Conditional Prediction ROC Bands for Graph Classification

Yujia Wu[1] **Bo Yang**[2] **Elynn Chen**[3] **Yuzhou Chen**[4] **Zheshi Zheng**[2]

Abstract

Graph classification in medical imaging and drug discovery requires accuracy and robust uncertainty quantification. To address this need, we introduce Conditional Prediction ROC (CP-ROC) bands, offering uncertainty quantification for ROC curves and robustness to distributional shifts in test data. Although developed for Tensorized Graph Neural Networks (TGNNs), CP-ROC is adaptable to general Graph Neural Networks (GNNs) and other machine learning models. We establish statistically guaranteed coverage for CP-ROC under *a local exchangeability condition*. This addresses uncertainty challenges for ROC curves under non-iid setting, ensuring reliability when test graph distributions differ from training data. Empirically, to establish local exchangeability for TGNNs, we introduce a data-driven approach to construct local calibration sets for graphs. Comprehensive evaluations show that CP-ROC significantly improves prediction reliability across diverse tasks. This method enhances uncertainty quantification efficiency and reliability for ROC curves, proving valuable for real-world applications with non-iid objects.

1 Introduction

Graph classification is essential in fields like medical image analysis and drug discovery, where high accuracy and robust uncertainty quantification (UQ) are critical. Graph Neural Networks and Tensorized Graph Neural Networks excel in label prediction accuracy by leveraging complex graph and tensor structures, enhancing statistical performance and scalability (Xia et al., 2021; Zhou et al., 2020; Wen et al., 2024). In addition, recent advancements in UQ for GNN include conformal methods for robust prediction sets (Zargarbashi et al., 2023; Huang et al., 2023), specialized models for travel demand and traffic risk assessment (Zhuang et al., 2022; Gao et al., 2022), and Bayesian frameworks for quantifying aleatoric and epistemic uncertainties (Munikoti et al., 2023). These advancements enhance the accuracy and uncertainty quantification of graph label prediction.

However, significant gaps remain. Firstly, simple UQ at the label level is insufficient for a comprehensive assessment of classifier reliability. The ROC curve, which provides a detailed illustration of classifier performance across thresholds, is especially useful in critical applications with imbalanced data and varying costs of false positives and negatives. Thus, the UQ of the ROC curve is needed for advanced graph classification tools like TGNNs. Secondly, most UQ for GNN does not address covariate shifts in graph features, which is common in large datasets. Existing methods for non-exchangeable cases, which adjust conformal prediction intervals for estimated covariate shifts (Tibshirani et al., 2019; Gibbs and Candes, 2021), are not directly applicable to GNNs where graph covariate shifts are hard to estimate and are not suitable for quantifying uncertainty of specific graph types.

To address these gaps, we propose Conditional Prediction ROC (CP-ROC) bands, which create prediction bands for ROC curves that are robust to distributional shifts in graph data. Developed for TGNNs, this method can be adapted to GNNs and other neural networks and machine learning models by modifying the similarity measure based on embedded features. Despite the challenge of constructing these prediction bands due to the complex structure of tensorized graph data, we design an efficient, data-driven method for calculating similarities between different tensorized graphs.

[1]Center for Data Science, New York University.
[2]Department of Biostatistics, University of Michigan, Ann Arbor.
[3]Stern School of Business, New York University.
[4]Department of Statistics, University of California, Riverside.

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

Our Contributions. We make three key advances in UQ for graph classification:

(1)*Conditional Prediction ROC (CP-ROC) Bands*: We propose a novel method providing conditional confidence bands for ROC curves, the first of its kind for graph classification.

(2)*Robust Coverage under Graph Covariate Shifts*: We offer statistically guaranteed conditional confidence bands robust to graph covariate shifts. This approach, requiring a local calibration set of similar graph data, significantly enhances model adaptability under a "local exchangeable" condition.

(3)*Empirical Method and Evaluation*: We propose a data-driven approach to construct local calibration sets for TGNNs that ensure local exchangeability. Extensive evaluations on 6 datasets demonstrate the efficacy and robustness of our method.

2 Related Work

Conditional Conformal Prediction for Deep Learning. Conformal Prediction (CP) (Vovk et al., 2005) generates prediction sets that include the true outcome with a specified probability $1-\alpha \in (0,1)$. CP's distribution-free nature makes it versatile in providing robust uncertainty estimates across various fields, including computer vision (Angelopoulos et al., 2020; Bates et al., 2021; Angelopoulos et al., 2022b), causal inference (Lei and Candès, 2021; Jin et al., 2023; Yin et al., 2024), time series forecasting (Gibbs and Candes, 2021; Zaffran et al., 2022), and drug discovery (Jin and Candès, 2023).

Recent work has applied conformal prediction to GNNs for node classification:Zargarbashi et al. (2023) propose a method using node-wise conformity scores, and Huang et al. (2023) introduce Conformalized Graph Neural Networks (CF-GNN) for rigorous uncertainty estimates. Our work differs by focusing on whole-graph classification using GNNs for representation learning. However, our proposed conditional prediction ROC bands can potentially integrate with these node classification methods.

"Conditional coverage" in Conformal Prediction (CP) aims for coverage at specific covariate values, which is challenging without additional assumptions (Lei and Wasserman, 2014; Vovk, 2012; Barber et al., 2021). Approaches for approximate conditional coverage include improved score functions (Romano et al., 2019, 2020b; Angelopoulos et al., 2022a), adapting to covariate shifts (Romano et al., 2020a; Gibbs et al., 2023), and using similar calibration data (Ding et al., 2023; Guan, 2023). However, these methods face limitations with GNNs due to difficulty estimating graph covariate shifts, unsuitability for certain graph types. Thus, a new method for conditional UQ for the ROC is needed.

UQ for ROC. Confidence bands for the ROC curve can be constructed using either point-wisely (Schafer, 1994; Hilgers, 1991) or globally (Jensen et al., 2000; Campbell, 1994). We focus on point-wise confidence band in this work because the global bands are often too wide to be useful. Most existing work on ROC confidence bands is designed for diagnostic testing settings (Nakas et al., 2023), which do not apply to our problem. Additionally, popular bootstrap-based UQ methods (Adler and Lausen, 2009) tend to underestimate uncertainty. Thus, we generalize the conformal-based method from Zheng et al. (2024), ensuring proper coverage of the oracle ROC curve.

Graph Neural Networks (GNN) are crucial for learning representations from graph-structured data, excelling in various applications (Kipf and Welling, 2016; Velickovic et al., 2017). Key developments include message-passing mechanisms (Gilmer et al., 2017) and attention-based models (Veličković and et al., 2018), addressing issues like over-smoothing and scalability (Li et al., 2018). Despite their effectiveness, traditional GNNs lack robust uncertainty quantification, which is critical for applications requiring reliable confidence intervals. While Bayesian approaches and ensemble methods have been explored to address this, they often lack theoretical guarantees and increase computational complexity (Arman Hasanzadeh et al., 2020).

In this paper, we focus on the classification of graph-structured objects, utilizing GNNs for representation learning alongside topological features, and introduce conditional prediction ROC (CP-ROC) bands for ROC curves that are robust to covariate shifts.

3 Problem Setups

An attributed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ consists of nodes $\mathcal{V}$, edges $\mathcal{E}$, and node feature matrix $\boldsymbol{X} \in \mathcal{R}^{N\times F}$. Its adjacency matrix $\boldsymbol{A} \in \mathcal{R}^{N\times N}$ has entries $a_{ij} = \omega_{ij}$ for connected nodes (weight $\omega_{ij} \equiv 1$ for unweighted graphs). The degree matrix $\mathcal{D}$ has $d_{ii} = \sum_j a_{ij}$.

In graph classification, we have graph-label pairs $\overline{\mathcal{G}}_i = (\mathcal{G}_i, y_i)$. The task is to predict a graph's label. A trained model $\widehat{f}(\mathcal{G})$ outputs $\hat{y}$, the most probable label from $\{1,\ldots,L\}$. Our goal is to obtain a prediction set $C(\mathcal{G},\alpha)$ with confidence level $\alpha \in (0,1)$ for label y. We split the dataset $\{\overline{\mathcal{G}}_1, \overline{\mathcal{G}}_2, \ldots, \overline{\mathcal{G}}_\aleph\}$ into four disjoint subsets: training $\overline{\mathcal{G}}_{\text{train}}$, validation $\overline{\mathcal{G}}_{\text{valid}}$, calibration $\overline{\mathcal{G}}_{\text{calib}}$, and test $\overline{\mathcal{G}}_{\text{test}}$. The calibration set is reserved for applying conformal prediction for uncertainty quantification.

Graph Classification. Our method is applicable to any trained model $\widehat{f}(\cdot)$. We demonstrate using Tensorized Graph Neural Networks (TTG-NN) Wen et al. (2024), which excels in whole graph classification. TTG-NN captures different levels of local and global representations in real-world graph data. Topological and graphical features of graphs from multi-filtrations and graph convolutions are aggregated together as a high-order tensor feature whose information are extracted automatically with integrated tensor decompositions. We consider the general Tensorized GNN where the input feature and hidden throughput are all in tensor forms. The input feature may be aggregated from more than two channels besides the topological and graphical channel considered in Wen et al. (2024). As a result, $\mathcal{G}_i$ can be viewed as the input tensor feature in the sequel. Mathematical formulations are detailed in Appendix 9.

ROC Curve and AUC. The *Receiver Operating Characteristic (ROC)* curve evaluates binary classifier performance by plotting True Positive Rate (TPR) against False Positive Rate (FPR) at various classifying thresholds $\lambda \in [0, 1]$:

$$\begin{aligned} \mathrm{TPR}(\lambda) &= \frac{\mathrm{TP}(\lambda)}{\mathrm{TP}(\lambda) + \mathrm{FN}(\lambda)} \\ \mathrm{FPR}(\lambda) &= \frac{\mathrm{FP}(\lambda)}{\mathrm{FP}(\lambda) + \mathrm{TN}(\lambda)} \end{aligned} \tag{1}$$

where $\mathrm{TP}(\lambda)$ (True Positives) is the number of correctly predicted positive instances; $\mathrm{FN}(\lambda)$ (False Negatives) is the number of positive instances incorrectly predicted as negative; $\mathrm{FP}(\lambda)$ (False Positives) is the number of negative instances incorrectly predicted as positive; and $\mathrm{TN}(\lambda)$ (True Negatives) is the number of correctly predicted negative instances.

Area Under the ROC Curve (AUC) is a scalar value that summarizes the overall performance of the classifier. It represents the probability that a randomly chosen positive instance is ranked higher than a randomly chosen negative instance. The AUC can be computed as the integral of the ROC curve:

$$\mathrm{AUC} = \int_0^1 \mathrm{TPR}(\lambda)\big(\mathrm{FPR}(\lambda)\big)\, d\big(\mathrm{FPR}(\lambda)\big).$$

AUC ranges from 0 to 1, with 1 indicating perfect classification, 0.5 random guessing, and < 0.5 worse than random.

The ROC curve visually represents classifier performance across thresholds, showing sensitivity-specificity trade-offs. It's particularly useful when false positives and negatives have different consequences. ROC curves enable classifier comparison, especially for imbalanced datasets, with AUC providing a concise performance summary.

This paper introduces two confidence bands for ROC curve uncertainty quantification: vertical bands for sensitivity (TPR) at fixed FPR, and horizontal bands for specificity (FPR) at fixed TPR, illustrated in Figure 1. These bands offer insights into classifier performance for positive and negative groups. AUC's confidence intervals can be derived by calculating the AUC of the upper and lower bounds of these ROC bands.

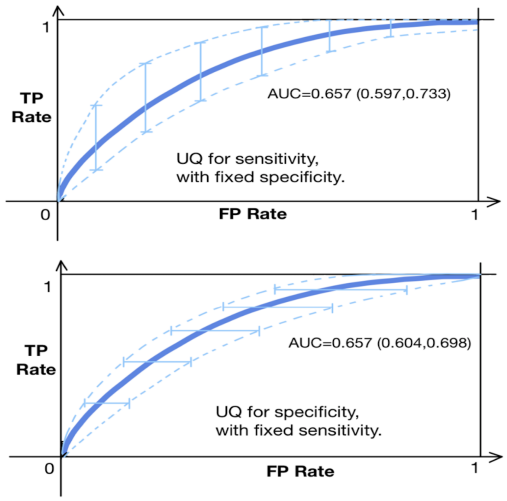


Figure 1: ROC bands for TPR (Top) and FPR (Bottom).

4 Conditional Prediction ROC Bands

This section introduces our method for constructing ROC bands, first for binary classification and then extending to multi-label classification. In binary classification, graph labels are 0 or 1, with an underlying positive probability $\pi(\mathcal{G}) = \mathbb{P}(y = 1|\mathcal{G})$. We assume a trained algorithm $\widehat{f}(\mathcal{G}) = \widehat{f}(\mathcal{G}|\mathcal{G}_{train})$ is available, estimating $\pi(\mathcal{G})$ based on training data $\mathcal{G}_{train}$.

4.1 Soft Conformal Prediction

Existing conformal prediction methods for predicting labels can result in large confidence set, which is uninformative for binary or few-category cases. To develop ROC bands, we first construct conformal prediction intervals for the soft probability $\pi(\mathcal{G})$, offering a more informative prediction with guaranteed coverage.

Under the common assumption of exchangeability among $\mathcal{G}_{valid}$, $\mathcal{G}_{calib}$, and $\mathcal{G}_{test}$, we construct a prediction interval $C(\mathcal{G})$ for any $\mathcal{G} \in \mathcal{G}_{test}$. Given a user-chosen error rate $\alpha \in [0, 1]$, we aim to ensure:

$$\mathbb{P}(\pi(\mathcal{G}) \in C(\mathcal{G})) \geq 1 - \alpha - o_{n_2}(1)$$

where n_2 is the size of calibration set.

To construct the prediction interval, we apply a calibration step involving the following procedures: First, we calculate the non-conformity score: $s_i = \tilde{\pi}(\mathcal{G}_i) - \widehat{f}(\mathcal{G}_i)$ for $\mathcal{G}_i \in \mathcal{G}_{\text{calib}}$, where $\tilde{\pi}(\mathcal{G}_i)$ is a consistent nonparametric estimator of $\pi(\mathcal{G}_i)$. Here we use K nearest neighbors for $\tilde{\pi}(\cdot)$ with distance defined by Topological Data Analysis (TDA) similarity matrix (Chazal and Michel, 2021).

Similarly we define $s_{\mathcal{G}}(\pi) = \pi - \hat{f}(\mathcal{G})$ and a conformal p-value

$$p^{\pi}(\mathcal{G}) = \frac{\sum_{j:\mathcal{G}_j \in \mathcal{G}_{calib}} \mathbf{1}(s_j < s_{\mathcal{G}}(\pi)) + 1}{|\mathcal{G}_{calib}|},$$

indicating a $(1-\alpha)$-level prediction set. Finally, we define the prediction set as $C(\mathcal{G}) = \{\pi : p^{\pi}(\mathcal{G}) \geq \alpha\} :=$

$$\big[\widehat{f}(\mathcal{G}) + q_{\alpha/2}(\{s_i\}), \widehat{f}(\mathcal{G}) + q_{1-\alpha/2}(\{s_i\})\big],$$

where s_i is the estimated non-conformity score for an object in the calibration set, and $q_{\gamma}(\mathcal{A})$ denotes the $\lfloor \gamma|\mathcal{A}| \rfloor$-th order statistic of set $\mathcal{A}$.

4.2 CP-ROC Bands for Exchangeable Data

In this subsection, we leverage the soft conformal prediction interval to construct ROC curve confidence bands under the exchangeable condition. Since that each point on the ROC curve represents TPR versus FPR, we need to develop two types of intervals: one for TPR at fixed specificity, and another for FPR at fixed sensitivity.

For any graph $\mathcal{G} \in \mathcal{G}_{test}$, we modify the previous procedure for developing $C(\mathcal{G})$ to obtain $C_k(\mathcal{G})$, conditioned on label $y = k$. Our aim is to ensure that for any user-chosen error rate $\alpha \in [0,1]$:

$$\mathbb{P}\big(\pi(\mathcal{G}) \in C_k(\mathcal{G}) | y = k\big) \geq 1 - \alpha - o_n(1),$$

where $n = \min(n_1, n_2)$, with n_1 and n_2 being the sizes of the training and calibration sets, respectively.

Given $y = k$, $\mathcal{G}$ and $\mathcal{G}^k_{calib} = \{\mathcal{G}_i \in \mathcal{G}_{calib} : y_i = k\}$ are exchangeable, $C_k(\mathcal{G})$ can be obtained from $\big[c_{lo}(\mathcal{G}, k), c_{up}(\mathcal{G}, k)\big] :=$

$$\big[\widehat{f}(\mathcal{G}) + q_{\alpha/2}(\{s_i\}), \widehat{f}(\mathcal{G}) + q_{1-\alpha/2}(\{s_i\})\big],$$

for $i : \mathcal{G}_i \in \mathcal{G}^k_{calib}$ and $y = k \in \{0,1\}$.

Next we combine all the confidence intervals together to construct confidence intervals $C^{sen}_{\lambda}(\mathcal{G}^1_{test})$ for sensitivity TPR(λ) and $C^{spe}_{\lambda}(\mathcal{G}^0_{test})$ for specificity FPR(λ) at any fixed threshold $\lambda \in [0,1]$:

$$C^{sen}_{\lambda}(\mathcal{G}^1_{test}) = \Big[\frac{1}{|\mathcal{G}^1_{test}|} \sum_{j:\mathcal{G}_j \in \mathcal{G}^1_{test}} \mathbf{1}(c_{lo}(\mathcal{G}, 1) > \lambda), \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j:\mathcal{G}_j \in \mathcal{G}^1_{test}} \mathbf{1}(c_{up}(\mathcal{G}, 1) > \lambda)\Big],$$

$$C^{spe}_{\lambda}(\mathcal{G}^0_{test}) = \Big[\frac{1}{|\mathcal{G}^0_{test}|} \sum_{j:\mathcal{G}_j \in \mathcal{G}^0_{test}} \mathbf{1}(c_{lo}(\mathcal{G}, 0) > \lambda), \frac{1}{|\mathcal{G}^0_{test}|} \sum_{j:\mathcal{G}_j \in \mathcal{G}^0_{test}} \mathbf{1}(c_{up}(\mathcal{G}, 0) > \lambda)\Big].$$

Finally, ROC bands can be plotted by collecting $C^{sen}_{\lambda}(\mathcal{G}^1_{test})$ and $C^{spe}_{\lambda}(\mathcal{G}^0_{test})$ with $\lambda \in [0,1]$.

4.3 CP-ROC Bands for Non-Exchangeable Data

Now we discuss the case where test data are non-exchangeable with calibration set, which is our primary goal. In this case, we need to modify $C_k(\mathcal{G})$ to the conditional prediction set $C^{cond}_k(\mathcal{G})$ for any graph $\mathcal{G} \in \mathcal{G}_{test}$ with label y, such that for any given a user-chosen error rate $\alpha \in [0,1]$,

$$\mathbb{P}\big(\pi(\mathcal{G} \in C^{cond}_k(\mathcal{G}) \mid \mathcal{G}, y = k\big) \geq 1 - \alpha - o_n(1).$$

To guarantee the conditional coverage given $\mathcal{G}_j \in \mathcal{G}_{test}$, our adopted strategy is to replace $\mathcal{G}_{calib}$ by $\mathcal{G}^j_{calib} = \{\mathcal{G}_i \in \mathcal{G}_{calib} : \mathcal{G}_i \text{ Similar to } \mathcal{G}_j\}$. While the "similarity" can be measured in different ways, here we use TDA similarity matrix to find the K nearest neighbors for our test point. Under the assumption that $\mathcal{G}_j$ is exchangeable with $\mathcal{G}^j_{calib}$, we extend the previous algorithm to the non-exchangeable case by using $\mathcal{G}^j_{calib}$ instead of $\mathcal{G}_{calib}$.

Specifically, we obtain $C^{cond}_k(\mathcal{G}_j)$ from $\big[c'_{lo}(\mathcal{G}_j, k), c'_{up}(\mathcal{G}_j, k)\big] :=$

$$\big[\widehat{f}(\mathcal{G}) + q_{\alpha/2}(\{s_i\}), \widehat{f}(\mathcal{G}) + q_{1-\alpha/2}(\{s_i\})\big],$$

where $i : \mathcal{G}_i \in \mathcal{G}^{j,k}_{calib}$, $\mathcal{G}^{j,k}_{calib} = \{\mathcal{G}_i \in \mathcal{G}^j_{calib} : y_i = k\}$ is a subset of $\mathcal{G}^j_{calib}$, and $y = k \in \{0,1\}$. Then we can combine all the conditional confidence intervals together to construct confidence intervals $C^{sen}_{cond,\lambda}(\mathcal{G}^1_{test})$ for sensitivity TPR(λ) and $C^{spe}_{cond,\lambda}(\mathcal{G}^0_{test})$ for speci-

ficity FPR(λ) at any fixed threshold $\lambda \in [0,1]$:

$$C^{sen}_{cond,\lambda}(\mathcal{G}^1_{test}) = \Bigg[\frac{1}{|\mathcal{G}^1_{test}|}\sum_{j:\mathcal{G}_j\in\mathcal{G}^1_{test}} \mathbf{1}(c'_{lo}(\mathcal{G}_j,1)>\lambda), \frac{1}{|\mathcal{G}^1_{test}|}\sum_{j:\mathcal{G}_j\in\mathcal{G}^1_{test}} \mathbf{1}(c'_{up}(\mathcal{G}_j,1)>\lambda)\Bigg],$$

$$C^{spe}_{cond,\lambda}(\mathcal{G}^0_{test}) = \Bigg[\frac{1}{|\mathcal{G}^0_{test}|}\sum_{j:\mathcal{G}_j\in\mathcal{G}^0_{test}} \mathbf{1}(c'_{lo}(\mathcal{G}_j,0)>\lambda), \frac{1}{|\mathcal{G}^0_{test}|}\sum_{j:\mathcal{G}_j\in\mathcal{G}^0_{test}} \mathbf{1}(c'_{up}(\mathcal{G}_j,0)>\lambda)\Bigg].$$

Finally, CP-ROC bands can be plotted by collecting $C^{sen}_{cond,\lambda}(\mathcal{G}^1_{test})$ and $C^{spe}_{cond,\lambda}(\mathcal{G}^0_{test})$, $\lambda \in [0,1]$.

4.4 Extension to Multi-label Classification

In this subsection, we consider a multi-label classification model with label $y \in \{1,2,\cdots,L\}$ for the graph $\mathcal{G}$. To plot the ROC curve in this case, the common practice in the literature is to construct separate ROC curves for each label. For a given label $k \in \{1,2,\cdots,L\}$, we define the binary outcome $y_{k,i} = \mathbf{1}(y_i = k)$ for any $\mathcal{G}_i$ in the observed data set, and let $\widehat{f}_k(\mathcal{G})$ be the estimated probability that the graph $\mathcal{G}$ is assigned label k by the algorithm trained on the training set. The procedure described in Section 4.3 can then be applied to visualize the CP-ROC for label k of the multi-label classification model.

5 Conditional CP-ROC Bands for Tensorized Graph Neural Networks

We provided a brief overview of the TGNN training process before delving into the construction of CP-ROC bands, which is detailed in Algorithm 1.

The TGNN, as outlined in Algorithm 2 in the Appendix, begins with randomly initialized parameters. The dataset is then divided into training and testing_calibration sets. The model undergoes a typical deep learning training regimen, which includes multiple epochs of batch processing, generating predictions, evaluating Binary Cross-Entropy loss, and updating parameters through backpropagation. This training process is designed to enhance TGNN's predictive accuracy for graph-structured inputs.

The ROC conformal prediction process employs an iterative approach, utilizing multiple splits of the test and calibration pools with varying random seeds. For exchangeable data, the entire calibration set is used for each test graph. For non-exchangeable data, a predefined topology similarity matrix D identifies the K nearest data points to form a new calibration set. Subsequently, for each calibration graph $\mathcal{G}_i$, the similarity matrix D is used again to calculate the mean probability of the K nearest datapoints, enabling the calculation of the non-parametric estimator $\tilde{\pi}(\mathcal{G}_i)$ for the non-conformity score, as outlined in Section 4.1.

Algorithm 1 Pseudo-code of ROC Bands for Tensorized Graph Neural Networks

```
train, test_calib_pool = split(dataset)
Set learning rate α, number of epochs N, batch size B, number of CP epochs M
                                                    ▷ TGNN Pre-Training
Call Appendix Algorithm 2 for TGNN pre-training.
                                                    ▷ ROC conformal prediction step
Load pre-trained TGNN model f, and TDA similarity distance D
Get probabilities p for each datapoint in dataset from pre-trained model f
for each iteration from 1 to M do
    test, calib = split(test_calib_pool, random_seed)
    for each test do
        if not exchangeable then
            Find K nearest calibration sets using D
            Set calib = K nearest calibration sets
        end if
        for each calib i do
            Find K nearest train sets using D
            f̂(G_i) = p[i]
            π̃(G_i) = mean(p[K nearest train sets])
            s_i = π̃(G_i) − f̂(G_i)
        end for
        Calculate lower and upper quantiles for sensitivity and specificity adjustments
    end for
end for
```

It's important to discuss how we calculate the topology similarity distance matrix D. In our work, we let $\mathcal{D}_{\mathcal{G}_i}$ and $\mathcal{D}_{\mathcal{G}_j}$ be the persistence diagrams for two graphs $\mathcal{G}_i$ and $\mathcal{G}_j$ by using persistent homology (Wasserman, 2016). The Wasserstein distance between these persistence diagrams, denoted by $\mathcal{W}_p(\mathcal{D}_{\mathcal{G}_i}, \mathcal{D}_{\mathcal{G}_j})$, is calculated as follow:

$$\mathcal{W}_p(\mathcal{D}_{\mathcal{G}_i}, \mathcal{D}_{\mathcal{G}_j}) = \inf_{\gamma\in\Gamma}\Big(\sum_{(x,y)\sim\gamma} \|x-y\|^p_\infty\Big)^{1/p}, \quad (2)$$

Our algorithm's final step involves computing lower and upper quantiles for both sensitivity and specificity. This process establishes robust ROC bands that quantifies the uncertainty in the model's predictions.

5.1 Theoretical Gaurantees

This section focuses on establishing the theoretical coverage of the proposed CP-ROC bands $C^{sen}_{cond,\lambda}(\mathcal{G}^1_{test})$ and $C^{spe}_{cond,\lambda}(\mathcal{G}^0_{test})$. Our objective is to quantify the uncertainty of $\widehat{f}(\mathcal{G})$, where the underlying truth is represented by the oracle probability $\pi(\mathcal{G})$. Consequently, our confidence bands aim to encompass

the oracle $\text{FPR}^{(o)}(\lambda)$ and $\text{TPR}^{(o)}(\lambda)$, which are defined as follows:

$$\text{TPR}^{(o)}(\lambda) = \frac{\sum_{j:\mathcal{G}_j \in \mathcal{G}^1_{test}} \mathbf{1}(\pi(\mathcal{G}_j) \geq \lambda)}{|\mathcal{G}^1_{test}|},$$
$$\text{FPR}^{(o)}(\lambda) = \frac{\sum_{j:\mathcal{G}_j \in \mathcal{G}^0_{test}} \mathbf{1}(\pi(\mathcal{G}_j) \geq \lambda)}{|\mathcal{D}^0_{test}|}$$

We aim to develop asymptotic $(1-\alpha)$-level coverage for our two target confidence intervals at a fixed confidence level α. We also aim to ensure that our developed confidence band consistently encompasses the ROC curve. This requires almost sure coverage of $\text{FPR}(\lambda)$ and $\text{TPR}(\lambda)$ based on the TGNN used in practice. We present all our theoretical findings in Theorem 1, with a detailed proof provided in Appendix 8.

Theorem 1. *Assume $\mathcal{G}_{test}$ are i.i.d data set, and each $\mathcal{G}_j \in \mathcal{G}_{test}$ is i.i.d with its K-nearest calibration sets $\mathcal{G}^j_{calib}$ and also K-nearest training sets $\mathcal{G}^j_{train}$. Let $F_{jk}(\cdot)$ denotes the CDF of $\{s_i\}_{i:\mathcal{G}_i \in \mathcal{N}_j, y_i = k}$, assume $F_{jk}(\cdot)$ is Lipschitz continuous , then*

$$\lim_{|\mathcal{G}_{train}|,|\mathcal{G}^1_{test}|,K\to\infty} P\big(\text{TPR}^{(o)}(\lambda) \in C^{sen}_{cond,\lambda}(\mathcal{D}^1_{tst}, \alpha)\big) \geq 1-\alpha, \text{ for } \lambda = \pi(\mathcal{G}_s), \mathcal{G}_s \in \mathcal{G}^1_{test}$$

$$\lim_{|\mathcal{G}_{train}|,|\mathcal{G}^0_{test}|,K\to\infty} P\big(\text{FPR}^{(o)}(\lambda) \in C^{spe}_{cond,\lambda}(\mathcal{D}^0_{tst}, \alpha)\big) \geq 1-\alpha, \text{ for } \lambda = \pi(\mathcal{G}_s), \mathcal{G}_s \in \mathcal{G}^0_{test}$$

In addition, we assume $F^{-1}_{jk}(\alpha/2) < 0 < F^{-1}_{jk}(1-\alpha/2)$. Then for any $\lambda \in (0,1)$, as $|\mathcal{G}^k_{test}|, |\mathcal{G}_{train}|, K \to \infty$, almost surely,

$$\text{TPR}(\lambda) \in C^{sen}_{cond,\lambda}(\mathcal{D}^1_{tst}, \alpha)$$
$$\text{FPR}(\lambda) \in C^{spe}_{cond,\lambda}(\mathcal{D}^0_{tst}, \alpha)$$

Notice that by definition, the ROC curve is a step function with jumps at $\lambda = \pi(\mathcal{G}), \mathcal{G} \in \mathcal{G}_{test}$. Thus, the first part of the theorem above shows that if we randomly choose a jump point on the ROC, the confidence interval with fixed sensitivity (for a point with a negative outcome) or with sensitivity (for a point with a positive outcome) will cover the oracle $\text{FPR}^{(o)}$ or $\text{TPR}^{(o)}$ with confidence level converging to $1-\alpha$.

As we use the K-nearest neighbors to guarantee the exchangeability as well as the consistency of $\tilde{\pi}(\cdot)$, this algorithm will largely depends on the similarity measure we choose and also the neighbor size K. In the theorem, we also requires $K \to \infty$, thus also requiring large size of calibration and training data.

The assumption that CDF $F_{jk}(\cdot)$ is Lipschitz continuous is weak because the underlying probability space of the graph is likely continuous. Finally, for coverage of $\text{TPR}(\lambda)$ and $\text{FPR}(\lambda)$, we assume the conformity scores have heavy left tails below 0 and right tails above 0. If an algorithm violates this assumption, we can calibrate it according to its prediction error, so that the conformity score is shifted to center at 0.

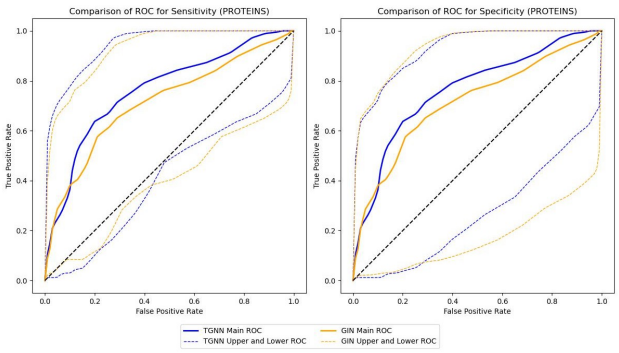


Figure 2: Example of exchangeable ROC bands for TGNN and GIN.

6 Experiments

6.1 Experiment Settings

Datasets and Baselines. We assess the uncertainty quantification performance of the TGNN model on graph classification tasks on chemical compounds and protein molecules. For chemical compounds, the datasets include DHFR, BZR, and COX2 (Sutherland et al., 2003; Kriege and Mutzel, 2012), which are composed of graphs representing chemical compounds with nodes as atoms and edges as chemical bonds. DHFR is used for enzyme-ligand binding affinity prediction, BZR for bioactivity against the benzodiazepine receptor, and COX2 for predicting enzyme inhibition, crucial in drug design. In the case of protein molecules, we use datasets including PROTEINS, D&D, PTC_MR and PTC_MM (Helma et al., 2001; Dobson and Doig, 2003; Borgwardt et al., 2005; Kriege and Mutzel, 2012), where each protein is depicted as a graph with nodes representing amino acids and edges denoting interactions like physical bonds or spatial proximity. These datasets aid in tasks such as protein structure classification and distinguishing chemical compounds based on carcinogenicity in male mice (MM). All selected datasets involve binary classification, making them suitable for testing ROC curve-based UQ methods. Our experimental setup includes a split ratio of 0.8/0.2 for the training and testing_calibration pool, respectively, and a further split of the testing_calibration pool into testing and calibration subsets with a ratio of 0.5/0.5. Table 2 in the Appendix summarizes the characteristics of these datasets.

Table 1: Classification and UQ performance on molecular and chemical graphs. Best results are in **bold**.

Datasets	TGNN_AUC	GIN_AUC	TGNN_SEN(i.i.d, Non i.i.d)	GIN_SEN (i.i.d, Non i.i.d)	TGNN_SPE (i.i.d, Non i.i.d)	GIN_SPE (i.i.d, Non i.i.d)
BZR	**0.8380**	0.8267	**0.3418**, **0.3393**	0.3834, 0.3693	0.6695, 0.6386	**0.5949**, **0.5611**
COX2	**0.7205**	0.7054	**0.5939**, **0.4765**	0.6157, 0.5035	0.6712, 0.5712	**0.6320**, **0.5330**
DHFR	0.8514	**0.8525**	0.4417, **0.3573**	**0.4360**, 0.3645	**0.5448**, **0.4810**	0.6079, 0.5422
PTC_MM	0.7191	**0.7202**	**0.6021**, **0.5223**	0.6109, 0.5538	0.7458, **0.6186**	**0.7414**, 0.6249
PTC_MR	**0.6881**	0.6359	**0.6122**, **0.5333**	0.6208, 0.5432	0.7367, **0.6077**	**0.7315**, 0.6156
D&D	**0.8020**	0.7888	**0.4291**, **0.3724**	0.4307, 0.3783	0.5546, 0.4571	**0.5326**, **0.4442**
PROTEINS	**0.7631**	0.7132	0.5396, 0.5091	**0.5378**, **0.5007**	**0.6611**, **0.6156**	0.7613, 0.7315

Experimental Setup. We implement our TGNN model within a Pytorch framework on a single NVIDIA Quadro RTX 8000 GPU with a 48GB memory capacity. We employ the Adam optimizer with a learning rate of 0.001 and utilize ReLU as the activation function across the model, except for the Softmax activation in the MLP classifier output. The resolution of the persistence image (PI) (Adams et al., 2017) (i.e., we vectorize a persistence diagram $\mathcal{D}_{\mathcal{G}_i}$ into the $\text{PI}_{\mathcal{D}_{\mathcal{G}_i}}$) is set to $P = 50$. We explore five different filtrations in our experiments: degree-based, betweenness-based, closeness-based, communicability-based, and eigenvector-based, with batch sizes maintained at 16. The graph convolution layers and MLPs are optimized to determine the best number of hidden units, chosen from $\{16, 32, 64, 128, 256\}$, with TTL featuring 32 hidden units. Our TGNN model incorporates three layers in the graph convolution blocks and two layers in the MLPs, with a consistent dropout rate of 0.5. For conformal prediction, $K = \{20, 30, 50\}$ neighbors are considered in the K-nearest neighbor approach, with an error rate α of 0.1, the exact number of neighbors being contingent upon the dataset size. Our code is available at Github

6.2 Results

In this section, we showcase experimental results of our novel uncertainty quantification method for ROC curves. As shown in Table 1, it highlights the areas under the ROC curve (AUC) for TGNN and GIN models (Xu et al., 2018), i.e., in the first two columns. The latter columns display the ROC bandwidths for sensitivity and specificity under both i.i.d and non-i.i.d conditions. We observe that our proposed CP-UQ method not only enhances the predictive reliability of graph neural networks across various binary classification tasks but extends to other machine learning methods, including logistic regression and SVMs, as well as additional deep learning models. Note that, unlike traditional uncertainty quantification methods which do not offer ROC bands, our approach generates ROC bands for both visual and quantitative measures of uncertainty.

Additionally, our results reveal that the non-i.i.d ROC bandwidths are consistently smaller than those in i.i.d scenarios. Due to the limited size of training and test dataset, the test data may be inconclusive, which makes the uncertainty by soft conformal prediction under the i.i.d setting larger than it should be. However, our proposed soft conformal prediction under the non-i.i.d settings show good capability handling this condition. For each test point $(\mathcal{G}_i, y_i) \in \mathcal{G}_{test}$, the subgroup of calibration dataset $\mathcal{N}_i = \{(\mathcal{G}, y) \in \mathcal{D}_{ca} : dist(\mathcal{G}, \mathcal{G}_i) \leq d\}$, which includes similar individuals as the prediction target, are chosen as a substitute of calibration set. $(\mathcal{G}_i, y_i)$ and corresponding $\mathcal{N}_i$ will be approximately i.i.d if $\mathcal{N}_i$ is correctly selected such that the uncertainty will be measured accurately Zheng et al. (2024). As a result, the uncertainty of our algorithm is smaller using the non-i.i.d soft conformal prediction.

Our findings also indicate that despite high AUC values from the models, some datasets exhibit moderate uncertainty. The quantification of uncertainty in TGNN is limited by dataset size, and as a compromise, the calibration dataset comprises 20% data. Soft conformal prediction, which ensures coverage, requires sufficient calibration points to achieve narrow and precise uncertainty estimation. The confidence interval narrows as more points from the target distribution are included in the calibration set. Despite these limitations and the potential for future enhancements with larger datasets, the current performance of the TGNN on these datasets is promising.

The estimated AUC does not always inversely correlate with the algorithm's uncertainty. While the AUC measures overall performance, including sensitivity and specificity, it is notable that the TGNN model achieves higher AUCs and shorter confidence intervals for sensitivity, which improves hit rates. However, there is a slight increase in the ROC bands and uncertainty for specificity. Essentially, the algorithm more reliably identifies true positives but with a marginal loss in certainty for rejecting false positives. Notably, the maximum increase in the confidence interval for specificity is less than 0.06, whereas the decrease in uncertainty for sensitivity is more pronounced. Moreover, expanding the dataset size is expected to enhance the precision of uncertainty quantification, potentially tightening the ROC bands for specificity further.

The ROC comparisons for the PROTEINS data, as illustrated in the Figure 2, visually corroborate the

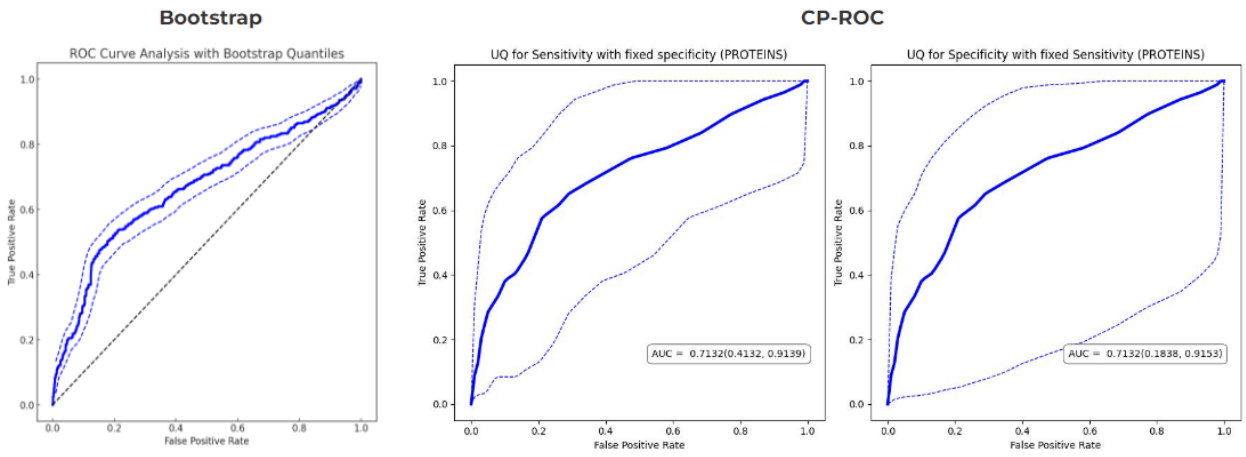


Figure 3: Comparison between Bootstrap and our UQ on PROTEIN dataset

numerical findings. That is, we observe that TGNN consistently outperforms GIN across most metrics, demonstrating higher true positive rates and maintaining tighter confidence intervals within the ROC bands. These visual representations align well with our quantitative analyses, reinforcing the superiority of TGNN in handling complex graph-based datasets. Please refer to figures in the Appendix13 for more details.

Furthermore, to test the robustness of our uncertainty quantification methodology, we applied it to the PROTEINS dataset using a bootstrap method with a resample size of 1000. Figure 3 presents the 95% confidence bands of the ROC curves based on bootstrap resampling, alongside the ROC bands generated by our method. We observe that although our area under the ROC curve (AUC) is approximately 0.7 (i.e., moderate predictive power), the uncertainty derived from the bootstrap method is relatively small. This small uncertainty can be misleading, especially given that the dataset we used is imbalanced, which naturally introduces higher uncertainty in model predictability. Additionally, there is no theoretical support for using the bootstrap method in uncertainty quantification, which raises concerns about its validity in these contexts.

6.3 Extension to Regression Task

To further demonstrate the versatility of our approach, we extend its application to regression tasks. This broader scope allows us to validate the robustness of our uncertainty quantification methodology across different types of predictive modeling scenarios beyond graph classification.

For the regression experiments, we simulate datasets by introducing complexities commonly found in real-world applications, such as missing covariates. We consider generating observed data from the following model:

$$\begin{aligned} Y &\sim \text{Ber}(\pi(X)) \\ \text{logit}(\pi(X)) &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 \end{aligned} \tag{3}$$

And we specify $X = (x_1, x_2, x_3)^T \in \mathbb{R}^3$ are the vector of independent covariates. And β_1, β_2 and β_3 are real values.

We investigate two different logistic regression models on different sample size n: a baseline model **M1**: $\text{logit}(\pi(X)) = \beta_0' + \beta_1' X_1 + \beta_2' X_2 + \beta_3' X_3$ with all relevant covariates correctly specified, and a misspecified model **M2**: $\text{logit}(\pi(X)) = \beta_0' + \beta_1' X_1 + \beta_2' X_2$ with one missing covariate. These models are trained on datasets of varying sizes to assess the scalability and stability of our method. Specifically, we compare their performance under different datasets sizes of $n \in \{50, 100, 1000\}$.

The top row of Figure 4 shows model performance comparison under independent (iid) test data using exchangeable CP, and the bottom row shows model performance under non-independent (non-iid) test data using non-exchangeable CP. By examining these different testing conditions, we demonstrate the method's robustness when applied to both iid and non-iid data, reflecting realistic conditions where data independence cannot always be assumed.

Scalability is a key concern when working with large datasets, and Figure 4 provides insight into how our proposed uncertainty quantification approach adapts as the training sample size increases. The method shows reliable performance even when covariates are missing in model M2, maintaining stable uncertainty estimates across varying complexities. A more thorough validation of this uncertainty quantification

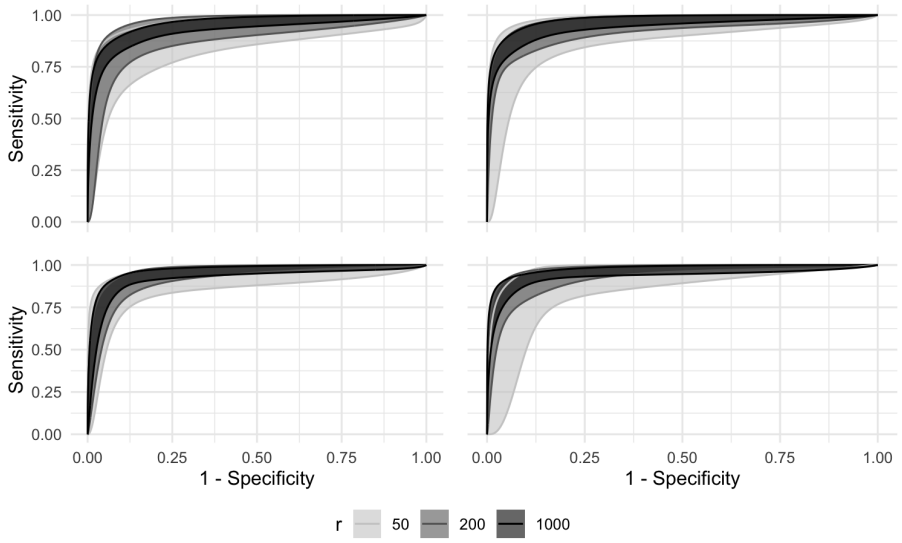


Figure 4: Performance comparison of two regression models across varying dataset sizes. The top row shows fixed sensitivity on i.i.d. test data; the bottom row shows fixed sensitivity on non-i.i.d. test data. The first column represents the correctly specified model (M1) performance, and the second column shows the misspecified model (M2) performance.

methodology can be found in the experiments of (Zheng et al., 2024).

By validating the uncertainty quantification methodology on regression tasks, we demonstrate its applicability beyond TGNN-based models. The experimental results, drawn from both synthetic and real-world datasets, indicate that our method can effectively handle missing data and scale efficiently to larger datasets. This extension reinforces the flexibility and robustness of our approach, establishing its potential for a wide range of predictive modeling tasks beyond classification. Moreover, the limitations observed in the bootstrap method further emphasize the importance of our approach, which offers a more theoretically grounded and robust framework for uncertainty quantification.

7 Conclusion

This paper introduces CP-ROC bands, a novel method designed for TGNNs to quantify the uncertainty of ROC curves, enhancing reliability in graph classification tasks. CP-ROC, adaptable to various neural networks, addresses challenges in uncertainty quantification due to covariate shifts, and its effectiveness is confirmed through extensive evaluations showing improved prediction reliability and efficiency. In the future, we will extend this idea to the spatio-temporal forecasting tasks.

Acknowledgments

This work was supported in part by the National Science Foundation under Grant DMS-2412577. Y.C. has been supported in part by the National Science Foundation grant NSF DMS-2335846/2335847. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Adams, H., Emerson, T., Kirby, M., Neville, R., Peterson, C., Shipman, P., Chepushtanova, S., Hanson, E., Motta, F., and Ziegelmeier, L. (2017). Persistence images: A stable vector representation of persistent homology. *Journal of Machine Learning Research*, 18.

Adler, W. and Lausen, B. (2009). Bootstrap estimated true and false positive rates and roc curve. *Computational statistics & data analysis*, 53(3):718–729.

Angelopoulos, A. N., Bates, S., Jordan, M., and Malik, J. (2020). Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*.

Angelopoulos, A. N., Kohli, A. P., Bates, S., Jordan, M., Malik, J., Alshaabi, T., Upadhyayula, S., and Romano, Y. (2022a). Image-to-image regression with distribution-free uncertainty quantification and applications in imaging. In *International Conference on Machine Learning*.

Angelopoulos, A. N. S. B., Malik, J., and Jordan, M. I. (2022b). Uncertainty sets for image classifiers using conformal prediction. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*.

Arman Hasanzadeh, E. H., Boluki, S., Zhou, M., Duffield, N., Narayanan, K., and Qian, X. (2020). Bayesian graph neural networks with adaptive connection sampling. *Proceedings of the 37th International Conference on Machine Learning, Online, PMLR 119, 2020*.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. (2021). The limits of distribution-free conditional predictive inference. *Information and Inference*, 10(2):455–482.

Bates, S., Angelopoulos, A., Lei, L., Malik, J., and Jordan, M. (2021). Distribution-free, risk-controlling prediction sets. *Journal of the ACM (JACM)*, 68(6):1–34.

Borgwardt, K. M., Ong, C. S., Schönauer, S., Vishwanathan, S., Smola, A. J., and Kriegel, H.-P. (2005). Protein function prediction via graph kernels. *Bioinformatics*, 21:i47–i56.

Campbell, G. (1994). Advances in statistical methodology for the evaluation of diagnostic and laboratory tests. *Statistics in medicine*, 13(5-7):499–508.

Chazal, F. and Michel, B. (2021). An introduction to topological data analysis: fundamental and practical aspects for data scientists.

Ding, T., Angelopoulos, A. N., Bates, S., Jordan, M. I., and Tibshirani, R. J. (2023). Class-conditional conformal prediction with many classes. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Dobson, P. D. and Doig, A. J. (2003). Distinguishing enzyme structures from non-enzymes without alignments. *Journal of Molecular Biology*, 330(4):771–783.

Gao, X., Jiang, X., Zhuang, D., Chen, H., Wang, S., and Haworth, J. (2022). Spatiotemporal graph neural networks with uncertainty quantification for traffic incident risk prediction.

Gibbs, I. and Candes, E. (2021). Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672.

Gibbs, I., Cherian, J. J., and Candès, E. J. (2023). Conformal prediction with conditional guarantees. *arXiv preprint arXiv:2305.12616*.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1263–1272.

Guan, L. (2023). Localized conformal prediction: A generalized inference framework for conformal prediction. *Biometrika*, 110(1):33–50.

Helma, C., King, R. D., Kramer, S., and Srinivasan, A. (2001). The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108.

Hilgers, R. (1991). Distribution-free confidence bounds for roc curves. *Methods of information in medicine*, 30(02):96–101.

Huang, K., Jin, Y., Candès, E., and Leskovec, J. (2023). Uncertainty quantification over graph with conformalized graph neural networks.

Jensen, K., Müller, H.-H., and Schäfer, H. (2000). Regional confidence bands for roc curves. *Statistics in medicine*, 19(4):493–509.

Jin, Y. and Candès, E. J. (2023). Selection by prediction with conformal p-values. *Journal of Machine Learning Research*, 24(244):1–41.

Jin, Y., Ren, Z., and Candès, E. J. (2023). Sensitivity analysis of individual treatment effects: A robust conformal inference approach. *Proceedings of the National Academy of Sciences*, 120(6):e2214889120.

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

Kriege, N. and Mutzel, P. (2012). Subgraph matching kernels for attributed graphs. In *Proceedings of the International Conference on Machine Learning*, pages 291–298.

Lei, J. and Wasserman, L. (2014). Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B*, 76(1):71–96.

Lei, L. and Candès, E. J. (2021). Conformal inference of counterfactuals and individual treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 83(5):911–938.

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Munikoti, S., Agarwal, D., Das, L., and Natarajan, B. (2023). A general framework for quantifying aleatoric and epistemic uncertainty in graph neural networks. *Neurocomputing*, 521:1–10.

Nakas, C. T., Bantis, L. E., and Gatsonis, C. A. (2023). *ROC analysis for classification and prediction in practice*. Chapman and Hall/CRC.

Romano, Y., Barber, R. F., Sabatti, C., and Candès, E. J. (2020a). With malice toward none: Assessing uncertainty via equalized coverage. *Harvard Data Science Review*, 2(2).

Romano, Y., Patterson, E., and Candès, E. J. (2019). Conformalized quantile regression. In *Advances in Neural Information Processing Systems*.

Romano, Y., Sesia, M., and Candès, E. J. (2020b). Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*.

Schafer, H. (1994). Efficient confidence bounds for roc curves. *Statistics in medicine*, 13(15):1551–1561.

Sutherland, J. J., O'brien, L. A., and Weaver, D. F. (2003). Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *Journal of Chemical Information and Computer Sciences*, 43(6):1906–1915.

Tibshirani, R. J., Foygel Barber, R., Candes, E., and Ramdas, A. (2019). Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32.

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

Veličković, P. and et al. (2018). Graph attention networks. In *International Conference on Learning Representations*.

Vovk, V. (2012). Conditional validity of inductive conformal predictors. In *Asian Conference on Machine Learning*.

Vovk, V., Gammerman, A., and Shafer, G. (2005). *Algorithmic learning in a random world*, volume 29. Springer.

Wasserman, L. (2016). Topological data analysis.

Wen, T., Chen, E., and Chen, Y. (2024). Tensor-view topological graph neural network.

Xia, F., Sun, K., Yu, S., Aziz, A., Wan, L., Pan, S., and Liu, H. (2021). Graph learning: A survey. *IEEE Trans AI*, 2(2):109–127.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. (2018). How powerful are graph neural networks? In *Proceedings of International Conference on Learning Representations*.

Yin, M., Shi, C., Wang, Y., and Blei, D. M. (2024). Conformal sensitivity analysis for individual treatment effects. *Journal of the American Statistical Association*, 119(545):122–135.

Zaffran, M., Féron, O., Goude, Y., Josse, J., and Dieuleveut, A. (2022). Adaptive conformal predictions for time series. In *International Conference on Machine Learning*, pages 25834–25866. PMLR.

Zargarbashi, S. H., Antonelli, S., and Bojchevski, A. (2023). Conformal prediction sets for graph neural networks. *Proceedings of the 40th International Conference on Machine Learning, PMLR 202:12292-12318, 2023.*

Zheng, Z., Yang, B., and Song, P. (2024). Quantifying uncertainty in classification performance: Roc confidence bands using conformal prediction. *arXiv preprint arXiv:2405.12953*.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81.

Zhuang, D., Wang, S., Koutsopoulos, H., and Zhao, J. (2022). Uncertainty quantification of sparse travel demand prediction with spatial-temporal graph neural networks.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes, please refer to algorithms 1 and 2 for GNN pre-train model and the UQ method, section 5.1 for the mathematical setting, assumption.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes, please refer to the section 6.2

(c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries.
Yes. The code is available at Github.

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

(a) Statements of the full set of assumptions of all theoretical results.
Yes, please refer to the section 5.1 and appendix 8

(b) Complete proofs of all theoretical results.
Yes, please refer to the section 5.1 and appendix 8

(c) Clear explanations of any assumptions.
Yes, please refer to the section 5.1 and appendix 8

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

(a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes, the code is available at Github.

(b) All the training details (e.g., data splits, hyperparameters, how they were chosen).
Yes, please refer to the section 6.1.

(c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times).
Yes, please refer to the section 6.2.

(d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider).
Yes, please refer to the section 6.1.

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

(a) Citations of the creator If your work uses existing assets.
Yes, all datasets used are cited.

(b) The license information of the assets, if applicable.
Not Applicable.

(c) New assets either in the supplemental material or as a URL, if applicable. Not Applicable.

(d) Information about consent from data providers/curators.
Yes, datasets are all public datasets

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content.
Not Applicable.

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

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

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

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

Supplementary Material

8 Proof of Theorem 1

For the first part of the theorem, we only need to show the coverage rate for $TPR^{(o)}(\lambda)$ and the coverage rate for $FPR^{(o)}(\lambda)$ is exactly the same.

$$
\begin{aligned}
&P\big(TPR^{(o)}(\lambda) \in C^{sen}_{cond,\lambda}(\mathcal{G}^1_{test})\big) = \\
&P\Bigg\{ \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[c'_{lo}(\mathcal{G}_j,1) > \lambda\big] \le \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{I}^1_{tst}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big], \text{ and} \\
&\qquad \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big] \le \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[c'_{up}(\mathcal{G}_j,1) > \lambda\big] \Bigg\}
\end{aligned} \tag{4}
$$

We only need to show the probability bound for one side in Equation 4, and the other side will be the same. For the left side,

$$
\begin{aligned}
&P\Bigg\{ \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big] < \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[c'_{lo}(\mathcal{G}_j,1) > \lambda\big] \Bigg\} \\
&= P\Bigg\{ \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big] < \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\widehat{f}(\mathcal{G}_j) + q_{\alpha/2}(\{s_i\}_{i:\mathcal{G}_i\in\mathcal{G}^{j,1}_{calib}}) > \lambda\big] \Bigg\}
\end{aligned}
$$

Define $s'_j = \pi(\mathcal{G}_j) - \widehat{f}(\mathcal{G}_j)$ for $\mathcal{G}_j \in \mathcal{G}^1_{test}$. For the right hand side inside the last term,

$$
\begin{aligned}
&\sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\widehat{f}(\mathcal{G}_j) + q_{\alpha/2}(\{s_i\}_{i:\mathcal{G}_i\in\mathcal{G}^{j,k}_{calib}}) > \lambda\big] \\
&= \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda + s'_j - q_{\alpha/2}(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}})\big]\mathbf{1}\big[\pi(\mathcal{G}_j) \le \lambda\big] \\
&\qquad + \big[\pi(\mathcal{G}_j) > \lambda + s'_j - q_{\alpha/2}(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}})\big]\mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big] \\
&\le \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[s'_j < q_{\alpha/2}(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}})\big]\mathbf{1}\big[\pi(\mathcal{G}_j) \le \lambda\big] \\
&\qquad + \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda + s'_j - q_{\alpha/2}(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}})\big]\mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big] \\
&\le \sum_{j\in\mathcal{I}^1_{tst}} \mathbf{1}\big[s'_j < q_{\alpha/2}\big(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}}\big)\big] \\
&\qquad + \sqrt{\sum_{j\in\mathcal{I}^1_{tst}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda + s'_j - q_{\alpha/2}(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}})\big] \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \lambda\big]}
\end{aligned}
$$

The last inequality comes from Cauchy inequality and also the fact that $\mathbf{1}^2(A) = \mathbf{1}(A)$ for any event A. From the assumption that $\mathcal{G}_j$ are iid with its corresponding $\mathcal{G}^j_{calib}$, so are $\mathcal{G}_j \in \mathcal{G}^1_{test}$ and $\mathcal{G}^1_{calib}$. From the proof of Theorem 3.1 in Zheng et al. (2024), we have

$$
\begin{aligned}
&\lim_{|\mathcal{G}^1_{test}|,|\mathcal{G}_{train}|,K\to\infty} \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j\in\mathcal{G}^1_{test}} \mathbf{1}\big[s'_j < q_{\alpha/2}\big(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}}\big)\big] \\
&= \lim_{|\mathcal{G}_{train}|,K\to\infty} \mathbb{E}\mathbf{1}(s'_j < q_{\alpha/2}\big(\{s_i\}_{i\in\mathcal{G}^{j,1}_{calib}}\big) < \alpha/2
\end{aligned}
$$

Now plug in $\lambda = \pi(\mathcal{G}_s)$ and we let $|\mathcal{G}^1_{test}|, |\mathcal{G}_{train}|, K \to \infty$, then

$$
\begin{aligned}
&P\Big\{ \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s)\big] < \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[c'_{lo}(\mathcal{G}_j, 1) > \pi(\mathcal{G}_s)\big] \Big\} \\
&\le P\Big\{ \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s)\big] < \alpha/2 + \\
&\qquad \sqrt{\sum_{j \in \mathcal{I}^1_{tst}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s) + s'_j - q_{\alpha/2}(\{s_i\}_{i \in \mathcal{G}^{j,1}_{calib}})\big] \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s)\big]} \Big\} \\
&= \mathbb{E} \frac{1}{|\mathcal{G}^1_{test}|} \sum_{s \in \mathcal{G}^1_{test}} \mathbf{1}\Big\{ \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s)\big] < \alpha/2 + \\
&\qquad \sqrt{\sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s) + s'_j - q_{\alpha/2}(\{s_i\}_{i \in \mathcal{G}^{j,1}_{calib}})\big] \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[\pi(\mathcal{G}_j) > \pi(\mathcal{G}_s)\big]} \Big\} \\
&\le \frac{1}{|\mathcal{G}^1_{test}|} \mathbb{E} \sum_{s: q_s \le \alpha/2} 1 = \alpha/2 \qquad (5)
\end{aligned}
$$

Here q_s denotes the quantile of $\pi(\mathcal{G}_s)$ among $\{\pi(\mathcal{G}_i)\}_{i \in \mathcal{G}^1_{test}}$, and (5) comes from the fact that quantile functions are convex. Thus prove the first part of Theorem 3.2.

For the second part, we only need to prove one side of the inequality in the following:

$$
\begin{aligned}
&\lim_{|\mathcal{G}^1_{test}|, |\mathcal{G}_{train}|, K \to \infty} TPR(\lambda) - \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big[c'_{up}(\mathcal{G}_j, 1) > \lambda\big] \\
&= \lim_{|\mathcal{G}^1_{test}|, |\mathcal{G}_{train}|, K \to \infty} \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \Big(\mathbf{1}\big(\widehat{f}(\mathcal{G}_j) > \lambda\big) - \mathbf{1}\big(c'_{up}(\mathcal{G}_j, 1) > \lambda\big) \Big) \\
&\le \lim_{|\mathcal{G}^1_{test}|, |\mathcal{G}_{train}|, K \to \infty} \frac{1}{|\mathcal{G}^1_{test}|} \sum_{j \in \mathcal{G}^1_{test}} \mathbf{1}\big(\widehat{f}(\mathcal{G}_j) > c'_{up}(\mathcal{G}_j, 1)\big) = 0
\end{aligned}
$$

The last steps comes from the second part of Theorem 3.1 in Zheng et al. (2024). Similarly, we have the other side of the inequality, thus proof the second part in this theorem.

9 Tensor Low-Rank Structures

Consider an M-th order tensor $\mathcal{X}$ of dimension $D_1 \times \cdots \times D_M$. If $\mathcal{X}$ assumes a (canonical) rank-R *CP low-rank* structure, then it can be expressed as

$$
\mathcal{X} = \sum_{r=1}^{R} c_r \, \mathbf{u}_{1r} \circ \mathbf{u}_{2r} \circ \cdots \mathbf{u}_{Mr}, \qquad (6)
$$

where $\circ$ denotes the outer product, $\mathbf{u}_{mr} \in \mathbb{R}^{D_m}$ and $\|\mathbf{u}_{mr}\|_2 = 1$ for all mode $m \in [M]$ and latent dimension $r \in [R]$. Concatenating all R vectors corresponding to a mode m, we have $\mathbf{U}_m = [\mathbf{u}_{m1}, \cdots, \mathbf{u}_{mR}] \in \mathbb{R}^{D_m \times R}$ which is referred to as the loading matrix for mode $m \in [M]$.

If $\mathcal{X}$ assumes a rank-$(R_1, \cdots, R_M)$ *Tucker low-rank* structure, then it writes

$$
\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \cdots \times_M \mathbf{U}_M = \sum_{r_1=1}^{R_1} \cdots \sum_{r_M=1}^{R_M} c_{r_1 \cdots r_M} (\mathbf{u}_{1r_1} \circ \cdots \circ \mathbf{u}_{Mr_M}),
$$

where $\mathbf{u}_{mr_m}$ are all D_m-dimensional vectors, and $c_{r_1 \cdots r_M}$ are elements in the $R_1 \times \cdots \times R_D$-dimensional core tensor $\mathcal{C}$.

Tensor Train (TT) low-rank approximates a $D_1 \times \cdots \times D_M$ tensor $\mathcal{X}$ with a chain of products of third order *core tensors* $\mathcal{C}_i$, $i \in [M]$, of dimension $R_{i-1} \times D_i \times R_i$. Specifically, each element of tensor $\mathcal{X}$ can be written as

$$x_{i_1,\cdots,i_M} = \mathbf{c}_{1,1,i_1,:}^\top \times \mathbf{c}_{2,:,i_2,:} \times \cdots \times \mathbf{c}_{M,:,i_M,:} \times \mathbf{c}_{M+1,:,1,1}, \tag{7}$$

where $\mathbf{c}_{m,:,i_m,:}$ is an $R_{m-1} \times R_m$ matrix for $m \in [M] \cup \{M+1\}$. The product of those matrices is a matrix of size $R_0 \times R_{M+1}$. Letting $R_0 = 1$, the first core tensor $\mathcal{C}_1$ is of dimension $1 \times D_1 \times R_1$, which is actually a matrix and whose i_1-th slice of the middle dimension (i.e., $\mathbf{c}_{1,1,i_1,:}$) is actually a R_1 vector. To deal with the "boundary condition" at the end, we augmented the chain with an additional tensor $\mathcal{C}_{M+1}$ with $D_{M+1} = 1$ and $R_{M+1} = 1$ of dimension $R_M \times 1 \times 1$. So the last tensor can be treated as a vector of dimension R_M.

CP low-rank (6) is a special case where the core tensor $\mathcal{C}$ has the same dimensions over all modes, that is $R_m = R$ for all $m \in [M]$, and is super-diagonal. TT low-rank is a different kind of low-rank structure and it inherits advantages from both CP and Tucker decomposition. Specifically, TT decomposition can compress tensors as significantly as CP decomposition, while its calculation is as stable as Tucker decomposition.

10 Tensorized Graph Neural Networks

The Tensor Transformation Layer (TTL) preserves the tensor structures of feature $\boldsymbol{\mathcal{X}}$ of dimension $D = \prod_{m=1}^M D_m$ and hidden throughput. Let L be any positive integer and $\mathbf{d} = \left[d^{(1)}, \cdots, d^{(L+1)}\right]$ collects the width of all layers. A *deep ReLU Tensor Neural Network* is a function mapping taking the form of

$$f(\mathcal{X}) = \mathcal{L}^{(L+1)} \circ \sigma \circ \mathcal{L}^{(L)} \circ \sigma \cdots \circ \mathcal{L}^{(2)} \circ \sigma \circ \mathcal{L}^{(1)}(\mathcal{X}), \tag{8}$$

where $\sigma(\cdot)$ is an element-wise activation function. Affine transformation $\mathcal{L}^{(\ell)}(\cdot)$ and hidden input and output tensor of the ℓ-th layer, i.e., $\mathcal{H}^{(\ell+1)}$ and $\mathcal{H}^{(\ell)}$ are defined by

$$\begin{aligned} \mathcal{L}^{(\ell)}(\mathcal{H}^{(\ell)}) &\triangleq \langle \mathcal{W}^{(\ell)}, \mathcal{H}^{(\ell)} \rangle + \mathcal{B}^{(\ell)}, \\ \text{and} \quad \mathcal{H}^{(\ell+1)} &\triangleq \sigma(\mathcal{L}^{(\ell)}(\mathcal{H}^{(\ell)})) \end{aligned} \tag{9}$$

where $\mathcal{H}^{(0)} = \boldsymbol{\mathcal{X}}$ takes the tensor feature, $\langle \cdot, \cdot \rangle$ is the tensor inner product, and *low-rank weight* tensor $\boldsymbol{\mathcal{W}}^{(\ell)}$ and a bias tensor $\mathcal{B}^{(\ell)}$. The tensor structure kicks in when we incorporate tensor low-rank structures such as *CP low-rank*, *Tucker low-rank*, and *Tensor Train low-rank*.

Tucker low-rank structure is defined by

$$\mathcal{X} = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \cdots \times_M \mathbf{U}_M + \mathcal{E}, \tag{10}$$

where $\mathcal{E} \in \mathbb{R}^{D_1 \times \cdots \times D_M}$ is the tensor of the idiosyncratic component (or noise) and $\boldsymbol{\mathcal{C}}$ is the latent core tensor representing the true low-rank feature tensors and $\mathbf{U}_m$, $m \in [M]$, are the loading matrices.

The complete definitions of three low-rank structures are given in Appendix 9. CP low-rank (6) is a special case where the core tensor $\mathcal{C}$ has the same dimensions over all modes, that is $R_m = R$ for all $m \in [M]$, and is super-diagonal. TT low-rank is a different kind of low-rank structure, which inherits advantages from both CP and Tucker decomposition. Specifically, TT decomposition can compress tensors as significantly as CP decomposition, while its calculation is as stable as Tucker decomposition.

11 TGNN Algorithm

The Algorithm 2 describes the pseudo-code for the training stage of TGNN.

12 Statistics of Dataset

Table 2 provides some statistics of the benchmark datasets.

Algorithm 2 Pseudo-code of Pre-training TGNN

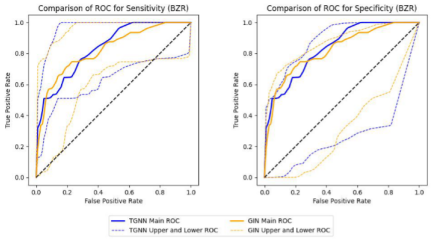
```
train, test_calib_pool = split(dataset)
Set learning rate α, number of epochs N, batch size B, number of CP epochs M
Initialize TGNN model f with random parameters θ
for epoch = 1 to N do
    for batch in LOADER(train, B) do
        loss = BCE_LOSS(f(batch.inputs), batch.labels)
        θ = θ − α × F.BACKWARD(loss)
    end for
end for
```

Table 2: Statistics of the benchmark datasets.

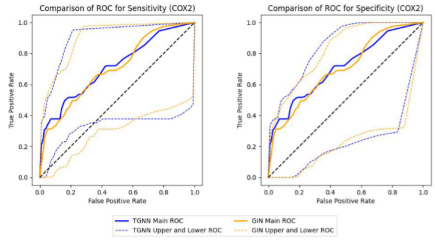
Dataset	**# Graphs**	**Avg.** $\lvert\mathcal{V}\rvert$	**Avg.** $\lvert\mathcal{E}\rvert$	**# Class**
BZR	405	35.75	38.36	2
COX2	467	41.22	43.45	2
DHFR	756	42.43	44.54	2
PTC_MR	344	14.29	14.69	2
PTC_MM	336	13.97	14.32	2
D&D	1178	284.32	715.66	2
PROTEINS	1113	39.06	72.82	2

13 Supplement ROC comparisons plots

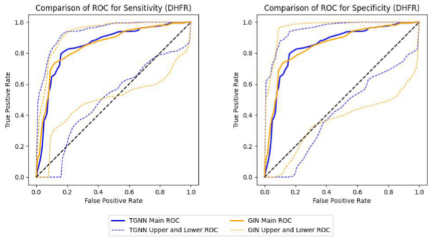
Figures 4 and 5 provide further ROC comparisons across multiple datasets. In most cases, TGNN outperforms GIN, showing higher true positive rates and tighter confidence bands, demonstrating its effectiveness in handling complex graph-based data. However, in some datasets, TGNN exhibits lower performance compared to GIN, which is reflected by larger confidence bands for TGNN. This variation highlights the effectiveness of our UQ methods, accurately capturing model uncertainty and performance fluctuations across different datasets. These figures, together with the quantitative results, emphasize the robustness and versatility of our approach.



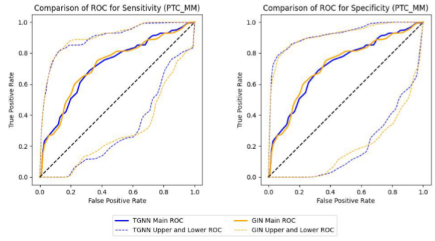
(a) BZR



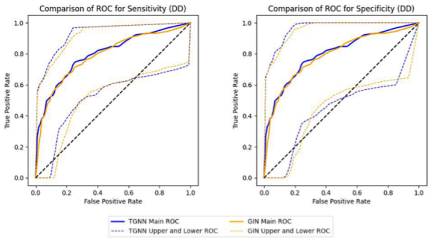
(b) COX2



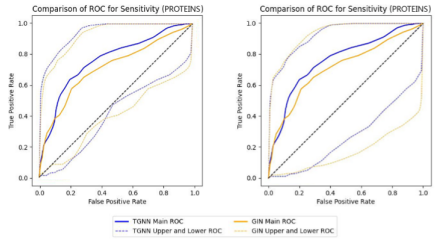
(c) DHFR



(d) PTC_MM

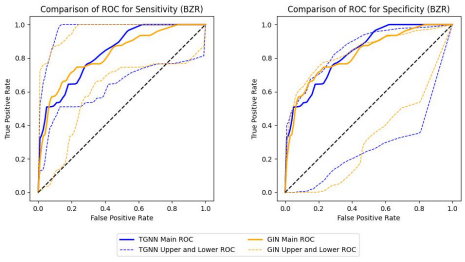


(e) D&D

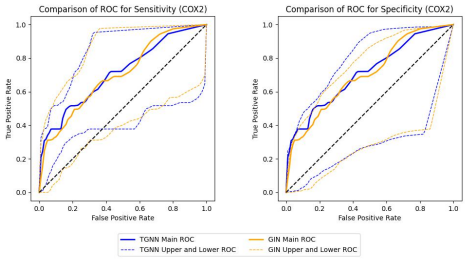


(f) PROTEINS

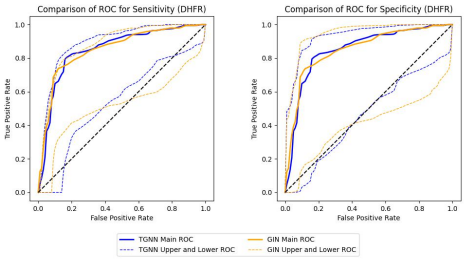
Figure 4: Exchangeable ROC bands of TGNN and GIN comparison.



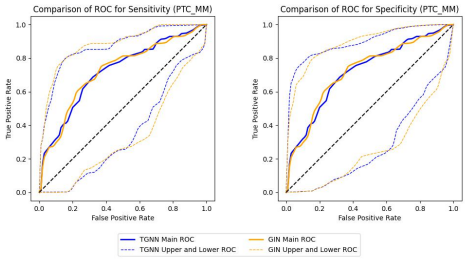
(a) BZR



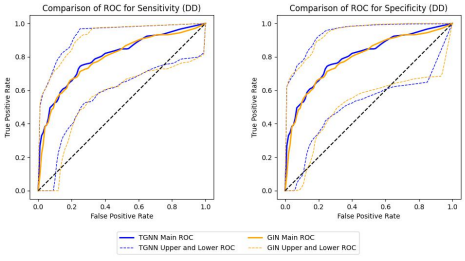
(b) COX2



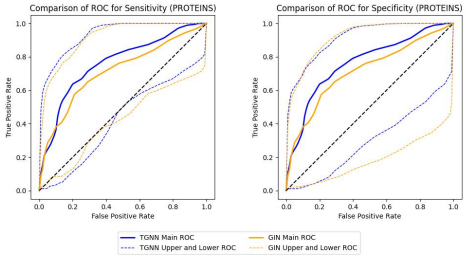
(c) DHFR



(d) PTC_MM



(e) D&D



(f) PROTEINS

Figure 5: Non-Exchangeable ROC band for TGNN and GIN comparison.