On the Global and Local Calibration of Graph Neural Networks

Francesco Ferrini[1] **Veronica Lachi**[3] **Antonio Longa**[3] **Cesare Barbera**[1,4]

Andrea Pugnana[1] **Andrea Passerini**[1] **Manfred Jaeger**[5]

[1]University of Trento [2]Fondazione Bruno Kessler [3]UiT The Arctic University of Norway

[4]University of Pisa [5]Aalborg University

Abstract

Recent work on calibration of Graph Neural Networks (GNNs) has largely concluded that GNNs are miscalibrated and typically under-confident on standard node classification benchmarks, motivating the development of graph-specific calibration methods evaluated in terms of Expected Calibration Error (ECE) on citation datasets such as Cora, Citeseer, and Pubmed. We revisit these conclusions and show that much of the reported miscalibration is explained by hyperparameter choices rather than intrinsic limitations of GNN architectures. Properly tuned classical GNNs achieve comparable ECE with respect to existing calibration methods. We further provide the first study of local calibration in graph neural networks by computing Local Calibration Error (LCE) on graph data. In particular, we adapt LCE to the graph setting by defining locality through distances in the node embedding space learned by the GNN. While global calibration errors are small, we observe higher local miscalibration. As future direction, calibration of GNNs should be further studied locally and on larger graph benchmarks rather than relying solely on global metrics on small datasets.

Workshop "Towards Trustworthy Predictions: Theory and Applications of Calibration for Modern AI" at AISTATS 2026, Tangier, Morocco.

1 Introduction

In many high-stakes applications, Machine Learning (ML) models are expected not only to be accurate but also well-calibrated - i.e., their predicted probabilities should reflect the true empirical frequencies of the corresponding classes. Following extensive studies on calibration in deep neural networks [Wang et al., 2021a, Wang, 2023, Luo et al., 2022, Guo et al., 2017], recent work has investigated the calibration properties of Graph Neural Networks (GNNs) [Hsu et al., 2022, Liu et al., 2022, Huang et al., 2025, Teixeira et al., 2019, Wang et al., 2024, Zhuang et al., 2024].

Most existing studies evaluate calibration of GNNs in the node classification setting on small citation network benchmarks such as Cora, Citeseer, and Pubmed [Yang et al., 2016]. Calibration performance is typically measured using Expected Calibration Error (ECE), and reliability diagrams are used for qualitative analysis. Across these works, a common conclusion has emerged: although GNNs often achieve strong predictive accuracy, their confidence estimates are generally miscalibrated and tend to be under-confident [Hsu et al., 2022, Huang et al., 2025, Teixeira et al., 2019, Zhuang et al., 2024].

Based on this observation, a growing literature has proposed calibration methods specifically designed for GNNs. These approaches include post-hoc methods such as temperature scaling variants adapted to graph structure, topology-aware calibration networks, and regularization techniques that modify the training objective to improve calibration. Many of these methods rely on additional models or calibration networks that operate on logits, node features, or graph structure to produce node-wise calibration parameters [Hsu et al., 2022, Huang et al., 2025]. Empirical results reported

in prior work suggest consistent improvements in ECE compared to uncalibrated GNN baselines on standard benchmarks.

In this paper we revisit the calibration properties of GNNs and challenge the widely accepted view that classical GNN architectures are inherently miscalibrated. We show that a large portion of the reported miscalibration can be attributed to suboptimal hyperparameter choices rather than intrinsic limitations of GNN models.

Our contributions are as follows:

1. We demonstrate that when compared against properly tuned GNN baselines, existing calibration methods often fail to provide big improvements. In many cases, standard GNN models with appropriate hyperparameters show comparable results with respecto to exsisting calibration approaches in terms of calibration error.

2. We provide the first systematic local calibration analysis for GNNs based on Local Calibration Error (LCE) [Luo et al., 2022]. While globally tuned GNNs exhibit low ECE, our results reveal that errors are primarily local rather than global.

Our findings suggest several directions for future research. First, calibration of GNNs should not be studied purely in terms of global metrics such as ECE. Instead, local calibration properties should be investigated in greater detail, following recent trends in calibration research for deep neural networks [Xiong et al., 2023]. Second, the small scale of commonly used citation benchmarks raises concerns about the reliability of calibration estimates. In contrast to calibration studies for standard deep learning models, which typically use large-scale datasets, node classification benchmarks often contain only a few dozen labelled nodes per class. This results in small bin supports for calibration metrics and potentially unstable estimates. Future work should therefore investigate calibration behaviour on larger graph benchmarks such as OGB [Hu et al., 2021].

Overall, our results indicate that classical GNN models are significantly better calibrated than previously believed, and that future research should focus on understanding local calibration phenomena and large-scale settings rather than designing increasingly complex global calibration methods.

2 Preliminaries

2.1 Node Classification with Graph Neural Networks

Let $G = (V, E)$ be a graph with node set V and edge set E. Each node $v \in V$ is associated with a feature vector $\mathbf{x}_v \in \mathbb{R}^d$ and, for a subset of nodes, a class label $y_v \in \{1, \dots, K\}$.

GNNs learn node representations through iterative message passing over the graph structure. Starting from initial node representations

$$\mathbf{h}_v^{(0)} = \mathbf{x}_v,$$

each layer updates node embeddings by aggregating information from neighboring nodes. A generic message passing layer can be written as

$$\mathbf{m}_v^{(l)} = \mathrm{AGG}^{(l)}\left(\{\mathbf{h}_u^{(l-1)} : u \in \mathcal{N}(v)\}\right),$$

$$\mathbf{h}_v^{(l)} = \mathrm{COMB}^{(l)}\left(\mathbf{h}_v^{(l-1)}, \mathbf{m}_v^{(l)}\right),$$

where $\mathcal{N}(v)$ denotes the set of neighbors of node v, $\mathrm{AGG}^{(l)}$ is an aggregation function such as mean or attention-based aggregation, and $\mathrm{COMB}^{(l)}$ is a parametric transformation typically implemented as a neural network layer.

After L layers, the final node embedding is

$$\mathbf{h}_v = \mathbf{h}_v^{(L)}.$$

Logits are produced by a parametric prediction layer

$$\mathbf{z}_v = \mathbf{W}\mathbf{h}_v + \mathbf{b},$$

or more generally

$$\mathbf{z}_v = g_\theta(\mathbf{h}_v),$$

where g_θ is a learnable function. Probabilities are obtained via the softmax function

$$\mathbf{p}_v = \mathrm{softmax}(\mathbf{z}_v).$$

The predicted label is

$$\hat{y}_v = \arg\max_k p_{v,k},$$

and the confidence score is defined as

$$c_v = \max_k p_{v,k}.$$

2.2 Calibration

There are multiple notions of calibration in the literature. The simplest notion is *confidence calibration*, which requires that the classifier's confidence matches the empirical accuracy, i.e.,:

$$\mathbb{P}(\hat{y}_v = y_v \mid c_v = p) = p, \quad \forall p \in [0,1].$$

In practice, confidence calibration is generally measured using Expected Calibration Error (ECE). Predictions are partitioned into M confidence bins $\{B_m\}_{m=1}^M$, and ECE is computed as

$$\mathrm{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{|V_{\text{test}}|} \left|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\right|.$$

ECE is the standard metric used in the literature for evaluating calibration of GNNs on node classification benchmarks.

2.3 Local Calibration Error

Global calibration metrics, such as ECE, measure average reliability across the entire dataset and may hide significant heterogeneity across individual predictions. To obtain a more fine-grained view of calibration, we consider Local Calibration Error (LCE) [Luo et al., 2022], which measures calibration within neighborhoods of similar samples.

Let $\hat{p}(x)$ denote the predicted confidence and $f(x)$ the predicted label. Given a kernel function $k_\gamma(x, x')$ with bandwidth γ, and letting $\beta(x)$ denote the set of samples whose predicted confidence lies in the same bin as x, the Local Calibration Error at a point x is

$$\mathrm{LCE}_\gamma(x) = \left| \frac{\sum_{i \in \beta(x)} \big(\hat{p}(x_i) - \mathbf{1}(f(x_i) = y_i)\big) k_\gamma(x, x_i)}{\sum_{i \in \beta(x)} k_\gamma(x, x_i)} \right|.$$

This definition interpolates between individual and global calibration: a small bandwidth γ corresponds to a more local calibration, while a large bandwidth approaches global calibration metrics.

The notion of locality in LCE is general and not specific to graph data. In the original formulation, locality is defined through similarity in a feature space via the kernel function $k_\gamma(x, x')$. For graph data, locality can instead be defined using node representations produced by the GNN itself.

Table 1: Global calibration on Planetoid citation benchmarks. ECE computed with 20 confidence bins (lower is better). Results are mean ± standard deviation. Cells highlighted in yellow indicate the best result, blue the second-best, and green the third-best for each dataset.

Model	Cora	Citeseer	Pubmed
CaGCN	4.01 ± 0.67	5.95 ± 0.72	4.05 ± 0.60
GATS	3.96 ± 0.46	7.26 ± 0.63	4.01 ± 0.13
AU-LS	4.32 ± 0.23	6.69 ± 1.76	6.48 ± 1.35
SCAR	3.35 ± 0.53	3.43 ± 0.58	3.81 ± 0.47
GNN	3.54 ± 0.39	6.31 ± 0.27	3.87 ± 0.27

Table 2: Local calibration on the test split of Cora for GNN. Mean LCE, as a function of kernel bandwidth γ and number of confidence bins (lower is better). Results are mean ± standard deviation.

	Bins		
γ	**10**	**15**	**20**
0.1	23.89 ± 0.54	23.89 ± 0.54	23.89 ± 0.54
0.5	22.52 ± 0.46	22.71 ± 0.51	22.98 ± 0.40
0.7	20.62 ± 0.57	21.24 ± 0.52	21.73 ± 0.33
1.0	17.21 ± 0.67	18.42 ± 0.66	19.23 ± 0.38
2.0	9.66 ± 1.11	10.93 ± 1.04	11.99 ± 0.77
5.0	6.21 ± 1.06	6.44 ± 1.07	6.62 ± 0.81

In this work we define similarity between nodes using distances in the embedding space learned by the GNN. Let $\mathbf{h}_v$ denote the embedding of node v. The kernel is defined as

$$k_\gamma(v, u) = \exp\left(-\frac{\|\mathbf{h}_v - \mathbf{h}_u\|^2}{\gamma}\right),$$

so that nodes are considered locally similar when their learned representations are close. This definition allows us to analyze calibration behavior within regions of the embedding space learned by the GNN.

3 Experiments

We evaluate calibration performance on the standard Planetoid node-classification benchmarks Cora, Citeseer, and Pubmed [Yang et al., 2016]. These datasets are widely used in prior work on GNN calibration and contain only a small number of labeled nodes per class, resulting in limited statistical support for calibration estimates. Further dataset and training details are provided in Appendix A.

We compare a classical GNN model with carefully tuned hyperparameters against several state-of-the-art

Table 3: Effective sample size (ESS) for LCE estimation on the CORA test split. ESS (mean ± standard deviation) within the local neighborhoods induced by kernel bandwidth γ, reported for different numbers of confidence bins. Larger ESS indicates more reliable local calibration estimates (a common rule of thumb is ESS $\gtrsim 30$).

	Bins		
γ	**10**	**15**	**20**
0.1	1.03 ± 0.00	1.03 ± 0.00	1.03 ± 0.00
0.5	1.86 ± 0.04	1.78 ± 0.03	1.71 ± 0.04
0.7	3.74 ± 0.25	3.40 ± 0.20	3.16 ± 0.18
1.0	9.00 ± 0.83	7.79 ± 0.67	7.06 ± 0.54
2.0	33.50 ± 2.68	27.20 ± 2.12	23.53 ± 1.61
5.0	124.80 ± 15.09	91.73 ± 11.53	73.70 ± 8.79

calibration methods.

The considered baselines include:

CaGCN [Wang et al., 2021b], a topology-aware post-hoc calibration method that trains a calibration GNN to propagate confidence values across the graph.

GATS [Hsu et al., 2022], an attention-based temperature scaling method that produces node-wise calibration parameters based on the graph structure and predictive distributions.

AU-LS [Wang et al., 2024], a training-time calibration method based on adaptive label smoothing designed to mitigate both under- and over-confidence.

SCAR [Huang et al., 2025], a calibration approach that adjusts confidence at the final representation layer by modifying the weight decay and performing node-level calibration.

Our model consists of a standard GNN architecture trained with hyperparameter tuning, including depth, normalization, and regularization choices. Implementation details are provided in Appendix A.

Calibration is evaluated using ECE with 20 bins and LCE as defined in Section 2, with 10, 15 and 20 bins and γ varying between 0.1 and 5.

3.1 Global Calibration

Table 1 reports ECE results on the three datasets. The tuned GNN baseline achieves comparable calibration performance on all datasets.

These results suggest that classical GNN architectures can achieve strong calibration performance when properly tuned. In particular, the global calibration errors observed here are substantially lower than typically reported in prior works, suggesting that miscalibration is largely influenced by hyperparameter choices.

Overall, these findings suggest that standard GNNs are effectively well-calibrated on Planetoid datasets without the need for specialized calibration methods.

At the same time, the Planetoid datasets are relatively small and contain limited numbers of labeled nodes. This leads to small bin supports when computing ECE and may result in unstable calibration estimates. An important direction for future work is therefore the evaluation of calibration on larger benchmarks such as OGB, which have not yet been systematically explored in the calibration literature.

3.2 Local Calibration

We next analyze calibration locally using LCE. Table 2 reports mean LCE values for different kernel bandwidths γ and numbers of bins.

To assess the reliability of the estimates, Table 3 reports the effective sample size (ESS) [Zhang et al., 2024] within each bin. Reliable estimation typically requires at least 30 samples per bin. The results show that for $\gamma \leq 1$ the effective sample size is generally too small, while values $\gamma \geq 2$ provide sufficient support for stable estimates.

For these reliable values of γ, the tuned GNN exhibits substantially larger LCE values than suggested by the global ECE results. This indicates that calibration errors are not uniformly distributed across the graph but instead concentrated in specific regions of the embedding space.

These results suggest that calibration in GNNs is primarily a local phenomenon: models that appear well-calibrated globally may still exhibit significant local miscalibration.

Overall, our findings indicate that future research should focus on local calibration methods rather than purely global approaches.

4 Conclusion

This paper revisits the calibration properties of GNNs for node classification. Contrary to the common view in the literature, we show that classical GNN architectures are not inherently miscalibrated. With appropriate hyperparameter choices, standard models are already well-calibrated without specialized calibration methods.

Our experiments show that existing calibration approaches do not show substantial improvements over properly tuned GNN baselines, suggesting that part

of the perceived calibration problem originates from weak reference models rather than fundamental limitations of GNNs. Using a notion of Local Calibration Error based on proximity in the GNN embedding space, we observe that calibration errors are primarily local rather than global: even well-calibrated models in terms of ECE exhibit high LCE.

These findings suggest two main directions for future research. First, calibration analysis for graph learning should focus on local calibration phenomena rather than relying exclusively on global metrics such as ECE. Second, calibration studies should move beyond small citation benchmarks toward larger graph datasets where calibration estimates are statistically more reliable.

References

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Hans Hao-Hsun Hsu, Yuesong Shen, Christian Tomani, and Daniel Cremers. What makes graph neural networks miscalibrated? *Advances in Neural Information Processing Systems*, 35:13775–13786, 2022.

Weihua Hu, Matthias Fey, Hongyu Ren, Maho Nakata, Yuxiao Dong, and Jure Leskovec. OGB-LSC: A large-scale challenge for machine learning on graphs. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL `https://openreview.net/forum?id=qkcLxoC52kL`.

Jincheng Huang, Jie Xu, Xiaoshuang Shi, Ping Hu, Lei Feng, and Xiaofeng Zhu. The final layer holds the key: A unified and efficient gnn calibration framework. *arXiv preprint arXiv:2505.11335*, 2025.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Tong Liu, Yushan Liu, Marcel Hildebrandt, Mitchell Joblin, Hang Li, and Volker Tresp. On calibration of graph neural networks for node classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

Rachel Luo, Aadyot Bhatnagar, Yu Bai, Shengjia Zhao, Huan Wang, Caiming Xiong, Silvio Savarese, Stefano Ermon, Edward Schmerling, and Marco Pavone. Local calibration: metrics and recalibration. In *Uncertainty in Artificial Intelligence*, pages 1286–1295. PMLR, 2022.

Leonardo Teixeira, Brian Jalaian, and Bruno Ribeiro. Are graph neural networks miscalibrated? *arXiv preprint arXiv:1905.02296*, 2019.

Cheng Wang. Calibration in deep learning: A survey of the state-of-the-art. *arXiv preprint arXiv:2308.01222*, 2023.

Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021a.

Min Wang, Hao Yang, Jincai Huang, and Qing Cheng. Moderate message passing improves calibration: A universal way to mitigate confidence bias in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 21681–21689, 2024.

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. *Advances in Neural Information Processing Systems*, 34:23768–23779, 2021b.

Miao Xiong, Ailin Deng, Pang Wei W. Koh, Jiaying Wu, Shen Li, Jianqing Xu, and Bryan Hooi. Proximity-informed calibration for deep neural networks. In *NeurIPS*, 2023.

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

Landan Zhang, Sylwia Bujkiewicz, and Dan Jackson. Three new methodologies for calculating the effective sample size when performing population adjustment. *BMC Medical Research Methodology*, 24(1): 287, 2024.

Dingyi Zhuang, Chonghe Jiang, Yunhan Zheng, Shenhao Wang, and Jinhua Zhao. Gets: Ensemble temperature scaling for calibration in graph neural networks. *arXiv preprint arXiv:2410.09570*, 2024.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. **Yes**. Section 2 defines the node classification setting, the GNN model, calibration metrics, and the Local Calibration Error formulation adapted to graphs.

(b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. **No**. The paper does not include a formal complexity analysis. In particular, the computational cost of LCE estimation and kernel computations is not analyzed explicitly. The complexity and statistical properties of LCE are discussed in the original work introducing the metric [Luo et al., 2022], which we follow in our implementation.

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. **No**. The code is not yet included. It will be released upon publication.

2. For any theoretical claim, check if you include:

(a) Statements of the full set of assumptions of all theoretical results. **Not Applicable**. The paper does not present new theoretical results. It is purely empirical.

(b) Complete proofs of all theoretical results. **Not Applicable**. No theoretical results are presented.

(c) Clear explanations of any assumptions. **Not Applicable**. No theoretical assumptions are required beyond standard experimental protocols.

3. For all figures and tables that present empirical results, check if you include:

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). **Yes**. Reproducibility materials are presented in appendix A.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen). **Yes**. Training details and dataset splits are described in Appendix A.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). **Yes**. ECE and LCE are formally defined in Section 2, and tables report mean and standard deviation across multiple runs.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). **Yes**. The used infrastructure are reported in appendix A.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

(a) Citations of the creator If your work uses existing assets. **Yes**. The paper cites the Planetoid datasets and prior calibration methods.

(b) The license information of the assets, if applicable. **No**. Dataset licenses are not specified.

(c) New assets either in the supplemental material or as a URL, if applicable. **Not Applicable**. No new datasets or benchmarks are introduced.

(d) Information about consent from data providers/curators. **Not Applicable**. The datasets are publicly available citation networks without human subject interaction.

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. **Not Applicable**. The datasets consist of scientific publications and do not contain sensitive personal data.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. **Not Applicable**

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. **Not Applicable**

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. **Not Applicable**

Appendix

A Experimental Details

All experiments were conducted on a dedicated server equipped with an Intel i9-13900K CPU (24 cores, 36MB cache), 192GB of RAM, and an NVIDIA GeForce RTX 4090 GPU with 24GB of memory.

All models were implemented in PyTorch and PyTorch Geometric. Training a single model required less than one hour on the largest dataset.

A.1 Model Architecture

All experiments use a standard Graph Convolutional Network architecture [Kipf and Welling, 2016]. We use GCN in order to match the experimental setting of prior calibration works, where calibration methods are typically evaluated on top of classical GNN architectures.

The final model outputs node probabilities through a softmax layer and confidence scores are defined as the maximum predicted probability.

A.2 Hyperparameter Tuning

Hyperparameters for our model were selected using grid search on the validation set. We explored hidden dimensions in $\{16, 32, 64\}$, numbers of GCN layers in $\{1, 2, 3, 4\}$, and dropout rates in $\{0.1, 0.3, 0.5\}$. All models were trained using the standard Planetoid data splits, and the best configuration was selected based on validation performance. Hyperparameter tuning was performed for our GNN baseline. For the calibration methods, we report the results published in the original papers and use their recommended experimental settings.

A.3 Reproducibility

All experiments use the fixed Planetoid splits for Cora, Citeseer, and Pubmed. Each experiment was repeated with multiple random seeds and results are reported as mean and standard deviation.

The code used to reproduce all experiments is available at Github

B Datasets and Experimental Splits

We evaluate our methods on three standard citation network benchmarks: Cora, CiteSeer, and PubMed. In these datasets, nodes represent scientific publications and edges represent citation links between documents. Each node is associated with a document feature vector derived from the text and a categorical label corresponding to the research topic. The task is node classification.

We use the standard fixed Planetoid splits that are commonly adopted in the graph neural network literature. A small labeled training set is used for model fitting, a validation set is used for hyperparameter selection and calibration tuning, and a disjoint test set is used for final evaluation. A summary of the datasets and splits is reported in Table 4.

The Planetoid splits follow a transductive semi-supervised setting in which only a small subset of nodes is labeled. The remaining nodes are treated as unlabeled during training but are still included in the graph structure and participate in message passing. This allows the model to exploit the full connectivity of the citation network while learning from a limited number of labeled examples. Consequently, the number of nodes assigned to the training validation and test sets does not sum to the total number of nodes in each dataset. The unlabeled nodes

Table 4: Summary of the citation network datasets and the standard train validation and test splits used in our experiments.

Dataset	Nodes	Edges	Classes	Features	Train	Val	Test
CORA	2708	5429	7	1433	140	500	1000
CITESEER	3327	4732	6	3703	120	500	1000
PUBMED	19717	44338	3	500	60	500	1000

are not used for supervision or evaluation but influence the learned node representations through the graph structure.