SimCalib: Graph Neural Network Calibration Based on Similarity between Nodes

Boshi Tang[1], **Zhiyong Wu**[1], **Xixin Wu**[2*],
Qiaochu Huang[1], **Jun Chen**[1], **Shun Lei**[1], **Helen Meng**[2]
[1]Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2]The Chinese University of Hong Kong, Hong Kong SAR, China
tbs22@mails.tsinghua.edu.cn, zywu@sz.tsinghua.edu.cn, wuxx@se.cuhk.edu.hk,
{hqc22, y-chen21, leis21}@mails.tsinghua.edu.cn, hmmeng@se.cuhk.edu.hk

Abstract

Graph neural networks (GNNs) have exhibited impressive performance in modeling graph data as exemplified in various applications. Recently, the GNN calibration problem has attracted increasing attention, especially in cost-sensitive scenarios. Previous work has gained empirical insights on the issue, and devised effective approaches for it, but theoretical supports still fall short. In this work, we shed light on the relationship between GNN calibration and nodewise similarity via theoretical analysis. A novel calibration framework, named *SimCalib*, is accordingly proposed to consider similarity between nodes at global and local levels. At the global level, the Mahalanobis distance between the current node and class prototypes is integrated to implicitly consider similarity between the current node and all nodes in the same class. At the local level, the similarity of node representation movement dynamics, quantified by nodewise homophily and relative degree, is considered. Informed about the application of nodewise movement patterns in analyzing nodewise behavior on the over-smoothing problem, we empirically present a possible relationship between over-smoothing and GNN calibration problem. Experimentally, we discover a correlation between nodewise similarity and model calibration improvement, in alignment with our theoretical results. Additionally, we conduct extensive experiments investigating different design factors and demonstrate the effectiveness of our proposed SimCalib framework for GNN calibration by achieving state-of-the-art performance on 14 out of 16 benchmarks.

Introduction

Graphs are ubiquitous in the real world, including social networks, knowledge graphs, traffic networks, among others. Due to the universality and expressive power of graph representations, the deep learning community has paid much attention to learning from graph-structured data and introduced various types of graph neural networks (GNNs) (Kipf and Welling 2016; Veličković et al. 2017; Hamilton, Ying, and Leskovec 2017). To date, GNNs have been successfully applied to various downstream applications with remarkable accuracy, such as drug discovery (Zhang et al. 2022b), fluid simulation (Liu et al. 2022), and recommendation system (Fan et al. 2019), to name a few.

However, in many applications, trustworthiness is as important (if no more) than accuracy, especially in safety-sensitive fields (Dezvarei et al. 2023). One of the promising solution to ensure trustworthiness of a trained model is aligning its prediction confidence with the ground truth accuracy, i.e., the model should provide the appropriate confidence to reveal whether the prediction should be trusted. Unfortunately, such an alignment is hardly achieved by modern neural networks (Guo et al. 2017; Wang et al. 2021). To mitigate the issue, a variety of calibration methods (Kull et al. 2019; Gupta et al. 2020; Zhang, Kailkhura, and Han 2020) have been proposed to calibrate pretrained deep neural networks (DNNs). However, calibration for GNNs is still underexplored. While it is possible to directly apply calibration methods designed for DNNs to GNNs by treating each node in the graph as an isolated sample, the specific challenges posed by GNN calibration remain unaddressed as the specific characteristics of graph structure, e.g., the relationship between nodes, is not well utilized for calibrating the GNN predictions.

Recently, a few works focus on GNN calibration, among which the most noticeable are CaGCN (Wang et al. 2021) and GATS (Hsu et al. 2022). Specifically, CaGCN produces nodewise temperatures by processing the pretrained classifier's logits with another graph convolutional network (GCN), in the hope that structural information can be implicitly integrated in model calibration. Following it, GATS empirically investigates factors that influence GNN calibration, and employs an attention network to account for the influential factors. However, to date, the efforts towards such a structured prediction problem (Nowozin, Lampert et al. 2011) have mostly concentrated on empirical aspects, suffering from a lack of theoretical supports.

In this paper, we make extensive efforts from both theoretical and practical aspects to tackle the aforementioned issues. Our main contributions are summarized as follows:

- We develop a theoretical approach for GNN calibration, and prove that by taking nodewise similarity into consideration we can reduce expected calibration error (ECE) effectively.
- We propose two similarity-oriented mechanisms to account for both global feature-level similarity and local nodewise representation movement dynamics similarity. By incorporating them into network designs, we propose

[*]Corresponding author.

SimCalib, a novel GNN calibration method that is data-efficient, easy to implement and highly expressive.

- We are the first to relate the oversmoothing problem to GNN calibration.
- We conduct comprehensive experiments investigating various design factors, and demonstrate the effectiveness of SimCalib by achieving new SOTA performance on 14 out of 16 benchmarks. Particularly, compared with the previous SOTA model, SimCalib on average reduces ECE by 10.4%.

Preliminary

Problem Setting

Herein we consider the problem of calibrating GNNs on semi-supervised node classification tasks. Specifically, given an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of nodes $\mathcal{V}$ and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, each node $v_i \in \mathcal{V}$ is associated with a feature vector $\boldsymbol{x}_i \in \mathbb{R}^d$. Moreover, a proper subset of nodes, denoted as $\mathcal{L} \subset \mathcal{V}$, is further associated with labels $\{y_i\}_{i:v_i \in \mathcal{L}}$, where $y_i \in \mathcal{Y} = \{1, \ldots, K\}$ is the ground-truth label for v_i. And the goal of semi-supervised node classification is to infer the labels for the unlabeled nodes $\mathcal{U} = \mathcal{V} \setminus \mathcal{L}$. A graph neural network approaches the problem by taking into account both nodewise features and structural information, i.e. adjacency matrix $\mathbf{A}$, and it predicts a probability distribution $\hat{p}_i$ over all the classes for each node v_i. The value on the j-th position of the distribution, i.e. $\hat{p}_i^{(j)}$, describes the estimated probability of v_i being in class j.

For each node, $\hat{p}_i$ induces the corresponding label prediction $\hat{y}_i := \texttt{argmax}_j \ \hat{p}_i^{(j)}$ and confidence $\hat{c}_i := \texttt{max}_j \ \hat{p}_i^{(j)}$. Perfect calibration is defined as (Wang et al. 2021):

$$\forall\, c \in [0, 1], \quad \mathtt{P}(y_i = \hat{y}_i | \hat{c}_i = c) = c. \tag{1}$$

In practice, perfect calibration cannot be estimated with a finite number of samples, therefore calibration quality is often quantified by expected calibration error (ECE) instead (Naeini, Cooper, and Hauskrecht 2015; Guo et al. 2017):

$$\mathtt{ECE} := \mathbb{E}_p\left[\left|\mathbb{E}[Y = k | \hat{\mathtt{p}}(Y = k|\boldsymbol{x}) = p] - p\right|\right], \tag{2}$$

where $\hat{\mathtt{p}}(Y = k|\boldsymbol{x})$ is the predicted probability of $\boldsymbol{x}$ being in class k, the inner expectation represents the ground-truth probability of $\boldsymbol{x}$ belonging to k, and the outer expectation iterates over all $p \in (0, 1)$.

Nodewise Temperature

To preserve the nodewise predictions, CaGCN calibrates logits by scaling them with nodewise temperatures, i.e.

$$\hat{z}_i' = \frac{\hat{z}_i}{T_i}, \tag{3}$$

where $\hat{z}_i$ is the nodewise logits for v_i produced by the pre-trained classifier and $T_i > 0$ is the temperature for v_i estimated by CaGCN. A noticeable property of such a mechanism is

$$\underset{j \in \mathcal{Y}}{\texttt{argmax}} \ \hat{z}_i' = \underset{j \in \mathcal{Y}}{\texttt{argmax}} \ \hat{z}_i, \tag{4}$$

which maintains the prediction accuracy of GNNs after calibration. In this work, we follow the practice and calibrate models in the same manner.

Theoretical Results

We consider the Gaussian mixture block model(Li and Schramm 2023), which is commonly used for theoretical analysis on graphs and neural network calibration(Carmon et al. 2019; Zhang et al. 2022a).

Definition 1. *(Gaussian model). For $\boldsymbol{\mu} \in \mathbb{R}^d$ and $\sigma > 0$, the Gaussian model is defined as a distribution over $(\boldsymbol{x}, y) \in \mathbb{R}^d \times \{-1, 1\}$:*

$$\boldsymbol{x}|y \sim \mathcal{N}(\boldsymbol{\mu} \cdot y, \sigma^2 \boldsymbol{I}), \tag{5}$$

where y follows the Bernoulli distribution $\mathtt{P}(y = 1) = \mathtt{P}(y = -1) = 1/2$.

Assumption 1. *For two graph nodes i and j with the Gaussian model parameterized by $\boldsymbol{x}_i|y_i \sim \mathcal{N}(\boldsymbol{\mu}_i \cdot y_i, \sigma^2 \boldsymbol{I}), \boldsymbol{x}_j|y_j \sim \mathcal{N}(\boldsymbol{\mu}_j \cdot y_j, \sigma^2 \boldsymbol{I})$, there exists a underlying linear relationship between the two nodes: $\boldsymbol{\mu}_j = a\boldsymbol{\mu}_i + \boldsymbol{b}$, where $a \in \mathbb{R}, \boldsymbol{b} \in \mathbb{R}^d$ are constants, and $||\boldsymbol{\mu}||^2 = d$.*

Parameter Setting 1. *We choose the model parameters that allow a classifier with non-trivial standard accuracy (e.g., $\geq 1\%$) to be learned with high probability, following the Theorem 4 of (Schmidt et al. 2018):*

$$||\boldsymbol{\mu}||^2 = d, \frac{||\boldsymbol{\mu}||^2}{\sigma^2} = \sqrt{\frac{d}{n}} \gg \frac{1}{\epsilon^2}, \epsilon \in (0, \frac{1}{2}) \tag{6}$$

Given n graphs as training samples $\{\boldsymbol{x}^{(k)}, y^{(k)}\}_{k=1\ldots n}$, the estimator for Gaussian distribution parameters of node i based on likelihood can be obtained as:

$$\bar{\boldsymbol{\mu}}_i = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_i^{(k)} y_i^{(k)}. \tag{7}$$

If nodes i and j are considered jointly, the estimator can then be obtained as:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_i &= \frac{1}{n(1+a^2)}\sum_{k=1}^{n}[y_i^{(k)}\boldsymbol{x}_i^{(k)} + a(y_j^{(k)}\boldsymbol{x}_j^{(k)} - \boldsymbol{b})] & (8)\\ &= \frac{1}{1+a^2}\bar{\boldsymbol{\mu}}_i + \frac{a^2}{1+a^2}\frac{\bar{\boldsymbol{\mu}}_j}{a} - \frac{a\boldsymbol{b}}{1+a^2}. & (9) \end{aligned}$$

Then, considering the ECE measure for the two estimators $\hat{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\mu}}$,

$$\begin{aligned} \mathtt{ECE}_{\hat{\boldsymbol{\mu}}} &= \mathbb{E}_p\left[\left|\mathbb{E}[Y = 1|\hat{\mathtt{p}}(Y = 1|\boldsymbol{x}) = p] - p\right|\right] & (10)\\ &= \mathbb{E}_{v=\hat{\boldsymbol{\mu}}_i^\top \boldsymbol{x}}\left[\left|\frac{1}{e^{-\frac{2\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2}v} + 1} - \frac{1}{e^{-2v} + 1}\right|\right] & (11)\\ \mathtt{ECE}_{\bar{\boldsymbol{\mu}}} &= \mathbb{E}_{v=\bar{\boldsymbol{\mu}}_i^\top \boldsymbol{x}}\left[\left|\frac{1}{e^{-\frac{2\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2}v} + 1} - \frac{1}{e^{-2v} + 1}\right|\right] & (12) \end{aligned}$$

we have the following theorem for the expected cost minimizing (ECM) classifier defined in App. Prop. 1:

Theorem 1. *Under the above parameter setting, there exist numerical constants c_0, c_2, with $d/n > c_0$ and $a^2 > (\frac{d}{n})^{1/4}/2$,*

$$\mathtt{ECE}_{\hat{\boldsymbol{\mu}}} \leq \mathtt{ECE}_{\bar{\boldsymbol{\mu}}} \textit{with probability } \geq 1 - e^{-c_2 d/32}. \tag{13}$$

Proof. We defer the detailed proof to the appendix in (Tang et al. 2023). Here we give a sketch of the proof. According to Lemma 2-5 (as proved in the appendix), with sufficiently large $d/n > c_0$ and high correlation a^2,

$$\mathtt{p}\left(1 \geq \frac{\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \geq \frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \geq \frac{1}{2}\right) \geq 1 - e^{-c_2 d/32},$$

and $\frac{1}{e^{-\frac{2\hat{\mu}^\top \boldsymbol{\mu}}{||\hat{\mu}||^2} v}+1}$ is always closer to $\frac{1}{e^{-2v}+1}$ than $\frac{1}{e^{-\frac{2\bar{\mu}^\top \boldsymbol{\mu}}{||\bar{\mu}||^2} v}+1}$ with various v. Thus,

$$\mathtt{p}(\mathtt{ECE}_{\hat{\boldsymbol{\mu}}} \leq \mathtt{ECE}_{\bar{\boldsymbol{\mu}}}) \geq 1 - e^{-c_2 d/32}.$$

□

The above theorem indicates that by jointly considering nodes with high correlation, the calibration error can be reduced effectively. This motivates our design of SimCalib which explicitly considers the similarity between nodes at both global and local levels.

Related Work

Calibration for Standard Multi-Class Classification

The model calibration task was first proposed in 2017 (Guo et al. 2017). About this problem, works can be roughly classified as post-hoc and training based methods. Post-hoc methods calibrate pretrained nodewise classifiers in ways that preserve predictions, as featured by temperature scaling (TS) (Guo et al. 2017), ensemble temperature scaling (ETS) (Zhang, Kailkhura, and Han 2020), multi-class isotonic regression (IRM) (Zhang, Kailkhura, and Han 2020), spline calibration (Gupta et al. 2020), Dirichlet calibration (Kull et al. 2019), etc. In contrast, instead of transforming logits from a pretrained classifier, training based methods modify either the model architecture or the training process itself. A plethora of methods based on evidential theory (Sensoy, Kaplan, and Kandemir 2018), model ensembling (Lakshminarayanan, Pritzel, and Blundell 2017), adversarial calibration (Tomani and Buettner 2021) and Bayesian approach (Hernández-Lobato and Adams 2015; Wen et al. 2018) belongs to the category. Whereas training based methods provide more flexibility compared to post-hoc ones, a limitation is that they hardly promise accuracy-preserving, thereby requiring careful trade-off between accuracy and calibration performance.

GNN Calibration

Comparatively, GNN calibration is currently less explored. Teixeira et al.(2019) empirically evaluate the post-hoc model calibration techniques developed for the standard i.i.d. setting on GNN calibration, and show that such a paradigm fails in the task due to an oversight of graph structural information. Afterwards, CaGCN (Wang et al. 2021) produces nodewise temperatures by processing nodewise logits via a graph convolutional network, to account for the graph structure. Additionally, GATS (Hsu et al. 2022) experimentally points out influential factors of GNN calibration and produces nodewise temperatures with an attention-based architecture. Furthermore, Hsu et al.(2022) propose edgewise calibration metrics. Recently, uncertainty quantification is also considered via conformal prediction (Huang et al. 2023; Zargarbashi, Antonelli, and Bojchevski 2023). However, our post-hoc calibration strategy differs from all the previous works with theoretical foundation and similarity-oriented mechanisms.

Methods

In light of our theorem, we aim at exploiting nodewise similarity in the process of GNN calibration. Thus we discuss two mechanisms, namely feature and representation movement similarities in this section.

Feature Similarity Propogation

An intuitive form of feature similarity is raw feature similarity. However, we find with experiments that it aligns badly with GNN predictions, and also the subsequent calibration process, thereby performing suboptimally in GNN calibration. Thus we calculate similarity based on intermediate features from the pretrained GNN classifier G_{pre}, which is to be calibrated, because the intermediate features are better clustered and better aligned with GNN predictions (Kipf and Welling 2016). Hereafter we denote the intermediate feature map from the l-th layer of G_{pre} as $X_{pre}^{(l)}$.

A naive solution would be to feed $X_{pre}^{(l)}$ directly into another GNN G_{feat} for feature processing, in the hope that G_{feat} implicitly takes feature similarity into account and produces nodewise temperatures T:

$$T = G_{feat}(X_{pre}^{(l)}, \mathbf{A}) \in \mathbb{R}^{|V|}. \tag{14}$$

Unfortunately, this renders the number of parameters for G_{feat} highly dependent on the number of $X_{pre}^{(l)}$'s dimensions. Specifically, if, for constants h , $h' \in \mathbb{N}$, the first layer of G_{feat} projects $X_{pre}^{(l)}$ from $\mathbb{R}^h$ to $\mathbb{R}^{h'}$, then G_{feat} will contain at least $h \times h'$ parameters, which can easily result in overfitting when h gets large. Thus, we desire a feature similarity mechanism that does not rely on input feature dimension or feature semantics.

Motivated by prototypical learning (Nassar et al. 2023; Snell, Swersky, and Zemel 2017), for each class k, we first take the average feature as a classwise template,

$$\forall k \in \mathcal{Y},\ \hat{\mu}_k := \frac{1}{|\mathcal{L}_k|} \sum_{i: v_i \in \mathcal{L}_k} x_i^{(l)}, \tag{15}$$

where $\mathcal{L}_k := \{v_i \in \mathcal{L} | y_i = k\}$, and $x_i^{(l)}$ is the intermediate feature for v_i in $X_{pre}^{(l)}$. Then, we define the similarity between $x_i^{(l)}$ and templates with the assistance of a distance measure $d(\cdot, \cdot)$:

$$s_i := \mathtt{sim}(x_i^{(l)}, \{\hat{\mu}_k\}) := \zeta(\{d(x_i^{(l)}, \hat{\mu}_k)\}), \tag{16}$$

where $\zeta(x) = \frac{x}{||x||_2}$ normalizes the input vector. Naturally, we consider s_i, the feature similarity between $x_i^{(l)}$ and template features, as a proxy of the similarity between $x_i^{(l)}$ and intermediate features of all the nodes from a class. Following Lee et al.(2018), we first compute variance matrix,

$$\hat{\Sigma} := \frac{1}{|\mathcal{L}|}\sum_k \sum_{i\in\mathcal{L}_k}(x_i^{(l)} - \hat{\mu}_k)(x_i^{(l)} - \hat{\mu}_k)^T, \tag{17}$$

and then induce the Mahalanobis distance (Mahalanobis 2018) accordingly:

$$d(x_i^{(l)}, \hat{\mu}_k) := (x_i^{(l)} - \hat{\mu}_k)^T\hat{\Sigma}^{-1}(x_i^{(l)} - \hat{\mu}_k). \tag{18}$$

Finally, we feed $\{s_i\}$ and $\mathbf{A}$ to G_{feat} to propagate feature-level similarities along the graph structure, i.e.,

$$T_{feat} := G_{feat}(\{s_i\}, \mathbf{A}) \in \mathbb{R}^{|\mathcal{V}|} \tag{19}$$

where T_{feat} is the nodewise temperature estimate by feature-level similarity.

Representation Movement Similarity

Recently, Yan et al.(2022) quantify nodewise representation movement dynamics of GNNs in node classification task, whose three cases of representation movement serve as the foundation of our representation movement similarity-aware mechanism. Hereafter, we provide the necessary backgrounds.

We denote the node degree of v_i as d_i, and the homophily of v_i is defined as: $h_i := \mathbb{P}(y_i = y_j | v_j \in \mathcal{N}_i)$, in which $\mathcal{N}_i$ is v_i's neighbor set. Finally, the expected relative degree of node v_i is $\overline{r_i} := \mathbb{E}_{\mathbf{A}|d_i}(\frac{1}{d_i}\sum_{j\in\mathcal{N}_i} r_{ij}|d_i)$, where $r_{ij} := \sqrt{\frac{d_i+1}{d_j+1}}$. Then, Yan claims that node representation dynamics can be grouped as three cases:

- Case 1: when h_i is low, node representations move closer to the representations of the other class, whatever value $\overline{r_i}$ takes.
- Case 2: when h_i is high but relative degree $\overline{r_i}$ is low, node representations still move closer to the other class but not as much as in the first case.
- Case 3: only when both h_i and $\overline{r_i}$ are high, node representations tend to move away from the other class.

The existence of such dynamics can easily obscure information of logits and features, e.g. v_i of case 1 from class 1 may end up having its logits similar to that of v_j, which is of case 2 from another class. Therefore, we desire our GNN calibrator to be able to decompose effects from similar nodewise representation movement behavior and nodewise input information, producing better calibrated results. Nonetheless, one difficulty of applying the theorem is the absence of ground-truth labels during training, making h_i unaccessible. Worse still, when applied to GNNs, both h_i and $\overline{r_i}$ are unable to differentiate messages from different neighbors, limiting the calibrator's model capacity. To circumvent these problems, we approximate homophily by $\hat{z}_i \cdot \hat{z}_j$. The relative degree information is considered by $\frac{d_i+1}{d_j+1}$. Furthermore, aware of correlation between ECE and distances to training nodes (Hsu et al. 2022), we estimate nodewise homophily and apply it to graph attention:

$$\alpha_{i,j} := \underset{j\in\mathcal{N}_i}{\mathtt{softmax}}(\sigma(\frac{1}{\eta_i\eta_j}\hat{z}_i \cdot \hat{z}_j)), \tag{20}$$

where η_i is the distance from v_i to the nearest training node, and $\sigma := \mathit{LeakyReLU}$ (Xu et al. 2015). Then messages from neighbors are weighted and summed:

$$T_{move} = \mathtt{softplus}(\sum_{j\in\mathcal{N}_i}\alpha_{i,j}\eta_j(\frac{d_i+1}{d_j+1})^t\tilde{z}_j^T W), \tag{21}$$

where t is a hyperparameter, $\tilde{z}$ sorts the logits (Rahimi et al. 2020) and $W \in \mathbb{R}^{|\mathcal{Y}|}$ is a trainable parameter modeling the message from v_j.

It is worth mentioning that representation movement dynamics was initially proven to relate to the oversmoothing problem(Yan et al. 2022), which refers to the problem that node features of GNNs converge towards the same values with the increase of model depth (Rusch, Bronstein, and Mishra 2023). Thus, the proceeding mechanism implies a relationship between oversmoothing and GNN calibration.

Our Model & Calibration Properties

We formalize our GNN calibrator, SimCalib, as:

$$\begin{aligned}\forall v_i \in \mathcal{V},\ \hat{p}_i' = \omega \cdot \mathtt{softmax}(\frac{\hat{z}_i}{T_{feat}}) \\ + (1-\omega)\cdot \mathtt{softmax}(\frac{\hat{z}_i}{T_{move}})\end{aligned} \tag{22}$$

where $\omega \in (0, 1)$ is a hyperparameter balancing feature and representation movement similarities. It is obvious that SimCalib is the composition of order-preserving functions and thus accuracy-preserving.

Experiments

In this section, we empirically demonstrate the effectiveness of our proposed method and evaluate the effects of various network designs.

Experimental Setup

In the experiments, we apply the commonly used equal-width binning scheme from Guo et al.(2017): for any node subset $\mathcal{N} \subset \mathcal{V}$, samples are regrouped into M equally spaced intervals according to their confidences, formally, $B_m := \{v_i \in \mathcal{N} | \frac{m-1}{M} < \hat{c}_i \leq \frac{m}{M}\}$, to compute the expected calibration error (ECE) of the GNN:

$$\mathtt{ECE} = \sum_{m=1}^{M}\frac{|B_m|}{|\mathcal{N}|}|\mathtt{acc}(B_m) - \mathtt{conf}(B_m)|, \tag{23}$$

where $acc(B_m)$ and $conf(B_m)$ are defined as:

$$\begin{aligned}\mathtt{acc}(B_m) &= \frac{1}{|B_m|}\sum_{i:v_i\in B_m}\mathbf{1}(y_i = \hat{y}_i), \\ \mathtt{conf}(B_m) &= \frac{1}{|B_m|}\sum_{i:v_i\in B_m}\hat{c}_i.\end{aligned} \tag{24}$$

To make a fair comparison, the evaluation protocol is mainly adopted from GATS. Specifically, we first train a series of GCNs (Kipf and Welling 2016) and GATs (Veličković et al. 2017) with node classification on eight widely used datasets: Cora (McCallum et al. 2000), Citeseer (Giles, Bollacker, and Lawrence 1998), Pubmed (Sen et al. 2008), CoraFull (Bojchevski and Günnemann 2017), and the four Amazon datasets (Shchur et al. 2018). Then we train calibrators on top of the pretrained nodewise classifiers to evaluate its calibration performance. After training, we evaluate models by ECE with $M = 15$ equally sized bins. To reduce the influence of randomness, we randomly assign 15% of nodes as $\mathcal{L}$, and the rest as $\mathcal{U}$, and we repeat this assignment process with randomness five times for each dataset. Once $\mathcal{L}$ has been sampled, we use three-fold cross-validation on it. Also, in each fold we randomly initialize our models five times. Therefore, this results in a total of 75 runs for experiment, the mean and standard deviation of which are finally reported. Full implementation details are presented in the Appendix.

Performance Comparison

We benchmark SimCalib against a variety of baselines on GNN calibration tasks:

- Temperature scaling(TS) applies a global temperature to scale every nodewise logits.
- Vector scaling(VS) scales and adds a bias to each class in a class-wise manner.
- Ensemble temperature scaling(ETS) softens probabilistic outputs by learning an ensemble of uncalibrated, TS-calibrated and uniform distribution.
- Graph convolution network as a calibration function (CaGCN) uses a GCN to process logits, producing nodewise temperatures.
- Graph attention temperature scaling (GATS) identifies factors that influence GNN calibration, and addresses them with graph attention mechanism.

We also report the ECEs for uncalibrated predictions as a reference. Among the baselines, TS, VS and ETS are designed for standard i.i.d. multi-class classification. CaGCN and GATS propagate information along the graph structure, and produce separate nodewise temperatures.

For all the experiments, the pretrained GNN classifiers will be frozen, and its first-layer feature map, together with the logits, will be fed into our calibration model as inputs. We train calibrators on validation sets by minimizing NLL loss, and validate it on the training set, following the common practice (Wang et al. 2021). We provide details of comparison settings and hyperparameters in Appendix. The calibration results are summarized in Table 1. We also provide the comparisons on adaptive calibration error (ACE) in Table 2.

Overall, SimCalib consistently produces well calibrated results for all the GNN backbones on every dataset. It sets a new SOTA for all experiments, with two exceptions of GAT on Citeseer(2nd best) and GAT on CS(2nd best), on which SimCalib performs worse than GATS by at most 3%. In contrast, SimCalib's improvements are more statistically significant, reducing average ECE by 10.4% compared to GATS.

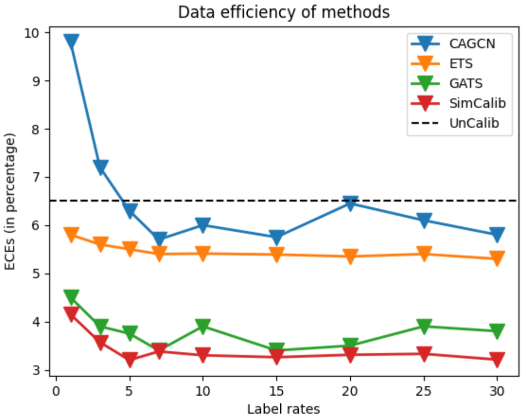

Figure 1: ECEs (%) of CaGCN, ETS, GATS and SimCalib, with different amounts of calibration data. For reference, we also plot the result for uncalibrated backbone as the dashed line.

With Wilcoxon signed test (Wilcoxon 1992) backed by scipy (Virtanen et al. 2020), we claim that our model is superior to the previous SOTA model, namely, GATS, with confidence 99.9% and $p = 7.63 \times 10^{-4}$.

Correlation between Feature Similarity and Calibration Improvement

Furthermore, we conduct experiments on CoraFull to assess the correlation between calibration improvement and nodewise similarity. We opt for CoraFull because it is the most complicated dataset of the eight, with 19,793 nodes, 126,842 edges, 70 classes and 8710 features, which makes it well representative of the real-world scenario.

In Fig. 2, we visually examine the correlation between feature similarity and calibration improvement by comparing the calibration performance of GATS, an uncalibrated GNN backbone and SimCalib. We group nodes by the Mahalanobis distances to the nearest template representation, and display the group-level ECEs as bars and ECE improvements as dashed lines. The figure shows that the miscalibration issue worsens with the decrease of feature similarity, while our calibration strategy calibrates the nodewise confidence in a consistent way. We believe that with the decrease of feature-level similarity, the samples become more outlying in the feature space, indistinguishable from samples from other classes, and thus it gets harder for the uncalibrated model to accurately tune its confidence in congruence with ground-truth prediction accuracy. In contrast, our model successfully overcomes such ambiguity in feature space by taking feature-level similarity into account, thereby consistenly calibrating samples of various feature similarity with similar expected calibration error. Therefore, this results in stronger improvement for SimCalib when fea-

Dataset	Backbone	UnCal	TS	VS	ETS	CaGCN	GATS	SimCalib
Cora	GCN	13.04±5.22	3.92±1.29	4.36±1.34	3.79±3.54	5.29±1.47	3.64±1.34	**3.32±0.99**
	GAT	23.31±1.81	3.69±0.90	3.30±1.12	3.54±1.01	4.09±1.06	3.18±0.90	**2.90±0.87**
Citeseer	GCN	10.66±5.92	5.15±1.50	4.92±1.44	4.65±1.69	6.86±1.41	4.43±1.30	**3.94±1.12**
	GAT	22.88±3.53	4.74±1.47	4.25±1.48	4.11±1.64	5.75±1.31	**3.86±1.56**	3.95±1.30
Pubmed	GCN	7.18±1.51	1.26±0.28	1.46±0.29	1.24±0.30	1.09±0.52	0.98±0.30	**0.93±0.32**
	GAT	12.32±0.80	1.19±0.36	1.00±0.32	1.20±0.32	0.98±0.31	1.03±0.32	**0.95±0.35**
Computers	GCN	3.00±0.80	2.65±0.57	2.70±0.63	2.58±0.70	1.72±0.53	2.23±0.49	**1.37±0.33**
	GAT	1.88±0.82	1.63±0.46	1.67±0.52	1.54±0.67	2.03±0.80	1.39±0.39	**1.08±0.33**
Photo	GCN	2.24±1.03	1.68±0.63	1.75±0.63	1.68±0.89	1.99±0.56	1.51±0.52	**1.36±0.59**
	GAT	2.02±1.11	1.61±0.63	1.63±0.69	1.67±0.73	2.10±0.78	1.48±0.61	**1.29±0.55**
CS	GCN	1.65±0.92	0.98±0.27	0.96±0.30	0.94±0.24	2.27±1.07	0.88±0.30	**0.81±0.30**
	GAT	1.40±1.25	0.93±0.34	0.87±0.35	0.88±0.33	2.52±1.04	**0.81±0.30**	0.83±0.32
Physics	GCN	0.52±0.29	0.51±0.19	0.48±0.16	0.52±0.19	0.94±0.51	0.46±0.16	**0.39±0.14**
	GAT	0.45±0.21	0.50±0.21	0.52±0.20	0.50±0.21	1.17±0.42	0.42 ±0.14	**0.40±0.13**
CoraFull	GCN	6.50±1.26	5.54±0.43	5.76±0.42	5.38±0.49	5.86±2.52	3.76±0.74	**3.22±0.74**
	GAT	4.73±1.39	4.00±0.50	4.17±0.43	3.89±0.56	6.55±3.69	3.54±0.63	**3.40±0.91**

Table 1: GNN calibration results of SimCalib and other baseline approaches in terms of ECE (%), where lower is better. For each experiment, the best result is displayed in bold. UnCal stands for the uncalibrated backbones.

Backbone	Uncal	GATS	SimCalib
GCN	16.23	15.80	**15.44**
GAT	16.27	15.52	**15.17**

Table 2: Mean and standard variation of ACE (%) of SimCalib and baselines on CoraFull.

ture similarity gets weaker. Although we discover a similar pattern for GATS, its ECE improvement is weaker than SimCalib when features become dissimilar. We attribute the performance gap to the feature-similarity-aware mechanism. The observation aligns with our theoretical hypothesis in that it suggests that feature similarity indeed plays a critical role in GNN calibration.

Data-Efficiency and Expressivity of SimCalib

In addition, we analyze the data-efficiency and expressivity of SimCalib for GNN calibration. For this, we reuse the GCN classifier pretrained on CoraFull, and compare the expected calibration errors between baselines and SimCalib, with different amounts of calibration data. The results are shown in Fig. 1. From the figure, we draw that SimCalib is both data-efficient and expressive. SimCalib does not require a lot of labels to perform decently, consistently outperforming all the baselines under all label rates. Moreover, SimCalib also expresses robustness to label rates.

Ablation Study

To understand the effects of the two similarity-oriented mechanisms, we conduct a thorough ablation study in this section. The results are shown in Table 3 and Table 4. Overall, each mechanism plays a critical role in GNN calibration and removing any will in general decrease performance while increasing variances.

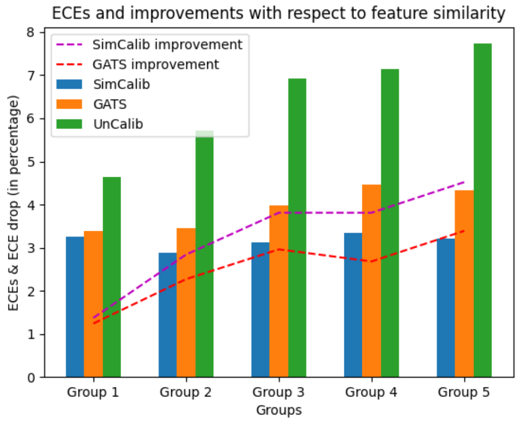

Figure 2: Figure of correlation between feature-level similarity and calibration improvement. We also illustrate the calibration improvement in terms of ECE (%) with the dashed lines. The groups are sorted in an ascending order with respect to nodewise Mahalanobis distances to the nearest templates.

Effect of feature similarity In order to decompose the effects from number of trainable parameters and feature-similarity mechanism, we investigate the performance of two different models: SimCalib with only representation movement similarity(SimCalib$_m$) and SimCalib with s_i replaced by G_{pre}'s output logits(SimCalib$_l$). Comparing SimCalib$_m$ with SimCalib$_l$, we see that the calibration performance slightly improves with more trainable parameters. However, the improvements are rather moderate, unable to match the performance of SimCalib. We hypothesize that since logits information has already been integrated into the calibration process in the nodewise representation move-

Dataset	Backbone	SimCalib$_l$	SimCalib$_m$	SimCalib$_h$	SimCalib$_r$	SimCalib$_n$	SimCalib
Cora	GCN	3.58±0.97	3.86±1.78	3.30±1.76	3.87 ± 1.77	4.16 ±1.79	3.32±0.99
	GAT	2.88±0.88	3.40±1.32	3.02±1.27	3.52 ±1.26	3.68±1.39	2.90±0.87
Citeseer	GCN	4.24±1.61	4.57±1.92	4.36±1.84	4.75±1.89	4.96 ±1.93	3.94±1.12
	GAT	4.22±1.51	4.41±2.29	3.93±2.25	4.65±2.21	4.73±2.20	3.95±1.30
Photo	GCN	1.50±0.50	1.42±0.63	1.38±0.71	1.52 ±0.65	1.58 ± 0.71	1.36±0.59
	GAT	1.39±0.58	1.44±0.55	1.36±0.54	1.55±0.54	1.52±0.59	1.29±0.55
CoraFull	GCN	3.47±0.79	3.91±0.79	3.76±0.82	3.18±0.73	3.33±0.90	3.22±0.74
	GAT	3.84±0.80	3.27±0.84	3.59±0.88	3.35±0.81	3.21±0.84	3.40±0.91

Table 3: Ablation study results in terms of ECE (%). Overall, all designs are critical and removing any of them results a general decrease in performance.

Backbone	w=0.5	w=0.6	w=0.8
GCN	3.34±0.84	3.22±0.74	3.32±0.69
GAT	3.52±1.02	3.40±0.91	3.46±0.79

Table 4: Mean and standard variation of ECE (%) of SimCalib on CoraFull with different w.

ment similarity, adding an extra branch of logits propagation only helps GNN calibration by introducing more parameters.

Effect of representation movement similarity Our representation movement similarity mechanism consists of two aspects of network designs, i.e. homophily term $\hat{z}_i \cdot \hat{z}_j$ and relative degree $(\frac{d_j+1}{d_i+1})^t$, therefore we design three models to analyze the effects of various components in the representation movement similarity mechanism. Particularly, we test the calibration performance of SimCalib$_h$ that disables relative degree, SimCalib$_r$ that disables homophily, and SimCalib$_n$ in which neither takes effects. We can easily draw from the experiments that the integrity of representation movement similarity mechanism is important to calibration performance and removing of any results in a worsened GNN calibrator. We attribute the performance drop to the inability of the calibrator to decompose effects from nodewise input information and effects from representation movement.

Conclusions

In this work, we provide theoretical analysis on the graph calibraion problem and prove that nodewise similarity plays an important role in the solution. We consider feature and nodewise representation movement similarities, which are quantified by Gaussian-induced Mahalanobis distances and homophily & relative degrees, respectively. Based on the mechanisms, we propose a novel calibrator, SimCalib, tailored for GNN calibration. SimCalib is data-efficient, expressive and accuracy-preserving at the same time. Our extensive experiments demonstrate the effectiveness of SimCalib by achieving state-of-the-art performances for GNN calibration on various datasets and for different backbones. Moreover, our experiments exhibit a correlational relationship between nodewise similarity and calibration improvement, in alignment with our theoretical results. Our work has the potential to be employed in cost-sensitive scenarios. Additionally our work is the first to reveal a non-trivial relationship between oversmoothing and GNN calibration problems.

Acknowledgements

This work is supported by National Natural Science Foundation of China (62076144, 62306260), Shenzhen Science and Technology Program (WDZC20220816140515001, JCYJ20220818101014030), and the Center for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission's InnoHK Scheme.

References

Bojchevski, A.; and Günnemann, S. 2017. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Duchi, J. C.; and Liang, P. S. 2019. Unlabeled data improves adversarial robustness. *Advances in neural information processing systems*, 32.

Dezvarei, M.; Tomsovic, K.; Sun, J. S.; and Djouadi, S. M. 2023. Graph Neural Network Framework for Security Assessment Informed by Topological Measures. *arXiv preprint arXiv:2301.12988*.

Fan, W.; Ma, Y.; Li, Q.; He, Y.; Zhao, E.; Tang, J.; and Yin, D. 2019. Graph neural networks for social recommendation. In *The world wide web conference*, 417–426.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 89–98.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

Gupta, K.; Rahimi, A.; Ajanthan, T.; Mensink, T.; Sminchisescu, C.; and Hartley, R. 2020. Calibration of neural networks using splines. *arXiv preprint arXiv:2006.12800*.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Hernández-Lobato, J. M.; and Adams, R. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, 1861–1869. PMLR.

Hsu, H. H.-H.; Shen, Y.; and Cremers, D. 2022. A graph is more than its nodes: Towards structured uncertainty-aware learning on graphs. *arXiv preprint arXiv:2210.15575*.

Hsu, H. H.-H.; Shen, Y.; Tomani, C.; and Cremers, D. 2022. What Makes Graph Neural Networks Miscalibrated? *Advances in Neural Information Processing Systems*, 35: 13775–13786.

Huang, K.; Jin, Y.; Candes, E.; and Leskovec, J. 2023. Uncertainty quantification over graph with conformalized graph neural networks. *arXiv preprint arXiv:2305.14535*.

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

Kull, M.; Perello Nieto, M.; Kängsepp, M.; Silva Filho, T.; Song, H.; and Flach, P. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *Advances in neural information processing systems*, 32.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in neural information processing systems*, 31.

Li, S.; and Schramm, T. 2023. Spectral clustering in the Gaussian mixture block model. *arXiv preprint arXiv:2305.00979*.

Liu, Q.; Zhu, W.; Jia, X.; Ma, F.; and Gao, Y. 2022. Fluid simulation system based on graph neural network. *arXiv preprint arXiv:2202.12619*.

Mahalanobis, P. C. 2018. On the generalized distance in statistics. *Sankhyā: The Indian Journal of Statistics, Series A (2008-)*, 80: S1–S7.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3: 127–163.

Naeini, M. P.; Cooper, G.; and Hauskrecht, M. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Nassar, I.; Hayat, M.; Abbasnejad, E.; Rezatofighi, H.; and Haffari, G. 2023. PROTOCON: Pseudo-label Refinement via Online Clustering and Prototypical Consistency for Efficient Semi-supervised Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11641–11650.

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

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703.

Rahimi, A.; Shaban, A.; Cheng, C.-A.; Hartley, R.; and Boots, B. 2020. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33: 13456–13467.

Rusch, T. K.; Bronstein, M. M.; and Mishra, S. 2023. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*.

Schmidt, L.; Santurkar, S.; Tsipras, D.; Talwar, K.; and Madry, A. 2018. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine*, 29(3): 93–93.

Sensoy, M.; Kaplan, L.; and Kandemir, M. 2018. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31.

Shchur, O.; Mumme, M.; Bojchevski, A.; and Günnemann, S. 2018. Pitfalls of graph neural network evaluation. *arXiv preprint arXiv:1811.05868*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Tang, B.; Wu, Z.; Wu, X.; Huang, Q.; Chen, J.; Lei, S.; and Meng, H. 2023. SimCalib: Graph Neural Network Calibration based on Similarity between Nodes. arXiv:2312.11858.

Teixeira, L.; Jalaian, B.; and Ribeiro, B. 2019. Are graph neural networks miscalibrated? *arXiv preprint arXiv:1905.02296*.

Tomani, C.; and Buettner, F. 2021. Towards trustworthy predictions from deep neural networks with fast adversarial calibration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9886–9896.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Virtanen, P.; Gommers, R.; Oliphant, T. E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S. J.; Brett, M.; Wilson, J.; Millman, K. J.; Mayorov, N.; Nelson, A. R. J.; Jones, E.; Kern, R.; Larson, E.; Carey, C. J.; Polat, İ.; Feng, Y.; Moore, E. W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E. A.; Harris, C. R.; Archibald, A. M.; Ribeiro, A. H.; Pedregosa, F.; van Mulbregt, P.; and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17: 261–272.

Wang, X.; Liu, H.; Shi, C.; and Yang, C. 2021. Be confident! towards trustworthy graph neural networks via confidence calibration. *Advances in Neural Information Processing Systems*, 34: 23768–23779.

Wen, Y.; Vicol, P.; Ba, J.; Tran, D.; and Grosse, R. 2018. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*.

Wilcoxon, F. 1992. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*, 196–202. Springer.

Xu, B.; Wang, N.; Chen, T.; and Li, M. 2015. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Yan, Y.; Hashemi, M.; Swersky, K.; Yang, Y.; and Koutra, D. 2022. Two sides of the same coin: Heterophily and oversmoothing in graph convolutional neural networks. In *2022 IEEE International Conference on Data Mining (ICDM)*, 1287–1292. IEEE.

Zargarbashi, S. H.; Antonelli, S.; and Bojchevski, A. 2023. Conformal Prediction Sets for Graph Neural Networks.

Zhang, J.; Kailkhura, B.; and Han, T. Y.-J. 2020. Mix-n-match: Ensemble and compositional methods for uncertainty calibration in deep learning. In *International conference on machine learning*, 11117–11128. PMLR.

Zhang, L.; Deng, Z.; Kawaguchi, K.; and Zou, J. 2022a. When and how mixup improves calibration. In *International Conference on Machine Learning*, 26135–26160. PMLR.

Zhang, Z.; Chen, L.; Zhong, F.; Wang, D.; Jiang, J.; Zhang, S.; Jiang, H.; Zheng, M.; and Li, X. 2022b. Graph neural network approaches for drug-target interactions. *Current Opinion in Structural Biology*, 73: 102327.

Appendix

Proofs

In this section, we illustrate our mathematic derivation of Thearom 1. The proof may seem horiffying, but the idea is simple, which is to compare the ECE of joint optimization and that of separated optimization based on probability concentation inequalities (of Gaussian and Chi-square variables). Then,

Proposition 1. *The learned model predicts $\hat{\boldsymbol{y}}_i = 1$ whenever $\hat{\boldsymbol{\mu}}_i \boldsymbol{x}_i \geq 0$.*

Proof. According to expected cost minimum (ECM) with equal mis-classification costs, the learned model predicts $\hat{\boldsymbol{y}}_i = 1$ if $\mathtt{p}(\boldsymbol{x}_i | \hat{\boldsymbol{\mu}}_i, \sigma_i \boldsymbol{I}) \geq \mathtt{p}(\boldsymbol{x}_i | -\hat{\boldsymbol{\mu}}_i, \sigma_i \boldsymbol{I})$, i.e., $\hat{\boldsymbol{\mu}}_i \boldsymbol{x}_i \geq 0$. □

Lemma 1. *For the learned model, the ECE measure is*

$$\mathtt{ECE} = \mathbb{E}_{v=\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}}\left[\left|\frac{1}{e^{-\frac{2\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2}v}+1} - \frac{1}{e^{-2v}+1}\right|\right] \tag{25}$$

Proof. Since $\hat{\mathtt{p}}(\mathtt{Y}=\mathtt{1}|\mathrm{x}; \hat{\boldsymbol{\mu}}) = \frac{1}{\mathsf{e}^{-2\hat{\boldsymbol{\mu}}^\top \mathrm{x}}+1}$,

$$\begin{aligned}
& \mathbb{E}_p[Y = 1 | \hat{\mathtt{p}}(\mathtt{Y}=\mathtt{1}|\mathrm{x}; \hat{\boldsymbol{\mu}}) = \mathtt{p}] && (26)\\
= \quad & \mathbb{E}_{v=\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}}[Y = 1 | \hat{\mathtt{p}}(\mathtt{Y}=\mathtt{1}|\mathrm{x}; \hat{\boldsymbol{\mu}}) = \tfrac{1}{\mathsf{e}^{-2\mathsf{v}+1}}] && (27)\\
= \quad & \mathbb{E}_{v=\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}}[Y = 1 | \hat{\boldsymbol{\mu}}^\top \boldsymbol{x} = v] && (28)\\
= \quad & \tfrac{\mathtt{p}(\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}=v|Y=1)}{\mathtt{p}(\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}=v|Y=1)+\mathtt{p}(\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}=v|Y=-1)} && (29)\\
= \quad & \tfrac{1}{e^{-\frac{2\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2}v}+1} && (30)
\end{aligned}$$

thus,

$$\begin{aligned}
\mathtt{ECE} \quad = \quad & \mathbb{E}_p\left[\left|\mathbb{E}[Y = 1 | \hat{\mathtt{p}}(\mathtt{Y}=\mathtt{1}|\mathrm{x}; \hat{\boldsymbol{\mu}}) = \mathtt{p}] - \mathtt{p}\right|\right] && (31)\\
= \quad & \mathbb{E}_{v=\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}}\left[\left|\mathbb{E}[Y = 1 | \hat{\mathtt{p}}(\mathtt{Y}=\mathtt{1}|\mathrm{x}) = \frac{1}{\mathsf{e}^{-2\mathsf{v}}+1}]\right.\right.\\
- \quad & \left.\left.\frac{1}{e^{-2v}+1}\right|\right]\\
= \quad & \mathbb{E}_{v=\hat{\boldsymbol{\mu}}^\top \boldsymbol{x}}\left[\left|\frac{1}{e^{-\frac{2\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2}v}+1} - \frac{1}{e^{-2v}+1}\right|\right] && (32)
\end{aligned}$$

□

where $\hat{\mathtt{p}}$ is the probability estimation from the learned model.

Lemma 2. *There exists numerical constant c_1, when d/n is sufficiently large, $\frac{\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \leq 1$, with high probability,*

$$\mathtt{p}\Big(\frac{\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \leq 1\Big) \geq 1 - e^{c_1 d/32} \tag{33}$$

Proof. Let $\boldsymbol{\epsilon}_i = \bar{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i$, $\boldsymbol{\epsilon}_j = \bar{\boldsymbol{\mu}}_j - \boldsymbol{\mu}_j$, then $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, \frac{\sigma^2}{n}\boldsymbol{I})$, $\boldsymbol{\epsilon}_j =\sim \mathcal{N}(0, \frac{\sigma^2}{n}\boldsymbol{I})$. Let $\boldsymbol{\delta} = \frac{1}{1+a^2}\boldsymbol{\epsilon}_i + \frac{a}{1+a^2}\boldsymbol{\epsilon}_j \sim \mathcal{N}(0, \frac{\sigma^2}{(1+a^2)n})$, from Eq. 9, we have $\hat{\boldsymbol{\mu}}_i = \boldsymbol{\mu}_i + \frac{1}{1+a^2}\boldsymbol{\epsilon}_i + \frac{a}{1+a^2}\boldsymbol{\epsilon}_j = \boldsymbol{\mu}_i + \boldsymbol{\delta}$, and $||\boldsymbol{\delta}||^2 \sim \frac{\sigma^2}{(1+a^2)n}\chi^2_d$.

Given $\frac{\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} = \frac{||\boldsymbol{\mu}||^2 + \boldsymbol{\mu}^\top \boldsymbol{\delta}}{||\boldsymbol{\mu}||^2 + ||\boldsymbol{\delta}||^2 + 2\boldsymbol{\mu}^\top \boldsymbol{\delta}} = \frac{1}{2} + \frac{1}{2}\frac{||\boldsymbol{\mu}||^2 - ||\boldsymbol{\delta}||^2}{||\boldsymbol{\mu}||^2 + ||\boldsymbol{\delta}||^2 + 2\boldsymbol{\mu}^\top \boldsymbol{\delta}}$, according to concentration inequalities $\mathtt{p}(||\boldsymbol{\delta}||^2 \leq \frac{d\sigma^2}{2(1+a^2)n}) \leq e^{-d/16}$, $\mathtt{p}(\boldsymbol{\mu}^\top \boldsymbol{\delta} \leq -\frac{d\sigma}{4(1+a^2)^{1/2}n^{1/2}}) \leq e^{-d/32}$, we have

$$\begin{aligned}
& \mathtt{p}\left(\frac{\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \geq \frac{1}{2} + \frac{1}{2}\frac{d - \frac{d\sigma^2}{2(1+a^2)n}}{d + \frac{d\sigma^2}{2(1+a^2)n} - 2\frac{d\sigma}{4(1+a^2)^{1/2}n^{1/2}}}\right)\\
\leq \quad & \mathtt{p}(||\boldsymbol{\delta}||^2 \leq \tfrac{d\sigma^2}{2(1+a^2)n}) + \mathtt{p}(\boldsymbol{\mu}^\top \boldsymbol{\delta} \leq -\tfrac{d\sigma}{4(1+a^2)^{1/2}n^{1/2}})\\
\leq \quad & 2e^{-d/16} + e^{-d/32}\\
\leq \quad & e^{-c_1 d/32}
\end{aligned}$$

thus,

$$\mathtt{p}(\frac{\hat{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \leq 1) \geq 1 - e^{-c_1 d/32} \tag{34}$$

with $(\frac{d}{n})^{1/2} \geq \frac{1+a^2}{4}$. □

Lemma 3. *With sufficiently large d/n, $\frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \geq 1/2$, with high probability,*

$$\mathtt{p}(\frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \geq 1/2) \geq 1 - 3e^{d/8} \tag{35}$$

Proof. Given $\frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} = \frac{||\boldsymbol{\mu}||^2 + \boldsymbol{\mu}^\top \boldsymbol{\epsilon}}{||\boldsymbol{\mu}||^2 + ||\boldsymbol{\epsilon}||^2 + 2\boldsymbol{\mu}^\top \boldsymbol{\epsilon}} = \frac{1}{2} + \frac{1}{2}\frac{||\boldsymbol{\mu}||^2 - ||\boldsymbol{\epsilon}||^2}{||\boldsymbol{\mu}||^2 + ||\boldsymbol{\epsilon}||^2 + 2\boldsymbol{\mu}^\top \boldsymbol{\epsilon}}$, $||\boldsymbol{\epsilon}||^2 \sim \frac{\sigma^2}{n}\chi^2_d$, according to concentration inequalities $\mathtt{p}(||\boldsymbol{\epsilon}||^2 \leq \frac{d\sigma^2}{4n}) \leq e^{-d/8}$, $\mathtt{p}(||\boldsymbol{\epsilon}||^2 \geq \frac{2d\sigma^2}{n}) \leq e^{-d/8}$, $\mathtt{p}(\boldsymbol{\mu}^\top \boldsymbol{\epsilon} \geq \frac{d\sigma}{2n^{1/2}}) \leq e^{-d/8}$, $\mathtt{p}(\boldsymbol{\mu}^\top \boldsymbol{\epsilon} \leq -\frac{d\sigma}{2n^{1/2}}) \leq e^{-d/8}$, we have

$$\begin{aligned}
& \mathtt{p}(\frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \leq \frac{1}{2} + \frac{1}{2}\frac{d - \frac{2d\sigma^2}{n}}{d + \frac{d\sigma^2}{4n} - 2\frac{d\sigma}{2n^{1/2}}})\\
\leq \quad & \mathtt{p}(||\boldsymbol{\epsilon}||^2 \geq \tfrac{2d\sigma^2}{n}) + \mathtt{p}(||\boldsymbol{\epsilon}||^2 \leq \tfrac{d\sigma^2}{4n}) + \mathtt{p}(\boldsymbol{\mu}^\top \boldsymbol{\epsilon} \leq -\tfrac{d\sigma}{2n^{1/2}})\\
\leq \quad & 3e^{-d/8}
\end{aligned}$$

hence,

$$\mathtt{p}\Big(\frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \geq \frac{1}{2}\Big) \geq 1 - 3e^{-d/8} \tag{36}$$

with sufficiently large $(\frac{d}{n})^{1/2} \geq n$. □

Lemma 4. *With the parameter setting and sufficiently large d/n,*

$$\mathtt{p}\left(\frac{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \leq \left(2 + \frac{\frac{d\sigma^2}{4n} - d}{d + \frac{d\sigma}{2n^{1/2}}}\right)^{-1}\right) \geq 1 - 2e^{-d/8} \tag{37}$$

Proof. $\frac{||\bar{\boldsymbol{\mu}}||^2}{\bar{\boldsymbol{\mu}}^\top \boldsymbol{\mu}} = \frac{||\boldsymbol{\mu}||^2 + ||\boldsymbol{\epsilon}||^2 + 2\boldsymbol{\mu}^\top \boldsymbol{\epsilon}}{||\boldsymbol{\mu}||^2 + \boldsymbol{\mu}^\top \boldsymbol{\epsilon}} = 2 + \frac{||\boldsymbol{\epsilon}||^2 - ||\boldsymbol{\mu}||^2}{||\boldsymbol{\mu}||^2 + \boldsymbol{\mu}^\top \boldsymbol{\epsilon}}$.

$$\mathtt{p}\left(\frac{||\bar{\boldsymbol{\mu}}||^2}{\bar{\boldsymbol{\mu}}^\top\boldsymbol{\mu}} \leq 2 + \frac{\frac{d\sigma^2}{4n}-d}{d+\frac{d\sigma}{2n^{1/2}}}\right) \tag{38}$$

$$\leq \quad \mathtt{p}(||\boldsymbol{\epsilon}||^2 \leq \tfrac{d\sigma^2}{4n}) + \mathtt{p}(\boldsymbol{\mu}^\top\boldsymbol{\epsilon} \geq \tfrac{d\sigma}{2n^{1/2}}) \tag{39}$$

$$\leq \quad 2e^{-d/8} \tag{40}$$

Hence,

$$\frac{\bar{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \leq \left(2 + \frac{\frac{d\sigma^2}{4n}-d}{d+\frac{d\sigma}{2n^{1/2}}}\right)^{-1} \tag{41}$$

with probability at least $1-2e^{-d/8}$. □

Lemma 5. *With sufficiently large d/n and a^2,*

$$\mathtt{p}\left(\frac{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \geq \left(2 + \frac{\frac{d\sigma^2}{4n}-d}{d+\frac{d\sigma}{2n^{1/2}}}\right)^{-1}\right) \geq 1-2e^{-d/8} \tag{42}$$

Proof. $\frac{||\hat{\boldsymbol{\mu}}||^2}{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}} = \frac{||\boldsymbol{\mu}||^2+||\boldsymbol{\delta}||^2+2\boldsymbol{\mu}^\top\boldsymbol{\delta}}{||\boldsymbol{\mu}||^2+\boldsymbol{\mu}^\top\boldsymbol{\delta}} = 2 + \frac{||\boldsymbol{\delta}||^2-||\boldsymbol{\mu}||^2}{||\boldsymbol{\mu}||^2+\boldsymbol{\mu}^\top\boldsymbol{\delta}}$

$$\mathtt{p}\left(\frac{||\hat{\boldsymbol{\mu}}||^2}{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}} \geq 2 + \frac{\frac{2d\sigma^2}{(1+a^2)n}-d}{d-\frac{d\sigma}{2(1+a^2)^{1/2}n^{1/2}}}\right) \tag{43}$$

$$\leq \quad \mathtt{p}(||\boldsymbol{\delta}||^2 \geq \tfrac{2d\sigma^2}{(1+a^2)n}) + \mathtt{p}(\boldsymbol{\mu}^\top\boldsymbol{\delta} \leq \tfrac{-d\sigma}{2(1+a^2)^{1/2}n^{1/2}}) \tag{44}$$

$$\leq \quad 2e^{-d/8} \tag{45}$$

with $(1+a^2)^{1/2} \geq \frac{t^2-4+\sqrt{16+256t+248t^2+64t^3+t^4}}{4t+8} \geq t/2, t = (d/n)^{1/4}$, we have $\frac{\frac{2d\sigma^2}{(1+a^2)n}-d}{d-\frac{d\sigma}{2(1+a^2)^{1/2}n^{1/2}}} \leq \frac{\frac{d\sigma^2}{4n}-d}{d+\frac{d\sigma}{2n^{1/2}}}$ Hence, with sufficient large a^2, i.e., large linear correlation between nodes,

$$\mathtt{p}\left(\frac{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \geq \left(2 + \frac{\frac{d\sigma^2}{4n}-d}{d+\frac{d\sigma}{2n^{1/2}}}\right)^{-1}\right) \geq 1-2e^{-d/8}. \tag{46}$$

□

Theorem 1. *Under the above parameter setting, there exist numerical constants c_0, c_2, with $d/n > c_0$ and $a^2 > (\frac{d}{n})^{1/4}/2$,*

$$\mathtt{ECE}_{\hat{\boldsymbol{\mu}}} \leq \mathtt{ECE}_{\bar{\boldsymbol{\mu}}} \textit{ with probability } \geq 1-e^{-c_2 d/32}. \tag{47}$$

Proof. According to Lemma 2, with sufficiently large $(\frac{d}{n})^{1/2} \geq \frac{1+a^2}{4}$, there exists numerical constant c_1, such that

$$\mathtt{p}\left(\frac{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \leq 1\right) \geq 1-e^{-c_1 d/32}.$$

According to Lemma 3, with sufficiently large $(\frac{d}{n})^{1/2} \geq n$,

$$\mathtt{p}\left(\frac{\bar{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \geq \frac{1}{2}\right) \geq 1-3e^{-d/8}$$

From Lemma 4 and 5, with sufficiently large $a^2 > (\frac{d}{n})^{1/4}/2$,

$$\mathtt{p}\left(\frac{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \geq \frac{\bar{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2}\right) \geq 1-4e^{-d/8}$$

hence, with sufficiently large $(\frac{d}{n})^{1/2} \geq c_0$,

$$\mathtt{p}\left(1 \geq \frac{\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2} \geq \frac{\bar{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2} \geq \frac{1}{2}\right) \geq 1-e^{-c_1 d/32}-7e^{-d/8},$$

and $\frac{1}{e^{-\frac{2\hat{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\hat{\boldsymbol{\mu}}||^2}v}+1}$ is always closer to $\frac{1}{e^{-2v}+1}$ than $\frac{1}{e^{-\frac{2\bar{\boldsymbol{\mu}}^\top\boldsymbol{\mu}}{||\bar{\boldsymbol{\mu}}||^2}v}+1}$ with various v. Thus,

$$\mathtt{p}(\mathtt{ECE}_{\hat{\boldsymbol{\mu}}} \leq \mathtt{ECE}_{\bar{\boldsymbol{\mu}}}) \geq 1-e^{-c_2 d/32}.$$

□

According to Theorem 1, considering the high correlation by jointly optimizing likelihoods of correlated nodes, leads to lower ECE than separately optimizing likelihoods, with high probability $\geq 1-e^{-c_2 d/32}$. For example, given 4 graphs ($n=4$) with 64-dim ($d=64$) features and linear correlation coefficients $|a| = 2.66$, the probability ≥ 0.84.

Implementation Details

Throughout all the experiments, we fix the global random seed to be 10, remaining the same with GATS. The seed eliminates randomness from python, numpy, pytorch and cuda. Our experiments are run on an Ubuntu 20.04 operating system, with a Nvidia V100 GPU, 64GB RAM and i9-13900K CPU. We mainly base our code on pytorch (Paszke et al. 2019) 1.12.1 and torch geometric 2.0.1. In all the experiments, G_{feat} is a GNN or GAT while G_{move} implements equation21 with at most 8 heads, although we find 1 or 2 heads are sufficient for most experiments. For each experiment, we do a small grid search on validation set to determine w and t, which can take values from $\{0.6, 0.8, 0.9\}$ and $\{0.3, 0.5, 1.0\}$ respectively. Following GATS, we train pretrained classifiers on Cora, Citeseer and Pubmed with a weight decay of 5e-4, and none on other datasets. We also conduct a coarse grid search to identify learning rate and number of heads on each dataset. All the hyperparameters are provided as a config file in our code appendix for reproducibility.

We train GATS and other baselines with the open-sourced code of GATS, and find that the outcomes do not exhibit any statistically significant difference with the results reported by GATS. Thus we adopt the reported performance from GATS as our baselines. It is worth mentioning that our network consumes twice the amount of trainable parameters as GATS, but we cannot report the performance of GATS with the same number of parameters because GATS gets its performance declined with twice the number of attention heads (Hsu et al. 2022).

Data efficiency of SimCalib

In Fig. 3, we evaluate the calibration performance of SimCalib and two baselines when applied to GAT. The uncalibrated ECE is also plotted as a dashed line as a reference. We omit CaGCN here as its poor calibration results make the performance gaps across other calibrators visually indistinguishable. From the figure, we find a slightly different phenomenon from that shown in Fig. 1, in that SimCalib performs worse (although still competitively) than ETS and

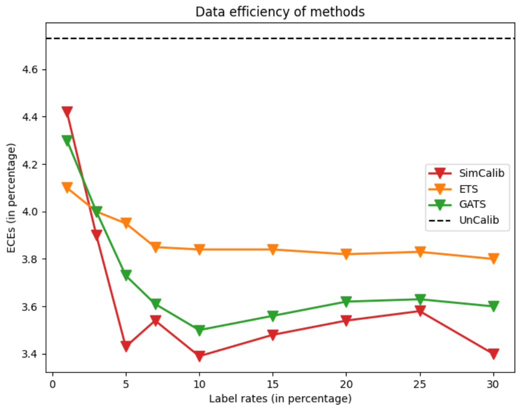

Figure 3: ECEs of different calibrators when applied to GAT.

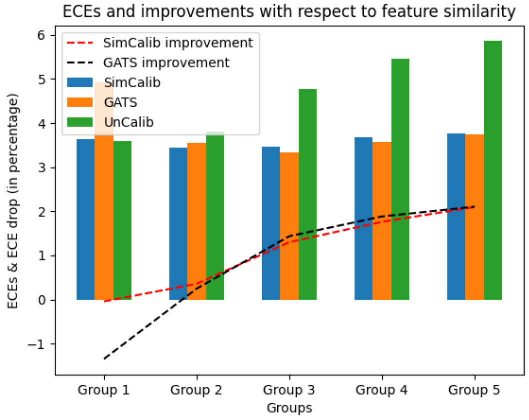

Figure 4: Correlation between feature similarity and calibration improvements for GAT backbone.

GATS at extremely low label rate (1%), but catches up and outperfoms the other baselines immediately. We attribute the inferior performance of SimCalib at low label rate to containing more trainable parameters. SimCalib performs decently even with a small amount of data. Noticeably, although SimCalib takes the form of model ensembling, it significantly outperforms the other ensembling baseline, namely ETS, at most label rates, which verifies the effectiveness of our information-blending design paradigm. The figure validates that our GNN calibrator consistently produces well calibrated confidences for various backbones. Also, SimCalib is expressive for its superior performance at larger label rates.

Feature similarity and calibration improvement

In Fig. 4, we illustrate the correlation between feature similarity and calibration performance & improvement for SimCalib and GATS, when applied to a pretrained GAT on CoraFull. The uncalibrated ECE worsens with increasing dissimilarity. Also, both GNN calibrators exhibit higher improvements with more dissimilarity. However, whereas SimCalib and GATS perform similarly across 4 out of the 5 groups, SimCalib outperforms GATS at the extremely low feature-similarity scenario, where GATS tangibly reduces calibration performance compared to the uncalibrated backbone. We believe that the design of feature similarity awareness renders SimCalib robust to feature similarities, as shown in the figure.

Reliability visualization

In this section, we provide reliability diagrams and confidence distributions for SimCalib and uncalibrated backbones so that readers can readily assess the improvements of SimCalib. Clearly, we can see that SimCalib consistently calibrates pretrained GNN classifiers.

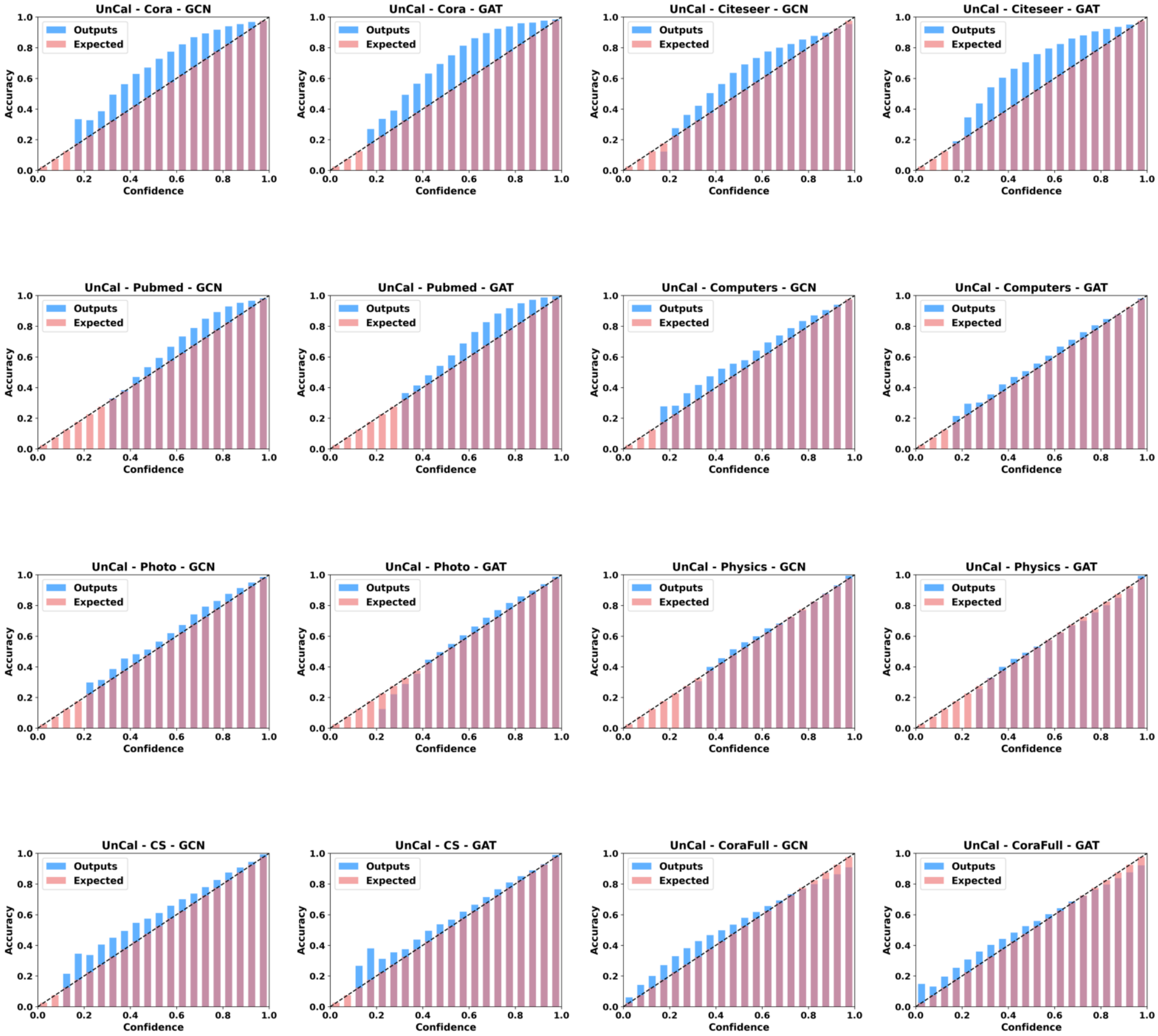

Figure 5: The reliability diagram for uncalibrated GNN classifiers. The horizontal axis represents confidences while the vertical axis is group-wise accuracy. For most datasets, we can see the underconfidence problem of GNNs.

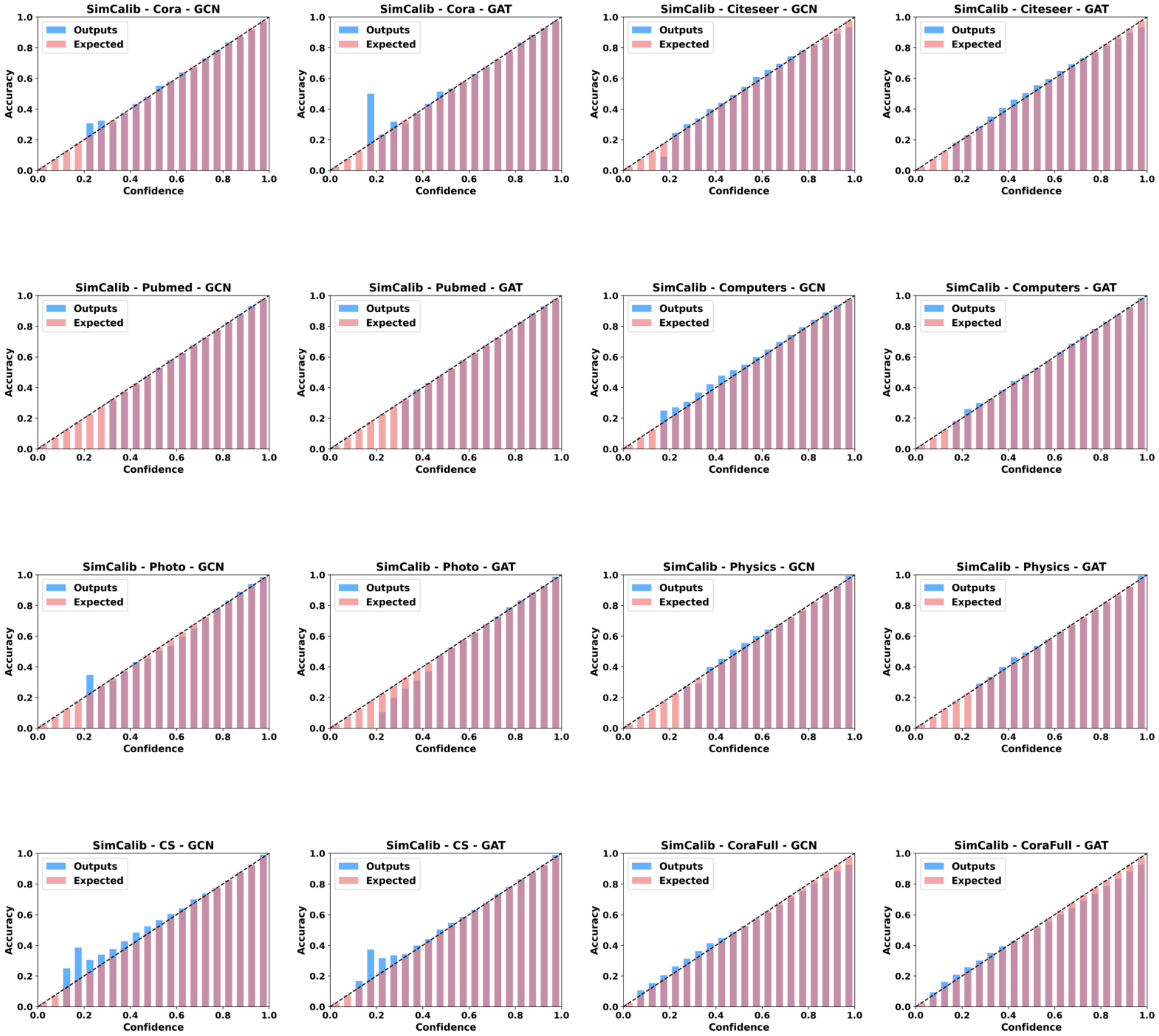

Figure 6: The reliability diagram for uncalibrated GNN classifiers. The horizontal axis represents confidences while the vertical axis is group-wise accuracy. Compared with the previous diagram, one can verify the effectiveness of SimCalib.

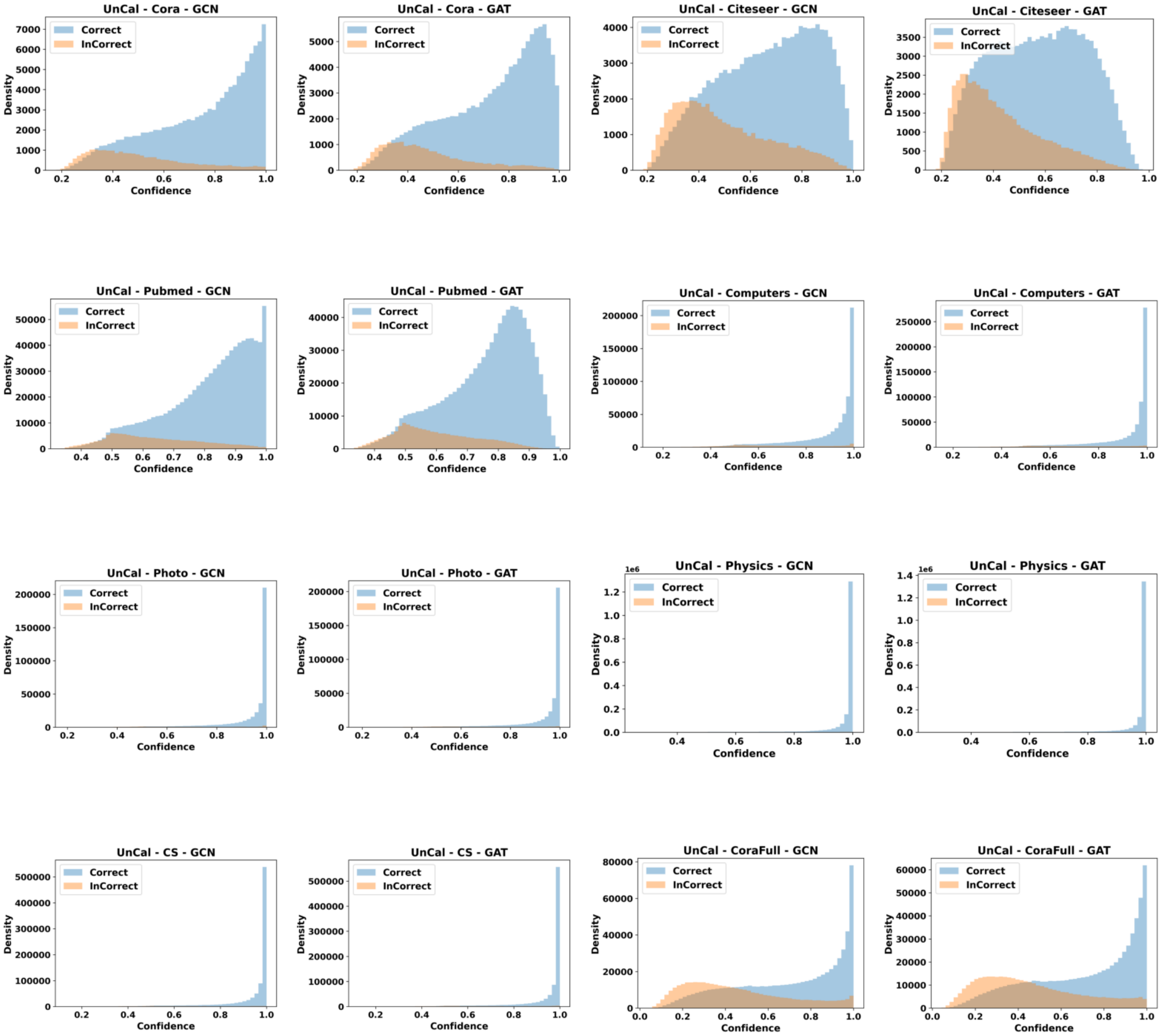

Figure 7: The confidence distributions for uncalibrated GNN classifiers. The horizontal axis represents confidences while the vertical axis is sample density.

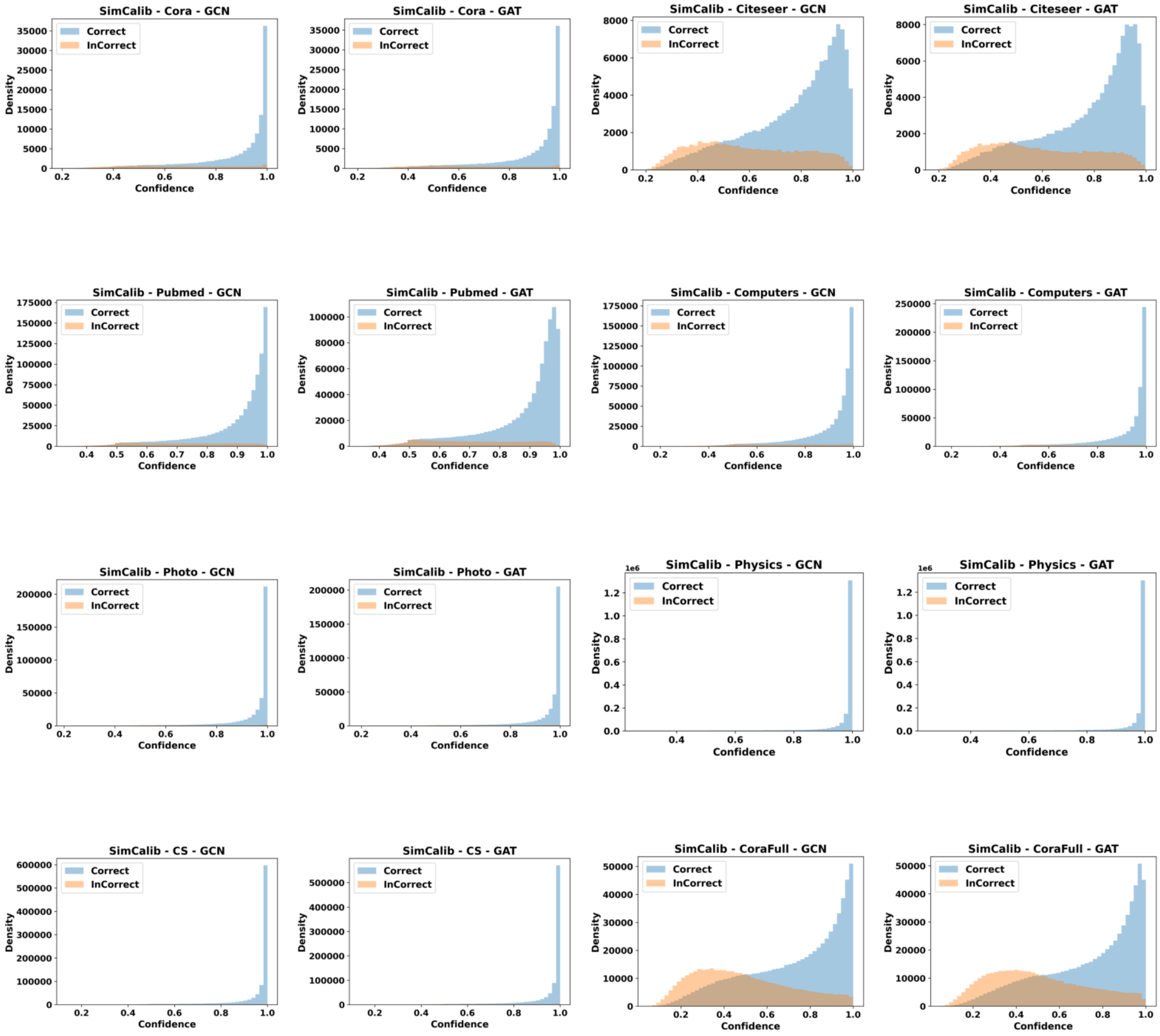

Figure 8: The confidence distributions for SimCalib-calibrated classifiers. The horizontal axis represents confidences while the vertical axis is sample density.