U|}\big|acc(U) - conf(U)\big| = 0. \tag{10}$$

For the chain case, the edgewise predictions are wrong for edge $(1, 2)$ and right for $(2, 3)$, and both edgewise predictions have confidence $\frac{2}{3} \times \frac{2}{3} = \frac{4}{9}$, thus its edgewise ECE is

$$ECE^e(Chain) = \frac{|\mathcal{E}_U|}{|\mathcal{E}_U|}\big|acc(\mathcal{E}_U) - conf(\mathcal{E}_U)\big| = |\frac{1}{2} - \frac{4}{9}| = \frac{1}{18}. \tag{11}$$

For the cycle case, all three edges $(1, 2), (1, 3), (2, 3)$ have confidence $\frac{4}{9}$ and only the edge $(2, 3)$ has correct label prediction, thus its edgewise ECE is

$$ECE^e(Cycle) = \frac{|\mathcal{E}_U|}{|\mathcal{E}_U|}\big|acc(\mathcal{E}_U) - conf(\mathcal{E}_U)\big| = |\frac{1}{3} - \frac{4}{9}| = \frac{1}{9}. \tag{12}$$

$(p_1, p_2, p_3) = (0.55, 0.8, 0.7)$: This is a non-uniform prediction that has zero edgewise ECE on the chain and nonzero edgewise ECE on the cycle. For both graphs, the label predictions are again $(\hat{y}_1, \hat{y}_2, \hat{y}_3) = (b, b, b)$ and their common nodewise ECE is computed as

$$ECE^n = \Big|\frac{2}{3} - \frac{0.55 + 0.8 + 0.7}{3}\Big| \approx 0.017. \tag{13}$$

For the chain case, one out of the two edges is correctly predicted, and its edgewise ECE is

$$ECE^e(Chain) = \Big|\frac{1}{2} - \frac{0.55 \times 0.8 + 0.8 \times 0.7}{2}\Big| = 0, \tag{14}$$

while for the cycle case, one out of its three edges is correctly predicted and its edgewise ECE is computed as

$$ECE^e(Cycle) = \Big|\frac{1}{3} - \frac{0.55 \times 0.8 + 0.8 \times 0.7 + 0.55 \times 0.7}{3}\Big| \approx 0.128. \tag{15}$$

The above examples highlight the difference between nodewise and edgewise ECE. Particularly, the edgewise ECEs depend on the graph structure while nodewise ECEs are structure agnostic and only depend on the node settings.

B Definitions of other edgewise metrics

Apart from ECE, other edgewise metrics can also be formulated. We provide the formulations for some common metrics in this section.

Accuracy While the commonly used nodewise accuracy is defined as

$$ACC^n = \frac{1}{|U|} \sum_{i \in U} \mathbf{1}(\hat{y}_i = y_i^*), \tag{16}$$

we can also define the edgewise accuracy

$$ACC^e = \frac{1}{|\mathcal{E}_U|} \sum_{(i,j) \in \mathcal{E}_U} \mathbf{1}(\hat{y}_i = y_i^*) \cdot \mathbf{1}(\hat{y}_j = y_j^*) \tag{17}$$

which computes the ratio of edges with correct label predictions.

Negative log-likelihood The nodewise negative log-likelihood (NLL), which recovers the joint NLL divided by $|U|$ under the naive mean field assumption, is defined as

$$NLL^n = -\frac{1}{|U|} \sum_{i \in U} \log \hat{p}(y_i^* | x, y_L). \tag{18}$$

Similarly, we can define the edgewise NLL as

$$NLL^e = -\frac{1}{|\mathcal{E}_U|} \sum_{(i,j) \in \mathcal{E}_U} \log \hat{p}(y_i^*, y_j^* | x, y_L). \tag{19}$$

Brier score The nodewise Brier score has the definition

$$Brier^n = \frac{1}{|U|} \sum_{i \in U} \Big(\big(1 - \hat{p}(y_i^* | x, y_L)\big)^2 + \sum_{y_i \neq y_i^*} \hat{p}(y_i | x, y_L)^2 \Big), \tag{20}$$

and we can formulate the edgewise Brier score as

$$Brier^e = \frac{1}{|\mathcal{E}_U|} \sum_{(i,j) \in \mathcal{E}_U} \Big(\big(1 - \hat{p}(y_i^*, y_j^* | x, y_L)\big)^2 + \sum_{(y_i, y_j) \neq (y_i^*, y_j^*)} \hat{p}(y_i, y_j | x, y_L)^2 \Big). \tag{21}$$

Note that the agree/disagree variants of the above edgewise metrics can be derived analogously by replacing $\mathcal{E}_U$ with $\mathcal{E}_A$ and $\mathcal{E}_D$ respectively. Moreover, while we focus on the ECE metric from Guo et al. [3], other variants of ECE exist [10] and their edgewise and agree/disagree variants can be formulated in a similar manner.

C Details of experimental setup

C.1 Dataset statistics

Table 2: Dataset statistics. The edge homophily presents the fraction of agreeing edges in the graphs.

Dataset	Nodes	Edges	Features	Classes	Edge homophily
Cora	2,708	10,556	1,433	7	81.0%
Citeseer	3,327	9,104	3,703	6	73.55%
Pubmed	19,717	88,648	500	3	80.24%

We evaluate our models on common citation networks: Cora [18], Citeseer [18], and Pubmed [13]. A summary of the dataset statistics is shown in Table 2. The edge homophily ratio from Zhu et al. [31] defines the fraction of edges in a graph which connects nodes with the same class label y^*:

$$Edge\,homophily(\mathcal{G}) = \frac{|\{(i,j) : (i,j) \in \mathcal{E} \mid y_i^* = y_j^*\}|}{|\mathcal{E}|} \tag{22}$$

The set in the numerator is similar to $\mathcal{E}_A$, but it collects agreeing edges from the whole edge set $\mathcal{E}$ instead of the test subset $\mathcal{E}_u$. Its complement is the fraction of disagreeing edges in the graph.

C.2 Details for model training

We randomly split the nodes to 15% as observed and 85% to unobserved set. We follow a training framework as [9, 5] to divide the observed set into three-fold where the two portions (10%) are the training set and the other portion (5%) is the validation set. For all the datasets, we do 5 random data splits, three-fold cross-validation per split, and together with 5 random model initializations. This results in 75 runs for each experiment. The models and training framework are implemented in PyTorch [16] and PyTorch Geometric [2].

We choose the commonly used GCN [7] and GAT [25] as the baseline models which do not consider the structured dependency between output nodes. For GCN, we use 2 layers with 64 hidden units. For GAT, we use 2 layers with 8 attention heads, and each consists of 8 hidden units. Dropout [21] is applied between layers with dropout rate 0.5. We train both models by minimizing NLL using Adam optimizer [6]. The learning rate is set to 0.01 with the weight decay 5e-4. We set the maximum epochs as 2000 and stop the training if the model does not improve on the validation set over the past 100 epochs.

For the structured prediction models we choose GMNN [17] and EPFGNN [28]. Both models are trained with the EM framework [14] to tackle the partially observed scenario of node classification on graphs and maximize the evidence lower bound instead of the intractable observed log-likelihood.

GMNN consists of one GCN as GNN backbone and another GCN to approximate the Conditional Random Field (CRF) [11]. Both GCNs have the same hyperparameters as the baseline GCN. We follow the training pipeline as the original paper. We first pre-train the GNN backbone for 200 epochs For M step we maximize the pseudolikelihood function [1] and update the approximated CRF with 100 epochs. For E step we update the GNN backbone for 100 epochs to approximate the true joint distribution. We observe the approximated CRF can achieve better performance than the GNN backbone by drawing multiple samples. In practice we draw 32 samples and compute the mean as the prediction for evaluation.

EPFGNN consists of one GCN backbone and a Markov Random Field (MRF) [8] which models the label dependency explicitly using a shared compatibility matrix. The GCN backbone produces as output the unary energies of the MRF. Same as GMNN, we first pretrain the GCN backbone for 200 epochs. For M step, we train the GCN backbone and MRF jointly by maximizing piecewise training objective [23]. Following Wang et al. [28], the training stops if the model does not improve its accuracy on the validation set over 10 epochs. For E step, we update a proposal distribution using mean field to approximate the true joint distribution. To get edge marginals from the MRF, we perform loopy belief propagation [12] with 100 iterations for evaluation.

D Additional experimental results

In this section, we collect additional experimental results. Table 3 summarizes the classification results in terms of nodewise, edgewise, agree, and disagree accuracies. And Figures 2, 3 and 4 display the reliability diagrams for GNN predictions on Cora, Citeseer and Pubmed.

Note the nodewise accuracies reported in Table 3 have significant discrepancies compared to the ones reported in Wang et al. [28]. This is caused by two main factors:

- At test time we employ the loopy belief propagation inference which also estimates edgewise marginals, while in Wang et al. [28] naive mean field is used instead;
- We have a different experimental setup and splitting (see Appendix C).

We report the corresponding results for EPFGNN using naive mean field in Table 4, which are more in line with the ones from Wang et al. [28]. We see that switching from naive mean field to loopy belief propagation yields better ECEs but worse accuracies. The decrease in accuracies might be explained by the mismatch of the inference method since naive mean field is used doing training.

Table 3: Classification results in terms of nodewise, edgewise, agree and disagree accuracies for various GNN models and datasets. For all metrics, higher is better.

Dataset	Model	Nodewise	Edgewise	Agree	Disagree
Cora	GCN	82.86±0.74	75.01±1.28	87.28±1.36	**23.93±2.50**
	GAT	**83.71±0.73**	**75.80±1.00**	**88.40±1.20**	23.33±1.89
	GMNN	82.73±0.86	75.46±1.39	88.33±1.35	21.92±2.16
	EPFGNN	76.48±1.16	65.07±1.85	76.32±1.74	18.25±2.51
Citeseer	GCN	**72.06±0.87**	**65.10±1.12**	87.15±1.11	**6.59±1.49**
	GAT	72.04±0.76	64.96±1.17	**87.59±0.87**	4.92±1.03
	GMNN	70.81±0.89	64.49±1.16	87.19±0.96	4.27±1.04
	EPFGNN	69.13±1.15	61.20±1.68	82.45±1.95	4.81±1.25
Pubmed	GCN	**86.43±0.26**	**79.07±0.38**	92.98±0.24	**22.08±0.95**
	GAT	85.12±0.36	78.38±0.47	**93.01±0.30**	18.47±1.10
	GMNN	84.23±0.29	77.70±0.46	92.82±0.29	15.75±0.98
	EPFGNN	82.51±0.58	74.35±0.67	88.40±0.65	16.81±1.22

Table 4: Results for EPFGNN when naive mean field is used for test time inference instead.

Dataset	Metric	Nodewise	Edgewise	Agree	Disagree
Cora	Accuracy	79.74 ± 1.09	70.02 ± 2.00	82.21 ± 1.88	19.30 ± 2.69
	ECE	9.16 ± 2.41	12.53 ± 3.31	20.03 ± 4.10	24.07 ± 4.40
Citeseer	Accuracy	71.77 ± 0.96	64.85 ± 1.30	87.56 ± 1.17	4.58 ± 1.20
	ECE	8.42 ± 6.01	12.32 ± 8.24	22.53 ± 13.97	44.61 ± 13.51
Pubmed	Accuracy	84.74 ± 0.55	76.76 ± 0.55	91.27 ± 0.52	17.31 ± 1.20
	ECE	6.13 ± 1.40	3.66 ± 1.56	11.26 ± 2.07	44.97 ± 2.23

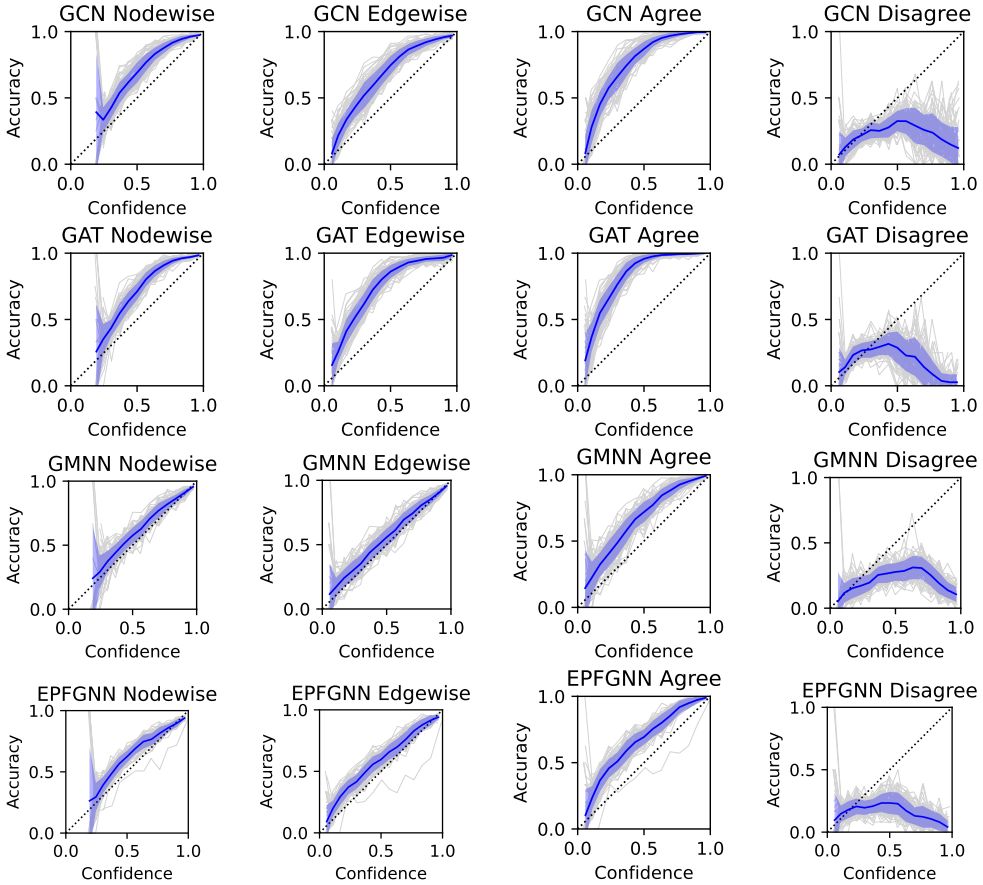


Figure 2: Reliability diagrams of GCN, GAT, GMNN, and EPFGNN trained on Cora. The light gray lines are the results of individual experiments. The solid blue line and the shaded blue area are the mean and standard deviation over 75 runs.

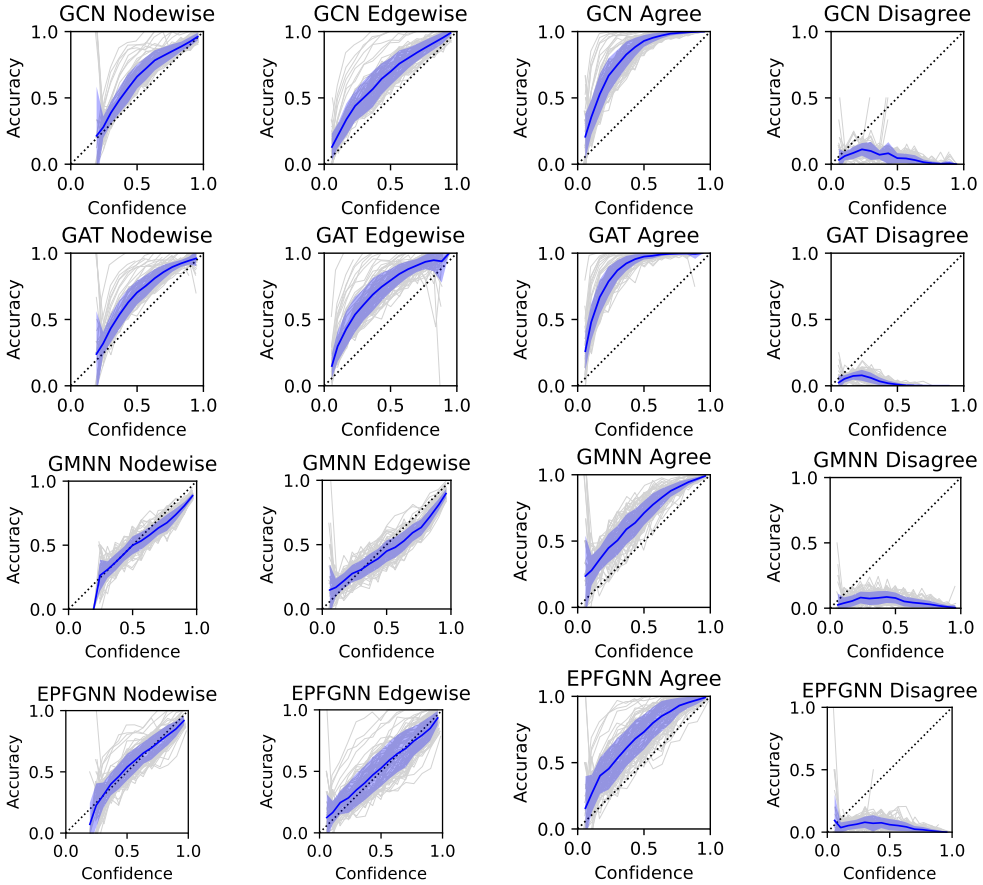


Figure 3: Reliability diagrams of GCN, GAT, GMNN, and EPFGNN trained on Citeseer. The light gray lines are the results of individual experiments. The solid blue line and the shaded blue area are the mean and standard deviation over 75 runs.

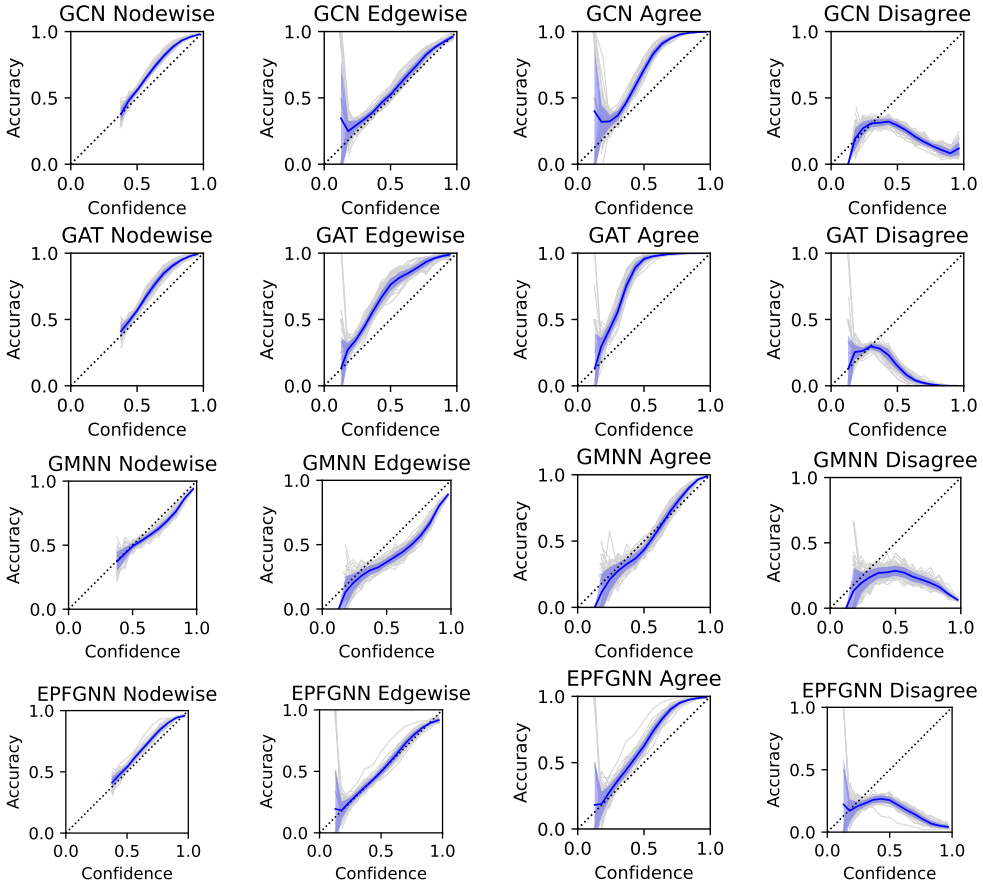


Figure 4: Reliability diagrams of GCN, GAT, GMNN, and EPFGNN trained on Pubmed. The light gray lines are the results of individual experiments. The solid blue line and the shaded blue area are the mean and standard deviation over 75 runs.