Conformal Edge-Weight Prediction in Latent Space

Akash Choudhuri* Yongjian Zhong* Mehrdad Moharrami* Christine Klymko†
Mark Heimann† Jayaraman J. Thiagarajan‡ Bijaya Adhikari*

Abstract

Predicting the edge weights of a graph is a critical task across many domains. Some examples include predicting traffic flow in transportation networks, strength of interactions in protein-protein networks, and collaboration frequency in co-authorship networks. Graph Neural Networks have been very successful in edge-weight prediction tasks. However, these predictions lack rigorous statistical uncertainty quantification. Recent work has demonstrated the efficacy of conformal inference in quantifying the uncertainties of the predictions made by graph neural networks. However, there has been limited research in conformal inference for edge-weight prediction.

A prior work has demonstrated that the powerful inductive bias of the representations learned by deep neural networks can be leveraged for robust uncertainty quantification. In this paper, we extend the traditional conformal inference paradigm to compute uncertainty estimates in the latent space exploiting the deep representations learned by graph neural networks. Specifically, our method (Edge-CP) computes the non-conformity scores in the latent feature space of the nodes and leverages the scores for bandwidth estimation for weighted edge prediction. Experiments on a wide variety of edge-weighted networks show that Edge-CP always achieves the pre-defined target marginal coverage and obtains up to 38.16% shorter bands than the nearest baseline. Additionally, Edge-CP achieves the best conditional coverage among all methods.

1 Introduction

Graph Neural Networks (GNN) have a wide variety of applications in numerous domains such as in traffic-forecasting [26], analyzing RNA sequences in biological networks [42], and drug discovery [24]. While GNNs offer better performance in predicting point estimates overall, deploying them in high-stakes environments is challenging as their predictions could be unreliable. Calibrating GNNs to provide uncertainty estimates of their predictions is a promising direction to resolve this issue. However, GNNs are notoriously under-confident [43], contrary to other deep-learning frameworks like Convolution Neural Networks (CNN), which are generally over-confident [36], traditional calibration techniques do not perform well for GNNs [28, 48]. This leads to the need to design calibration techniques tailored to networks and the corresponding encodings by GNN which are used as predictors. While there has been some research interest in GNN-specific calibration, these are either computationally expensive or fail to provide statistical guarantees of the effectiveness of the uncertainty estimates [12, 13, 43].

The field of conformal inference, pioneered by [41] provides prediction regions along with point predictions, thus providing a notion of model uncertainty. Recently, there has been some interest in conformal inference for GNN predictions for both inductive and transductive settings [6, 14, 45, 46]. However, all of these works only perform node-level prediction. Although node-level predictive tasks are generally used in various applications, edge-level predictions are particularly important in biological and drug networks. For example, in protein-protein networks, the interactions unravel the cellular behavior and its functionality [17]. Similarly, an edge in a drug–target interaction network indicates that a drug binds to a target protein [15]. Therefore, uncertainty quantification is equally important for edge-level tasks.

One naive approach of extending node-level uncertainty quantification for edge-level tasks would transform each graph to its corresponding line-graph [5] and use the previously developed conformal inference methods on the transformed graph. However, this approach has some drawbacks. First, this technique will not scale to larger graphs where the number of edges is order of magnitude larger than the number of nodes. Second, the predictive performance of GNNs can vastly differ between the original graph and the corresponding line-graph, thus affecting the uncertainty estimates for their predictions. This leads to the need for the development of a conformal inference algorithm tailored to edge predictions on the original graph itself.

To address the research gaps mentioned above, we

*Department of Computer Science, University of Iowa. {akash-choudhuri, yongjian-zhong, mehrdad-moharrami, bijaya-adhikari}@uiowa.edu

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory. {klymko1, heimann2}@llnl.gov

‡Apple Inc. jjthiagarajan@gmail.com

propose a conformal inference approach for edge-weight prediction that operates on the original graph. We assume an encoder-decoder framework as the mean estimator, where the encoder can be a GNN that aggregates node information while the decoder performs edge-weight prediction using the latent embeddings of the aggregated information of the nodes. We extend the notion of 'surrogate features' [37] to edge-weight prediction and develop a general non-conformity score that quantifies uncertainty. This non-conformity score measures the deviation between the surrogate feature embeddings and the embeddings produced by the encoder in the latent dimension. We then transfer the heuristic notion of uncertainty in the latent embedding space to the output space and correspondingly ensure the general marginal coverage guaranteed by conformal inference methods. Our contributions can be summarized as follows:

- We present a principled approach to directly apply conformal inference for uncertainty quantification for weighted edge prediction using the latent node representations learned by GNNs.
- We perform extensive experiments on four real-world datasets, and our proposed approach outperforms all the baselines in three out of the 4 datasets in terms of band efficiency.
- We perform experiments to show that our approach obtains reasonable conditional coverage, which is an indicator of the adaptability of the bands to the nature of every individual sample.

2 Related Work

We discuss the literature related to our present work here. Prior works can be characterized into the following categories:

2.1 Uncertainty Quantification in Graph Neural Networks: Multiple methods perform model-agonistic risk estimation in Graph Neural Networks (GNNs) for both classification and regression tasks [28, 33, 48], while other works leverage the structural properties of graphs with [13, 43] by exploring underconfidence of GNNs for calibration. [8] interprets dropout training in deep neural networks as approximate Bayesian inference within deep Gaussian Processes while [12,21] identify depth, width, weight decay, batch normalization and other temperature scaling methods for calibration. [39, 40] develop and apply a stochastic centering method for calibration in GNNs.

2.2 Conformal Inference: The distribution-free nature of the coverage guarantee provided by conformal inference has allowed applications in various domains like model calibration [35], passenger booking systems [44], computer vision [1,4] and time series forecasting [9,23]. Given a user-specified miscoverage level (uncertainty level) $\alpha \in (0,1)$, it leverages a set of 'calibration' data to output prediction sets/intervals for new test points that provably include the true outcome with probability at least $1-\alpha$. Different non-conformity scores have been proposed by prior works [16,31,32] for both classification tasks with a recent work proposing a score in the latent feature space [37]. While exchangeability remains an important assumption in standard conformal inference, works like [3, 9, 23, 38] extend the standard conformal inference beyond exchangeability in cases of label or covariate shift or dependent data.

2.3 Conformal Inference in Graph Neural Networks: The application of conformal inference to network data has gained popularity recently with the first application being in the inductive setting [6], which showed that the non-conformity scores are not exchangeable. On the other hand [14, 25, 45] operate in the transductive setting, where the non-conformity scores are exchangeable. Both these works utilize the local neighborhood structure of the graph to improve effectiveness while obtaining good efficiency. A recent work [46] builds the notation of node-exchangeability and edge-exchangeability in growing graphs in the inductive setting and applies non-conformity scores based on the structure of the graph thus formed at each step.

3 Preliminaries

Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y})$ be a digraph, where $\mathcal{V}$ is a set of nodes, $\mathcal{E}$ is a set of edges, and $\mathcal{X} = \{\mathbf{x}_v\}_{v\in\mathcal{V}}$ is the set of node attributes, where $\mathbf{x}_v \in \mathbb{R}^d$ is a d-dimensional feature vector for node $v \in \mathcal{V}$. Let $\mathcal{Y} = \{y_{u,v}\}_{(u,v)\in\mathcal{E}}$ be the set of edge weights, and $y_{u,v} \in \mathbb{R}$ is the weight of an edge $e_{u,v}$ connecting a pair of nodes v_u and v_v. Our paper focuses on regression problems, but our theory and method can be easily extended to classification problems. To perform point predictions, we are given a mean estimator μ that predicts the edge weight $\hat{y}_{\mathrm{u,v}}$ given the node embeddings x_u and x_v that form the edge.

3.1 Transductive Setting: We focus on the transductive weighted edge prediction problem with random data split. In this setting, we partition the edge weights into three disjoint sets: *training* $D_{\text{train}} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y}_{\text{train}})$, *calibration* $D_{\text{cal}} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y}_{\text{cal}})$, and *testing* $D_{\text{test}} = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y}_{\text{test}})$. In particular, during

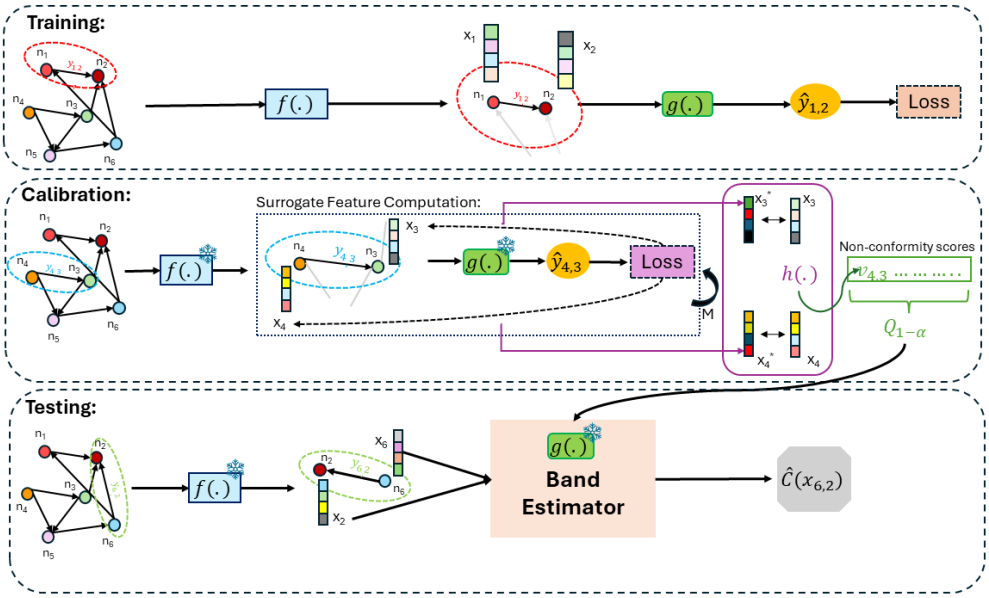

Figure 1: Schematic Representation of Edge-CP (best viewed in color): During **training**, the encoder f(·) learns node representations while the decoder g(·) concatenates pairs of node embeddings to predict edge weights for the training labeled edges. During **calibration**, the weights of f(·) and g(·) are frozen and for each calibration edge, the surrogate feature is computed by steps 4-8 of Algorithm 1. Then the non-conformity score is represented by a function of the norm of deviation between the surrogate features and the encoded features of the nodes. The non-conformity scores of the calibration edges are sorted and $Q_{1-\alpha}$ is computed, which is then sent as the perturbation limit to the band estimator during **testing**. The band estimator also takes in the encoded node embeddings of the corresponding test edge and returns the upper and lower uncertainty bands for each test edge.

training, the model can access $\mathcal{V}, \mathcal{E}, \mathcal{X}$, but only the training weights $\mathcal{Y}_{\text{train}}$ are revealed to the model. Abusing the notation, we use $\mathcal{E}_{\text{train}}$ to denote elements of $\mathcal{E}$ for which the edge weights are in $\mathcal{Y}_{\text{train}}$. We follow the same notation throughout the paper. After training, the calibration data $\{y_{u,v}\}_{(u,v)\in\mathcal{E}_{\text{cal}}}$ is used to apply conformal prediction. Finally, we predict the weights of the remaining edges (i.e., $\mathcal{E}_{\text{test}}$).

3.2 Conformal Inference: In this paper we focus on split conformal inference [14, 45], where given a predefined miscoverage rate $\alpha \in [0, 1]$, for a given non-conformity score that quantifies the heuristic notion of uncertainty, the method proceeds in the following steps: (1) **Training:** Train the mean estimator μ on the training data D_{train}. (2) **Score Computation:** For each edge $e = (u, v)$ connecting nodes u and v in $\mathcal{E}_{\text{cal}}$, compute the non-conformity scores $\{V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v})\}_{e\in\mathcal{E}_{\text{cal}}}$ and create a distribution from the scores. (3) **Quantile Creation:** Compute the $(1-\alpha)^{\text{th}}$ quantile $\hat{Q}_{1-\alpha}$ of the distribution $\frac{1}{|\mathcal{E}_{cal}|+1}\sum_{e\in\mathcal{E}_{cal}} \delta_{V_e} + \delta_\infty$, where δ_a is Dirac Delta distribution at point a, and V_e when $e = (u, v)$ is shorthand for $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v})$. (4) **Predictive Band Creation:** Given a new test point $e_{u,v}$, a prediction set/interval $\hat{C}(e_{u,v}) = \{y \in \mathcal{Y} : V(\mathbf{x}_u, \mathbf{x}_v, y) \leq \hat{Q}_{1-\alpha}\}$ is constructed. The notion of transferring the prediction bands computed on the calibration data to the points in test data relies on the following assumption on permutation invariance [14, 46].

Assumption 1. *For any permutation π on the calibration and test edges, the non-conformity score V obeys*

$$V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}; \{y_{a,b}\}_{(a,b)\in\mathcal{E}_{train}}, \mathcal{X}, \mathcal{V}, \mathcal{E}) = V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}; \{y_{a,b}\}_{(a,b)\in\mathcal{E}_{train}}, \mathcal{X}, \mathcal{V}, \mathcal{E}_\pi)$$

This means that the non-conformity scores of edges in a graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{X}, \mathcal{Y})$ are exchangeable.

Assumption 1 imposes the permutation invariance condition for the GNN training to later compute the non-conformity scores for edge prediction, which means that the model output/ non-conformity score is invariant to permuting the order of the calibration and test edges on the graph. Note that this is an extension of the assumption [14] makes for nodes, but for edges. Typically, an edge is constructed by a pair of nodes. As the non-conformity scores for nodes are exchangeable due to GNNs obeying the permutation invariance assumption [19], so would the scores that are constructed by a function that takes in a pair of node scores.

Lemma 1. *(Coverage Guarantee for Conformal Inference) [14, 38, 41] Under Assumption 1, for any $\alpha > 0$, the confidence band returned by the conformal inference algorithm satisfies:*

$$\mathbb{P}(y_{u,v} \in \hat{C}_{1-\alpha}(e_{u,v})) \geq 1 - \alpha \tag{3.1}$$

where the probability is taken over the calibration fold D_{cal} and the testing point $(e_{u,v}, y_{u,v})$.

Here, $\mathbb{P}(y_{u,v} \in \hat{C}_{1-\alpha}(e_{u,v}))$ denotes the **coverage** that the true label $y_{u,v}$ lies in the predictive band. The proof of the lemma is provided in the Appendix.

4 Methodology

We now describe our proposed approach, which we call Edge-Conformal Prediction (Edge-CP) to reduce inefficiency (trivially large predictive band lengths) while maintaining valid coverage. Edge-CP leverages the individual node-level uncertainties to predict the edge-weights between given pairs of nodes. The key idea of Edge-CP is to construct the non-conformity score function in the latent dimension and transfer the notion of non-conformity in the latent dimension to the output space. We will now define the mean estimator as a composition of functions.

4.1 Mean Estimator: We define the mean estimator μ for Edge-CP using two components, namely an encoder (represented by function $\mathrm{f}(\cdot)$:$\mathbb{R}^n \to \mathbb{R}^{n_L}$) and a decoder (represented by function $\mathrm{g}(\cdot,\cdot)$:$\mathbb{R}^{n_L} \times \mathbb{R}^{n_L} \to \mathbb{R}$). The point prediction for each sample can be obtained by passing it via a composition of these 2 functions. The details about the encoder and the decoder functions are, as follows: **(1) Encoder: Graph Neural Network:** Graph Neural Networks (GNNs) aggregate neighborhood information [10] via a sequence of propagation layers where the k^{th} layer consists of a Message Passing Step, and a Node Update Step.**(2) Decoder:** After the node representations are learned via the encoder, the decoder predicts the edge weight between a pair of nodes i and j by concatenating them and passing it via a feed-forward neural network.

4.2 Node-Uncertainty Enhanced Edge Non-Conformity Score: In this section, we elaborate on our proposed non-conformity score for the weighted edge prediction problem in graphs. While different norm-based scores can be used as non-conformity scores to quantify the non-conformity of samples [29, 30], our approach leverages a heuristic notion of node level uncertainty in the latent feature space [37] to quantify uncertainty in edge weight prediction. We transfer the heuristic idea of non-conformity in the latent feature space and transfer that notion to the output space. To this end, we first define the surrogate node feature embeddings in the latent feature space, which can serve as the ground truth proxy of the label value in the output space.

Algorithm 1 Non-Conformity Feature Score

Require: Pair of node features with their corresponding edge weight label $(\mathbf{x}_u, \mathbf{x}_v, y_{u,v})$, trained encoder $\hat{f}(\cdot)$, trained decoder $\hat{g}(\cdot,\cdot)$, NC function $h(\cdot,\cdot)$, step size η, number of steps M

$\mathbf{s}_u \leftarrow \hat{f}(\mathbf{x}_u)$
$\mathbf{s}_v \leftarrow \hat{f}(\mathbf{x}_v)$
$m \leftarrow 0$
while $m < M$ **do**
  $\mathbf{s}_u \leftarrow \mathbf{s}_u - \eta \frac{\partial ||\hat{g}(\mathbf{s}_u, \mathbf{s}_v) - y_{u,v}||^2}{\partial \mathbf{s}_u}$
  $\mathbf{s}_v \leftarrow \mathbf{s}_v - \eta \frac{\partial ||\hat{g}(\mathbf{s}_u, \mathbf{s}_v) - y_{u,v}||^2}{\partial \mathbf{s}_v}$
  $m \leftarrow m + 1$
end while
return $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}) = h(||\mathbf{s}_u - \hat{f}(\mathbf{x}_u)||, ||\mathbf{s}_v - \hat{f}(\mathbf{x}_v)||)$

Definition 1: (Surrogate Node Embeddings) *Given a trained mean estimator $\hat{\mu} = \hat{g}(\hat{f}(\cdot), \hat{f}(\cdot))$ denoting a composition of two functions, where $\hat{f}(\cdot)$ denotes the encoder function and $\hat{g}(\cdot,\cdot)$ denotes the decoder function, for a given sample $(\mathbf{x}_u, \mathbf{x}_v)$ with label $y_{u,v}$, $(\hat{\mathbf{s}}_u, \hat{\mathbf{s}}_v)$ denote the latent node feature embeddings. The surrogate node feature embeddings are the latent feature embeddings $(\mathbf{s}_u, \mathbf{s}_v)$ where $g(\mathbf{s}_u, \mathbf{s}_v) = y_{u,v}$.*

With the help of the surrogate node feature embeddings, we can now define a non-conformity score in the feature space. The score is given as $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}) = h(||\mathbf{s}_u - \hat{f}(\mathbf{x}_u)||, ||\mathbf{s}_v - \hat{f}(\mathbf{x}_v)||)$. Here $h(\cdot,\cdot)$ is a binary operator on the deviation between the surrogate node feature embeddings and the latent node embeddings for each pair of nodes constituting an edge. The surrogate node embeddings serve as a proxy for the node embeddings that are needed to predict the given edge weight correctly. The deviation in turn creates a ℓ_p ball around those perfect node embeddings with the radius of $\|\mathbf{s}_v - \hat{f}(\mathbf{x}_v)\|, \forall v \in \mathcal{V}$. However, the presence of the infimum operator in the non-conformity score makes it intractable in practice. So, we use Algorithm 1, which applies gradient descent from the encoder's learned node embeddings to the surrogate node feature embeddings independently to compute the upper bound of the proposed non-conformity score.

4.3 Band Estimation: Having constructed the notion of non-conformity in the latent feature space where we leverage the heuristic of deviation between the ideal

and predicted node embedding for perfect prediction of edge weights and derive a confidence band based on it, we will need a way to transfer this confidence band to the output space as well.

To transfer the confidence bands to the output space, we can use the decoder $\hat{g}(\cdot,\cdot)$. Since the decoder $\hat{g}(\cdot,\cdot)$ is non-linear, it is harder to estimate the confidence band explicitly. So, we use Neural Network Robustness Certification Methods like Interval Bound Propagation (IBP) [11] and CROWN [47]. Both these methods utilize interval arithmetic to propagate bounds through each layer of the network. For a test sample passed through the encoder with node embeddings $(\mathbf{x}_u, \mathbf{x}_v)$ and a given perturbation value ϵ, the method computes upper and lower bounds on the activations at each layer of the decoder, ensuring that the output logits respect these bounds. Mathematically, if we assume that $\hat{g}(\cdot,\cdot)$ is a m-layer neural network with $n_k \forall k \in [m]$ neurons in each layer k-th layer weight matrix be $\mathbf{W}^{(k)} \in \mathbb{R}^{n_k \times n_{k-1}}$. If $(\mathbf{x}_{\pi(u)}, \mathbf{x}_{\pi(v)})$ are the perturbed versions of node embeddings $(\mathbf{s}_u, \mathbf{s}_v)$ within an ϵ-bounded ℓ_p-ball centered at $(\mathbf{s}_u, \mathbf{s}_v)$, denoted as $\mathbf{x}_{\pi(u)} \in \mathbb{B}_P(\mathbf{s}_u, \epsilon)$ and $\mathbf{x}_{\pi(v)} \in \mathbb{B}_P(\mathbf{s}_v, \epsilon)$ with $\mathbb{B}_P(x_O, \epsilon) = \{x : ||x - x_O||_\infty \leq \epsilon\}$ these methods construct two explicit functions $\hat{g}^L : \mathbb{R}^{n_L} \times \mathbb{R}^{n_L} \to \mathbb{R}$ and $\hat{g}^U : \mathbb{R}^{n_L} \times \mathbb{R}^{n_L} \to \mathbb{R}$ such that the inequality $\hat{g}^L(\mathbf{x}_{\pi(u)}, \mathbf{x}_{\pi(v)}) \leq \hat{g}(\mathbf{x}_{\pi(u)}, \mathbf{x}_{\pi(v)}) \leq \hat{g}^U(\mathbf{x}_{\pi(u)}, \mathbf{x}_{\pi(v)})$ holds. Thus, these robustness certification methods can provide the upper and lower bounds (uncertainty estimate) in the output space, given the node embeddings in the feature space and a perturbation value. Our overall method is presented in Algorithm 2. Here the perturbation is $Q_{1-\alpha}$ from Step 7 of Algorithm 2 computed after estimating scores on the calibration data Having defined the overall framework, we will now theoretically demonstrate the benefits of our proposed framework over general conformal inference methods.

4.4 Theoretical Guarantees

Theorem 1. *(Edge-CP is provably more efficient than Vanilla-CP) Assume that the node feature space satisfies the following conditions:*
(1) Length Preservation: *The loss of information for Edge CP in the latent feature space is bounded by the information obtained in the output space.*
(2) Expansion: *The Band Estimation operator expands the differences between individual length and their quantiles.*
(3) Quantile Stability: *Given a calibration set D_{cal}, the quantile of the band length is stable in both feature space and output space*
Then Edge CP provably outperforms Vanilla CP in terms of average band length where the expectation is taken over the calibration fold and the testing point.

An informal sketch of the proof will be to start with the statement of Expansion which assumes that the difference between the quantile and each individual is smaller in the feature space than that in the output space. We display empirical results to verify this claim later. We will then rearrange the terms, but unlike [37] cannot directly use the Holder condition as our nonconformity score is not simply a norm-type function but is instead a function of two norm type functions. Thus we leverage the Lipschitz continuous property coupled with the Minkowski Inequality and then use the statement of Length Preservation and Quantile Stability to get the final form of the proof statement. For the detailed proof, refer to the Appendix.

Algorithm 2 Weighted Edge Conformal Prediction (Edge-CP)

Require: Level α, Graph G, D_{train}, D_{cal}, and Test point $e_{u,v}$

1: Randomly split the dataset D into training (D_{train}) and calibration fold (D_{cal})
2: Train a base ML model $\hat{g}(\hat{f}(\cdot), \hat{f}(\cdot))$ using D_{train}
3: Freeze the weights of $\hat{f}(\cdot)$ and $\hat{g}(\cdot,\cdot)$
4: **for** each edge (i,j) with weight $y_{i,j}$ in $\mathcal{E}_{cal}$ **do**
5: Get NC score $V(\mathbf{x}_i, \mathbf{x}_j, y_{i,j})$ via Algorithm 1
6: **end for**
7: Calculate the $(1-\alpha)^{\text{th}}$ quantile $\hat{Q}_{1-\alpha}$ of the distribution:

$$\frac{1}{|\mathcal{E}_{cal}| + 1} \sum_{e \in \mathcal{E}_{cal}} \delta_{V_e} + \delta_\infty$$

8: Apply Band Estimation on test data features with perturbation $\hat{Q}_{1-\alpha}$ and prediction head $g(\cdot,\cdot)$, which returns $\hat{C}^{ecp}_{1-\alpha}(e_{u,v})$
9: **return** $\hat{C}^{ecp}_{1-\alpha}(e_{u,v})$

5 Experiment

5.1 Setup: We conduct experiments to demonstrate the advantages of Edge CP over other Conformal Uncertainty Quantification methods in achieving empirical marginal coverage for graph data and report the average band length. We also evaluate the conditional coverage of Edge CP and conduct parameter analysis with different score functions.
Evaluation: For the task of weighted edge prediction, we follow a standard semi-supervised learning evaluation procedure [19], where we randomly split data into train and test folds with 80:20 split ratio. Then we equally split the train data into real train and calibration folds randomly. We adopt the following metrics to evaluate the algorithmic empirical performance:

- **Empirical Coverage (Effectiveness):** It is the empirical probability that a test point falls into the predicted confidence band. A good predictive inference method should achieve empirical coverage slightly larger than $1-\alpha$ for a given significance level α.
- **Band Length (Efficiency):** Given the empirical coverage being larger than $1-\alpha$, we want the confidence band to be as short as possible. The band length should be compared under the regime of empirical coverage being larger than $1-\alpha$, otherwise one can always set the confidence band to empty to get a zero band length.

To calculate the coverage for Edge CP, we first apply Band Estimation on the test point $e_{u,v}$, with label $y_{u,v}$) to detect whether $y_{u,v}$ is in $\hat{\mathcal{C}}^{\text{ecp}}_{1-\alpha}(e_{u,v})$, and then calculate its average value to obtain the empirical coverage. Also, since the explicit expression for confidence bands is intractable for the proposed algorithm, we could only derive an estimated band length via Band Estimation. Concretely, we first use band estimation to estimate the confidence interval, which returns a band with explicit formulation, and then calculate the average length across each dimension.

Table 1: Description of the 4 Datasets

Dataset	|V|	|E|	Label Range
HS-PI	17,849	633,460	(1.77-4.90)
cond-mat	16,264	47,594	(0.05-22.33)
astro-ph	16,046	121,251	(0.01-16.50)
BZR-MD	6,520	137,734	(1.14-16.64)

Although conformal prediction methods only theoretically guarantee empirical coverage [7], it is desirable to have adaptive confidence bands based on the 'hardness' of samples (referred to as conditional coverage) [32]. Conditional coverage asks for $\mathbb{P}(y_{u,v} \in \hat{\mathcal{C}}(e_{u,v}) \mid e_{u,v} = e) \approx 1-\alpha, \forall e$.

Baselines: As smaller effectiveness always leads to higher efficiency, for a fair comparison, we can only compare methods on efficiency that achieve the same effectiveness. Thus, we do not evaluate other uncertainty quantification baselines as they do not produce exact effectiveness and are thus not comparable. We report the details about the baselines below:

- **Monte-Carlo Dropout (MCDropout) [8]:** After the base estimator is trained, this method turns on dropout during evaluation and produces K predictions. We then take the 90% quantile of the predicted distribution.
- **Conformal Quantile Regression (CQR) [31]:** This method corrects the upper and lower quantiles produced by the base estimator with the scored correction term.
- **Conformalized GNN (CF-GNN) [14]:** This method adds a topology-aware correction model on top of the base estimator that updates the node predictions based on their neighbors, thus leading to shorter bands.
- **Vanilla-CP (Vanilla) [41]:** This is a classical conformal inference method for regression on the original edge-weighted graph which is constructed by evaluating the MSE for each edge weight in the output space for calibration dataset, finding the $1-\alpha$ quantile of the scores and use that for test data to construct prediction band $\hat{\mathcal{C}}^{\text{vanilla}}_{1-\alpha}(e_{u,v})$ for a test point $e_{u,v}$.
- **Error Reweighted Conformal CQR (CQR-ERC) [25]:** This method is a recent work using conformal inference for weighted edge prediction. To address data heteroscedasticity, the authors employ error reweighting and CQR.

For the baselines that perform on the line graph as well as CQR-ERC, we use a 2-layer GCN. In contrast, for our method and Vanilla CP, we use a 2-layer GCN as the encoder followed by concatenation of embeddings and predicting via FFN layers as the decoder.

Datasets: We evaluate Edge CP on 4 weighted edge prediction datasets of diverse network types. We summarize some basic statistics in Table 1 and give more details below:

- **Human Protein-Protein Physical Interaction Network (HS-PI):** The Human Protein-Protein physical interaction network compiled by the HumanNet project [18] where the interaction strengths were compiled from years of protein-protein interaction mappings from a variety of sources.
- **Astrophysics Collaboration Network (astro-ph) [27]:** This network contains the collaboration network of scientists posting preprints on the astrophysics archive on arXiv during 1995-1999 and was compiled by M. Newman.
- **Condensed-Matter Physics Collaboration Network (cond-mat) [27]:** This network contains the collaboration network of scientists posting preprints on the condensed matter archive on arXiv during 1995-1999 and was compiled by M. Newman.

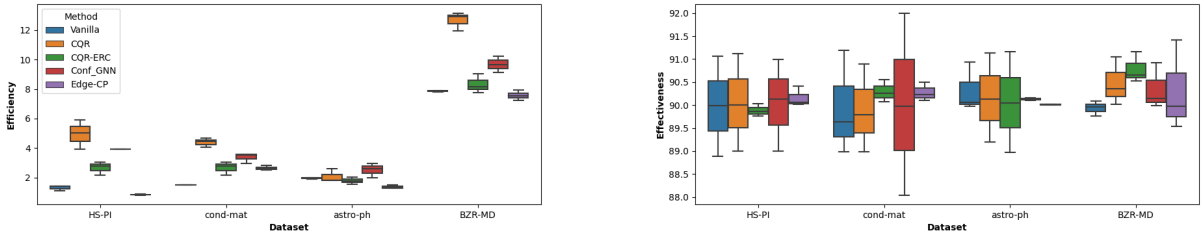

Figure 2: Variations of Effectiveness (right) and Efficiency (left) across seeds for the 4 datasets with $\alpha = 0.1$

Table 2: Mean Effectiveness and Band Length for 4 Datasets, $\alpha = 0.1$

Method	Effectiveness (higher is better)				Efficiency (lower is better)			
	HS-PI	cond-mat	astro-ph	BZR-MD	HS-PI	cond-mat	astro-ph	BZR-MD
Vanilla	89.96 ✗	90.13 ✓	90.27 ✓	89.93 ✗	1.31	**1.51**	1.93	7.86
MCDropout	30.26 ✗	68.57 ✗	72.24 ✗	73.44 ✗	1.16	2.06	0.28	4.11
CQR	90.04 ✓	89.89 ✗	89.99 ✗	90.42 ✓	4.93	4.39	2.08	12.83
CQR-ERC	89.98 ✗	90.16 ✓	90.04 ✓	90.77 ✓	2.73	3.58	1.73	8.21
CF-GNN	90.04 ✓	90.01 ✓	90.13 ✓	90.25 ✓	3.93	3.36	2.51	9.47
Edge-CP	90.37 ✓	90.38 ✓	90.00 ✓	90.30 ✓	**0.81**	2.60	**1.30**	**7.56**

- **Benzodiazepine Receptor (BZR) Network (BZR-MD) [20, 34]:** This network contains interactions between a set of 405 ligands of the Benzodiazepine Receptor (BZR).

5.2 Results: From Tables 2 and 3, we observe several important findings:

Table 3: Mean SSC for 4 Datasets, $\alpha = 0.1$

Dataset	Model	Mean SSC
HS-PI	**CQR**	83.08
	CQR-ERC	85.21
	CF-GNN	83.98
	Edge-CP (Ours)	**88.37**
cond-mat	**CQR**	80.13
	CQR-ERC	81.38
	CF-GNN	81.16
	Edge-CP (Ours)	**82.79**
astro-ph	**CQR**	81.22
	CQR-ERC	**83.58**
	CF-GNN	78.59
	Edge-CP (Ours)	82.03
BZR-MD	**CQR**	74.33
	CF-GNN	79.67
	CQR-ERC	79.91
	Edge-CP (Ours)	**80.73**

Edge-CP always achieves empirical effectiveness: The left side of Table 2 shows the mean effectiveness for all the methods for the task of weighted edge prediction for a target effectiveness of 90% across 4 datasets. While MCDropout, CQR, and CF-GNN operate on the line graph, Edge-CP, CQR-ERC, and Vanilla operate on the original graph itself. Firstly, we observe that none of the uncertainty quantification methods other than CF-GNN and Edge-CP achieves the desired target effectiveness across all datasets. The results thus validate the validity of our theory behind Edge-CP for uncertainty quantification. Also, notice that Edge-CP consistently achieves the greatest effectiveness. We believe that the slight over-coverage of Edge-CP also helps it to achieve better conditional coverage.

Edge-CP achieves competitive efficiency: The right side of Table 2 shows the mean effectiveness for all the methods for the task of weighted edge prediction for a target effectiveness of 90% across 4 datasets. We observe that our method shows improvement over Vanilla (we do not compare MCDropout as it has very low Effectiveness for all the datasets) for HS-PI, astro-ph and BZR-MD datasets and loses to it in the cond-mat dataset. However, Edge-CP still performs better than CQR, CQR-ERC and CF-GNN (both of which also have conditional coverage while Vanilla does not) across all four datasets. This means that Edge-CP is not just returning trivial prediction bands with extremely high effectiveness. We hypothesize that CF-GNN does worse than our method as in the line graph, the original graph's edges now become nodes and in the Topology-aware Conformal Correction Process, the information gathered from the local subgraph is that

of edges instead of nodes. On the other hand, CQR-ERC is simply a variant of Vanilla but with adaptive bands that only computes the NC scores in the output space and fails to leverage the rich information from the latent feature space of the nodes. However, Edge-CP directly leverages node uncertainty in the score function to compute the non-conformity scores.

Edge-CP achieves the best Overall Conditional Coverage over all baselines: While Edge-CP achieves marginal effectiveness, it is highly desirable to have a method that achieves reasonable conditional coverage, which was the motivation of APS [32] and CQR [31]. We evaluate conditional coverage for the methods by estimating the Size-stratified coverage metric [2]. Lower SSC indicates lower conditional coverage and consequently lower adaptiveness of the conformal method to the nature of the noise in the data sample. Table 3 shows the mean SSC for the UQ methods. Vanilla has equal-sized predictive bands and thus cannot be evaluated for conditional coverage. On the other hand, we do not evaluate MCDropout as it has very little effectiveness. We observe that Edge-CP has the highest SSC across all datasets compared to CQR and CF-GNN and achieves SSC very close to $1-\alpha$ in all of them, obtaining it for STRING-H. We notice that CQR-ERC has better SSC than our method for astroph. However note that this 1.89% gain in SSC comes with a 48.21% loss in marginal effectiveness as shown in Table 2.

Table 4: Results for different latent feature-based score functions for 2 datasets, $\alpha = 0.1$

	Score	Effectiveness	Efficiency	SSC
HS-PI	**Cat**	90.68	0.83	87.98
	Max	90.31	0.82	88.51
	Sum	91.75	0.98	88.92
	Min	90.37	0.81	88.37
cond-mat	**Cat**	90.38	2.60	82.79
	Max	90.58	2.79	80.36
	Sum	93.35	3.17	90.52
	Min	90.13	2.65	78.95

5.3 Ablation Study: Variants of Edge-CP Non-Conformity Scores We performed additional experiments by altering the score functions of Edge-CP for HS-PI and cond-mat datasets. For this experiment, the variants of the binary-operator score function $h(\cdot,\cdot)$ were:

1. Cat: In this score function, for step 9 of Algorithm 1, we first concatenated the node embeddings $\hat{f}(\mathbf{x}_u)$ and $\hat{f}(\mathbf{x}_v)$ and then concatenated the Surrogate Node Feature Embeddings $\mathbf{s}_u$ and $\mathbf{s}_v$. Then the non-conformity score function is given as $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}) = ||(\mathbf{s}_u|\mathbf{s}_v) - (\hat{f}(\mathbf{x}_u)|\hat{f}(\mathbf{x}_v))||$. Here $|$ denotes the concatenation operation.

2. Max: We used the maximum node deviation given as $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}) = \max(||\mathbf{s}_u - \hat{f}(\mathbf{x}_u)||, ||\mathbf{s}_u - \hat{f}(\mathbf{x}_v)||)$ as the NC score function.

3. Sum: We used the sum given as $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}) = ||\mathbf{s}_u - \hat{f}(\mathbf{x}_u)|| + ||\mathbf{s}_v - \hat{f}(\mathbf{x}_v)||$ as the NC score function.

4. Min: We used the maximum node deviation given as $V(\mathbf{x}_u, \mathbf{x}_v, y_{u,v}) = \min(||\mathbf{s}_u - \hat{f}(\mathbf{x}_u)||, ||\mathbf{s}_v - \hat{f}(\mathbf{x}_v)||)$ as the NC score function.

The results of the experiment are in Table 4. The results indicate that observing the maximum/minimum deviation between the Surrogate Node Feature Embeddings and the predicted node embeddings performs very similarly to concatenating the node features and then computing the deviation. Note that the theory of conformal inference does not guarantee conditional effectiveness, and even though Sum obtains the highest SSC for HS-PI and cond-mat, we reject it due to its high marginal efficiency. We also note that Min does better than Max and this is to be expected as the band estimator generally gives looser bands for samples. This led us to experiment with Min and Cat as the score functions for all the experiments and display the best result in Table 4.

6 Conclusion

In this work, we extend conformal inference in GNN to weighted edge prediction tasks. Our method leverages the latent node embeddings to construct a NC score. We show that our method[1] achieves the theoretical guarantees achieved by traditional conformal inference methods. Potential future directions would be to explore the validity of this method in higher-order interaction structures like hypergraphs and the connection to locally valid coverage [22].

Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344, Lawrence Livermore National Security, LLC. This work was partially funded by the CDC MInD Healthcare Network grant U01CK000594 and the associated COVID-19 supplemental funding, NSF CyberTraining grant 2320980, and the LLNL-LDRD Program under Project No. 22-SI-004 with IM release number LLNL-CONF-850978.

[1] Code and Supplement: `https://github.com/Soothysay/Edge-CP.git`

References

[1] A. Angelopoulos, S. Bates, J. Malik, and M. I. Jordan. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*, 2020.

[2] A. N. Angelopoulos, S. Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023.

[3] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

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

[5] L. Cai, J. Li, J. Wang, and S. Ji. Line graph neural networks for link prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5103–5113, 2021.

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

[7] R. Foygel Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 10(2):455–482, 2021.

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

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

[10] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[11] S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

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

[13] H. H.-H. Hsu, Y. Shen, C. Tomani, and D. Cremers. What makes graph neural networks miscalibrated? *Advances in Neural Information Processing Systems*, 35:13775–13786, 2022.

[14] K. Huang, Y. Jin, E. Candes, and J. Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.

[15] K. Huang, C. Xiao, L. M. Glass, M. Zitnik, and J. Sun. Skipgnn: predicting molecular interactions with skip-graph networks. *Scientific reports*, 10(1):21092, 2020.

[16] R. Izbicki, G. T. Shimizu, and R. B. Stern. Flexible distribution-free conditional predictive bands using density estimators. *arXiv preprint arXiv:1910.05575*, 2019.

[17] K. Jha, S. Saha, and H. Singh. Prediction of protein–protein interaction using graph neural networks. *Scientific Reports*, 12(1):8360, 2022.

[18] C. Y. Kim, S. Baek, J. Cha, S. Yang, E. Kim, E. M. Marcotte, T. Hart, and I. Lee. Humannet v3: an improved database of human gene networks for disease research. *Nucleic acids research*, 50(D1):D632–D639, 2022.

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

[20] N. Kriege and P. Mutzel. Subgraph matching kernels for attributed graphs. *arXiv preprint arXiv:1206.6483*, 2012.

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

[22] Z. Lin, S. Trivedi, and J. Sun. Locally valid and discriminative prediction intervals for deep learning models. *Advances in Neural Information Processing Systems*, 34:8378–8391, 2021.

[23] Z. Lin, S. Trivedi, and J. Sun. Conformal prediction with temporal quantile adjustments. *Advances in Neural Information Processing Systems*, 35:31017–31030, 2022.

[24] Y. L. Liu, Y. Wang, O. Vu, R. Moretti, B. Bodenheimer, J. Meiler, and T. Derr. Interpretable chirality-aware graph neural network for quantitative structure activity relationship modeling in drug discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14356–14364, 2023.

[25] R. Luo and N. Colombo. Conformal load prediction with transductive graph autoencoders. *arXiv preprint arXiv:2406.08281*, 2024.

[26] T. Mallick, P. Balaprakash, E. Rask, and J. Macfarlane. Transfer learning with graph neural networks for short-term highway traffic forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 10367–10374. IEEE, 2021.

[27] M. E. Newman. The structure of scientific collaboration networks. *Proceedings of the national academy of sciences*, 98(2):404–409, 2001.

[28] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019.

[29] H. Papadopoulos, K. Proedrou, V. Vovk, and A. Gammerman. Inductive confidence machines for regression. In *Machine learning: ECML 2002: 13th European conference on machine learning Helsinki, Finland, August 19–23, 2002 proceedings 13*, pages 345–356. Springer, 2002.

[30] H. Papadopoulos, V. Vovk, and A. Gammerman. Con-

formal prediction with neural networks. In *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*, volume 2, pages 388–395. IEEE, 2007.

[31] Y. Romano, E. Patterson, and E. Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.

[32] Y. Romano, M. Sesia, and E. Candes. Classification with valid and adaptive coverage. *Advances in Neural Information Processing Systems*, 33:3581–3591, 2020.

[33] N. Seedat, J. Crabbé, and M. van der Schaar. Datasuite: Data-centric identification of in-distribution incongruous examples. In *International Conference on Machine Learning*, pages 19467–19496. PMLR, 2022.

[34] J. J. Sutherland, L. A. O'brien, and D. F. Weaver. 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, 2003.

[35] D. Sweidan and U. Johansson. Probabilistic prediction in scikit-learn. In *The 18th International Conference on Modeling Decisions for Artificial Intelligence, Online (from Umeå, Sweden), September 27-30, 2021.*, 2021.

[36] K. Tang, D. Miao, W. Peng, J. Wu, Y. Shi, Z. Gu, Z. Tian, and W. Wang. Codes: Chamfer out-of-distribution examples against overconfidence issue. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1153–1162, 2021.

[37] J. Teng, C. Wen, D. Zhang, Y. Bengio, Y. Gao, and Y. Yuan. Predictive inference with feature conformal prediction. In *The Eleventh International Conference on Learning Representations*, 2022.

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

[39] P. Trivedi, M. Heimann, R. Anirudh, D. Koutra, and J. J. Thiagarajan. A stochastic centering framework for improving calibration in graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2023.

[40] P. Trivedi, D. Koutra, and J. J. Thiagarajan. On estimating link prediction uncertainty using stochastic centering. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6810–6814. IEEE, 2024.

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

[42] J. Wang, A. Ma, Y. Chang, J. Gong, Y. Jiang, R. Qi, C. Wang, H. Fu, Q. Ma, and D. Xu. scgnn is a novel graph neural network framework for single-cell rna-seq analyses. *Nature communications*, 12(1):1882, 2021.

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

[44] H. Werner, L. Carlsson, E. Ahlberg, and H. Boström. Evaluation of updating strategies for conformal predictive systems in the presence of extreme events. In *Conformal and Probabilistic Prediction and Applications*, pages 229–242. PMLR, 2021.

[45] S. H. Zargarbashi, S. Antonelli, and A. Bojchevski. Conformal prediction sets for graph neural networks. In *International Conference on Machine Learning*, pages 12292–12318. PMLR, 2023.

[46] S. H. Zargarbashi and A. Bojchevski. Conformal inductive graph neural networks. In *The Twelfth International Conference on Learning Representations*, 2023.

[47] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.

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