On Estimating the Epistemic Uncertainty of Graph Neural Networks using Stochastic Centering

Puja Trivedi [1] Mark Heimann [2] Rushil Anirudh [2] Danai Koutra [1] Jayaraman J. Thiagarajan [2]

Abstract

Safe deployment of graph neural networks (GNNs) under distribution shift requires models to provide accurate confidence indicators. However, while it is well-known in computer vision that CI quality diminishes under distribution shift, this behavior remains understudied for GNNs. Hence, we begin with a case-study on CI calibration under controlled structural and feature distribution shifts and demonstrate that increased expressivity or model size is not effective for improving CI performance. Consequently, we instead advocate for the use of epistemic uncertainty quantification (UQ) methods to modulate CIs and propose G-ΔUQ, a new single model UQ method that extends the recently proposed stochastic centering framework to support structured data and partial stochasticity. Evaluated across covariate, concept, and graph size shifts, G-ΔUQ not only outperforms several popular UQ methods in obtaining calibrated CIs, but also outperforms alternatives when CIs are used for generalization gap prediction or OOD detection. Overall, our work not only introduces a new, flexible GNN UQ method, but also provides novel insights into GNN CIs on safety-critical tasks.

1. Introduction

As graph neural networks (GNNs) are increasingly deployed in critical applications with test-time distribution shifts (Zhang & Chen, 2018; Gaudelet et al., 2020; Yang et al., 2018; Yan et al., 2019; Zhu et al., 2022), it becomes necessary to expand model evaluation to include safety-centric metrics, such as as calibration errors (Guo et al., 2017), out-of-distribution (OOD) rejection rates (Hendrycks & Gimpel, 2017), and generalization gap estimates (Jiang et al., 2019), to holistically understand model performance under such shifted regimes (Hendrycks et al., 2022b; Trivedi et al., 2023b). Notably, such additional metrics often rely upon on *confidence indicators* (CI), such as maximum softmax or predictive entropy, which can derived from prediction probabilities. However, while it is well-known in computer vision that CI quality can significantly deteriorate under distribution shifts (Wiles et al., 2022; Ovadia et al., 2019) and that factors such as model size or expressitivity can significantly affect this deterioration (Minderer et al., 2021), this behavior remains less-explored for GNNs.

Indeed, there is an expectation that more advanced or expressive architectures (Chuang & Jegelka, 2022; Alon & Yahav, 2021; Topping et al., 2022; Rampášek et al., 2022; Zhao et al., 2022) would improve GNN CI calibration on graph classification tasks. Yet, we find that using graph transformers (Rampášek et al., 2022) or positional encodings (Dwivedi et al., 2022; Wang et al., 2022b; Li et al., 2020) does not significantly improve CI calibration over vanilla message-passing GNNs under controlled, label-preserving distribution shifts. Notably, when CIs are not well-calibrated, GNNs with high accuracy may perform poorly on the additional safety metrics, leading to unseen risks during deployment. Given that using advanced architectures is not an immediately viable solution for improving CI calibration, we instead adovocate for modulating CIs using epistemic *uncertainty estimates*.

Uncertainty quantification (UQ) methods (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017; Blundell et al., 2015) have been extensively studied for vision models (Guo et al., 2017; Minderer et al., 2021), and have been used to improve vision model CI performance under distribution shifts. Our work not only studies the effectiveness of such methods on improving GNN CIs, but also proposes a novel UQ method, G-ΔUQ, which extends the recently proposed, state-of-the-art stochastic data-centering or "anchoring" framework (Thiagarajan et al., 2022; Netanyahu et al., 2023) to support partial stochasticity and structured data. In brief, stochastic centering provides a scalable alternative to highly effective, but prohibitively expensive deep ensembles (DeepEns) by efficiently sampling a model's hypothesis space, in lieu of training multiple, independently

[1] CSE Department, University of Michigan, Ann Arbor [2] Lawrence Livermore National Laboratory. Correspondence to: Puja Trivedi <pujat@umich.edu>.

DMLR Workshop at International Conference on Machine Learning, Copyright 2023 by the author(s).

trained models. When using the uncertainty-modulated confidence estimates from G-ΔUQ, we outperform other popular UQ methods, on not only improving the CI calibration under covariate, concept and graph size shifts, but also in improving generalization gap prediction and OOD detection performance.

Proposed Work. This work studies the effectiveness of GNN CIs on the graph classification tasks with distribution shifts, and proposes, a novel uncertainity-based method for improving CI performance. Our contributions can be summarized as follows:

Sec. 3: Case Study on CI Calibration. We find that improving GNN expressivity does not mitigate CI quality degradation under distribution shifts.

Sec. 4: (Partially) Stochastic Anchoring for GNNs. We propose G-ΔUQ, a novel, flexible stochastic UQ method, that extends stochastic centering to support GNNs and partial stochasticity.

Sec. 5: Evaluating Uncertainity-Modulated CIs under Distribution Shifts. Across covariate, concept and graph-size shifts and evaluation protocols (calibration, OOD rejection, generalization gap prediction), we demonstrate the effectiveness of G-ΔUQ.

2. Preliminaries

In this section, we introduce several tasks that rely upon reliable confidence indicators and discuss recent efforts to improve them.

Notations. Let $\mathcal{G} = (\mathbf{X}, \mathbf{E}, \mathbf{A}, Y)$ be a graph with node features $\mathbf{X} \in \mathbb{R}^{N \times d_\ell}$, (optional) edge features $\mathbf{E} \in \mathbb{R}^{m \times d_\ell}$, adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, and graph-level label $Y \in \{0,1\}^c$, where N, m, d_ℓ, c denote the number of nodes, number of edges, feature dimension and number of classes, respectively. We use i to index a particular sample in the dataset, e.g. $\mathcal{G}_i, \mathbf{X}_i$.

Then, we can define a graph neural network consisting of ℓ message passing layers (MPNN), a graph-level readout function (READOUT), and classifier head (MLP) as follows:

$$\mathbf{X}_M^{\ell+1},\, \mathbf{E}^{\ell+1} = \texttt{MPNN}_e^{\ell}\left(\mathbf{X}^{\ell}, \mathbf{E}^{\ell}, \mathbf{A}\right), \tag{1}$$

$$\mathbf{G} = \texttt{READOUT}\left(\mathbf{X}_M^{\ell+1}\right), \tag{2}$$

$$\hat{Y} = \texttt{MLP}\left(\mathbf{G}\right), \tag{3}$$

where $\mathbf{X}_M^{\ell+1}, \mathbf{E}^{\ell+1}$ are intermediate node and edge representations, and $\mathbf{G}$ is the graph representation. We focus on a graph classification setting throughout our paper.

2.1. Using Confidence Estimates in Safety Critical Tasks

The safe deployment of graph machine learning models in critical applications requires that GNNs not only generalize to ID and OOD datasets, but that they do so safely. To this end, recent works (Hendrycks et al., 2022b; 2021; Trivedi et al., 2023b) have expanded model evaluation to include additional robustness metrics to provide a holistic view of model performance. Notably, while reliable confidence indicators are critical to success on these metrics, the impact of distributions shift on GNN confidence estimates remains under-explored. Below, we introduce the different robustness metrics that we use to understand their behavior.

Calibration: Calibrated models should provide confidence estimates such that they match the true probabilities of the classes being predicted (Naeini et al., 2015; Guo et al., 2017; Ovadia et al., 2019). Poorly calibrated models are over/under confident, making it difficult to trust their predictions. By computing the top-1 label expected calibration error (ECE) (Kumar et al., 2019; Detlefsen et al., 2022), we can directly evaluate the quality of GNN confidence indicators as follows. Let p_i be the top-1 probability, c_i be the predicted confidence, b_i a uniformly sized bin in [0,1]. Then, $ECE := \sum_i^N b_i \|(p_i - c_i)\|$.

Generalization Error Prediction: Accurate estimation of the expected generalization error on unlabeled datasets allows models with unacceptable performance to be pulled from production. To this end, generalization error predictors (GEPs) (Garg et al., 2022; Ng et al., 2022; Jiang et al., 2019; Trivedi et al., 2023a; Guillory et al., 2021) which assign sample-level scores, $S(x_i)$ which are then aggregated into dataset-level error estimates, have become popular. We use maximum softmax probability and a simple thresholding mechanism as the GEP (since we are interested in understanding the behavior of confidence indicators), and report the error between the predicted and true target dataset accuracy: $GEPError := ||\text{Acc}_{target} - \frac{1}{|X|}\sum_i \mathbb{I}(\mathrm{S}(\bar{\mathrm{x}}_i; \mathrm{F}) > \tau)||$, where τ is tuned by minimizing GEP error on the validation dataset.

Out-of-Distribution Detection: By reliably detecting OOD samples and abstaining from making predictions, models can avoid over extrapolating to distributions which are not relevant. While many scores have been proposed for detection (Hendrycks et al., 2019; 2022a; Lee et al., 2018; Wang et al., 2022a; Liu et al., 2020), flexible, popular baselines, such as maximum softmax probability and predictive entropy (Hendrycks & Gimpel, 2017), can be derived from confidence indicators relying upon prediction probabilities. Here, we report the AUROC for the binary classification task of detecting OOD samples using the maximum softmax probability (Kirchheim et al., 2022).

We briefly note that while more sophisticated scores can be used, our focus is on the reliability of GNN confidence indicators and thus we choose scores directly related to those estimates. Moreover, since sophisticated scores can often be derived from prediction probabilities, we expect their performance would also be improved with better estimates.

2.2. Improving Confidence Indicators

While success on the above tasks requires reliable prediction confidence indicators, it is well-known in computer vision that such estimates are often unreliable or uncalibrated directly out-of-the-box (Guo et al., 2017), especially under distribution shifts (Ovadia et al., 2019; Wiles et al., 2022; Hendrycks et al., 2019). To this end, many strategies have been proposed to improve calibration (Lakshminarayanan et al., 2017; Guo et al., 2017; Gal & Ghahramani, 2016; Blundell et al., 2015). We note that such strategies might also help GEPs and OOD detectors as the scores are more informative. One particularly effective strategy is to create a Deep Ensemble (DEns) (Lakshminarayanan et al., 2017) by training a set of independent models (e.g., different hyper-parameters, random-seeds, data order, etc) where the mean predictions over the set is noticeably better calibrated. However, since DEns requires training multiple models, in practice, it can be prohibitively expensive to use. To this end, we focus on single model strategies.

Single model strategies attempt to reliably estimate uncertainty in a scalable way, which can then optionally be used to re-calibrate the prediction probabilities. Here, the intuition is that when the epistemic uncertainties are large in a data regime, confidence estimates can be tempered so that they better reflect the accuracy degradation during extrapolation (e.g., training on small-sized graphs but testing on large-sized graphs). Some popular strategies include: Monte Carlo dropout (MCD) (Gal & Ghahramani, 2016) which performs Monte Carlo dropout at inference time and takes the average prediction to improve calibration, temperature scaling (Temp) (Guo et al., 2017) which rescales logits using a temperature parameters computed from a validation set, and Blundell et al. which proposes a variational method for estimating uncertainty. While such methods are more scalable than DeepEns, in many cases, they do struggle to match its performance (Ovadia et al., 2019). However, the recently proposed stochastic centering paradigm is able to simulate an ensemble by sampling from different hypotheses using anchoring. We introduce this paradigm in detail below as we will be extending it to accommodate both structured, discrete graph data, and partial stochasticity (see Sec. 4).

2.3. Stochastic Centering for Uncertainty Quantification

In a recent work (Thiagarajan et al., 2022), it was found that applying a (random) constant bias to vector-valued (and image) data leads to non-trivial changes in the resulting solution. This behavior was attributed to the lack of shift-invariance in the neural tangent kernel (NTK) induced by conventional neural networks such as MLPs and CNNs. Building upon this observation, Thiagarajan et al. proposed a single model uncertainty estimation method, Δ-UQ, based on the principle of *anchoring*. Conceptually, anchoring is the process of creating a relative representation for an input sample x in terms of a random anchor c (which is used to perform the *stochastic centering*), $[c, x - c]$. By choosing different anchors randomly in each iteration, Δ-UQ emulates the process of sampling different solutions from the hypothesis space (akin to an ensemble). During inference time, for a test sample, it aggregates multiple predictions obtained via different random anchors (same anchor distribution as training) and produces uncertainty estimates. Further, another recent study (Netanyahu et al., 2023), suggests that anchoring can also be utilized to improve the extrapolation behavior of deep neural networks. While an attractive paradigm, there are several challenges to using stochastic centering and anchoring for GNNs/graph data. We discuss and remedy these in Sec. 4, and also propose several partially stochastic variants.

3. Case Study on GNN Calibration

In this section, we perform a motivational study on the calibration of widely-adopted GNN architectures. We consider a simple, structural distribution shift on a standard benchmark (Dwivedi et al., 2020) to emphasize that reliable prediction confidence remains an important, open problem of study despite improvements in architectures (He et al., 2022; Corso et al., 2020; Zhao et al., 2022) and expressivity (Wang et al., 2022b; Dwivedi et al., 2022).

Experimental Set-up: Data. Superpixel-MNIST (Dwivedi et al., 2020; Knyazev et al., 2019; Velickovic et al., 2018) is a popular graph benchmark that converts MNIST images into k nearest-neighbor graphs of superpixels (Achanta et al., 2012). We select this benchmark as it allows for (i) a diverse set of well-trained models without requiring independent, extensive hyper-parameter search and (ii) controlled, label preserving, structural distortion distribution shifts. Inspired by (Ding et al., 2021), we create structurally distorted but valid graphs by rotating MNIST images by a fixed number of degrees and then creating the super-pixel graphs from these rotated images. Since superpixel segmentation on these rotated images will yield different superpixel k-nn graphs but not harm class information, we can create label-preserving structural distortion shifts. Indeed, models are trained only on the original (without any rotation) graphs.

Experimental Set-up: Models. While improving the expressivity of GNNs is an active area of research, positional encodings and graph-transformer architectures have proven to be particularly popular due to their effectiveness, and flexibility. Here, we consider the effects of (i) incorporating equivariant and stable positional encodings (Wang et al., 2022b) (ii) utilizing message passing vs. graph transformer architectures and (iii) changing depth/width on calibration

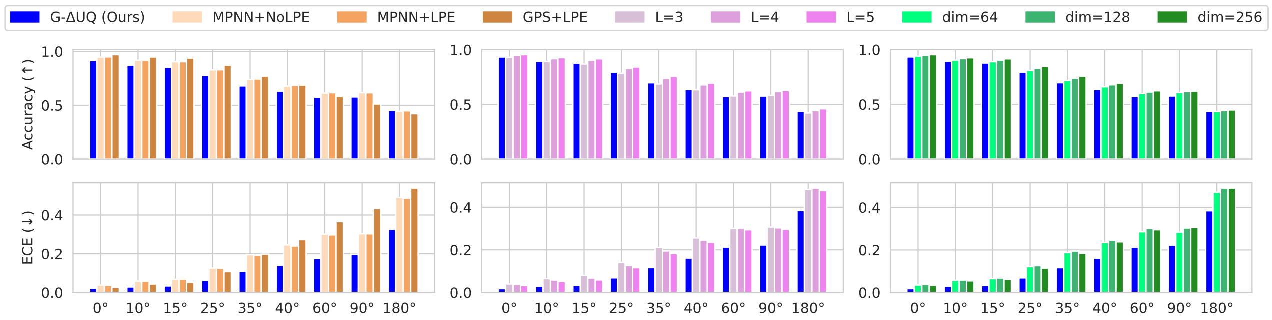

Figure 1. **Calibration on Structural Distortion Distribution Shifts.** On a controlled graph structure distortion shift, we evaluate models trained on the standard superpixel MNIST benchmark (Dwivedi et al., 2020) on super-pixel k-nn graphs created from rotated MNIST images. While accuracy is expected to decrease as distribution shift increases, we observe that the expected calibration error also grows significantly worse. Importantly, this trend is persistent when considering transformer architectural variants (GPS (Rampášek et al., 2022)), as well as different depths and widths. In contrast, our proposed G-ΔUQ method achieves substantial improvement in ECE without significantly compromising on accuracy.

under distribution shift. These parameters are considered with respect to the GatedGCN backbone (Dwivedi et al., 2020). For the graph transformer, we utilize the recently state-of-the-art, flexible GPS architecture (Rampášek et al., 2022).

Observations. In Fig. 1, we present our results and make the following observations.

We begin by noting that graph transformer architectures have shown better handling of over-smoothing (a phenomenon where graph neural networks lose discriminative power) and over-squashing (a phenomenon where graph neural networks collapse node representations) (Alon & Yahav, 2021; Topping et al., 2022). However, when it comes to the task of obtaining calibrated predictions under distribution shifts, GPS (“general, powerful, scalable") graph transformer performs noticeably worse compared to its message-passing variants, despite having comparable accuracies. This is apparent particularly at high degrees (60°, 90°, 180°). Furthermore, we find that positional encodings have minimal effects on both calibration and accuracy under distribution shift. This suggests that while these techniques may enhance theoretical and empirical expressivity, they do not necessarily transfer to the safety-critical task of obtaining calibrated predictions under distribution shifts. In addition, we investigate the impact of model depth and width on calibration performance, considering that model size has been known to affect the calibration of vision models (Guo et al., 2017; Minderer et al., 2021) and the propensity for over-squashing in graph neural networks (Xu et al., 2021). Our observations reveal that increasing the number of message passing layers ($L = 3 \to L = 5$) can improve accuracy, but it may also marginally decrease calibration error. Moreover, we find that increasing the width of the model can lead to slightly worse calibration at high levels of shift (90°, 180°), although accuracy scores are not compromised.

Notably, when we apply our proposed method G-ΔUQ, (see Sec. 4), to the simple message-passing backbone with no positional encodings, it significantly improves the calibration over more expressive variants (GPS, LPE), across all levels of distribution shifts, while maintaining comparable accuracy. We briefly note that we did not tune the hyperparameters to our method to ensure a fair comparison, so expect that accuracy could be improved with tuning. Overall, our results emphasize that effective uncertainty-based prediction calibration remains a difficult problem that cannot be easily solved through advancements in architectures and expressivity.

4. Graph-ΔUQ: Uncertainty-based Prediction Calibration

Motivated by the calibration study conducted on various GNN architectures in the previous section, we propose to perform uncertainty-based calibration of prediction probabilities in GNNs. To accomplish this objective, we will extend the recently introduced $\Delta-$UQ model to graph-structured data. By doing so, we will illustrate how this extended model can effectively enhance the reliability of confidence indicators, even in the presence of difficult distribution shifts.

As mentioned in Section 2, the concept of anchoring has proven to be effective in training deep models with enhanced extrapolation capabilities (Netanyahu et al., 2023). Additionally, it has shown promise in facilitating single model uncertainty estimation (Thiagarajan et al., 2022). However, previous research has primarily focused on traditional vision models and relied on straightforward input space transformations to construct anchored representations. Moreover, the distribution shifts encountered in graph datasets exhibit distinct characteristics compared to those typically examined in the vision literature. Hence, it becomes imperative to gain new insights into uncertainty quantification for GNNs.

Graph datasets possess structured, discrete, and variable-size characteristics, making it extremely challenging to devise appropriate anchors that are capable of effectively sampling the underlying model hypothesis space. Consequently, in this section, we not only provide a conceptual introduction to the application of stochastic centering in the context of GNNs but also explore the potential benefits of incorporating partial stochasticity and pretraining to further improve the performance of anchored GNNs. In Section 5, we present empirical evidence to substantiate the advantages of this approach.

4.1. Node Feature Anchoring

We begin by reminding that in the Δ-UQ framework, input samples are transformed into an anchored representation, where the anchors are randomly drawn from the training dataset itself. During inference, we marginalize the predictions over multiple anchor choices to obtain mean and uncertainty estimates. While this is easy to implement for vector valued data or images, due to the variability and discreteness of graph sizes, performing a residual operation on the anchor/query pair $\mathbf{A}$ would introduce artificial edge weights and connectivity artifacts. To this, we create anchors using node features as an input-space analog to the subtraction and channel-concatenation operation in Δ-UQ.

To accomplish this, we first draw anchors from a Gaussian distribution ($\mathcal{N}(\mu, \sigma)$) fitted to the training node features. During training, we randomly sample an anchor for each node. Mathematically, given the anchor $c \sim \mathcal{N}(\mu, \sigma)$, we create the anchor/query node feature pair $[\mathbf{X}_i || \mathbf{X}_i - c_i]$, where $||$ denotes concatenation. During inference, we sample a fixed set of k anchors and compute residuals for all nodes with respect to the same anchor. For datasets with categorical node features, it is more beneficial to perform the anchoring operation after embedding the node features in a continuous space. Alternatively, considering the advantages of positional encodings in enhancing model expressivity (Wang et al., 2022b), one can compute positional information for each node and perform anchoring based on these encodings. While using only node features for anchoring neglects information about the underlying structure, incorporating positional encodings provides a straightforward approach to include some form of relational priors in anchor construction.

4.2. Hidden Layer Anchoring

While our G-ΔUQ approach can be used with any GNN architecture, we explore three different variants for improving both the flexibility and scalability of stochastic centering-based UQ. Intuitively, performing anchoring in the input space creates a fully stochastic neural network as all parameters are learned using the randomized input, and it emulates the process of sampling different solutions from a hypothesis space. However, recent evidence with Bayesian neural networks shows that relaxing the assumption of fully stochastic neural networks and defining partially stochastic models (Sharma et al., 2023) leads to strong computational benefits, and in many cases, improved calibration performance. Motivated by this observation, we propose to extend the family of functions supported in G-ΔUQ by anchoring in intermediate layers, in lieu of the inputs. This allows for *partially stochastic* models, wherein the layers prior to the anchoring step are deterministic. Moreover, this intermediate anchoring has the additional benefit that anchors will be able to sample hypotheses that consider both topological and node feature information due to `MPNN` steps.

Intermediate MPNN Anchoring: Given a GNN containing ℓ `MPNN` layers, let $k \leq \ell$ be the layer at which we perform node feature anchoring. We obtain the anchor/query pair by computing the intermediate node representations from the first k `MPNN` layers. We then randomly shuffle the node features over the entire batch, concatenate the residuals, and proceed with the `READOUT` and `MLP` layers as with the standard $\Delta-$UQ model. Note that, we do not consider the gradients of the query sample when updating the parameters, and the $\texttt{MPNN}^{k+1}$ layer is modified to accept inputs of dimension $d_\ell \times 2$ (to take in anchored representations as inputs). Another difference from the input space implementation is that, we fix the set of anchors and subtract a single anchor from all node representations in an iteration (instead of sampling uniquely).

Intermediate Read Out Anchoring: While `READOUT` anchoring is conceptually similar to intermediate `MPNN` anchoring, we now only obtain a different anchor for each hidden graph representation, instead of individual nodes. This allows us to sample hypotheses after all node information has been aggregated.

Pretrained Anchoring: Lastly, we note that, in order to be compatible with the stochastic centering framework (the input layer or chosen intermediate layer), one needs to fully redesign the network architecture and retrain from scratch. To circumvent this, we consider a variant of `READOUT` anchoring with a pretrained GNN backbone. Here, the final `MLP` layer of a pretrained model is discarded, and reinitialized to accommodate query/anchor pairs. We then freeze the `MPNN`, and only train the anchored classifier head. This allows for an inexpensive, limitedly stochastic GNNs (see Sec. 5.2).

While these variants can lead to vastly different uncertainty estimates, the complexity of the task and the nature of the

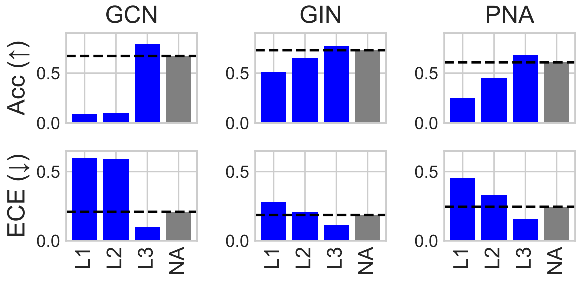

Figure 2. **Impact of Layer Selection on G-ΔUQ.** Performing anchoring at different layers leads the sampling of different hypothesis spaces. On D&D, we see that later layer anchoring corresponds to a better inductive bias and can lead to dramatically improved performance.

distribution shift will determine which of the variants is best suited.

5. Uncertainty-based Prediction Calibration under Distribution Shift

In this section, we conduct experiments on size generalization (Sec. 5.1) and the recently proposed, Graph-Out-of-Distribution (GOOD) benchmark to rigorously study uncertainty estimation of GNNs under distribution shifts and the benefits of stochastic anchoring.

5.1. Size Generalization

While GNNs are well-known to struggle when generalizing to larger size graphs (Buffelli et al., 2022; Yehudai et al., 2021; Chen et al., 2022), their predictive uncertainty behavior with respect to such shifts remains under studied. Given that such shifts can be expected at deployment, reliable uncertainty estimates under this setting are important for safety critical applications. We note that while sophisticated training strategies can be used to improve size generalization (Buffelli et al., 2022; Bevilacqua et al., 2021), our focus is primarily on the quality of uncertainity estimates, so we do not consider such techniques. However, we note that G-ΔUQ can be used in conjunction with such techniques.

Experimental Set-up. Following the procedure of (Buffelli et al., 2022; Yehudai et al., 2021), we create a size distribution shift by taking the smallest 50%-quantile of graph size for the training set, and reserving the larger quantiles (>50%) for evaluation. Unless, otherwise noted, we report results on the largest 10% quantile to capture performance on the largest shift. We utilize this splitting procedure on four well-known benchmark binary graph classification datasets from the TUDataset repository (Morris et al., 2020): D&D, NCI1, NCI09, and PROTEINS. Please see the Supplementary for dataset statistics. We consider three different backbone GNN models, GCN (Kipf & Welling, 2017), GIN (Xu et al., 2019), and PNA (Corso et al., 2020) and report their performance with and without stochastic anchoring. All models contain three message passing layers and the same sized hidden representation.

Results. As noted in Sec. 4, stochastic anchoring can be applied at different layers, leading to the sampling of different hypothesis spaces and inductive biases. In order to empirically understand this behavior, we compare the performance of stochastic centering when applied at different layers on the D&D dataset, which comprises the most severe size shift from training to test set (see Fig. **??**). We observe that applying stochastic anchoring after the `READOUT` layer (L3) dramatically improves both accuracy and calibration as the depth increases. While this behavior is less pronounced on other datasets (see Supplementary), we find overall that applying stochastic anchoring at the last layer yields competitive performance on size generalization benchmarks and better convergence compared to stochastic centering performed at earlier layers.

Indeed, in Fig. 3, we compare the performance of last-layer anchoring against the baseline model on 4 datasets. We observe that G-ΔUQ improves calibration performance on most datasets, while generally maintaining or even improving the accuracy. Indeed, improvement is most pronounced on the largest shift (D&D), further emphasizing the benefits of stochastic centering.

5.2. Evaluation under Concept and Covariate Shifts

Conventional neural networks are well-known to behave unpredictably under different types of distribution shifts. Therefore, in this section, we seek to understand behavior of GNN predictive uncertainity under controlled covariate and concept shifts. Moreover, we expand our evaluation to include the utility of predictive uncertainties in the OOD detection (Hendrycks & Gimpel, 2017; Hendrycks et al., 2019) and generalization gap prediction tasks (Guillory et al., 2021; Ng et al., 2022; Trivedi et al., 2023a; Garg et al., 2022). We begin by introducing our data and additional tasks, and then present our results.

Experimental Set-up: Data In brief, concept shift corresponds to a change in the conditional distribution of labels given input from the training to evaluation datasets, while covariate shift corresponds to change in the input distribution. We use the recently proposed Graph Out-Of Distribution (GOOD) benchmark (Gui et al., 2022) to obtain four different datasets (GOODCMNIST, GOODMotif-basis, GOODMotif-size, GOODSST2) with their corresponding in-/out- of distribution concept and covariate splits. To ensure fair comparison, we use the architectures and hyperparameters suggested by the benchmark when training. Please see the supplementary for more details.

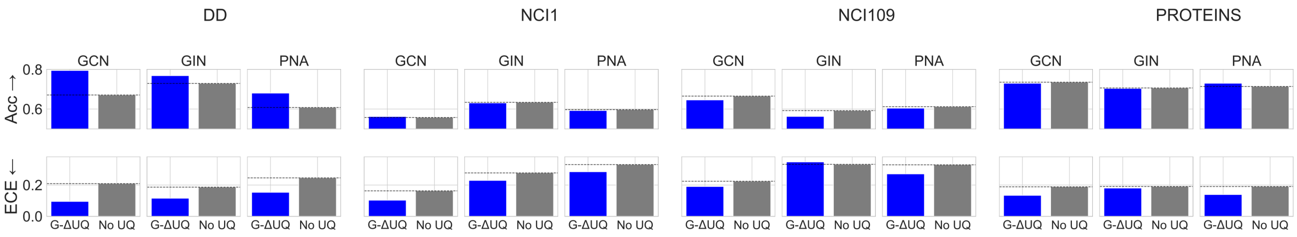

Figure 3. **Predictive Uncertainty under Size Distribution Shifts.** When evaluating the accuracy and calibration error of models trained with and without stochastic anchoring on dataset with a graph size distribution shift, we observe that stochastic centering decreases calibration error while improving or maintaining accuracy across datasets and different GNNs.

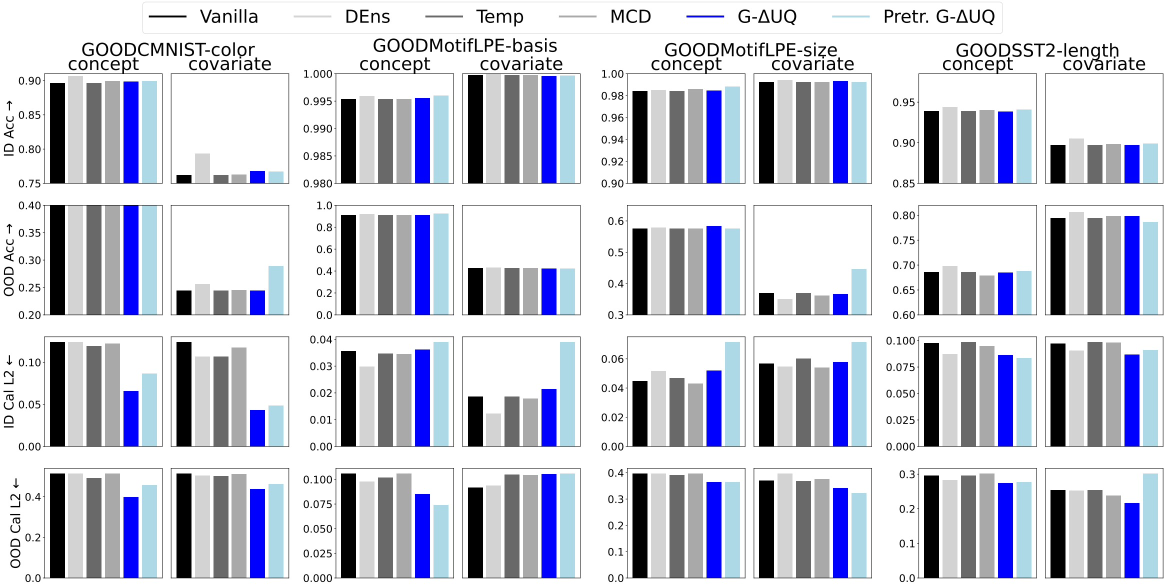

Figure 4. **Predictive Uncertainity under Concept and Covariate Shifts.** Stochastic anchoring leads to competitive in-distribution and out-of distribution test accuracy while improving calibration, across domains and shifts. This is particularly true when comparing to other single-model UQ methods.

Baselines: We consider the following methods in our analysis: Deep Ensembles (Lakshminarayanan et al., 2017), Monte Carlo Dropout (MCD) (Gal & Ghahramani, 2016), and our proposed G-ΔUQ, including the pretrained variant. DeepEns is well known to be a performative baseline on uncertainty estimation tasks, but we emphasize that it requires training multiple models. This is in contrast to single model estimators, such as MCD and G-ΔUQ. We note that while MCD and G-ΔUQ can be applied at intermediate layers; we present results on the best performing layer but include the full results in the supplementary.

Tasks: In addition to reporting expected calibration error, we also report the mean absolute error obtained when attempting to predict the generalization accuracy, and the AUROC when attempting to detect out-of-distribution samples. To estimate the generalization error, we use the confidences obtained by the different baselines as sample-level scores, $\mathrm{S}(\mathrm{x}_i)$ corresponding to the model's expectation that a sample is correct. We then threshold and aggregate sample-level scores to obtain a dataset-level estimate: $\frac{1}{|X|}\sum_i \mathbb{I}(\mathrm{S}(\mathrm{x_i}) > \tau)$, where the threshold hyperparameter τ is identified by training a regressor to recover the true accuracy on a pre-defined, validation dataset. We report the MAE between the estimated error and true error on both in- and out-of -distribution test splits provided by the GOOD benchmark. For OOD detection, we create a binary classification task where the objective to correctly reject OOD test splits from the benchmark.

Results: *Accuracy & Calibration.* We notice that using stochastic anchoring via G-ΔUQ yields competitive accuracy, especially in comparison to other single-model methods such as MCD, temperature scaling, or the base GNN model: in-distribution accuracy is higher on 6 out of 8 dataset/shift combinations, and out-of-distribution accuracy is higher on 5 out of 8 combinations. While Deep Ensembles is the most accurate method on a majority of datasets,

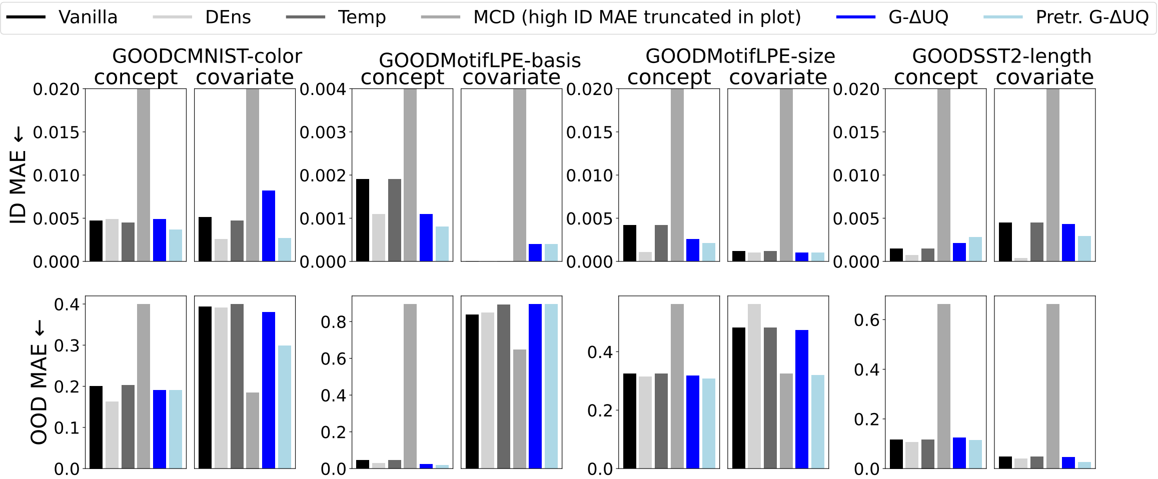

Figure 5. **Generalization Gap Prediction**.The mean absolute error when using scores obtained from different baselines in the challenging (and to the best of our knowledge, yet unexplored for graphs) task of generalization error prediction are reported. While there is not a dominant method, stochastic anchoring is very competitive, and yields among the lowest MAE of single-model UQ estimators. Notably, pretrained G-ΔUQ is particularly effective and outperforms the end-to-end variant.

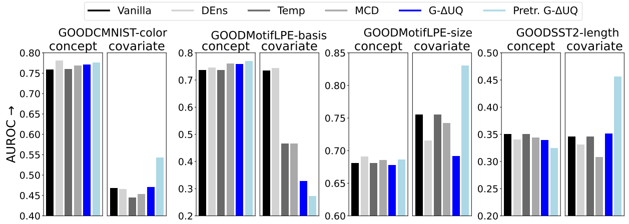

Figure 6. **OOD Detection.** The AUROC is reported for the task of detecting out-of-distribution samples. Under concept shift, the proposed G-ΔUQ variants are very competitive with other baselines, including DeepEns. Under covariate shifts, except for GOODMotif-basis, pretrained G-ΔUQ produces significant improvements over all baselines, including end-to-end G-ΔUQ training.

they are known to be computationally expensive. Moreover, the simpler stochastic anchoring procedure generally comes close to the accuracy of Deep Ensembles, and in a few cases (covariate shift on GOODCMNIST and GOODMotif-size datasets), can noticeably outperform it. Stochastic anchoring also excels in improving calibration, improving in-distribution calibration compared to all baselines on 4 out of 8 combinations. ***Most importantly, out-of-distribution calibration error is decreased by stochastic anchoring on 7 of 8 dataset/shift combinations compared to* all *other methods (single-model or ensemble).***

Results: Generalization Gap Prediction. Next, we study all of our methods on the GOOD benchmarks for the task of generalization gap prediction, and report the results in Fig. 5. On this challenging task, there is no clear winner across all benchmarks. However, stochastic anchoring methods are consistently competitive in MAE, and yield among the lowest MAE (across the board lower than other single-model UQ methods). In particular, ***the pretrained G-ΔUQ variant produces on average the lowest MAE for generalization gap estimation.***

Results: OOD Detection. Finally, we consider the task of detecting out-of-distribution samples. In Fig. 6, we see that the performance of stochastic anchoring methods under concept shift is generally very competitive with other UQ methods. For covariate shifts, except for the case of GOODMotif-basis use-case, stochastic anchoring produces high AUROC scores. In particular, on the GOODCMNIST-color, GOODSST2-length and GOODMotif-size benchmarks, the pretrained variant of G-ΔUQ produces significantly improved AUROC scores. Finally, on GOODMotif-basis, however, both have lower AUROC than other baselines; we suspect the reason for this to be the inherent simplicity of this dataset and that G-ΔUQ was prone to shortcuts.

6. Conclusion

In this work, we take a closer look at confidence estimation under distribution shifts in the context of graph neural networks. We begin by demonstrating that techniques for improving GNN expressivity, such as transformer architectures and using positional encodings, do not necessarily improve the estimation performance on a simple structural distortion shift benchmark. To this end, we seek to improve the uncertainty estimation of GNNs by adapting the principle of stochastic anchoring for discrete, structured settings. We propose several G-ΔUQ variants, and demonstrate the benefits of partial stochasticity when estimating uncertainty. Our evaluation is extensive, spanning multiple types of distribution shift (size, concept, covariate) while considering multiple safety critical tasks that require reliable estimates (calibration, generalization gap prediction, and OOD detection.) The proposed G-ΔUQ improves estimation performance on a number of tasks, while remaining scalable. Overall, our paper rigorously studies uncertainty estimation for GNNs, identifies several shortcomings in existing approaches and proposes a flexible framework for reliable estimation.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012.

Alon, U. and Yahav, E. On the bottleneck of graph neural networks and its practical implications. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Bevilacqua, B., Zhou, Y., and Ribeiro, B. Size-invariant graph representations for graph classification extrapolations. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2021.

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2015.

Buffelli, D., Liò, P., and Vandin, F. Sizeshiftreg: a regularization method for improving size-generalization in graph neural networks. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2022.

Chen, Y., Zhang, Y., Bian, Y., Yang, H., Ma, K., Xie, B., Liu, T., Han, B., and Cheng, J. Learning causally invariant representations for out-of-distribution generalization on graphs. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2022.

Chuang, C.-Y. and Jegelka, S. Tree mover's distance: Bridging graph metrics and stability of graph neural networks. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2022.

Corso, G., Cavalleri, L., Beaini, D., Liò, P., and Velickovic, P. Principal neighbourhood aggregation for graph nets. In *NeurIPS*, 2020.

Detlefsen, N. S., Borovec, J., Schock, J., Harsh, A., Koker, T., Liello, L. D., Stancl, D., Quan, C., Grechkin, M., and Falcon, W. Torchmetrics - measuring reproducibility in pytorch, 2022. URL https://github.com/Lightning-AI/torchmetrics.

Ding, M., Kong, K., Chen, J., Kirchenbauer, J., Goldblum, M., Wipf, D., Huang, F., and Goldstein, T. A closer look at distribution shifts and out-of-distribution generalization on graphs. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. *CoRR*, 2020.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2016.

Garg, S., Balakrishnan, S., Lipton, Z. C., Neyshabur, B., and Sedghi, H. Leveraging unlabeled data to predict out-of-distribution performance. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Gaudelet, T., Day, B., Jamasb, A. R., Soman, J., Regep, C., Liu, G., Hayter, J. B. R., Vickers, R., Roberts, C., Tang, J., Roblin, D., Blundell, T. L., Bronstein, M. M., and Taylor-King, J. P. Utilising graph machine learning within drug discovery and development. *CoRR*, abs/2012.05716, 2020.

Gui, S., Li, X., Wang, L., and Ji, S. GOOD: A graph out-of-distribution benchmark. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS), Benchmark Track*, 2022.

Guillory, D., Shankar, V., Ebrahimi, S., Darrell, T., and Schmidt, L. Predicting with confidence on unseen distributions. In *ICCV*, 2021.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proc. of the Int. Conf. on Machine Learning, (ICML)*, 2017.

He, X., Hooi, B., Laurent, T., Perold, A., LeCun, Y., and Bresson, X. A generalization of vit/mlp-mixer to graphs. *CoRR*, abs/2212.13350, 2022.

Hendrycks, D. and Gimpel, K. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017.

Hendrycks, D., Mazeika, M., and Dietterich, T. G. Deep anomaly detection with outlier exposure. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2019.

Hendrycks, D., Carlini, N., Schulman, J., and Steinhardt, J. Unsolved problems in ML safety. *CoRR*, abs/2109.13916, 2021.

Hendrycks, D., Basart, S., Mazeika, M., Zou, A., Kwon, J., Mostajabi, M., Steinhardt, J., and Song, D. Scaling out-of-distribution detection for real-world settings. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2022a.

Hendrycks, D., Zou, A., Mazeika, M., Tang, L., Li, B., Song, D., and Steinhardt, J. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

Jiang, Y., Krishnan, D., Mobahi, H., and Bengio, S. Predicting the generalization gap in deep networks with margin distributions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

Kirchheim, K., Filax, M., and Ortmeier, F. Pytorch-ood: A library for out-of-distribution detection based on pytorch. In *Workshop at the Proc. Int. Conf. on Computer Vision and Pattern Recognition CVPR*, 2022.

Knyazev, B., Taylor, G. W., and Amer, M. R. Understanding attention and generalization in graph neural networks. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2019.

Kumar, A., Liang, P., and Ma, T. Verified uncertainty calibration. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2019.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2017.

Lee, K., Lee, K., Lee, H., and Shin, J. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2018.

Li, P., Wang, Y., Wang, H., and Leskovec, J. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2020.

Liu, W., Wang, X., Owens, J. D., and Li, Y. Energy-based out-of-distribution detection. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2020.

Minderer, M., Djolonga, J., Romijnders, R., Hubis, F., Zhai, X., Houlsby, N., Tran, D., and Lucic, M. Revisiting the calibration of modern neural networks. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2021.

Morris, C., Kriege, N. M., Bause, F., Kersting, K., Mutzel, P., and Neumann, M. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond (GRL+ 2020)*, 2020. URL www.graphlearning.io.

Naeini, M. P., Cooper, G. F., and Hauskrecht, M. Obtaining well calibrated probabilities using bayesian binning. In *Proc. Conf. on Adv. of Artificial Intelligence (AAAI)*, 2015.

Netanyahu, A., Gupta, A., Simchowitz, M., Zhang, K., and Agrawal, P. Learning to extrapolate: A transductive approach. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Ng, N., Hulkund, N., Cho, K., and Ghassemi, M. Predicting out-of-domain generalization with local manifold smoothness. *CoRR*, abs/2207.02093, 2022.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2019.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a General, Powerful, Scalable Graph Transformer. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2022.

Sharma, M., Farquhar, S., Nalisnick, E., and Rainforth, T. Do bayesian neural networks need to be fully stochastic? In *AISTATS*, 2023.

Thiagarajan, J. J., Anirudh, R., Narayanaswamy, V., and Bremer, P.-T. Single model uncertainty estimation via stochastic data centering. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2022.

Topping, J., Giovanni, F. D., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. In *Proc. Int. Conf. on Learning Representations ICLR*, 2022.

Trivedi, P., Koutra, D., and Thiagarajan, J. J. A closer look at scoring functions and generalization prediction. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023a.

Trivedi, P., Koutra, D., and Thiagarajan, J. J. A closer look at model adaptation using feature distortion and simplicity bias. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023b.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *ICLR*, 2018.

Wang, H., Li, Z., Feng, L., and Zhang, W. Vim: Out-of-distribution with virtual-logit matching. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022a.

Wang, H., Yin, H., Zhang, M., and Li, P. Equivariant and stable positional encoding for more powerful graph neural networks. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022b.

Wiles, O., Gowal, S., Stimberg, F., Rebuffi, S.-A., Ktena, I., Dvijotham, K. D., and Cemgil, A. T. A Fine-Grained Analysis on Distribution Shift. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? In *ICLR*, 2019.

Xu, K., Zhang, M., Jegelka, S., and Kawaguchi, K. Optimization of graph neural networks: Implicit acceleration by skip connections and more depth. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2021.

Yan, Y., Zhu, J., Duda, M., Solarz, E., Sripada, C. S., and Koutra, D. Groupinn: Grouping-based interpretable neural network for classification of limited, noisy brain data. In *Proc. Int. Conf. on Knowledge Discovery & Data Mining, KDD*, 2019.

Yang, J., Lu, J., Lee, S., Batra, D., and Parikh, D. Graph R-CNN for scene graph generation. In *Proc. Euro. Conf. on Computer Vision (ECCV)*, 2018.

Yehudai, G., Fetaya, E., Meirom, E., Chechik, G., and Maron, H. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pp. 11975–11986. PMLR, 2021.

Zhang, M. and Chen, Y. Link prediction based on graph neural networks. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2018.

Zhao, L., Jin, W., Akoglu, L., and Shah, N. From stars to subgraphs: Uplifting any GNN with local structure awareness. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Zhu, Y., Du, Y., Wang, Y., Xu, Y., Zhang, J., Liu, Q., and Wu, S. A survey on deep graph generation: Methods and applications. In *Learning on Graphs Conference (LoG)*, 2022.

A. Appendix

A.1. Details on Super-pixel Experiments

We provide an example of the rotated images and corresponding super-pixel graphs in Fig. 7, as well as additional resulting using the GINE backbone.

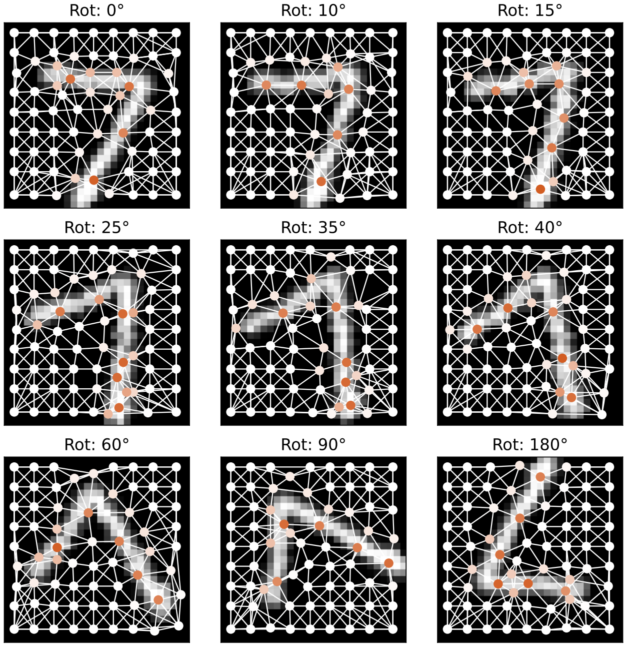

Figure 7. **Rotated Super-pixel MNIST.** Rotating images prior to creating super-pixels to leads to some structural distortion (Ding et al., 2021). We can see that the class-discriminative information is preserved, despite rotation. This allows for simulating different levels of distribution shifts, while still ensuring that samples are valid.

A.2. Dataset Statistics

The statistics for the size generalization experiments (see Sec. 5.1) are provided below in Table 1.

Table 1. **Size Generalization Dataset Statistics:** This table is directly reproduced from (Buffelli et al., 2022), who in turn used statistics from (Yehudai et al., 2021; Bevilacqua et al., 2021).

	NCI1			NCI109		
	all	Smallest 50%	Largest 10%	all	Smallest 50%	Largest 10%
Class A	49.95%	62.30%	19.17%	49.62%	62.04%	21.37%
Class B	50.04%	37.69%	80.82%	50.37%	37.95%	78.62%
Num of graphs	4110	2157	412	4127	2079	421
Avg graph size	29	20	61	29	20	61

	PROTEINS			DD		
	all	Smallest 50%	Largest 10%	all	Smallest 50%	Largest 10%
Class A	59.56%	41.97%	90.17%	58.65%	35.47%	79.66%
Class B	40.43%	58.02%	9.82%	41.34%	64.52%	20.33%
Num of graphs	1113	567	112	1178	592	118
Avg graph size	39	15	138	284	144	746

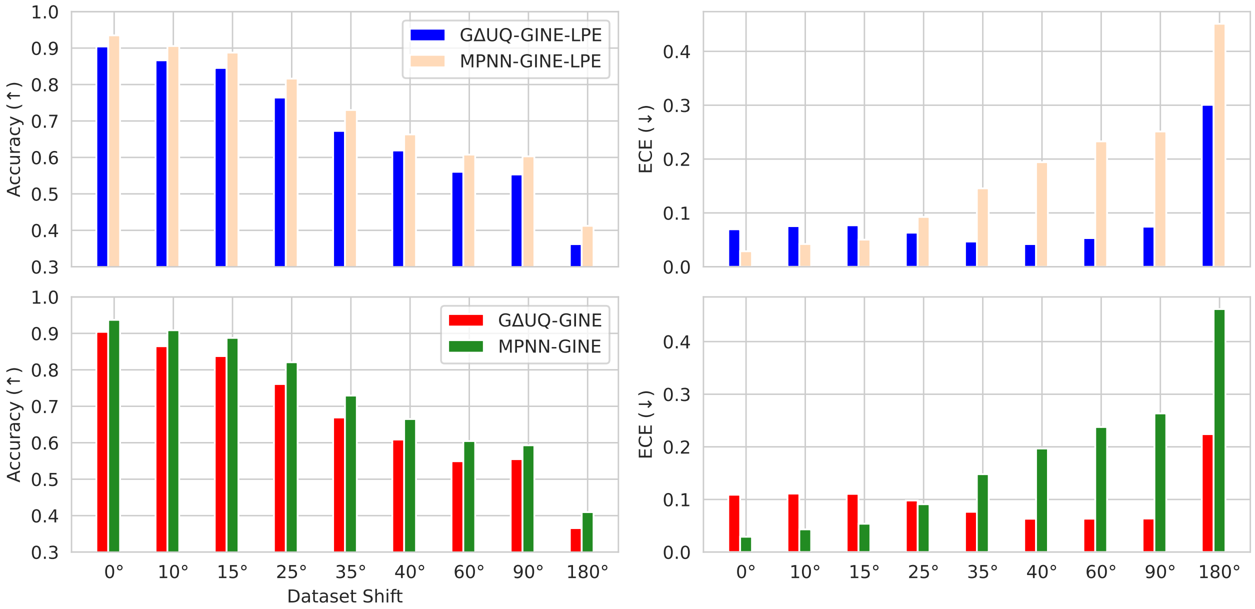

Figure 8. **Rotated Super-pixel MNIST, GINE Backbone.** We report additional results for performance on rotated-superpixel MNIST using a GINE backbone and `READOUT` stochastic ancoring. While G-ΔUQ does lose some accuracy, we see at higher levels of distortion, that it is significantly better calibrated.

A.3. Layer Selection Size Generalization

As discussed in Sec. 5.1, the choice of anchoring layer can have a disparate effect on the accuracy and calibration of a dataset. For example, in Fig. 9, we see that while the choice of layer is very influential for DD, it is significantly less important for the other datasets. However, overall, we see that anchoring after `READOUT` leads to good performance.

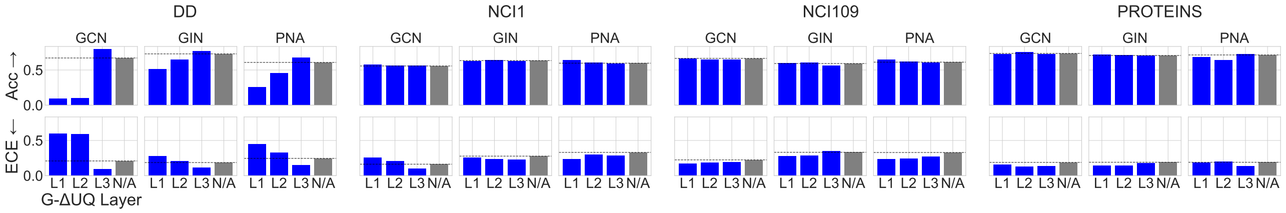

Figure 9. **Effect of Anchoring Layer on Performance.** Performing stochastic anchoring at different layers leads to sampling of different inductive biases and hypothesis spaces. Here, we see that the choice of anchoring layer is important to the the D&D dataset, but is not as influential on the other datasets. Given that last-layer anchoring generally performs well, we suggest performing stochastic anchoring in the last-layer when models are expected to encounter size distribution shifts.

A.4. GOOD Benchmark Experimental Details

For our experiments in Sec. 5.2, we utilize the in/out-of-distribution covariate and concept splits provided by (Gui et al., 2022). Furthermore, we use the suggested models and architectures provided by their package. In brief, we use GIN models with virtual nodes (except for GOODMotif) for training, and average scores over 3 seeds. When performing stochastic anchoring at a particular layer, we double the hidden representation size for that layer. Subsequent layers retain the original size of the vanilla model.

We use 10 samples when computing uncertainties using Monte Carlo Dropout, and manually set individual layers to “train" in order to perform layer-wise dropout. When performing stochastic anchoring, we use 10 fixed anchors randomly drawn from the in-distribution validation dataset. We also train the G-ΔUQ for an additional 50 epochs to ensure that models are able to converge. Please see our code repo for the full details.