On the Temperature of Bayesian Graph Neural Networks for Conformal Prediction

Seohyeon Cha
KAIST
kaitjgus@kaist.ac.kr

Honggu Kang
KAIST
khg13@kaist.ac.kr

Joonhyuk Kang
KAIST
jkang@kaist.ac.kr

Abstract

Accurate uncertainty quantification in graph neural networks (GNNs) is essential, especially in high-stakes domains where GNNs are frequently employed. Conformal prediction (CP) offers a promising framework for quantifying uncertainty by providing *valid* prediction sets for any black-box model. CP ensures formal probabilistic guarantees that a prediction set contains a true label with a desired probability. However, the size of prediction sets, known as *inefficiency*, is influenced by the underlying model and data generating process. On the other hand, Bayesian learning also provides a credible region based on the estimated posterior distribution, but this region is *well-calibrated* only when the model is correctly specified. Building on a recent work that introduced a scaling parameter for constructing valid credible regions from posterior estimate, our study explores the advantages of incorporating a temperature parameter into Bayesian GNNs within CP framework. We empirically demonstrate the existence of temperatures that result in more efficient prediction sets. Furthermore, we conduct an analysis to identify the factors contributing to inefficiency and offer valuable insights into the relationship between CP performance and model calibration.

1 Introduction

Graph neural networks (GNNs) have demonstrated their impressive abilities to learn from graph-structured data in a wide range of domains, including social sciences, chemistry, knowledge graphs, and recommendation systems [18, 34, 12, 39, 8]. Meanwhile, with the increasing applications of GNNs in safety-critical tasks, there is a growing demand for trustworthy and reliable uncertainty estimates from GNN outputs [14, 37]. However, many existing uncertainty quantification methods are not applicable to GNNs, since they rely on independent and identically distributed (i.i.d.) data assumption, which is violated in graph-structured data [1, 16]. Recent studies have addressed uncertainty in graph-related tasks, but these approaches introduce additional model architecture for density estimation or knowledge distillation. [31, 43].

Conformal prediction (CP) [36] is a promising framework for obtaining uncertainty estimates under the only assumption of data exchangeability[1]. CP constructs prediction sets that are guaranteed to contain a ground-truth with a desired coverage level. Formally, consider a training dataset $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y},\ i \in \{1, \cdots, n\}$, and a test point (x_{n+1}, y_{n+1}) drawn exchangeably from an underlying distribution P. Then, for a given a pre-defined coverage $1-\alpha$, CP generates a prediction set $\mathcal{C}_{n,\alpha}(x_{n+1}) \subseteq \mathcal{Y}$ for a test input x_{n+1} that satisfies

$$\mathbb{P}[y_{n+1} \in \mathcal{C}_{n,\alpha}(x_{n+1})] \geq 1-\alpha \tag{1}$$

where the probability is taken over $n+1$ data samples $\{(x_i, y_i)\}_{i=1}^{n+1}$. A prediction set satisfying the coverage condition in the equation (1) is said to be *valid*. While CP guarantees a valid prediction for

[1]The data exchangeability assumes that every permutation of data samples have the same probability, which is a weaker assumption than i.i.d assumption.

NeurIPS 2023 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2023).

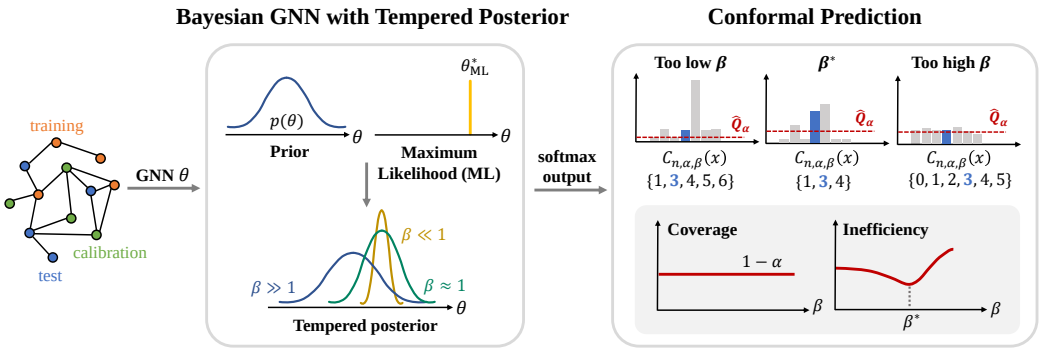


Figure 1: **Conformal prediction (CP) of Bayesian GNNs with tempered posteriors**. In Bayesian GNNs, the temperature parameter β controls the spread of the posterior over model parameters θ. Once we train Bayesian GNNs with a specific temperature, CP generates prediction sets by comparing softmax output of a test sample to a threshold obtained from calibration set. In the right box of this figure, the blue bar represents the true label, and CP guarantees that prediction sets contain the true label with a pre-defined coverage. We show that Bayesian GNNs with an appropriate temperature produce more efficient prediction sets, reducing inefficiency, while keeping the desired coverage.

any classifier, the size of a prediction set, called *inefficiency*, is largely affected by the underlying classifier and the data generating process [7, 40].

Meanwhile, Bayesian learning [3, 9, 38] (see also [29, Ch. 12] for an overview) is a traditional way for building a predictive interval. In contrast to frequentist learning, which aims to find an optimal point of model parameters, Bayesian learning produces a posterior estimate for model parameters. Consequently, it can naturally generate credible regions by simply taking $(1-\alpha) \cdot 100$ % confidence region of the posterior distribution. However, the region can only be *valid* without model misspecification, which means that prior, likelihood, and computational capability should be correctly specified in posterior inference [23, 38]. To address the issue of model misspecification when constructing a posterior credible region, a recent study has introduced a scaling parameter that controls the spread of posterior [33] (see also [19] for related discussion on *generalized Bayesian learning*). Their findings demonstrate that the control of the scaling parameter can yield a credible region with an appropriate size but provides approximate validity, whereas CP offers a rigorous guarantee of validity.

Building upon insights from previous works, we study the benefits of Bayesian GNNs in CP framework to obtain valid and efficient uncertainty estimates. While recent CP approaches for GNNs have focused on ensuring validity and reducing inefficiency, these methods are applicable to specific tasks [6, 41] or require correction dataset for additional GNN training [15]. In our work, we propose the use of a temperature parameter in Bayesian GNNs to allow for flexible control of inefficiency within CP framework. We show that Bayesian GNNs improve the performance of CP, especially the inefficiency, compared to frequentist GNNs while preserving the validity. Furthermore, our experiments on both node classification and graph classification tasks demonstrate the existence of temperatures of Bayesian GNNs that lead to more efficient prediction sets. While previous studies have explored the temperature parameter in Bayesian learning mainly for calibration purposes [33, 42], our study investigates the impact of temperatures on CP performance. Moreover, we provide an analysis that explores the connection between the CP performance and model calibration, providing valuable insights on the varying inefficiency depending on the temperature.

The main contributions of this study are summarized as follows:

- We show that Bayesian GNNs outperforms in CP compared to frequentist GNNs.
- Our experiments on graph-structured data demonstrate the existence of temperatures that result in better efficiency[2].
- We analyze the connection between the inefficiency and model calibration to verify the factors that contribute to the inefficiency.

[2]Throughout the paper, we use the terms "lower inefficiency" and "better efficiency" interchangeably.

2 Background

2.1 Conformal Prediction

Conformal prediction (CP) [36] aims to construct a set of candidate labels that is likely to contain the true label. Formally, given a test example $x \in \mathcal{X}$ and for every candidate label $y \in \mathcal{Y}$, CP either rejects or accepts the candidate pair (x, y) to be included in a prediction set. This decision is done based on a statistic, called *non-conformity score* $s((x, y)|\theta_{\mathcal{D}})$, a real-valued function that takes input-ouput pair (x, y) as an input given a pre-trained model parameter $\theta_{\mathcal{D}}$. The score $s((x, y)|\theta_{\mathcal{D}})$ quantifies the disagreement between data sample x and a candidate label y. One typical example of non-conformity score broadly used in classification task is negative log-likelihood ($-\log p_\theta(y|x)$).

In this work, we use one most common type of CP, called *split conformal prediction (SCP)* [36], which splits available dataset into training and calibration sets. Given a trained model on training set $\mathcal{D}$, SCP creates a prediction set $\mathcal{C}_{n,\alpha}(x)$ for a test input x using the calibration set $\{(x_i, y_i)\}_{i=1}^n$. Specifically, we first compute the non-conformity scores of calibration samples as $\{s((x_i, y_i)|\theta_{\mathcal{D}})\}_{i=1}^n$. Then, the $(1-\alpha)$-quantile of the set $\{s((x_1, y_1)|\theta_{\mathcal{D}}), \ldots, s((x_n, y_n)|\theta_{\mathcal{D}})\} \cup \{\infty\}$ becomes the threshold that determines whether to accept or reject candidate label $y \in \mathcal{Y}$ to be contained in the set. The threshold can be obtained by selecting $\lceil (1-\alpha)(n+1) \rceil$-th smallest value from the set $\{s((x_1, y_1)|\theta_{\mathcal{D}}), \ldots, s((x_n, y_n)|\theta_{\mathcal{D}})\} \cup \{\infty\}$, denoted as $Q_\alpha(\{s((x_i, y_i)|\theta_{\mathcal{D}})\}_{i=1}^n)$. Consequently, the corresponding prediction set for a test sample x is defined as

$$\mathcal{C}_{n,\alpha}(x) = \{y \in \mathcal{Y} : s((x, y)|\theta_{\mathcal{D}}) \leq Q_\alpha(\{s((x_i, y_i)|\theta_{\mathcal{D}})\}_{i=1}^n)\}\,. \tag{2}$$

Theorem 1 *Given that calibration set $\{(x_i, y_i)\}_{i=1}^n$ and a test data point (x, y) are exchangeable random variables, for any coverage level $1-\alpha$, for any trained model $\theta_{\mathcal{D}}$, and for any nonconformity score $s(\cdot|\theta_{\mathcal{D}})$, a conformal prediction set $\mathcal{C}_{n,\alpha}(x)$ defined in the equation* (2) *satisfies the equation* (1) [36].

Theorem 1 provides a marginal coverage guarantee for all x on average. Roughly speaking, the SCP can induce any classifier to produce prediction sets guaranteed to satisfy the desired coverage level. This implies that, on average, these sets have a pre-defined probability of containing the true label.

2.2 Bayesian Learning

We introduce two learning paradigms, *frequentist learning* and *Bayesian learning*, to find a classifier which predicts an output $y \in \mathcal{Y}$ for a given input $x \in \mathcal{X}$. A conventional learning that finds an optimal model parameter θ and the corresponding single classifier $p(y|x, \theta)$ given training set $\mathcal{D} = \{x_i, y_i\}_{i=1}^n$, is called *frequentist learning* [25]. In general, it optimizes the model parameter by minimizing the training loss as

$$\theta^* = \operatorname*{argmin}_\theta L_{\mathcal{D}}(\theta) = \sum_{(x,y)\in\mathcal{D}} -\log p(y|x, \theta). \tag{3}$$

Meanwhile, *Bayesian learning* [25, 3, 29] updates prior beliefs about model parameter θ that fits the dataset $\mathcal{D}$ while accounting for a prior belief $p(\theta)$ about the values of θ. Instead of providing a single point estimate for θ, it infers a posterior distribution over possible values of parameters θ based on Bayes' rule by incorporating likelihood from data and prior knowledge on θ. Hence, it is readily capable of capturing and quantifying the uncertainty about model parameter θ. Concretely, the Bayesian posterior can be obtained by minimizing the *free energy criterion* [20, 17],

$$q_\beta^*(\theta|\mathcal{D}) = \operatorname*{argmin}_q \mathbb{E}_{q(\theta)}[L_{\mathcal{D}}(\theta)] + \beta \cdot \mathrm{KL}[q(\theta)||p(\theta)], \tag{4}$$

where $\mathrm{KL}[q(\theta)||p(\theta)]$ is the Kullback-Leibler (KL) divergence between two distributions q and p, and $\beta > 0$ is the *temperature* parameter which creates a tempered posterior $q_\beta^*(\theta|\mathcal{D})$.

One advantage of posterior inference in Bayesian learning is that it can naturally produce credible regions from the posterior. For example, we simply can take 90 % confidence region of posterior distribution if we want to get a *well-calibrated* (or *valid*) region which satisfies the pre-defined coverage probability of 90 % to contain a target output in the region, as in the equation (1). However, it fails in common when the model is *misspecified* [10, 23, 19]. Model misspecification occurs from

three assumptions in posterior inference, which are prior belief, likelihood, and computational power required to perform inference on intractable posterior [19]. A model might converge to an incorrect solution if a true model is not included in the prior hypothesis space, or even if the space contains the true solution, a model may struggle to find a good solution unless it benefits from reasonable inductive biases from the likelihood [38]. Hence, if there is model misspecification, the resulting posterior can be poorly calibrated and creates undesirable uncertainty quantification. (refer to Fig. 3 in [19]). There has been a study that introduces a scaling parameter to address the model misspecification when constructing a posterior credible region [33]. The scaling parameter controls the spread of the posterior distribution for the credible region to be *efficient* in terms of its size and approximately *well-calibrated*. Note that our work differs from previous studies in that we investigate how the temperature parameter β affects the size of prediction sets obtained by CP when applied on top of Bayesian learning.

3 Conformal Prediction for Bayesian GNNs

3.1 Bayesian GNNs with Tempered Posterior

We begin with the explanation on how to implement Bayesian graph convolutional networks (GCNs). We use Graph DropConnect (GDC), a stochastic regularization technique with adaptive connection sampling [13]. The adaptive connection sampling is a generalized technique that encompasses a range of other regularization and sampling techniques, including DropOut [30], DropEdge [27], and Node Sampling [5]. It operates by applying binary random masks to the adjacency matrix, randomly masking edges, nodes, and channels. They show that connection sampling in GDC can be interpreted as Bayesian extensions of GNNs by transferring randomness from sampling to the model parameter space. Consequently, by implementing GDC within GNNs, we are able to get an appropriate posterior estimate for GNN model parameters and further adjust its temperature.

In detail, for every edge $e = (v, u) \in \mathcal{E}$, we generate a random mask vector denoted as $\mathbf{z}_{v,u}^{(l)} \in \{0, 1\}^{1\times f_l}$ by sampling from Bernoulli distribution with a drop rate π_l, denoted as $\text{Bern}(\pi_l)$. Here, f_l represents the number of features in the l-th layer of GNN. We define the degree of node v as c_v and its neighborhoods as $\mathcal{N}(v)$. Then, for every l-th layer with the weight parameter matrix $\mathbf{w}^{(l)}$ and activation function $\sigma(\cdot)$, we can represent its output feature of each node v when connection sampling is applied as the equation (5) where $\mathbf{w}_{v,u}^{(l)} := \text{diag}(\mathbf{z}_{v,u}^{(l)})\mathbf{w}^{(l)}$ and $\mathbf{h}_u^{(l)}$ denote output feature of node u at l-th layer, resulting in learning different random weight parameter $\mathbf{w}_{v,u}^{(l)}$ for each edge $(v, u) \in \mathcal{E}$, i.e.,

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\frac{1}{c_v}(\sum_{u\in\mathcal{N}(v)\cup\{v\}} \mathbf{z}_{v,u}^{(l)} \odot \mathbf{h}_u^{(l)})\mathbf{w}^{(l)}\right) = \sigma\left(\frac{1}{c_v}\sum_{u\in\mathcal{N}(v)\cup\{v\}} \mathbf{h}_u^{(l)}\mathbf{w}_{v,u}^{(l)}\right). \tag{5}$$

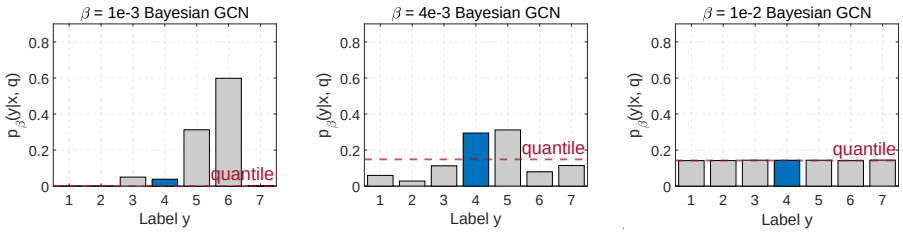
In Bayesian GNNs, we estimate the posterior distribution of GNN weight parameters considering them as random parameters. Since GDC incorporates random parameters including weight parameters $\mathbf{W}^{(l)} = \{\mathbf{w}_e^{(l)}\}_{e=1}^{|\mathcal{E}|}$, random masks $\mathbf{Z}^{(l)} = \{\mathbf{z}_e^{(l)}\}_{e=1}^{|\mathcal{E}|}$, and the corresponding dropout rate π_l, we infer posteriors on these parameters, denoted as $\boldsymbol{\theta} = \{\mathbf{W}^{(l)}, \mathbf{Z}^{(l)}, \pi_l\}_{l=1}^{L}$. Note that the drop rates are also learnable to enhance the flexibility of the model [4]. Then, the free energy criterion for training Bayesian GNNs is defined as

$$F_\beta(q) = \mathbb{E}_{q(\boldsymbol{\theta})}\left[\sum_{(x,y)\in\mathcal{D}} -\log p(y|x, \boldsymbol{\theta})\right] + \beta \cdot \sum_{l=1}^{L} \text{KL}\left[q(\boldsymbol{\theta})||p(\boldsymbol{\theta})\right] \tag{6}$$

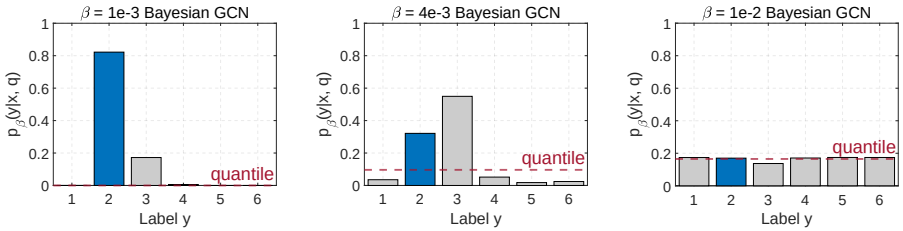
where the *temperature* parameter β controls the spread of posterior distribution, as illustrated in Fig. 1. The posterior with $\beta = 1$ corresponds to standard Bayesian posterior, and the $\beta \gg 1$ and $\beta \ll 1$ depend more on the prior assumption and information from data, respectively. Consequently, the *tempered posterior* can be obtained by minimizing the equation (6) as

$$q_\beta(\boldsymbol{\theta}|\mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{(x,y)\in\mathcal{D}} p(y|x, \boldsymbol{\theta})^{1/\beta}. \tag{7}$$

Though estimating the posterior distribution in the equation (7) requires intractable computation, *variational inference (VI)* [9] suggests a good approximation for it, where the variational distribution



(a) Label distribution on Cora dataset.



(b) Label distribution on Citeseer dataset.

Figure 2: **Example of label distribution depending on the temperature of Bayesian GCNs.**

from a defined family of distributions (e.g., Gaussian distributions) approximates true posterior distribution. Details on prior assumptions and VI algorithm for GDC are stated in Appendix D. Suppose we have the tempered posterior $q_\beta(\boldsymbol{\theta}|\mathcal{D})$ or its VI approximate, then the predictive distribution on a test sample x can be obtained by averaging over all likely models as

$$p_\beta(y|x,q) = \int p(y|x,\boldsymbol{\theta}) q_\beta(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta} \approx \frac{1}{T}\sum_{t=1}^{T} p(y|x,\hat{\boldsymbol{\theta}}_t) \tag{8}$$

where the distribution can be approximated by the average over T sets of i.i.d. samples $\hat{\boldsymbol{\theta}}_t$ drawn from $q_\beta(\boldsymbol{\theta}|\mathcal{D})$. Since we use negative log-loss as non-conformity score for CP, i.e., $-\log p_\beta(y|x,q)$, the prediction sets, especially the size of sets is directly affected by the temperature. Accordingly, we discuss the adjustment of temperature of Bayesian GNNs for CP in the following.

3.2 Temperature Control for Conformal Prediction

We now present how to exploit the benefits of Bayesian GNNs with tempered posterior in split conformal prediction (SCP). As illustrated in Fig. 1, we first train Bayesian GNNs using loss function in the equation (6) with a pre-determined temperature β using training set. Afterward, for a given calibration set $\{(x_i, y_i)\}_{i=1}^n$, we compute the non-conformity score of calibration data as negative log-loss based on the predictive distribution obtained from the tempered posterior as the equation (8). Subsequently, for a given pre-defined coverage level of $1-\alpha$, the prediction set for a test input x, generated using SCP on top of Bayesian GNNs with tempered posterior, is defined as follows:

$$\mathcal{C}_{n,\alpha,\beta}(x) = \{y \in \mathcal{Y} : -\log p_\beta(y|x,q) \le Q_\alpha(\{-\log p_\beta(y_i|x_i,q)\}_{i=1}^n)\}\,. \tag{9}$$

This prediction set satisfies the desired coverage condition, as stated in Theorem 2. Additionally, as depicted in Fig. 1, we suggest that employing the tempered posterior in Bayesian GNNs, where the temperature controls its spread as means to overcome model misspecification, can generate an efficient prediction set within CP. While the credible region introduced in [33] provides a coverage guarantee in an asymptotic sense, SCP provides a rigorous guarantee without relying on specific distribution assumptions, as outlined in Theorem 1. Given that conformal predictors alone may not necessarily create the smallest prediction sets, adjusting the temperature provides greater freedom to reduce the size of the prediction sets. Please refer to Appendix B for the proofs of the theorems.

Theorem 2 *Given that calibration set $\{(x_i, y_i)\}_{i=1}^n$ and a test data point (x, y) are exchangeable random variables, for any coverage level $1-\alpha$, and for any pre-determined temperature parameter $\beta \in [0, \infty)$, a prediction set $\mathcal{C}_{n,\alpha,\beta}(x)$ defined in the equation* (9) *satisfies the equation* (1) *with additional marginalization over posterior sampling in the equation* (8) *when assuming a finite number of sampling T.*

To suggest a brief insight into how the temperature affects the inefficiency, we provide an example of label distribution from Bayeisan GNNs at different temperatures, extracted from real data samples of Cora and Citeseer. In Fig. 2, note that there are some cases where the true label is included in prediction set, but not selected as a point prediction. While prediction sets from all models include the true label, the size largely depends on the temperature of Bayesian GNNs. As depicted in Fig. 1, when the temperature parameter β is set too low, the label distribution tends to reflect an over-confident behavior compared to other models. This leads to extremely low quantiles and prediction set containing a large number of candidate labels. Conversely, if the temperature is set too high, it results in labels having nearly identical confidence levels, leading to prediction sets with less informative labels. However, when an appropriate temperature is selected between these extremes, Bayesian GNNs generate efficient prediction sets that contain fewer number of informative labels, resulting in better efficiency.

4 Experimental Results

In this section, we study (i) the conformal prediction performance of tempered posterior in GCN, in terms of empirical coverage and inefficiency, and analyze (ii) the contributing factors of inefficiency, considering the connection between inefficiency and model calibration.

Experimental settings We consider two major tasks on graph-structured data, which are graph classification and node classification tasks. We focus on transductive setting in node classification task that ensures the validity [15]. We have selected public graph datasets with enough number of classes that can effectively reflect the change of set size depending on the temperature. We use CIFAR10-superpixel [21, 8] and ENZYMES [24] for graph classification task, and Cora and Citeseer for node classification task [28]. Overall dataset statistics used for model training and CP are stated in Table 1. Our models are based on GCN [18] and Bayesian GCNs are trained with Graph DropConnect (GDC) assuming the same priors for random parameters as [13]. We have selected temperature parameters for each dataset that make reasonable test accuracy. To ensure the exchangeability of split conformal prediction, we randomly split calibration and test set as illustrated in Table 1, and CP results are averaged over 50 runs for CIFAR10, and 100 runs for other datasets. Further details on experimental settings and exchangeability conditions for GNNs can be seen in Appendix A, C.

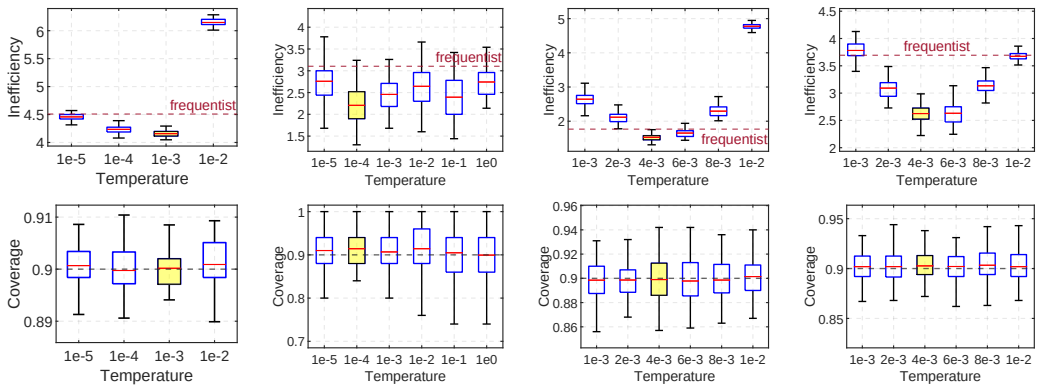
Evaluation metrics We use two metrics of conformal prediction, empirical coverage and empirical inefficiency. A conformal predictor is considered to be valid when the empirical coverage over test set $\mathcal{D}_{\text{test}}$, denoted as $\widehat{cov}$, satisfies the pre-defined coverage level (we use 0.9). Note that achieving higher coverage is not necessarily advantageous if it leads to an increase in set size, quantified as higher inefficiency. Another metric is empirical inefficiency which measures the average size of prediction sets over test set, denoted as $\widehat{ineff}$. Lower inefficiency indicates that a conformal predictor is more efficient since it outputs smaller number of candidate labels while keeping the same pre-defined coverage level.

$$\widehat{cov} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} \mathbb{1}\{y \in C_{n,\alpha}(x)\}, \ \widehat{ineff} = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} |C_{n,\alpha}(x)| \tag{10}$$

Calibration measures To assess the calibration of GCNs, we employ commonly used calibration measures, along with CP measures. For each data sample (x, y), suppose we have a point prediction from a model θ as $\hat{y} = \operatorname{argmax}_c p(c|x, \theta)$, then accuracy and confidence for every sample are defined as $\mathbb{1}\{\hat{y} = y\}$ and $p(c = \hat{y}|x, \theta)$, respectively. We use reliability diagram, which depicts the average sample accuracy against confidence levels. If a model is perfectly calibrated, where the estimated

Dataset	#Nodes	#Edges	#Graphs	#Feat	#Classes	#Train	#Cal	#Test
Cora	2708	5429	-	1433	7	140	500	1000
Citeseer	3327	4732	-	3703	6	120	500	1000
CIFAR10	-	-	50000	5	10	10000	5000	10000
ENZYMES	-	-	600	21	6	500	50	50

Table 1: Dataset statistics for Cora, Citeseer, CIFAR10, ENZYMES



(a) CIFAR10-superpixel (b) ENZYMES (c) Cora (d) Citeseer

Figure 3: **Empirical coverage and inefficiency of Bayesian GCNs with different temperatures**. Model that produces smallest sized prediction sets on average is denoted as yellow box. The exact average coverage and inefficiency of Bayesian models are represented as red line inside each box.

probability of true label is exactly the same as the confidence for all levels, the reliability diagram takes the form of an identity function. In our experiments, we divide samples into $M = 20$ confidence bins, each denoted as B_m, whose samples have confidence within the range $(\frac{m-1}{M}, \frac{m}{M}]$. The average accuracy and confidence over samples in B_m are denoted as $\text{acc}(B_m)$ and $\text{conf}(B_m)$, respectively. We additionally use two measures obtained from reliability diagram, expected calibration error (ECE) and maximum calibration error (MCE) [11], defined as

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \ \text{MCE} = \max_{m \in \{1,\cdots,M\}} \left|\text{acc}(B_m) - \text{conf}(B_m)\right|, \quad (11)$$

where n is the number of total samples. ECE is a weighted average of differences between confidence and accuracy across all bins, while MCE represents the largest of these differences among all bins.

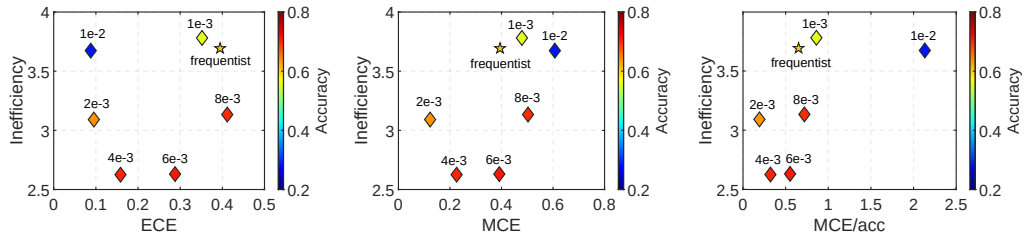
4.1 Efficiency of Tempered Posteriors

We evaluated our frequentist and Bayesian GCNs using CP evaluation metrics, empirical coverage and empirical inefficiency in Fig. 3. As theoretically guaranteed, Bayesian models of all temperatures satisfy the pre-defined coverage level of 0.9. Since coverage is guaranteed empirically across all models, the empirical inefficiency determines the performance of conformal predictors. A conformal predictor producing a smaller prediction set implies that the set conveys valuable information. In Fig. 3, we have verified that Bayesian GCN gives benefits to the lower inefficiency, while guaranteeing the desired coverage. *Notably, there exists a model with a temperature that makes better inefficiency, compared to other temperatures*, as illustrated as yellow boxes in Fig. 3. One may question that model with high accuracy might create smaller sets, and results in lower inefficiency. However, it is worth noting that while the inefficiency gets affected by the model performance, we have observed that this is not always the case. In the following sections, we will delve into this matter further by examining the relationship between inefficiency and model calibration.

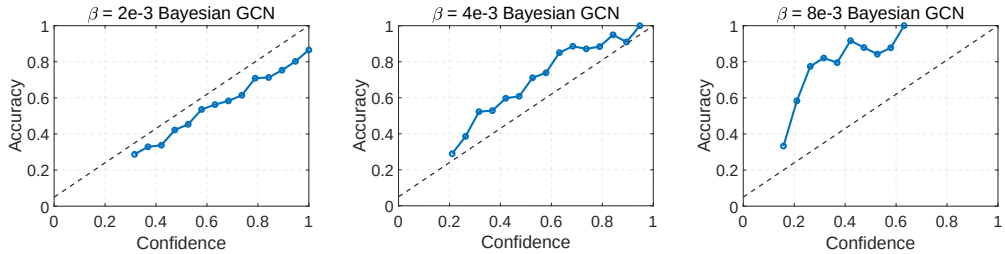
4.2 Contributing Factors to Inefficiency

Our experiments have shown that inefficiency is influenced by not only model performance but also model calibration. For example in Fig. 4a, the Bayesian model with the temperature $\beta = 2 \cdot 10^{-3}$ has lower inefficiency than the model with $\beta = 8 \cdot 10^{-3}$, even though the test accuracy is lower. Motivated by this observation, we conducted following analysis to answer to the quenstion *"What factors contribute to inefficiency in CP due to the changes in temperature?"*

Relation between inefficiency and model calibration To provide a deeper understanding of the connection between prediction sets and the underlying model, we conducted evaluation of Bayesian GCNs with different temperatures, assessing calibration through reliability diagram, ECE and MCE.

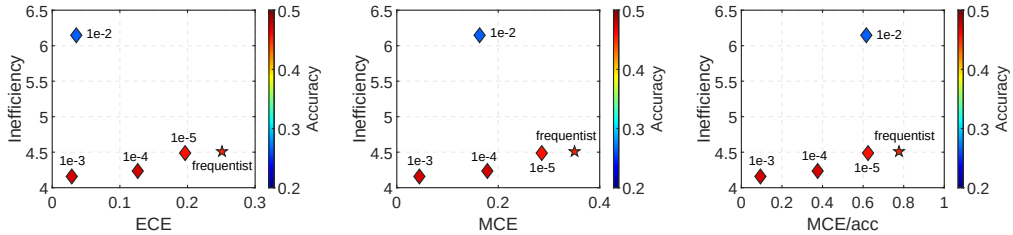


(a) Relation between inefficiency and calibration measures

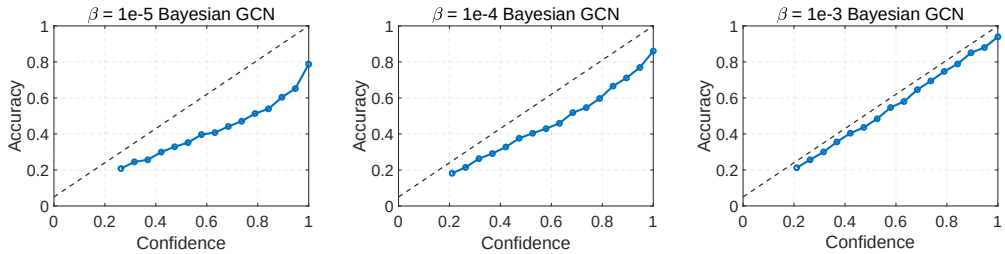


(b) Reliability diagrams of trained Bayesian GCNs on Citeseer

Figure 4: **Results of inefficiency and calibration on Citeseer**. (a) Model with lower inefficiency implies that it generates prediction sets with smaller size on average. Bayesian GCNs with $\beta = 4 \cdot 10^{-3}$ has the lowest inefficiency. (b) The line $y = x$ on reliability diagrams represents a perfectly calibrated model.



(a) Relation between inefficiency and calibration measures



(b) Reliability diagram of trained Bayesian GCNs on CIFAR10

Figure 5: **Results of inefficiency and calibration on CIFAR10**. (a) Model with the lower inefficiency implies that it generates prediction sets with smaller size on average. Bayesian GCNs with $\beta = 10^{-3}$ has the lowest inefficiency. (b) The line $y = x$ on reliability diagrams represents a perfectly calibrated model.

Overall, we have verified that employing an appropriate temperature parameter β enhances the calibration of GCNs. This improvement is evident in reliabillity diagrams, as shown in Fig. 5b. Bayesian GCN model which has the lowest inefficiency at $\beta = 10^{-3}$ is comparably well-calibrated than other models.

Furthermore, our experiments imply two major observations. First, *when two models exhibit similar model accuracy levels, the better calibrated model tends to have lower inefficiency, even if the*

difference can be slight. In Fig. 4a, the Bayesian model with $\beta = 4 \cdot 10^{-3}$, although having similar accuracy with the model $\beta = 8 \cdot 10^{-4}$, exhibits reduced inefficiency. This can be attributed to its more robust calibration, as evidenced by ECE, MCE, and the reliability diagram. Similar case can be observed from Fig. 5 between two Bayesian models with $\beta = 10^{-3}$ and $\beta = 10^{-4}$.

The second finding is that *a well-calibrated model, even when its accuracy is relatively low, yields reasonably efficient prediction sets*. From the comparison of Bayesian models with $\beta = 2 \cdot 10^{-3}$ and $\beta = 8 \cdot 10^{-3}$, we clearly see that the well-calibrated one with $\beta = 2 \cdot 10^{-3}$ results in lower inefficiency, despite its lower test accuracy. Note that the case of $\beta = 2 \cdot 10^{-3}$, the two calibration measures ECE and MCE well reflect the reliability diagram as seen in Fig. 4a. However, in the case of a model with $\beta = 10^{-2}$, while MCE reasonably captures the low inefficiency resulting from its poor model quality, whereas ECE does not. Hence, in the following, we explore how previous calibration measures, ECE and MCE, reflect CP inefficiency and propose insights on temperature selection through a new calibration measure.

Combined measure for temperature selection While our previous findings have indicated the existence of temperatures that lead to efficient prediction sets through CP, it is more preferable to find such good temperatures that yield low inefficiency, without adding computational burden from CP. For this purpose, MCE proves to be better than ECE, and this can be attributed to the sensitivity of inefficiency to difficult samples when making prediction sets. However, it is important to note that neither ECE nor MCE perfectly reflects inefficiency, making points such as $\beta = 10^{-2}$ in Fig. 4 and $\beta = 10^{-2}$ in Fig. 5. Consequently, recognizing that both model performance and calibration contribute to set size, we propose that a combined measure involving accuracy and calibration measure could offer a better chance to capture the inefficiency of models with different temperatures. In the rightmost figures of Fig. 4 and Fig. 5, we present the relation between the combined measure, calculated as MCE divided by accuracy, and inefficiency. These figures show that the combined measure better accounts for the inefficiency of models with high temperatures, addressing the limitations of ECE and MCE. While we have provided a preliminary exploration of the combined measure's ability to represent the set size in CP, we plan to further investigate a measure that well explains the inefficiency. Additionally, in our future research, we aim to develop methods for optimizing the temperature through CP-aware loss [15, 40, 32, 26] during training process.

5 Conclusion

In this work, we explore the impact of temperature on CP in Bayesian GNNs, specifically focusing on the inefficiency of prediction sets. We show that Bayesian GNNs improve the performance in CP compared to frequentist GNNs while providing more flexibility to control the inefficiency via temperature. Our experiments demonstrate that there exist temperatures of Bayesian GNNs that generate efficient prediction sets within CP. Moreover, our analysis on the connection between inefficiency and model calibration suggests further possibility for a measure capable of accurately capturing the inefficiency. Assessing the CP performance using such measure can reduce the computational burden induced by CP. Furthermore, the investigation of a novel method for training the temperature parameter using CP-aware loss [15, 40, 32, 26] remains as an interesting future work. We anticipate that these findings can provide guidance to trustworthy graph learning that studies reliable uncertainty estimates employing CP.

Acknowledgement

The authors want to thank Sangwoo Park and Osvaldo Simeone for providing the main idea of this work that studies the impact of temperature in Bayesian learning when used in conjunction with CP under GNN architecture. This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-0-01787 and IITP-2023-RS-2023-00259991) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation).

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297, 2021.

[2] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

[3] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR, 2015.

[4] Shahin Boluki, Randy Ardywibowo, Siamak Zamani Dadaneh, Mingyuan Zhou, and Xiaoning Qian. Learnable bernoulli dropout for bayesian deep learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3905–3916. PMLR, 2020.

[5] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[6] Jase Clarkson. Distribution free prediction sets for node classification. In *International Conference on Machine Learning*, pages 6268–6278. PMLR, 2023.

[7] Guneet S Dhillon, George Deligiannidis, and Tom Rainforth. On the expected size of conformal prediction sets. *arXiv preprint arXiv:2306.07254*, 2023.

[8] Vijay Prakash Dwivedi, Chaitanya K Joshi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *arXiv preprint arXiv:2003.00982*, 2020.

[9] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[10] Peter Grünwald and Thijs Van Ommen. Inconsistency of bayesian inference for misspecified linear models, and a proposal for repairing it. 2017.

[11] 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.

[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[13] Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *International conference on machine learning*, pages 4094–4104. PMLR, 2020.

[14] 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.

[15] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. *arXiv preprint arXiv:2305.14535*, 2023.

[16] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110:457–506, 2021.

[17] Sharu Theresa Jose and Osvaldo Simeone. Free energy minimization: A unified framework for modeling, inference, learning, and optimization. *IEEE Signal Processing Magazine*, 38(2):120–125, 2021.

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

[19] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

[20] Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. An optimization-centric view on bayes' rule: Reviewing and generalizing variational inference. *The Journal of Machine Learning Research*, 23(1):5789–5897, 2022.

[21] Boris Knyazev, Graham W Taylor, and Mohamed Amer. Understanding attention and generalization in graph neural networks. *Advances in neural information processing systems*, 32, 2019.

[22] Arun Kumar Kuchibhotla. Exchangeability, conformal prediction, and rank tests. *arXiv preprint arXiv:2005.06095*, 2020.

[23] Andres Masegosa. Learning under model misspecification: Applications to variational and ensemble methods. *Advances in Neural Information Processing Systems*, 33:5479–5491, 2020.

[24] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.

[25] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[26] Sangwoo Park, Kfir M Cohen, and Osvaldo Simeone. Few-shot calibration of set predictors via meta-learned cross-validation-based conformal prediction. *arXiv preprint arXiv:2210.03067*, 2022.

[27] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. *arXiv preprint arXiv:1907.10903*, 2019.

[28] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

[29] Osvaldo Simeone. *Machine learning for engineers*. Cambridge university press, 2022.

[30] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[31] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. Graph posterior network: Bayesian predictive uncertainty for node classification. *Advances in Neural Information Processing Systems*, 34:18033–18048, 2021.

[32] David Stutz, Ali Taylan Cemgil, Arnaud Doucet, et al. Learning optimal conformal classifiers. *arXiv preprint arXiv:2110.09192*, 2021.

[33] Nicholas Syring and Ryan Martin. Calibrating general posterior credible regions. *Biometrika*, 106(2):479–486, 2019.

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

[35] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

[36] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*, volume 29. Springer, 2005.

[37] 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, 2021.

[38] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

[39] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

[40] Yachong Yang and Arun Kumar Kuchibhotla. Finite-sample efficient conformal prediction. *arXiv preprint arXiv:2104.13871*, 2021.

[41] Soroush H Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. Conformal prediction sets for graph neural networks. 2023.

[42] Matteo Zecchin, Sangwoo Park, Osvaldo Simeone, Marios Kountouris, and David Gesbert. Robust pacm: Training ensemble models under model misspecification and outliers. *arXiv preprint arXiv:2203.01859*, 2022.

[43] Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems*, 33:12827–12836, 2020.

Supplementary Material

A Exchangeability Conditions for GNNs

The only requirement to ensure a predefined coverage guarantee for the prediction set of CP is the exchangeability between calibration and test data. This is easily satisfied in graph classification tasks where the graph data is sampled in i.i.d. manner from the same distribution of graphs. However, it requires more careful consideration in node classification tasks.

Recent studies [15, 41] have proposed that for common permutation-invariant GNNs, the exchangeability of non-conformity scores for calibration and test data is satisfied in transductive node classification tasks. To be simple, since the selection of calibration nodes does not change the set of non-conformity scores of entire nodes in transductive settings, non-conformity scores remain invariant to the permutation of calibration and test data. Hence, CP can produce valid prediction sets from GNNs. In contrast, in the inductive setting where GNNs are trained only on the training nodes and make prediction on unseen nodes during test time, the exchangeability is violated. This is because the non-conformity scores for calibration nodes are influenced by changes in the graph. In response, a recent work [6] has proposed employing non-exchangeable conformal prediction [2] in inductive setting, but this approach is not applicable to the transductive setting. In our work, we focus on graph classification tasks and transductive node classification tasks, where the exchangeability and validity of CP are satisfied.

Randomness control for exchangeability of Bayeisan GNNs Especially for Bayesian GNNs trained using VI loss in the equation (6), ensuring the exchangeability of non-conformity scores requires further control over the randomness involved in Bayesian inference. When generating a predictive distribution $p_\beta(y|x, q)$ from the tempered posterior with β as in the equation (8), we ensure that the same drop rate and corresponding random masks are employed for both calibration and test data to ensure the exchangeability.

We sample T sets of drop rates for each layer $l \in \{1, \cdots, L\}$ as $\{\pi_t^{(l)}\}_{t=1}^T$ and generate corresponding random masks $\{\mathbf{Z}_t^{(l)}\}_{t=1}^T$. This process yields a fixed set of random model parameters $\{\mathbf{W}_t^{(l)}\}_{t=1}^T$ for each layer, shared across calibration and test set. We then perform T number of forward passes through the GNNs with a fixed set of random model parameters. The resulting softmax outputs are averaged across T runs to create a single predictive distribution, used to calculate non-conformity scores for both calibration and test data.

B Proofs for Theorem 1, 2

Assuming a fixed temperature β in a Bayesian model, we can unify the proofs for both Theorem 1 and Theorem 2 since the validity condition is satisfied for any non-conformity score. Therefore, we present a comprehensive proof of the validity condition as outlined in Theorem 1. The subsequent parts involve demonstrating the exchangeability of non-conformity scores between calibration set and a test data sample, followed by proving coverage guarantees for prediction sets.

Assumption 1 *Given that calibration set $\{z_i\}_{i=1}^n = \{(x_i, y_i)\}_{i=1}^n$ and a test data point $z_{n+1} = (x_{n+1}, y_{n+1})$ are exchangeable, i.e., the joint distribution $p(z_1, \ldots, z_n, z_{n+1})$ is permutation invariant with respect to $\{z_1, \ldots, z_n, z_{n+1}\}$. Formally, the equality $p(z_1, \ldots, z_n, z_{n+1}) = p(z_{\pi(1)}, \ldots, z_{\pi(n)}, z_{\pi(n+1)})$ is satisfied for any permutation function $\pi(\cdot)$.*

If we achieve the exchangeability as described in Assumption 1, we can establish that the non-conformity scores $s_i = s(z_i|\theta_\mathcal{D})$ are exchangeable because they are only conditioned on the deterministic parameter $\theta_\mathcal{D}$, i.e.,

$$\begin{aligned} &(z_1, \ldots, z_n, z_{n+1}) \overset{d}{=} (z_{\pi(1)}, \ldots, z_{\pi(n)}, z_{\pi(n+1)}) \\ \iff &(s_1, \ldots, s_n, s_{n+1}) \overset{d}{=} (s_{\pi(1)}, \ldots, s_{\pi(n)}, s_{\pi(n+1)}) \end{aligned} \tag{12}$$

Lemma 1 *If $s_1, \ldots, s_n, s_{n+1}$ are exchangeable random variables, then for any $1-\alpha \in (0,1)$, then the following inequality holds*

$$\mathbb{P}[s_{n+1} \leq Q_\alpha(\{s_i\}_{i=1}^n)] \geq 1-\alpha \tag{13}$$

where $Q_\alpha(\{s_i\}_{i=1}^n)$ is the $\lceil (1-\alpha)(n+1) \rceil$-th smallest value from the set $\{s_1, \ldots, s_n\} \cup \{\infty\}$.

Proof. By the exchangeability of random variables $\{s_1, \ldots, s_n, s_{n+1}\}$, we have the equality [22]

$$\mathbb{P}[s_{n+1} \leq Q_\alpha(\{s_i\}_{i=1}^{n+1})] \geq 1-\alpha. \tag{14}$$

Furthermore, we have the following equivalence [35]

$$s_{n+1} > Q_\alpha(\{s_i\}_{i=1}^n) \iff s_{n+1} > Q_\alpha(\{s_i\}_{i=1}^{n+1}) \tag{15}$$

as s_{n+1} cannot be strictly larger than itself or ∞. It ensures that the non-conformity score s_{n+1} maintains its position regardless of whether the quantile is considered with or without the test data point.

To sum up, given the exchangeability of nonconformity scores on calibration set and a test data point, along with Lemma 1, we can derive the inequality that satisfies the validity conditions in both Theorem 1 and Theorem 2 as

$$\mathbb{P}[y_{n+1} \in \mathcal{C}_{n,\alpha}(x_{n+1})] = \mathbb{P}[s(z_{n+1}|\theta_{\mathcal{D}}) \leq Q_\alpha(\{s(z_i|\theta_{\mathcal{D}})\}_{i=1}^n)] \geq 1-\alpha. \tag{16}$$

C Experimental Details

C.1 Model training

Node classification We show results on two real citation graph datasets, *Cora* and *Citeseer* datasets. For Bayesian GCN training, we followed the instructions in `https://github.com/armanihm/GDC` [13]. We train for 2500 epochs using Adam optimizer with a learning rate of 0.005. To obtain predictive distribution using Bayesian marginalization in the equation (8), we set the number of runs as $T = 20, 12$ for Cora and Citeseer, respectively. For frequentist GCN model, we additionally use weight decay of 0.005 and dropout [30], to reproduce reasonable accuracy though it can be considered as one kind of Bayesian learning. We utilize a 4-layer GCN with hidden layer dimension of 128 following [13] for node classification tasks, which is relatively larger than commonly used 2-layer architecture. However, larger GCNs are prone to over-fitting and over-smoothing issues, hence we demonstrate that not only the Bayesian model effectively addresses these challenges, and also temperature scaling provides even more flexibility in making efficient prediction sets in CP. We focus on the transductive setting, where graph topology and features of every node are given to model during training while labels of test nodes are unobserved.

Graph classification We use two graph classification datasets, with enough number of labels, *CIFAR10-superpixel* [8] and *ENZYMES* [24]. For CIFAR10-superpixel dataset, we train both BBGDC-GCN and frequentist GCN models for 1000 epochs using Adam optimizer with a learning rate of 0.001. GCN model consists of 5 layers of output feature dimension 180 and averaging readout function to make graph-level representation. We mainly followed the code implementation of `https://github.com/graphdeeplearning/benchmarking-gnns` [8]. For ENZYMES dataset which can be accessed by TUDataset [24], we train for 3000 epochs using Adam optimizer with a learning rate of 0.001. GCN model consists of 3 layers of output feature dimension 128 and sum readout function. To obtain predictive distribution using Bayesian marginalization in the equation (8), we set the number of runs as $T = 10$ for both datasets.

D Variational Beta-Bernoulli Graph DropConnect (BBGDC)

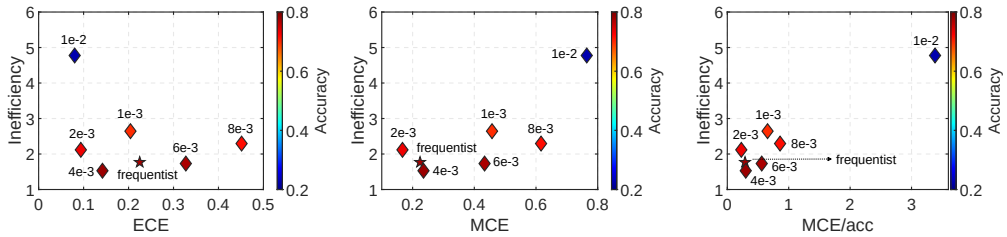
To perform variational inference (VI) for GNN weight parameters, GDC random masks, and the drop rate, we have assumed a beta-Bernoulli prior for random masks and replace the Beta prior of drop rate for each layer into Kumaraswamy distribution, following assumptions in [13]. To enable the optimization of drop rates together with weight parameters, we used Augment-REINFORCE-Merge (ARM) estimator which can obtain the gradient of loss with respect to drop rates with two forward passes [4].

E Results on Cora and ENZYMES

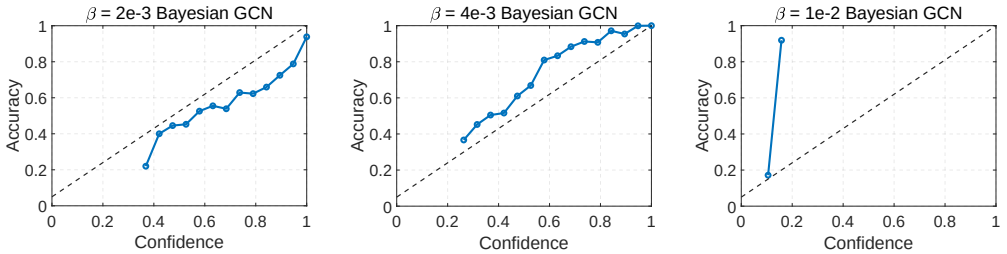
We additionally present results of the relation between inefficiency and model calibration on Cora and ENZYMES datasets. Two observations that (i) a well-calibrated model between models with similar accuracy shows better efficiency, and (ii) a model with a relatively low accuracy can achieve improved efficiency if it is well-calibrated, suggested in 4.2 can be seen from these datasets as well.

On Cora dataset, as shown in Fig. 6, among Bayesian GCNs achieving similar test accuracies in the temperature sets $\{10^{-3}, 2 \cdot 10^{-3}, 8 \cdot 10^{-3}\}$ and $\{4 \cdot 10^{-3}, 6 \cdot 10^{-3}\}$, the better calibrated models, with $\beta = 2 \cdot 10^{-3}$ and $\beta = 4 \cdot 10^{-3}$, in terms of both ECE and MCE exhibit improved efficiency, showing our first observation. In the case of GCNs with $\beta = 10^{-2}$ that achieves relatively low test accuracy, it shows the lowest ECE and the highest MCE among models. This is due to the behavior observed in the reliability diagram, where the majority of samples have both low confidence and low accuracy, while a few samples have significantly higher accuracy within a specific bin, leading to a high MCE. This highlights the limitation of ECE to capture the inefficiency in CP and emphasizes the necessity for a new measure.

On ENZYMES dataset, we can observe our second finding that a model with relatively low test accuracy can achieve improved efficiency if the model is well-calibrated, in terms of ECE, MCE, and reliability diagram. Specifically in Fig. 7, in the case of the model with $\beta = 1$, despite having lower model accuracy compared to other models, it results in similar or even lower inefficiency than a model with higher accuracy ($\beta = 10^{-5}$) due to its low ECE, MCE, and well-calibrated behavior as observed in reliability diagram. However, it was hard to capture the first observation on ENZYMES dataset, since the GCNs with temperatures $\{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ that acheive similar accuracy did not exhibit strong calibration as visualized in Fig. 7b.

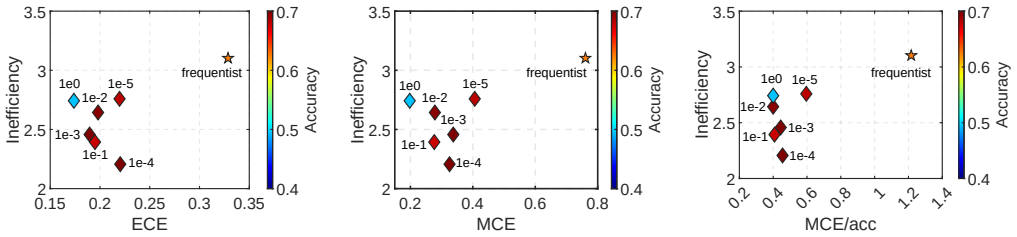


(a) Relation between inefficiency and calibration measures

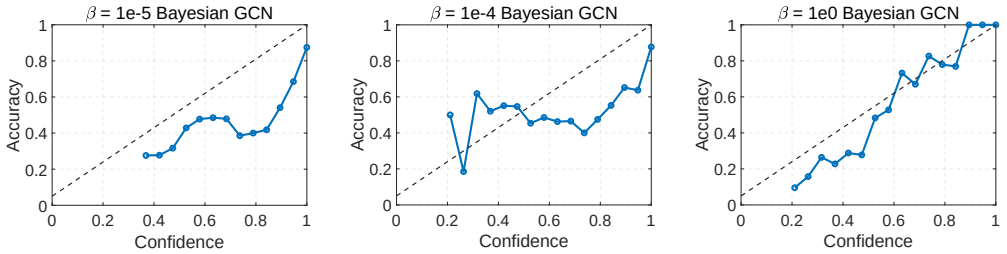


(b) Reliability diagram of trained GCNs on Cora

Figure 6: **Results of inefficiency and calibration on Cora**. (a) Model with the lower inefficiency implies that it generates prediction sets with smaller size on average. Bayesian GCNs with $\beta = 4 \cdot 10^{-3}$ has the lowest inefficiency. (b) The line $y = x$ on reliability diagrams represents a perfectly calibrated model.



(a) Relation between inefficiency and calibration measures



(b) Reliability diagram of trained GCNs on ENZYMES

Figure 7: **Results of inefficiency and calibration on ENZYMES**. (a) Model with the lower inefficiency implies that it generates prediction sets with smaller size on average. Bayesian GCNs with $\beta = 10^{-4}$ has the lowest inefficiency. (b) The line $y = x$ on reliability diagrams represents a perfectly calibrated model.