Node Classification With Reject Option

Anonymous authors
Paper under double-blind review

Abstract

One of the key tasks in graph learning is node classification. While Graph neural networks have been used for various applications, their adaptivity to reject option settings has not been previously explored. In this paper, we propose NCwR, a novel approach to node classification in Graph Neural Networks (GNNs) with an integrated reject option. This allows the model to abstain from making predictions when uncertainty is high. We propose cost-based and coverage-based methods for classification with abstention in node classification settings using GNNs. We perform experiments using our method on three standard citation network datasets Cora, Citeseer and Pubmed and compare with relevant baselines. We also model the Legal judgment prediction problem on the ILDC dataset as a node classification problem, where nodes represent legal cases and edges represent citations. We further interpret the model by analyzing the cases in which it abstains from predicting and visualizing which part of the input features influenced this decision.

1 Introduction

The recent decade has witnessed a surge of interest in using GNNs in various domains such as computer vision Satorras & Estrach (2018), natural language processing Schlichtkrull et al. (2017), and bioinformatics Xia & Ku (2021), to name a few. GNNs Kipf & Welling (2017) capture structural aspects of the data in the form of nodes and edges to perform any prediction task. It learns node, edge, and graph-level embeddings to get high-dimensional features using the message-passing mechanism in the GNN layer.

GNNs are being utilized in deploying systems to real-world tasks Derrow-Pinion et al. (2021). GNNs are also used in high-risk applications such as legal judgment prediction Dong & Niu (2021), disease prediction Sun et al. (2021), financial fraud prediction Xu et al. (2021), etc., with a high cost of incorrect predictions. In such high-risk situations, standard GNN models are not very cost-effective while handling decisions on difficult examples.

In literature, uncertainty estimation approaches are used to measure the prediction uncertainty involved in high-risk applications. Conformal prediction methods are very popular among them Gawlikowski et al. (2023); Wang et al. (2024); Angelopoulos & Bates (2021). In these approaches, a confidence parameter is specified and at the time of prediction, these algorithms predict a label set instead of a single label. The idea is that the label set contains the actual label with very high confidence which is pre-specified. Such approaches have been fine-tuned for GNNs also. Huang et al. (2023).

In contrast to conformal prediction methods, another possible approach used in high-risk situations is to avoid making any decisions on difficult and confusing examples. Consider the case of the diagnosis of a patient for a specific disease. In case of confusion, the physician might choose not to risk misdiagnosing the patient. She might instead recommend further medical tests to the patient or refer her to an appropriate specialist. The primary response in these cases is to "reject" the example. Such flexibility of the classifier to avoid taking any decision is called the reject option. Reject option classifiers have been extensively used in various high-risk applications such as healthcare Hanczar & Dougherty (2008); da Rocha Neto et al. (2011), finance Rosowsky & Smith (2013) etc. Rejection option classifiers can be categorized into two broad classes (a) cost-based and (b) coverage-based. In cost-based approaches Kalra et al. (2021); Charoenphakdee et al. (2021); Ramaswamy et al. (2018); Cao et al. (2022), the goal of the algorithm is to find an optimal classifier

by minimizing a loss which also incorporates the cost of rejection along with the cost of misclassification. The cost of rejection is much smaller compared to misclassification. Overall, these approaches aim to minimize the number of rejected examples as well as minimise the misclassifications of unrejected samples. On the other hand, in the coverage-based method Geifman & El-Yaniv (2019; 2017), a coverage parameter is pre-specified and the algorithm tries to maintain the fraction of unrejected samples the same as coverage. Both algorithms try to reject difficult examples.

In this paper, we propose integrating a reject option in GNNs for the node classification problem. We propose two variants corresponding to cost-based and coverage-based approaches. We also present how this method can be used in real-world scenarios by working on prediction tasks of high-risk domains such as Healthcare and Law.

Our contributions are as follows: *i*) We extend and generalize GNNs to train for node features with cost-based and coverage-based abstention models. *ii*) We perform an empirical study to evaluate our models on popular benchmark datasets for node classification tasks and compare them with baseline methods. *iii*) We show extensive results of our method on the Indian Legal Documents Corpus (ILDC) dataset for the LJP task. *iv*) To understand why our model chooses to reject certain cases, we further examine these cases with the help of Shapley Additive Explanations (SHAP) Lundberg & Lee (2017).

2 Related Work

2.1 Node Classification

Node classification is a fundamental task related to machine learning for graphs and network analysis. GNN methods can be broadly classified into three categories that perform node classification as the primary task. The first set of models introduced convolution-based GNN architectures by extending original CNNs to graphs Scarselli et al. (2008), Defferrard et al. (2016), Hamilton et al. (2017), Kipf & Welling (2017) Bresson & Laurent (2017). Secondly, proposed attention and gating mechanism-based architectures using anisotropic operations on graphs Veličković et al. (2018). The third category focuses on the theoretical limitations of previous types Xu et al. (2018), Morris et al. (2019), Maron et al. (2019), Chen et al. (2019).

2.2 Reject Option Classification

There are two broad categories of approaches for reject option classifiers: coverage-based and cost-based. Coverage is defined as the ratio of samples that the model does not reject. For a given coverage, the model finds the best examples that can give the best performance. SelectiveNet is a coverage-based method proposed for learning with abstention El-Yaniv et al. (2010); Geifman & El-Yaniv (2019). SelectiveNet is a deep neural network architecture that optimizes prediction and selection functions to model a selective predictor. As this approach does not consider rejection cost d in their objective function, it can avoid rejecting hazardous examples.

Cost-based approaches assume that the reject option involves a cost of d. In Kalra et al. (2021), the authors propose a deep neural network-based reject option classifier for two classes that learn instance-dependent rejection functions. In Ramaswamy et al. (2018), multiclass extensions of the hinge-loss with a confidence threshold are considered for reject option classification. Ni et al. (2019) prove calibration results for various confidence-based smooth losses for multiclass reject option classification. Charoenphakdee et al. (2021) prove that K-class reject option classification can be broken down into K binary cost-sensitive classification problems. They subsequently propose a family of surrogates, the ensembles of arbitrary binary classification losses. Cao et al. (2022) propose a general recipe to convert any multiclass loss function to accommodate the reject option, calibrated to loss l_{0d1}. They treat rejection as another class.

2.3 Uncertainty Estimation Using Conformal Prediction

Uncertainty in Deep Neural Networks Gawlikowski et al. (2023) studies the sources of uncertainty like data uncertainty and model uncertainty, and the estimation of uncertainty measures to further be used in high-risk applications. This is further explored in the Graph setting by Wang et al. (2024) exploring both sources

of uncertainty and estimating uncertainty. In the GNN setting uncertainty measures are further utilized for downstream tasks like OOD detection, outlier identification and Trustworthy GNNs. Conformal Prediction Angelopoulos & Bates (2021) is a setting which uses these uncertainty measures to predict an interval instead of one class which will have the true label with a fixed probability with statistical guarantees. This setting uses Distribution-Free Uncertainty Quantification which makes it accessible to a wide range of applications. Huang et al. (2023) proposed CF-GNN extending conformal prediction to GNNs and node classification.

3 Method

GNN models like GAT Veličković et al. (2018) focus on learning effective and efficient representations of nodes to perform any downstream task. Let $\mathcal{X}$ be the instance space and $\mathcal{Y} \in \{1, \ldots, K\}$ be the label space. We represent the embedding space learned using GNN by $\mathcal{H}$. GNN treats each instance as a node and learns embedding for each node. We used GAT as the base GNN Architecture but our method is model agnostic and can be replaced with any GNN architecture to learn the node embeddings.

3.1 NCwR-Cov: Coverage Based Node Classifier With Rejection

NCwR-Cov uses coverage-based logic to learn node classifiers with a reject option. We use similar ideas to SelectiveNet Geifman & El-Yaniv (2019) to learn the coverage-based rejection function. Figure 1 shows the architecture of NCwR-Cov. Node representations are learned using the first GNN layer and given as input to the second GNN layer which follows the softmax layer. The second GNN layer and softmax layer combined learn mapping $\mathbf{f} : \mathcal{H} \rightarrow \Delta_{K-1}$ where Δ_{K-1} is K-dimensional simplex. Function $\mathbf{f}$ is used to predict the class of a node. There are two more fully connected layers after the softmax layer (having 512 nodes and one node) to model the selection function $g : \mathcal{H} \rightarrow \{0, 1\}$. Selection function g decides whether to predict a given example or not. Selection function $g(\mathbf{h})$ is a single neuron with a sigmoid activation. At the beginning, a threshold of 0.5 is set for the selection function, which means $\mathbf{f}(\mathbf{h})$ predicts $\mathbf{h}$ if and only if $g(\mathbf{h}) \geq 0.5$. The auxiliary prediction head implements the prediction task $a(\mathbf{h})$ without the need for coverage to get a better representation of examples with low confidence scores, which are usually ignored by the prediction head. This head is only used for training purposes. We use cross-entropy loss l_{ce} to capture the error made by the prediction function $\mathbf{f}(\mathbf{h})$. The empirical risk of the model is captured as follows.

$$r(\mathbf{f}, g|S_n) = \frac{\frac{1}{n}\sum_{i=1}^{n} l(f(\mathbf{h}_i), y_i) g(\mathbf{h}_i)}{\phi(g|S_n)}$$

where $\phi(g|S_n)$ is empirical coverage computed as $\phi(g|S_n) = \frac{1}{n}\sum_{i=1}^{n} g(\mathbf{h}_i)$. An optimal selective model could be trained by optimizing the selective risk given constant coverage. We use the following error function to optimize $\mathbf{f}(.)$ and $g(.)$.

$$E(\mathbf{f}, g) = r(\mathbf{f}, g|S_n) + \lambda\Psi(c - \phi(g|S_n))$$

where $\Psi(a) = \max(0, a)^2$ is a quadratic penalty function, c is the target coverage, and λ controls the importance of coverage constraint. The loss function used at the auxiliary head is standard cross entropy loss l_{ce} without any coverage constraint. Thus, the empirical risk function corresponding to the auxiliary head is $E(\mathbf{f}) = 1/n\sum_{i=1}^{n} l_{ce}(\mathbf{f}(\mathbf{h}_i), y_i)$. The final error function of NCwR-Cov is a convex combination of $E(f, g)$ and $E(\mathbf{f})$ as follows, $E = \alpha E(\mathbf{f}, g) + (1 - \alpha)E(\mathbf{f})$, where $\alpha \in (0, 1)$. When the data is trained over a training set using a coverage constraint, this constraint is violated on the test set. The constraint requires the true coverage $\phi(g)$ to be larger than the given coverage constraint c, which is usually violated. To get the optimal actual coverage, we calibrate the threshold τ to select the example in $g(h')$ using this validation set, which results in coverage as close as possible to target coverage.

3.2 NCwR-Cost: Cost Based Node Classifier With Rejection

In the cost-based method, the cost of rejection d is pre-specified. The goal here is to learn an optimal node classifier with rejection for a given d value. The architecture of NCwR-Cost is presented in Figure 2. The first block in NCwR-Cost consists of two GNN layers. The output of the second GNN layer is fed to a softmax

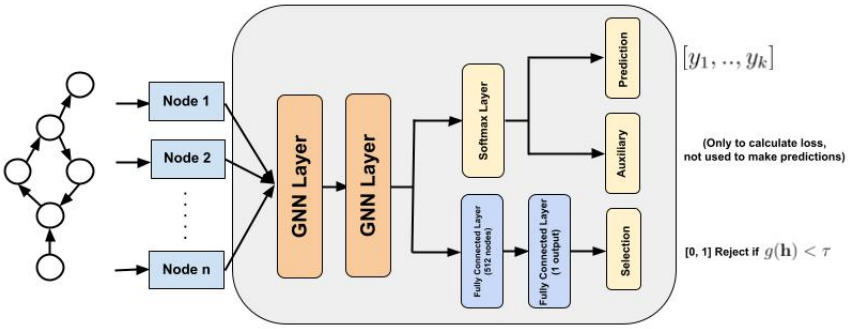


Figure 1: Architecture of NodeCwR-Cov: Coverage based node classifier with rejection.

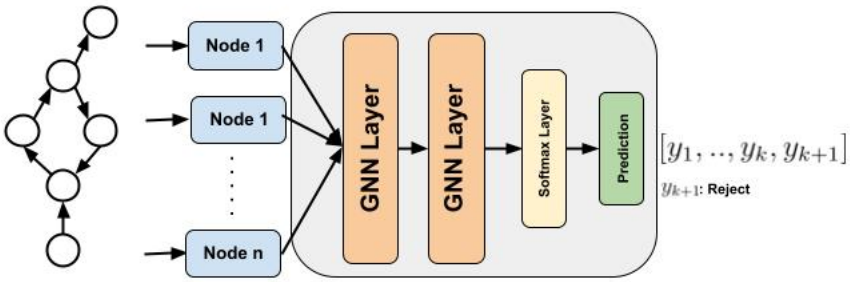


Figure 2: Architecture of NodeCwR-Cost: Cost based node classifier with rejection.

layer with $(K+1)$ nodes. Note that we assume rejection as the $(K+1)^{th}$ class here. The second GNN layer and softmax layer combined learn prediction function $f : \mathcal{H} \rightarrow \Delta_K$ where Δ_K is $(K+1)$-dimensional simplex. Note that $(K+1)^{th}$ output corresponds to the reject option in this architecture. Let $\mathbf{e}_j$ denote $K+1$-dimensional vector such that its jth element is one and other elements are zero. Note that $(K+1)^{th}$ element never becomes one as we do not get a rejection label in the training data. We use the following variant of cross-entropy loss, which also incorporates the cost of rejection Cao et al. (2022).

$$l_{ce}^{d}(f(\mathbf{h}), \mathbf{e}_y) = l_{ce}(f(\mathbf{h}), \mathbf{e}_y) + (1-d) l_{ce}(f(\mathbf{h}), \mathbf{e}_{K+1}) = -\log f_y(\mathbf{h}) - (1-d) \log f_{K+1}(\mathbf{h}) \tag{1}$$

Here $f_{K+1}(\mathbf{h})$ is the output corresponding to the reject option, and $f_y(\mathbf{h})$ is the output related to the actual class. For very small values of d, the model focuses more on maximizing $f_{K+1}(\mathbf{h})$ to prefer rejection over misclassification. Note that loss l_{ce}^{d} is shown to be consistent with the l_{0d1} loss Cao et al. (2022). For $d = 1$, the loss l_{ce}^{d} becomes the same as standard cross entropy loss l_{ce}.

4 Experimental Setup

In this section, we provide details of the experimental setup, which primarily talks about the datasets used, baselines used, implementation details of the proposed approaches NCwR-Cost and NCwR-Cov and implementation details of baselines used.

4.1 Datatsets Used

We evaluate our model on three standard citation network datasets, Cora, Citeseer, and Pubmed Sen et al. (2008). In these datasets, each document is represented by a node, and the class label represents the category of the document. Undirected edges represent citations. We follow standard practice Kipf & Welling (2017); Veličković et al. (2018) for training node classifier. We use 20 nodes per class for training. Thus, the number of nodes for training varies for each dataset depending on the number of classes. We use 500 nodes for validation and 1000 for testing on all the datasets.

4.2 Base Graph Neural Network Architecture Used

We can use any GNN as the base architecture for both methods. In our experiment section, we use GAT as the base architecture due to its effectiveness and popularity. We also note that in our experiments changing the base GNN Architectures and their parameters like the number of layers did not affect the results by much. We present these results in Appendix A.

For the GAT architecture, we closely follow the experimental setup mentioned in Veličković et al. (2018). We modify the open-source GAT implementation by Antognini (2021) for our approach. We first apply dropout Srivastava et al. (2014) on node features with $p = 0.6$. These node features, along with the adjacency matrix, are passed through a GAT Layer having 8 attention heads, where each head produces eight features per node. We use LeakyReLU as the activation function inside the GAT Layer with $\alpha = 0.2$. These outputs are concatenated for the first layer (64 features per node). Another dropout layer with the same probability follows this. The dropout layer's output is passed through the final GAT layer with a single attention head, which takes 64 features per node and outputs k features per node, where k is the number of classes. It is passed to ELU Clevert et al. (2015) activation function. The network output is passed through a softmax layer.

4.3 Details of NCwR-Cov Implementation

We use the GAT architecture to integrate the coverage-based reject option into the model as mentioned in Geifman & El-Yaniv (2019). The output from the softmax layer is separately given to both prediction head f and auxiliary head h. The output from the second GAT Layer is passed through a Fully Connected Hidden Layer with 512 nodes. This is passed through Batch Normalization Ioffe & Szegedy (2015), ReLU, and a Fully Connected Output Layer with one node. This is passed through Sigmoid Activation to get a selection score $[0, 1]$. The Prediction head and Selection head are concatenated together, and the selective loss is performed on this output. We set $\lambda = 32$ as the constraint on coverage to calculate this loss. Cross Entropy Loss is performed on the output of the Auxiliary head but is not used for making predictions. A convex combination of these two loss values with $\alpha^l = 0.5$ is used to optimize the model. We trained the model with early stopping with a patience of 100 epochs. Error on the validation set is used to implement the stopping condition. We observed that the model trains for approximately 1800 epochs to train the NCwR-Cov model. Once the model is trained, the coverage on the test data set when $\tau = 0.5$ will vary. However, since we have the selection scores of each node in the validation set, we sort them and select a τ value that matches the expected coverage.

4.4 Details of NCwR-Cost Implementation

We trained NCwR-Cost by taking the output of the GAT network. In the cost-based approach, we treat the reject option as an integrated class in the model. Hence, for a k class classification problem, we change the model architecture from giving k outputs to $k+1$ outputs. In this model, we have to perform CwR Loss (see eq.(1)) for optimization. From the output, every node that gets $k+1$ as the output will be rejected. We use early stopping with a patience of 100 epochs. Model trains for approximately 1000 epochs for NCwR-Cost.

4.5 Baselines Used

To the best of our knowledge, we are the first to use reject option classifiers for node classification on high-risk applications. This makes it tough to compare with existing baselines to show the importance of our contribution. However, we introduce some changes in existing methods to model them as Reject Option Classifiers and compare them with the following two baselines.

- Softmax Response (SR): We treat the Softmax scores of the Vanilla GNN as an uncertainty measure and reject examples when the predicted class score is lower than some threshold. As we change the thresholds, the rejection rate changes. We used the following threshold values to get different reject option classifiers using Softmax-Response: $[0.5, 0.6, 0.7, 0.8, 0.9]$.

- CF-GNN Huang et al. (2023): Conformal classifier with coverage parameter α predicts a label set such that the probability of actual label being present in the predicted label set is greater than or equal to $1 - \alpha$. We model the state-of-the-art Conformal Prediction method for GNNs as Reject Option Classifiers as follows. We simply reject those examples for which the CF-GNN predicts more than one class label. Rejecting such examples makes sense because of the label ambiguity and it aligns with the basic principles used for rejection. Note that as we increase the coverage parameter α in the CF-GNN, the predicted label set size will reduce. For a higher value of α, the CF-GNN predicts smaller label sets. Thus, the rejection rate should decrease as we increase α. Different α values used are: $[0.1, 0.125, 0.15, 0.175, 0.2]$.

For both baselines, we used the same GAT architecture that we used for NCwR-Cost and NCwR-Cov.

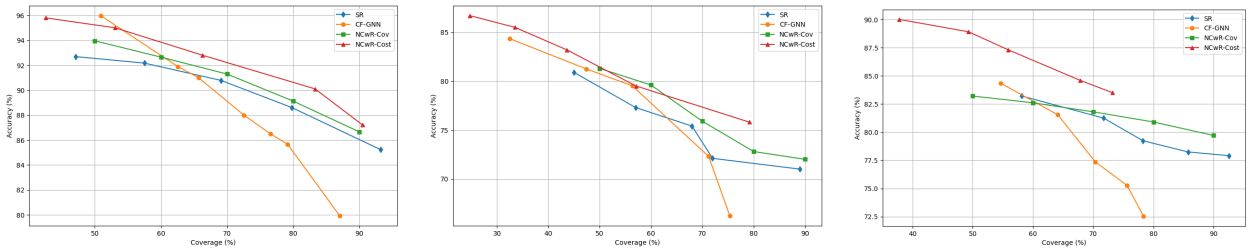
5 Experimental Results

Here, we present experimental results for proposed approaches NCwR-Cost and NCwR-Cov. We also compare them with different baselines to highlight the importance of the proposed approaches. We repeat each experiment 10 times with random initialization and report the average accuracies.

5.1 Comparison Results with Baselines

Figure 3 shows comparison results with baselines. We observe that NCwR-Cost always outperforms the Softmax-Response based approach in terms of accuracy on unrejected samples for all coverage values. NCwR-Cost also outperforms the CF-GNN based approach for all coverage values and for all datasets except for one case in the Cora dataset. For the Cora dataset, for coverage of 50%, CF-GNN has marginally better accuracy. Thus, NCwR-Cost is a superior model compared to the baseline models.

We see that NCwR-Cov also outperforms the Softmax-Response based approach for all datasets and coverage values except for one case with Pubmed dataset (coverage value 60%). For coverage value 60% with the Pubmed dataset, Softmax-Response has slightly better accuracy than NCwR-Cov. Compared to CF-GNN, NCwR-Cov always performs better for coverage values greater than 60%. For coverage values smaller than 60%, CF-GNN performs marginally better than NCwR-Cov.



(a) Cora (b) Citeseer (c) Pubmed

Figure 3: Comparison of NCwR-Cost and NCwR-Cov with baselines.

5.2 Comparison between NCwR-Cov and NCwR-Cost

In Table 1, we report the performance of NCwR-Cov models trained for coverage rates ranging in $[0.1, \ldots, 0.9]$. Although we can calibrate the threshold to cover any number of examples irrespective of the training coverage, it is preferred to train the model on the same coverage rates and then calibrate τ to the same to get the best results. We observe that for all the datasets, as we increase the coverage, the accuracy on unrejected samples decreases. This pattern is expected as we reject lesser examples, there will be more difficult examples to classify.

In Table 2, we report the performance of NCwR-Cost models trained for rejection cost (d) taking values in $[0.5, 0.6, 0.7, 0.8, 0.85]$. As the cost of rejection d increases, the rejection rate decreases. Decreasing the

Coverage	Cora	Citeseer	Pubmed
0.5	93.96 ± 1.45	81.3 ± 2.19	83.2 ± 3.34
0.6	92.65 ± 0.5	79.6 ± 2.43	82.6 ± 1.48
0.7	91.29 ± 0.45	75.9 ± 2.86	79.8 ± 2.46
0.8	89.12 ± 0.8	72.8 ± 1.05	80.9 ± 1.24
0.9	86.65 ± 0.7	72 ± 0.69	79.7 ± 0.59
1	81.65	70.12	76.7

Table 1: Accuracy of NCwR-Cov for various coverage rates on Cora, Citeseer and Pubmed.

	Cora			Citeseer			Pubmed		
d	Acc	Cov	0-d-1	Acc	Cov	0-d-1	Acc	Cov	0-d-1
0.5	95.8 ± 0.05	42.6 ± 0.02	0.305 ± 0.11	91.6 ± 0.12	9.7 ± 0.05	0.460 ± 0.04	88.9 ± 0.02	49.3 ± 0.08	0.309 ± 0.16
0.6	95.0 ± 0.04	53.1 ± 0.03	0.308 ± 0.06	87.9 ± 0.09	17.6 ± 0.09	0.516 ± 0.06	84.6 ± 0.05	67.8 ± 0.05	0.298 ± 0.08
0.7	92.8 ± 0.04	66.3 ± 0.08	0.283 ± 0.07	85.5 ± 0.01	33.5 ± 0.02	0.514 ± 0.04	-	-	-
0.8	90.1 ± 0.08	83.3 ± 0.04	0.216 ± 0.08	79.5 ± 0.11	57.1 ± 0.09	0.460 ± 0.02	-	-	-
0.85	87.2 ± 0.06	90.5 ± 0.07	0.196 ± 0.06	75.8 ± 1.61	79.2 ± 0.04	0.368 ± 0.15	-	-	-
1	81.65	100	0.178	70.12	100	0.299	76.7	100	0.233

Table 2: Accuracy of NCwR-Cost for various cost of rejection values on Cora, Citeseer and Pubmed. We only experimented with $d = [0.5, 0.6]$ in Pubmed as it has only $k = 3$ classes and $d < \frac{k-1}{k}$.

rejection rate will increase coverage. As the coverage increases, the model will misclassify more samples, which decreases the performance of unrejected samples.

Comparison: Figure 3 shows the coverage versus accuracy plots for both cost-based and coverage-based approaches on different datasets. The cost-based approach shows a clear advantage in terms of accuracy for most coverage rates. The reason is as follows. The coverage constraint in NCwR-Cov does not ensure the rejection of those examples that are hard to classify correctly. Thus, it may reject some of the easy examples. Thus, every coverage value may include more hard examples. Label smoothing makes this situation worse for NCwR-Cov due to soft labels. We also observe a very high standard deviation in the performance of NCwR-Cov. On the other hand, NCwR-Cost prefers to reject hard examples first by assigning a cost to rejection. This makes NCwR-Cost perform better than NCwR-Cov.

5.3 Node Embedding Visualization

We plot t-SNE plots to represent the predicted class of each node. It is noticeable in Figure 4 that in both NCwR-Cost and NCwR-Cov models, the rejected examples (represented using black colour) are usually the nodes that highly overlap between two or more classes. We can also notice that as the model coverage decreases, the number of examples it rejects increases and covers more overlapping boundaries between classes. It is worth noting that although the coverage and accuracy are almost comparable in both models, the examples that each model chooses to reject are from different overlapping classes. NCwR-Cost tries to reject those examples which are in the overlapping regions of different classes. On the other hand, NCwR-Cov sometimes rejects examples which are predominantly from particular classes. For example, for coverage of 50.4%, NCwR-Cov rejects more examples of classes 3 and 4.

6 Application: Legal Judgement Prediction

Legal Judgment Prediction Feng et al. (2022); Cui et al. (2023) is an active area of research in the field of Machine Learning and Natural Language Processing. Automating the Judgment Prediction Process can be of huge value for various reasons. While these models perform extremely well already due to the recent surge

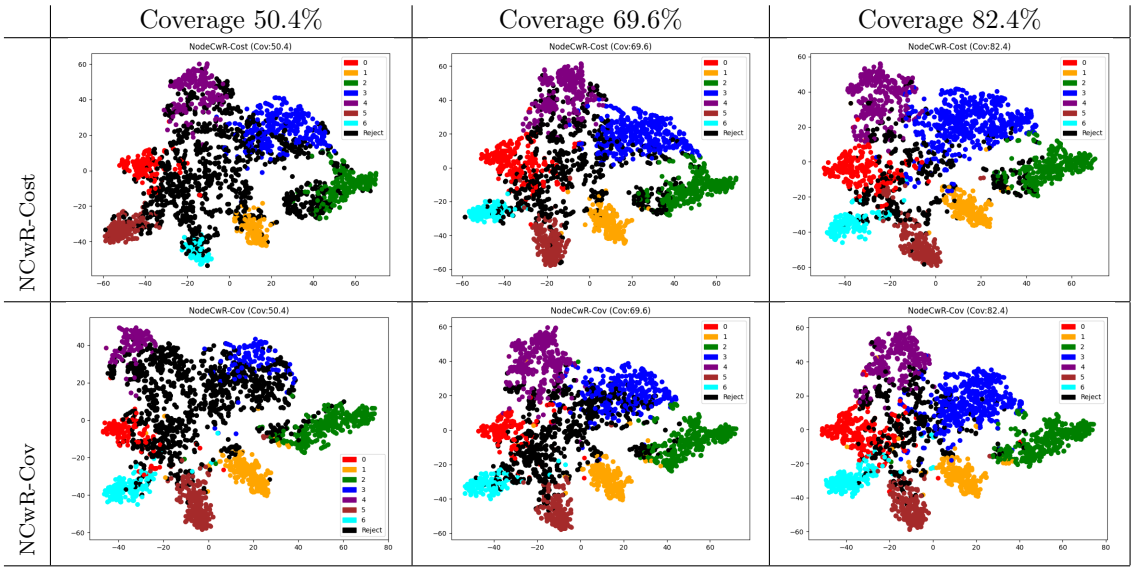


Figure 4: t-SNE plots representing predictions on Cora dataset (black - reject option).

in NLP progress, one issue that still remains is the reliability of these models to push to real world scenarios. Unlike many current fields that NLP models are automating tasks, Legal Judgment Prediction is a very high risk application in which the cost of misclassification is very high. In such high risk applications, performing reliably well for a small set of examples is much more valuable than giving a prediction for every sample.

Indian Legal Documents Corpus (ILDC) Malik et al. (2021) Indian Legal Documents Corpus is a dataset of collection of case proceedings in English from the Supreme Court of India (SCI), covering the period from 1947 to April 2020 presented by Malik et al. (2021). The raw dataset poses significant pre-processing challenges due to unstructured document formats, spelling errors, and the need to remove meta-information and direct decision statements from the texts. The ILDC dataset is divided into two subsets: ILDC$_{single}$ - single petition cases, and ILDC$_{multi}$ - cases with multiple petitions leading to different decisions. We worked with the ILDC$_{single}$ subset for our experimental setting. This subset contains a total of 7,593 cases, split into train/test/development sets (5,082/1,517/994).

This dataset was expanded by adding another $24,907$ cases without a final verdict and the citations within these cases with existing cases using ikanoon API by Khatri et al. (2023). This paper presented Legal Judgment Prediction as a semi-supervised node classification task where each node represents the text of the case proceedings and link represents a citation between cases. The additional cases without a label essentially enable the model with message passing through citation networks and are not part of the training set of nodes.

6.1 Experimental Setup

We follow the experimental setup presented in Khatri et al. (2023) and use a pretrained XLNet model from Malik et al. (2021) to extract language embeddings from all the new cases in the dataset. We formulate a graph where each node represents a case (Cases part of the ILDC have a label and additional cases extracted by Khatri et al. (2023) do not have a label) and the citations between the cases as links. Khatri et al. (2023) presents that using directed or undirected edges does not affect the model performance by a lot, hence we formulate the citations as undirected edges. On this graph, we train the GAT model for node classification and replicate the result presented in Khatri et al. (2023). On top of this, we perform experiments on this graph using our architectures. In Table 3 we report the performance for different cost of rejection values and coverage rates on ILDC dataset.

	NCwR-Cost		NCwR-Cov	
d	Acc (%)	Cov (%)	Acc (%)	Cov
0.25	87.24 ± 2.45	67.00 ± 3.30	97.55 ± 0.62	0.5
0.35	82.32 ± 2.72	86.34 ± 4.61	94.94 ± 1.20	0.6
0.40	79.95 ± 1.52	93.44 ± 7.17	90.58 ± 1.24	0.7
0.45	80.38 ± 1.74	97.83 ± 0.76	86.01 ± 1.09	0.8
0.50	79.94 ± 2.09	98.99 ± 0.20	81.87 ± 1.03	0.9

Table 3: Accuracy of NCwR-Cost and NCwR-Cov for various cost of rejection values and coverage rates respectively on ILDC dataset.

6.2 Explainability

Explainability methods in machine learning aim to make model predictions understandable to humans, especially in high-stakes domains like law. SHAP (Shapley Additive Explanations) Lundberg & Lee (2017) is one of the most reliable and widely used approaches for model interpretability, based on cooperative game theory and Shapley values.In this work, SHAP is applied to explain predictions for legal judgment tasks, where understanding which parts of the legal text drive a model's decision is very useful for transparency. SHAP provides visual explanations by highlighting portions of the legal text. Specifically, red highlights indicate text that pushes the model toward a positive outcome (e.g., the model is more confident in it's classification), while blue highlights signify text that supports a negative outcome (e.g., text that contradicts model's classification). The intensity of these colors represents the magnitude of the contribution—the darker the shade, the stronger the influence of the text segment on the final prediction.

In our experiments, we use the last 512 tokens of a petition in Supreme Court of India (SCI) Proceedings between the appellant and respondent, where the 'label' contains either '0' or '1'. A label of '0' represents petitions that have been rejected, while a label of '1' represents petitions that have been accepted.We have demonstrated two examples below: one where our model is very confident in its prediction and predicts correctly in Figure 6, and another where the model's confidence is lower, leading to an incorrect prediction in Figure 5.

6.2.1 Case 1: Explanation of the SCI Proceedings Where Model is Highly Confident

This case involves an appeal by A2, who was convicted under Section 380 of the Indian Penal Code (IPC) by the Trial Court for the theft of a gold chain. Two co-accused, A1 and A3, were acquitted of all charges. The appellant (A2) challenged the conviction in the Sessions Court and High Court, both of which upheld the conviction. The appellant then approached the Supreme Court, contesting the credibility of the prosecution's evidence, including the recovery of the stolen gold chain and the delay in filing the First Information Report (FIR).

The Supreme Court examined the evidence and found that the prosecution's case lacked credibility, particularly due to the significant delay in lodging the FIR (16 days) and the questionable recovery of the chain. As a result, the Supreme Court acquitted the appellant.

For this example, our model is very confident in its prediction and predicts the label '1' correctly. Explanation from SHAP is given in Figure 6.

Sentences Leading to the Decision in Red

- **Sentence 1** *"It is inconceivable that she would not realize that she had ..."*
  Highlights the Court's skepticism regarding the plausibility of the theft.

- **Sentence 2** *"As we have already noted that FIR was registered after about ..."*
  The Supreme Court found the delay in filing the FIR highly suspicious.

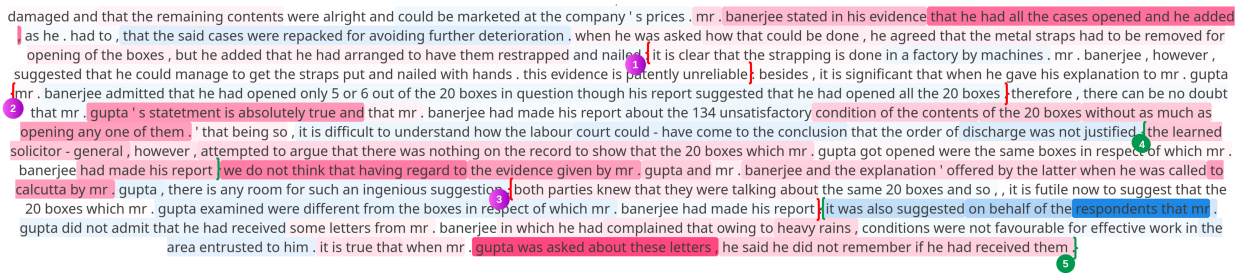


Figure 5: SHAP explanation of Case where model prediction is wrong and confidence is low.

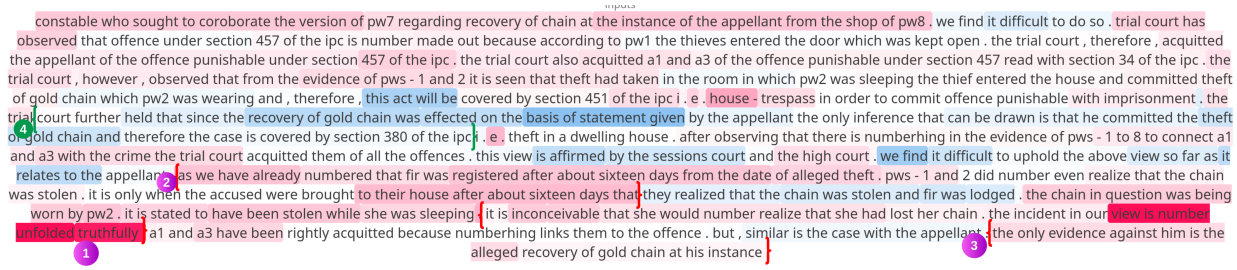


Figure 6: SHAP explanation of Case where model prediction is right and confidence is high.

- **Sentence 3** *"The only evidence against him is the alleged recovery of the gold chain . . . "*
  The Supreme Court questioning the reliability of the sole evidence used to convict the appellant.

Sentence Contradicting the Decision in blue

- **Sentence 3** *"The Trial Court further held . . . "*
  Indicates that the recovery of the chain was crucial in linking the appellant to the crime.

6.2.2 Case 2: Explanation of the SCI Proceedings Where Model is very Low Confident

The case involves a dispute regarding the condition of certain boxes of goods. Mr. Gupta discovered that few boxes were damaged, while the rest were intact and marketable. Mr. Banerjee, who was responsible for handling the boxes, claimed to have opened and repacked all 20 boxes, but his evidence was found unreliable. He admitted that he only opened a few boxes and fabricated his report on the condition of all 20 boxes. The Labour Court had previously concluded that the discharge order against Mr. Banerjee was unjustified. The Supreme Court reviewed the evidence and found Mr. Banerjee's actions to be questionable. The Supreme Court's decision favored the appellant, overturning the Labour Court's conclusion that the discharge order was unjustified.

For this example, our model's confidence is lower, leading to rejection. Explanations from SHAP for this example are given in Figure 5 where the explanation is based on wrong prediction. Below is the explanation based on actual label. Thus red and blue coloured explanations are reversed in Figure 5 with respect to actual label.

Sentences Leading to the Decision in Blue

- **Sentence 1** *It is clear that the strapping is done . . . "*
  Highlights the unreliability of Mr. Banerjee's claim about repacking the boxes.

- **Sentence 2** *"Mr. Banerjee admitted that he had opened only 5 or 6 . . . "*
  Directly challenges Mr. Banerjee's report and supports Mr. Gupta's statement regarding the condition of the boxes.

- **Sentence 3** *"Both parties knew that they were talking about the same 20 boxes . . ."*
  Reinforces the argument that Mr. Banerjee's report was inaccurate.

Sentence Contradicting the Decision in Red

- **Sentence 4** *The learned Solicitor-General, however, attempted to argue . . ."*
  Suggests a possible gap in evidence regarding whether the boxes Mr. Gupta examined were indeed the same as those reported on by Mr. Banerjee.
- **Sentence 5** *"It was also suggested on behalf of the respondents that . . ."*
  Implies that Mr. Gupta might not have acknowledged all relevant correspondence from Mr. Banerjee.

7 Conclusion

We propose NCwR-Cost and NCwR-Cov, novel GNN architectures for integrating reject option in node classification. These models can reject from making predictions whenever they are not certain about predicting an example. Our experimental results show that the proposed models perform better than the baseline methods. Such results show the importance of separate GNN architecture having a reject option integrated into it. Our results on the LJP task show that these models are very effective in such applications.

References

Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.

Diego Antognini. pygat. `https://github.com/Diego999/pyGAT`, 2021.

Xavier Bresson and Thomas Laurent. Residual gated graph convnets. *arXiv preprint arXiv:1711.07553*, 2017.

Yuzhou Cao, Tianchi Cai, Lei Feng, Lihong Gu, Jinjie GU, Bo An, Gang Niu, and Masashi Sugiyama. Generalizing consistent multi-class classification with rejection to be compatible with arbitrary losses. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 521–534. Curran Associates, Inc., 2022.

Nontawat Charoenphakdee, Zhenghang Cui, Yivan Zhang, and Masashi Sugiyama. Classification with rejection based on cost-sensitive classification. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1507–1517. PMLR, 18–24 Jul 2021.

Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. *Advances in neural information processing systems*, 32, 2019.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Junyun Cui, Xiaoyu Shen, and Shaochun Wen. A survey on legal judgment prediction: Datasets, metrics, models and challenges. *IEEE Access*, 2023.

Ajalmar R. da Rocha Neto, Ricardo Sousa, Guilherme de A. Barreto, and Jaime S. Cardoso. Diagnostic of pathology on the vertebral column with embedded reject option. In *Pattern Recognition and Image Analysis*, pp. 588–595, 2011.

Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Austin Derrow-Pinion, Jennifer She, David Wong, Oliver Lange, Todd Hester, Luis Perez, Marc Nunkesser, Seongjae Lee, Xueying Guo, Brett Wiltshire, et al. Eta prediction with graph neural networks in google maps. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 3767–3776, 2021.

Qian Dong and Shuzi Niu. Legal judgment prediction via relational learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 983–992, 2021.

Ran El-Yaniv et al. On the foundations of noise-free selective classification. *JMLR*, 11(5), 2010.

Yi Feng, Chuanyi Li, and Vincent Ng. Legal judgment prediction: A survey of the state of the art. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 5461–5469. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/765. Survey Track.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *NIPS*, pp. 4878–4887, 2017.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *ICML*, pp. 2151–2159, 2019.

William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning on Large Graphs. In *NIPS*, pp. 1024–1034, 2017.

Blaise Hanczar and Edward R. Dougherty. Classification with reject option in gene expression data. *Bioinformatics*, 24(17):1889–1895, 2008.

Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. *NeurIPS*, 2023.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Bhavya Kalra, Kulin Shah, and Naresh Manwani. Risan: Robust instance specific deep abstention network. In *Proceedings of the 37th Conference on Uncertainty in Artificial Intelligence*, volume 161, pp. 1525–1534, 27–30 Jul 2021.

Mann Khatri, Mirza Yusuf, Yaman Kumar, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. Exploring graph neural networks for indian legal judgment prediction. *arXiv preprint arXiv:2310.12800*, 2023.

Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*, 2017.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf`.

Vijit Malik, Rishabh Sanjay, Shubham Kumar Nigam, Kripabandhu Ghosh, Shouvik Kumar Guha, Arnab Bhattacharya, and Ashutosh Modi. ILDC for CJPE: Indian legal documents corpus for court judgment prediction and explanation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4046–4062, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.313. URL `https://aclanthology.org/2021.acl-long.313`.

Haggai Maron, Heli Ben-Hamu, Hadar Serviansky, and Yaron Lipman. Provably powerful graph networks. *Advances in neural information processing systems*, 32, 2019.

Christopher Morris, Martin Ritzert, Matthias Fey, William L Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Chenri Ni, Nontawat Charoenphakdee, Junya Honda, and Masashi Sugiyama. On the calibration of multiclass classification with rejection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Ross Quinlan. Thyroid Disease. UCI Machine Learning Repository, 1986. DOI: https://doi.org/10.24432/C5D010.

Harish G Ramaswamy, Ambuj Tewari, Shivani Agarwal, et al. Consistent algorithms for multiclass classification with an abstain option. *Electronic Journal of Statistics*, 12(1):530–554, 2018.

Yasin I. Rosowsky and Robert E. Smith. Rejection based support vector machines for financial time series forecasting. In *Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 1161–1167, Dallas, Texas, August 2013.

Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *International conference on learning representations*, 2018.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

M. Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*, 2017. URL `https://api.semanticscholar.org/CorpusID:5458500`.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93, Sep. 2008. doi: 10.1609/aimag.v29i3.2157. URL `https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2157`.

J. W. Smith, J. E. Everhart, W. C. Dickson, W. C. Knowler, and R. S. Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pp. 261–265, 1988.

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.

Zhenchao Sun, Hongzhi Yin, Hongxu Chen, Tong Chen, Lizhen Cui, and Fan Yang. Disease prediction via graph neural networks. *IEEE Journal of Biomedical and Health Informatics*, 25(3):818–826, 2021. doi: 10.1109/JBHI.2020.3004143.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. In *ICLR*, 2018.

Fangxin Wang, Yuqing Liu, Kay Liu, Yibo Wang, Sourav Medya, and Philip S Yu. Uncertainty in graph neural networks: A survey. *arXiv preprint arXiv:2403.07185*, 2024.

Tian Xia and Wei-Shinn Ku. Geometric graph representation learning on protein structure prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1873–1883, 2021.

Bingbing Xu, Huawei Shen, Bingjie Sun, Rong An, Qi Cao, and Xueqi Cheng. Towards consumer loan fraud detection: Graph neural networks with role-constrained conditional random field. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4537–4545, 2021.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2018.

A Comparison with different base GNN

We compare the results of our method on various GNN Architectures and Hyperparameters. We show that our method is GNN agnostic and results in similar performance for most of these architectures. Results are presented in Table 4.

Coverage	GCN	GAT	GraphSAGE	GATv2	GAT (3 layers)	GAT (4 layers)
0.7	88.86	88.71	85.86	89.14	88.75	89.21
0.75	88.00	87.73	85.47	88.60	87.35	88.25
0.8	85.62	86.87	84.50	86.37	87.75	86.18
0.85	85.53	85.06	83.06	85.65	85.96	85.81
0.9	83.89	84.67	82.56	84.22	84.45	84.03

Table 4: Accuracy comparison of our method with various base GNN Architectures. Unless mentioned, we use two GNN layers in each architecture.

B Applications on medical domain

We added applications on two more tabular health datasets: UCI Thyroid Quinlan (1986) dataset and Pima Indians Diabetes dataset Smith et al. (1988). Each tabular dataset was transformed into a graph structure to enable graph-based learning. Initially, the dataset was standardized to normalize the feature values across samples, facilitating the k-nearest neighbours (KNN) process. Using KNN, an edge was created between each data point and its closest neighbours in feature space which formed a sparse graph. This approach effectively connected each node (data point) with a fixed number of neighbours (taken $k = 5$), creating edges that reflect the similarity in feature space.

To represent this as a graph, each data point was treated as a node, where its feature vector constituted node attributes, and the class label served as the target variable. Edges were defined based on KNN relationships, where nodes shared edges if they were among each other's nearest neighbours. This graph was then formatted into a structure suitable for graph neural networks, containing nodes, edges and labels to support message-passing and learning across connected nodes.

The graph was constructed by concatenating the training (85%) and test (15%) sets, treating them as a unified structure to facilitate transductive learning. Approximately 10% of the training data was reserved as a validation set, ensuring a fair evaluation of the model's performance during training. While the full graph was available for message passing during training and testing, only the labels of the training nodes were utilized for model optimization. We have shown the results for NCwR-Cost and NCwR-Cov models on the UCI Thyroid dataset in Table 5 and on the Pima Indians Diabetes dataset in Table 6.

NCwR-Cost			NCwR-Cov	
d	Acc (%)	Cov (%)	Cov	Acc (%)
0.05	96.56 ± 0.68	60.52 ± 9.21	99.66 ± 0.09	0.5
0.10	96.40 ± 0.13	82.37 ± 2.39	99.53 ± 0.06	0.6
0.20	95.29 ± 0.40	94.09 ± 0.84	99.31 ± 0.10	0.7
0.30	94.37 ± 0.21	98.20 ± 0.18	98.75 ± 0.09	0.8
0.40	93.75 ± 0.18	99.76 ± 0.13	97.48 ± 0.04	0.9

Table 5: Accuracy of NCwR-Cost and NCwR-Cov for various cost of rejection values and coverage rates respectively on UCI Thyroid dataset. Shown mean and standard deviation for 5 runs.

NCwR-Cost			NCwR-Cov	
d	Acc (%)	Cov (%)	Acc (%)	Cov
0.30	88.54 ± 0.68	64.00 ± 2.57	94.38 ± 1.40	0.5
0.35	87.40 ± 0.13	73.23 ± 2.79	90.77 ± 1.40	0.6
0.40	86.46 ± 0.40	84.00 ± 3.19	88.89 ± 1.57	0.7
0.45	84.11 ± 0.21	92.92 ± 1.75	87.31 ± 1.72	0.8
0.50	81.79 ± 0.18	99.69 ± 0.69	85.86 ± 2.25	0.9

Table 6: Accuracy of NCwR-Cost and NCwR-Cov for various cost of rejection values and coverage rates respectively on Pima Indians Diabetes dataset. Shown mean and standard deviation for 5 runs.