Estimating Epistemic Uncertainty of Graph Neural Networks using Stochastic Centering

Anonymous Author(s)
Affiliation
Address
email

Abstract

While graph neural networks (GNNs) are widely used for node and graph representation learning tasks, the reliability of GNN uncertainty estimates under distribution shifts remains relatively under-explored. Indeed, while *post-hoc* calibration strategies can be used to improve in-distribution calibration, they need not also improve calibration under distribution shift. However, techniques which produce GNNs with better *intrinsic* uncertainty estimates are particularly valuable, as they can always be combined with post-hoc strategies later. Therefore, in this work, we propose G-ΔUQ, a novel training framework designed to improve intrinsic GNN uncertainty estimates. Our framework adapts the principle of stochastic data centering to graph data through novel graph anchoring strategies, and is able to support partially stochastic GNNs. While, the prevalent wisdom is that fully stochastic networks are necessary to obtain reliable estimates, we find that the functional diversity induced by our anchoring strategies when sampling hypotheses renders this unnecessary and allows us to support G-ΔUQ on pretrained models. Indeed, through extensive evaluation under covariate, concept and graph size shifts, we show that G-ΔUQ leads to better calibrated GNNs for node and graph classification. Further, it also improves performance on the uncertainty-based tasks of out-of-distribution detection and generalization gap estimation. Overall, our work provides insights into uncertainty estimation for GNNs, and demonstrates the utility of G-ΔUQ in obtaining reliable estimates.

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 calibration errors (Guo et al., 2017), out-of-distribution (OOD) rejection rates (Hendrycks & Gimpel, 2017), and generalization error predictions (GEP) (Jiang et al., 2019), to holistically understand model performance in such shifted regimes (Hendrycks et al., 2022b; Trivedi et al., 2023b). Notably, improving on these additional metrics often requires reliable uncertainty estimates, such as maximum softmax or predictive entropy, which can be derived from prediction probabilities. Although there is a clear understanding in the computer vision literature that the quality of uncertainty estimates can noticeably deteriorate under distribution shifts (Wiles et al., 2022; Ovadia et al., 2019), the impact of such shifts on graph neural networks (GNNs) remains relatively under-explored.

Post-hoc calibration methods (Guo et al., 2017; Gupta et al., 2021; Kull et al., 2019; Zhang et al., 2020), which use validation datasets to rescale logits to be obtain better calibrated models, are an effective, accuracy-preserving strategy for improving uncertainty estimates and model trustworthiness. Indeed, several post-hoc calibration strategies (Hsu et al., 2022; Wang et al., 2021) have been recently proposed to explicitly account for the non-IID nature of node-classification

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

datasets. However, while these methods are effective at improving uncertainty estimate reliability on in-distribution (ID) data, they have not been evaluated on OOD data, where they may become unreliable. To this end, training strategies which produce models with better intrinsic uncertainty estimates are valuable as they will provide better out-of-the-box ID and OOD estimates, which can then be further combined with post-hoc calibration strategies if desired.

The Δ-UQ training framework (Thiagarajan et al., 2022) was recently proposed as a scalable, single model alternative for vision models ensembles and has achieved state-of-the-art performance on calibration and OOD detection tasks. Central to Δ-UQ's success is the concept of *anchored* training, where models are trained on stochastic, relative representations of input samples in order to simulate sampling from different functional modes at test time (Sec. 2.) While, on the surface, Δ-UQ also appears as a potentially attractive framework for obtaining reliable, intrinsic uncertainty estimates on graph-based tasks, there are several challenges that arise from the structured, discrete, and variable-sized nature of graph data that must be resolved first. Namely, the anchoring procedure used by Δ-UQ is not applicable for graph datasets, and it is unclear how to design alternative anchoring strategies such that sufficiently diverse functional modes are sampled at inference to provide reliable epistemic uncertainty estimates.

Proposed Work. Thus, our work proposes G-ΔUQ, a novel training paradigm which provides better intrinsic uncertainty estimates for both graph and node classification tasks through the use of newly introduced graph-specific, anchoring strategies. Notably, our anchoring strategies support partially stochastic GNNs (instead of only fully stochastic Δ-UQ models). We demonstrate that not only is partially stochasticity is empirically valuable in calibrated GNNs across different distribution shifts and architectures, it also supports a light-weight uncertainty aware fine-tuning strategy for pretrained models and reduced the computational burden of training a fully stochastic model. Our contributions can be summarized as follows:

• **(Partially) Stochastic Anchoring for GNNs.** We propose G-ΔUQ, a novel training paradigm that improves the reliability of uncertainty estimates on GNN-based tasks. Our novel graph-anchoring strategies support partial stochasticity GNNs as well as training with pretrained models. (Sec. 3).

• **Evaluating Uncertainty-Modulated CIs under Distribution Shifts.** Across covariate, concept and graph-size shifts, we demonstrate that G-ΔUQ leads to better calibration. Moreover, G-ΔUQ's performance is further improved when combined with post-hoc calibration strategies on several node and graph-level tasks, including new safety-critical tasks (Sec. 5).

• **Fine-Grained Analysis of G-ΔUQ.** We study the calibration of architectures of varying expressivity and G-ΔUQ 's ability to improve them under varying distribution shift. We further demonstrate its utility as a lightweight strategy for improving the calibration of pretrained GNNs (Sec. 6).

2 Related Work

While uncertainty estimates are useful for a variety of safety-critical tasks (Hendrycks & Gimpel, 2017; Jiang et al., 2019; Guo et al., 2017), DNNs are well-known to provide poor uncertainty estimates directly out of the box (Guo et al., 2017). To this end, there has been considerable interest in building calibrated models, where the confidence of a prediction matches the probability of the prediction being correct. Notably, since GEP and OOD detection methods often rely upon transformations of a model's logits, improving calibration can in turn improve performance on these tasks as well. Due to their accuracy-preserving properties, post-hoc calibration strategies, which rescale confidences after training using a validation dataset, are particularly popular. Indeed, several methods (Guo et al., 2017; Gupta et al., 2021; Kull et al., 2019; Zhang et al., 2020) have been proposed for DNNs in general and, more recently, dedicated node-classifier calibration methods (Hsu et al., 2022; Wang et al., 2021) have also been proposed to accommodate the non-IID nature of graph data. (See App. A.7 for more details.) Notably, however, such post-hoc methods do not lead to reliable estimates under distribution shifts, as enforcing calibration on ID validation data does not directly lead to reliable estimates on OOD data (Ovadia et al., 2019; Wiles et al., 2022; Hendrycks et al., 2019).

Alternatively, Bayesian methods have been proposed for DNNs (Hernández-Lobato & Adams, 2015; Blundell et al., 2015), and more recently GNNs (Zhang et al., 2019; Hasanzadeh et al., 2020), as inherently "uncertainty-aware" strategies. However, not only do such methods often lead to performance loss, require complicated architectures and additional training time, they often struggle to outperform the simple Deep Ensembles (DEns) baseline (Lakshminarayanan et al., 2017). By training a collection of independent models, DEns is able to sample different functional modes of the hypothesis space, and thus, capture epistemic variability to perform uncertainty quantification (Wilson & Izmailov, 2020). Given that DEns requires training and storing multiple models, the SoTA Δ-UQ framework (Thiagarajan et al., 2022) was recently proposed to sample different functional modes using only a single model, based on the principle of *anchoring*. Conceptually, anchoring is the process of creating a relative representation for an input sample in terms of a random "anchor." By randomizing anchors throughout training (e.g., stochastically centering samples with respect to different anchors), Δ-UQ emulates the process of sampling different solutions from the hypothesis space. Given Δ-UQ's success in improving calibration and generalization (Netanyahu et al., 2023) under distribution shifts on computer vision tasks and the limitations of existing post-hoc strategies, stochastic centering appears as a potentially attractive framework for obtaining reliable uncertainty estimates when performing GNN-based graph and node classification tasks under distribution shifts. However, as we will discuss in Sec. 3, there are several challenges that arise from the structured, discrete, and variable-sized nature of graph data, which necessitate novel anchoring strategies to ensure that the underlying functional hypothesis space is effectively sampled.

Preliminaries. Here, we formally introduce stochastic centering. Let $\mathbf{C} := \mathbf{X}_{train}$ be the anchor distribution, $x \in \mathbf{X}_{test}$ be a test sample, and anchor $c \in \mathbf{C}$ be a single anchor. Since, previous research on stochastic centering has focused on vision models (CNNs, ResNets, ViT), straightforward input space transformations were used to construct anchored representations. Namely, anchored image samples were created by subtracting and channel-wise concatenating two images: $[\mathbf{X} - \mathbf{C}, \mathbf{C}]$). Then, the corresponding stochastically centered model can be defined as $f_\theta : [\mathbf{X} - \mathbf{C}, \mathbf{C}] \to \hat{\mathbf{Y}}$. Like ensembles, predictions and uncertainties are aggregated over different hypotheses. Given K random anchors, the mean target class prediction, $\boldsymbol{\mu}(y|\mathrm{x})$, and the corresponding variance, $\boldsymbol{\sigma}(y|\mathrm{x})$ are computed as: $\boldsymbol{\mu}(y|\mathrm{x}) = \frac{1}{K}\sum_{k=1}^{K} f_\theta([\mathrm{x} - \mathrm{c}_k, \mathrm{c}_k])$ and $\boldsymbol{\sigma}(y|\mathrm{x}) = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}(f_\theta([\mathrm{x} - \mathrm{c}_k, \mathrm{c}_k]) - \boldsymbol{\mu})^2}$. Since the variance over K anchors captures epistemic uncertainty by sampling different hypotheses, these estimates can be used to modulate the predictions: $\boldsymbol{\mu}_{\text{calib.}} = \boldsymbol{\mu}(1 - \boldsymbol{\sigma})$. The rescaled logits and uncertainty estimates have led to state-of-the-art performance on image outlier rejection and extrapolation (Anirudh & Thiagarajan, 2022).

3 Graph-ΔUQ: Uncertainty-Aware Predictions

As discussed in Sec. 2, the stochastic centering paradigm has demonstrated significant promise in computer vision; but there are several challenges that must be addressed prior to applying it to GNNs (and graph data). Foremost, it is unclear how to define graph-specific anchoring strategies such that stochastic centering is able to sample appropriately diverse, yet effective, GNN functional hypotheses. Indeed, trivial input transformations (e.g., subtraction/channel concatenation) are not possible when working with structured, discrete, variable-sized and potentially non-IID graphs. Moreover, we hypothesize and empirically demonstrate (Sec. 5) that fully stochastic GNNs, as induced by input space anchoring, are, in fact, not necessary for obtaining reliable uncertainty estimates. To this end, we propose `MPNN` and `READOUT` anchoring as alternative, scalable anchoring strategies for improving graph classifier calibration. Next, we first introuce the key notations that we use in the remainder of the paper, and then we conceptually describe the different anchoring strategies.

Notations. Let $\mathcal{G} = (\mathbf{X}, \mathbb{E}, \mathbf{A}, Y)$ be a graph with node features $\mathbf{X} \in \mathbb{R}^{N \times d_\ell}$, (optional) edge features $\mathbb{E} \in \mathbb{R}^{m \times d_\ell}$, 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. $\mathcal{G}_i, \mathbf{X}_i$. We can then define a GNN consisting of ℓ message passing layers (`MPNN`), a graph-level readout function (`READOUT`), and classifier head (`MLP`), respectively, as : $\mathbf{X}_M^{\ell+1}$, $\mathbb{E}^{\ell+1} = \mathtt{MPNN}_e^\ell\left(\mathbf{X}^\ell, \mathbb{E}^\ell, \mathbf{A}\right)$, $\mathbf{G} = \mathtt{READOUT}\left(\mathbf{X}_M^{\ell+1}\right)$, and $\hat{Y} = \mathtt{MLP}\left(\mathbf{G}\right)$, where $\mathbf{X}_M^{\ell+1}, \mathbb{E}^{\ell+1}$ are intermediate node and edge representations, and $\mathbf{G}$ is the graph representation. When performing node classification, we do not include the `READOUT` layer, and instead output node-level predictions: $\hat{Y}^{|Nxc|} = \mathtt{MLP}\left(\mathbf{X}_M^{\ell+1}\right)$.

3.1 Node Feature Anchoring

Due to the discrete nature and potential size variability in graphs, performing a structural residual operation, $[\mathbf{A} - \mathbf{A}_c, \mathbf{A}_c]$ with respect to a graph sample, $\mathcal{G} = (\mathbf{X}, \mathbb{E}, \mathbf{A}, Y)$, and another anchor graph, $\mathcal{G}_c = (\mathbf{X}_c, \mathbb{E}_c, \mathbf{A}_c, Y_c)$, would be ineffective at inducing a stochastically centered GNN. Indeed, such a transform would introduce artificial edge weights and connectivity artifacts, harming convergence. Likewise, when performing graph classification, we cannot directly anchor over node features, $[\mathbf{X} - \mathbf{X}_c, \mathbf{X}_c]$, since graphs are different sizes. Taking arbitrary subsets of node features is also inadvisable as node features cannot be considered IID, and due to iterative message passing, the network may not be able to converge after aggregating k hops of stochastic node representations. (This is in contrast to images, where only a single anchor is used to induce to stochasticity).

To address these challenges, we instead fit a Gaussian distribution, ($\mathcal{N}(\mu, \sigma)$), over the training dataset node features to help manage the combinatorial stochasticity induced by message passing and issues relating to differing graph sizes. We emphasize that this distribution is only used for anchoring and does not assume that the dataset's node features are normally distributed. During training, we randomly sample an anchor from that distribution for each node. Mathematically, given an anchor $\mathbf{C}^{N\times d} \sim \mathcal{N}(\mu, \sigma)$, we create the anchor/query node feature pair $[\mathbf{X}_i - \mathbf{C}||\mathbf{X}_i]$, where $||$ denotes concatenation, and i is the node index. During inference, we sample a fixed set of K anchors and compute residuals for all nodes with respect to the same anchor, e.g., $\mathbf{c}^{1\times d}{}_k \sim \mathcal{N}(\mu, \sigma)$ ($[\mathbf{X}_i - c_k||\mathbf{X}_i]$), with appropriate broadcasting. For datasets with categorical node features, anchoring can be performed after embedding the node features into a continuous space. If node features are not available, anchoring can still be performed via positional encodings (Wang et al., 2022b), which are known to improve the expressivity and performance of GNNs (Dwivedi et al., 2022a).

Performing anchoring with respect to node features is the most analogous extension of Δ-UQ to graphs as it results in fully stochastic GNNs. This is particularly true on node classification tasks where each node (with its corresponding feature and label) can be viewed as an individual sample, similar to an image in the original Δ-UQ formulation. Indeed, in Sec. 4, we show that our above formulation can be straightforwardly used to improve the behavior of node-classifiers under distribution shifts, and can be combined with various post-hoc calibration strategies to further improve the calibration.

3.2 Hidden Layer Anchoring for Graph Classification

While node feature anchoring can leveraged even for graph classification tasks, there are several nuances that may limit its effectiveness. Notably, since each sample (and label) is at a graph-level, NFA not only effectively induces multiple anchors per sample, it also ignores structural information that may be useful in sampling more *functionally diverse* hypotheses, e.g., hypotheses which capture functional modes that rely upon different high-level semantic, non-linear features. To improve the quality of hypothesis sampling, we introduce hidden layer anchoring below, which incorporates structural information into anchors at the expense of full stochasticity in the network (See Fig. 1.)

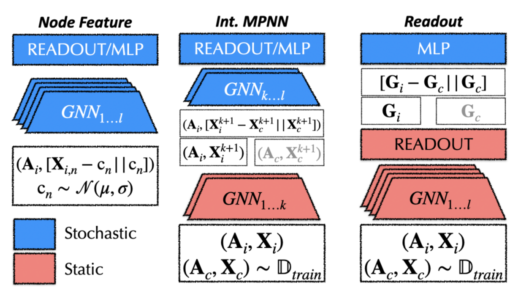

Figure 1: **Overview of G-ΔUQ.** We propose three different stochastic centering variants that induce varying levels of stochasticity in the underlying GNN. Notably, READOUT stochastic centering allows for using pretrained models with G-ΔUQ.

Hidden Layer and Readout Anchoring: Given a GNN containing ℓ MPNN layers, let $r \leq \ell$ be the layer at which we perform anchoring. The anchor/sample pair is obtained from the intermediate node representations from the first r MPNN layers. We then randomly shuffle the node features over the entire *batch*, ($\mathbf{C} = \text{SHUFFLE}(\mathbf{X}_i^{r+1})$), concatenate the residuals, and proceed with the READOUT and MLP layers as usual. Note the gradients of the query sample are not considered when updating parameters, and the $\texttt{MPNN}^{r+1}$ layer is modified to accept inputs of dimension $d_r \times 2$ (to take in anchored representations as inputs). For improved convergence, we fix the set of anchors and subtract a single anchor from all node representations in an iteration (instead of sampling uniquely), e.g., $\mathbf{c}^{1\times d} = \mathbf{X}_c^{r+1}[n,:]$ and $[\mathbf{X}_{i,n}^{r+1} - \mathbf{c}||\mathbf{c}]$. This process induces the following GNN (requires appropriate broadcasting): $\mathbf{X}^{r+1} = \texttt{MPNN}^{1\ldots r}$, $\mathbf{X}^{r+1} = \texttt{MPNN}^{r+1\ldots\ell}\left([\mathbf{X}^{r+1} - \mathbf{C}, \mathbf{X}^{r+1}], \mathbf{A}\right)$, and $\hat{Y} = \texttt{MLP}(\texttt{READOUT}\left(\mathbf{X}^{\ell+1}\right))$.

Not only do hidden layer anchors aggregate structural information over r hops, they induce a GNN that is now partially stochastic, as layers $1\ldots r$ are deterministic. Interestingly, it was recently demonstrated that relaxing the assumption of full stochasticity to partial stochasticity in Bayesian neural networks (BNNs) not only leads to strong computational benefits, but may also improve calibration (Sharma et al., 2023). Indeed, by reducing network stochasticity, it is naturally expected that hidden layer anchoring will reduce the diversity of the hypotheses, but by sampling more *functionally diverse* hypotheses through deeper, semantically expressive anchors, it is possible that *naively* maximizing diversity is in fact not required for reliable uncertainty estimation. To validate this hypothesis, we thus propose the final variant, `READOUT` anchoring for graph classification tasks. While conceptually similar to hidden layer anchoring, here, we simultaneously minimize GNN stochasticity (only the classifier is stochastic) and maximize anchor expressivity (anchors are graph representations pooled after ℓ rounds of message passing). Notably, `READOUT` anchoring is also compatible with pretrained GNN backbones, as the final `MLP` layer of a pretrained model is discarded (if necessary), and reinitialized to accommodate query/anchor pairs. Given the frozen `MPNN` backbone, only the anchored classifier head is trained.

In Sec. 5, we empirically verify the effectiveness of our proposed G-ΔUQ variants and demonstrate that fully stochastic GNNs are, in fact, unnecessary to obtain highly generalizable solutions, meaningful uncertainties and improved calibration on graph classification tasks. Moreover, in addition to strong calibration, we demonstrate in Sec. 6 that G-ΔUQ provides estimates that are useful for safety-critical OOD detection and generalization gap prediction tasks.

4 Node Classification Experiments: G-ΔUQ Improves Calibration

In this section, we demonstrate that G-ΔUQ improves uncertainty estimation in GNNs, particularly when evaluating *node classifiers* under distribution shifts. To the best of our knowledge, GNN calibration has not been extensively evaluated under this challenging setting, where uncertainty estimates are known to be unreliable (Ovadia et al., 2019). We demonstrate that G-ΔUQ not only directly provides better estimates, but also that combining G-ΔUQ with existing post-hoc calibration methods further improves performance.

Experimental Setup. We use the concept and covariate shifts for WebKB, Cora and CBAS datasets provided by Gui et al. (2022), and follow the recommended hyperparameters for training. In our implementation of node feature anchoring, we use 10 random anchors to obtain predictions with G-ΔUQ. All our results are averaged over 5 seeds and post-hoc calibration methods (described further in App. A.7) are fitted on the in-distribution validation dataset. The expected calibration error and accuracy on the unobserved "OOD test" split are reported.

Results. A subset of our results (Cora-Degree) are presented in Table 1 (remaining results are in the supplementary Table 10). We observe that G-ΔUQ is substantially better calibrated than the vanilla model under both concept (0.307 vs. 0.13) and covariate shift (0.348 vs. 0.141), while maintaining comparable, if not better accuracy. Most notably, we see that G-ΔUQ outperforms vanilla models that have been calibrated with graph-specific techniques CaGCN and GATS. Not only does this suggest that G-ΔUQ inherently provides more robust estimates but that there is substantial room for improving the OOD calibration of post-hoc GNN calibrators. Further, we can combine G-ΔUQ with post-hoc calibration strategies leading to even better performance. Our observations are generally consistent across the other datasets as well.

5 Graph Classification Uncertainty Experiments with G-ΔUQ

While applying G-ΔUQ to node classification tasks was relatively straightforward, performing stochastic centering with graph classification tasks is more nuanced. As discussed in Sec. 3, different anchoring strategies can introduce varying levels of stochasticity, and it is unknown how these strategies affect uncertainty estimate reliability. Therefore, we begin by demonstrating that fully stochastic GNNs are not necessary for producing reliable estimates (Sec. 5.1). We then extensively evaluate the calibration of partially stochastic GNNs on covariate and concept shifts with and without post-hoc calibration strategies (Sec. 5.2), as well as for different UQ tasks (Sec. 5.3). Lastly, we demonstrate that G-ΔUQ's uncertainty estimates remain reliable when used with different architectures and pretrained backbones (Sec. 6).

5.1 Is Full Stochasticity Necessary for G-ΔUQ?

By changing the anchoring strategy and intermediate anchoring layer, we can induce varying levels of stochasticity in the resulting GNNs. As discussed in Sec. 3, we hypothesize that the decreased stochasticity incurred by performing anchoring at deeper network layers will lead to more functionally diverse hypotheses, and consequently more reliable uncertainty estimates. We verify this hypothesis here, by studying the effect of anchoring layer on calibration under graph-size distribution shift. Namely, we find that `READOUT` anchoring sufficiently balances stochasticity and functional diversity.

Experimental Setup. We study the effect of different anchoring strategies on graph classification calibration under graph-size shift. Following the procedure of (Buffelli et al., 2022; Yehudai et al.,

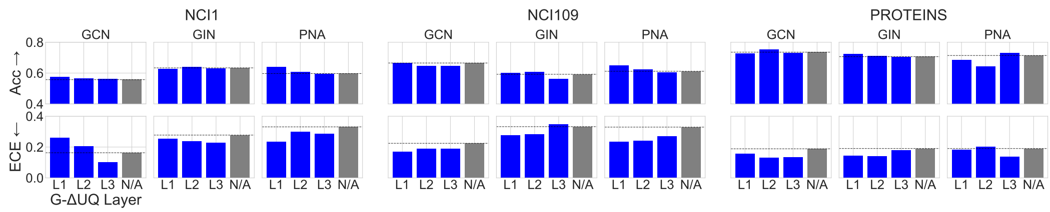

Figure 2: **Effect of Anchoring Layer.** Anchoring at different layers induces different hypotheses spaces. `READOUT` anchoring generally performs well across datasets and architectures.

2021), we create a size distribution shift by taking the smallest 50%-quantile of graph size for the training set, and evaluate on the largest 10% quantile. Following (Buffelli et al., 2022), we apply this splitting procedure to NCI1, NCI09, and PROTEINS (Morris et al., 2020), consider 3 GNN backbones (GCN (Kipf & Welling, 2017), GIN (Xu et al., 2019), and PNA (Corso et al., 2020)) and use the same architectures/parameters. (See Appendix A.5 for dataset statistics.) The accuracy and expected calibration error over 10 seeds on the largest-graph test set are reported for models trained with and without stochastic anchoring.

Results. We compare the performance of anchoring at different layers in Fig. 2. We find overall that applying anchoring at the `READOUT` layer yields competitive performance on size generalization benchmarks and better convergence compared to stochastic centering performed at earlier layers. Notably, the success of `READOUT` anchoring validates our hypothesis that full stochasticity is not necessary for reliable estimates. This finding is also practically useful as such models are faster to train and able to support pretrained models. Given these benefits and its empirical performance, we perform `READOUT` anchoring for all following experiments.

Table 1: **Calibration under Covariate and Concept shifts.** G-ΔUQ leads to better calibrated models for node-(GOODCora) and graph-level prediction tasks under different kinds of distribution shifts. Notably, G-ΔUQ can be combined with post-hoc calibration techniques to further improve calibration. The expected calibration error (ECE) is reported. Best, Second.

			Shift: Concept				Shift: Covariate			
			Accuracy (↑)		ECE (↓)		Accuracy (↑)		ECE (↓)	
Dataset	Domain	Calibration	No G-Δ UQ	G-Δ UQ	No G-Δ UQ	G-Δ UQ	No G-Δ UQ	G-Δ UQ	No G-Δ UQ	G-Δ UQ
GOODCora	Degree	×	0.581±0.003	0.595±0.003	0.307±0.009	0.13±0.011	0.47±0.002	0.518±0.014	0.348±0.032	0.141±0.008
		CAGCN	0.581±0.003	**0.597**±0.002	0.135±0.009	0.128±0.025	0.47±0.002	0.522±0.025	0.256±0.08	0.231±0.025
		Dirichlet	0.534±0.007	0.551±0.004	0.12±0.004	0.196±0.003	0.414±0.007	0.449±0.01	0.163±0.002	0.356±0.01
		ETS	0.581±0.003	0.596±0.004	0.301±0.009	0.116±0.018	0.47±0.002	**0.523**±0.003	0.31±0.077	0.141±0.003
		GATS	0.581±0.003	0.596±0.004	0.185±0.018	0.229±0.039	0.47±0.002	0.521±0.011	0.211±0.004	0.308±0.011
		IRM	0.582±0.002	0.597±0.002	0.125±0.001	0.102±0.002	0.469±0.001	0.522±0.004	0.194±0.005	0.13±0.004
		Orderinvariant	0.581±0.003	0.592±0.002	0.226±0.024	0.213±0.049	0.47±0.002	0.498±0.027	0.318±0.042	0.196±0.027
		Spline	0.571±0.003	0.595±0.003	0.080±0.004	**0.068**±0.004	0.459±0.003	0.52±0.004	0.158±0.01	**0.098**±0.004
		VS	0.581±0.003	0.596±0.004	0.306±0.004	0.127±0.002	0.47±0.001	0.522±0.005	0.345±0.005	0.146±0.005
GOODCMNIST	Color	×	0.499±0.003	0.497±0.002	0.439±0.078	0.334±0.066	0.348±0.009	0.355±0.034	0.551±0.147	0.423±0.172
		Dirichlet	0.495±0.009	0.510±0.008	0.303±0.012	0.304±0.007	0.350±0.053	0.335±0.059	0.542±0.091	0.406±0.076
		ETS	0.499±0.011	0.500±0.013	0.433±0.014	0.359±0.013	0.348±0.037	0.336±0.067	0.538±0.077	0.467±0.088
		IRM	0.499±0.006	0.500±0.010	0.285±0.004	**0.283**±0.008	0.348±0.049	0.336±0.071	0.416±0.084	0.425±0.093
		Orderinvariant	0.499±0.030	0.500±0.028	0.379±0.050	0.386±0.042	0.348±0.036	0.337±0.059	0.475±0.077	0.542±0.104
		Spline	0.495±0.008	0.497±0.010	0.29±0.007	0.291±0.008	0.346±0.051	0.335±0.071	0.414±0.085	0.425±0.093
		VS	0.499±0.007	0.500±0.012	0.439±0.006	0.377±0.009	0.349±0.037	0.336±0.067	0.549±0.071	0.468±0.089
		Ensembling	0.505±0.001	**0.509**±0.004	0.437±0.082	0.343±0.004	0.397±0.005	**0.408**±0.006	0.423±0.017	**0.327**±0.013
GOODMotif	Basis	×	0.925±0.001	0.925±0.003	0.095±0.014	0.078±0.007	0.691±0.001	0.689±0.002	0.329±0.274	0.342±0.266
		Dirichlet	0.925±0.011	0.923±0.010	0.081±0.015	0.103±0.007	0.686±0.009	0.681±0.009	0.337±0.067	0.316±0.047
		ETS	0.925±0.009	0.927±0.012	0.095±0.010	0.096±0.013	0.691±0.011	0.699±0.016	0.314±0.041	0.304±0.049
		IRM	0.925±0.014	0.93±0.013	0.087±0.018	0.097±0.010	0.691±0.011	0.698±0.016	0.316±0.051	0.305±0.045
		Orderinvariant	0.925±0.010	0.928±0.011	0.091±0.009	0.093±0.007	0.691±0.011	0.690±0.011	0.321±0.050	0.319±0.041
		Spline	0.925±0.010	0.927±0.011	0.091±0.008	0.089±0.012	0.691±0.010	0.689±0.016	0.324±0.055	0.313±0.051
		VS	0.925±0.009	0.927±0.012	0.095±0.010	0.095±0.013	0.683±0.013	0.680±0.018	0.326±0.057	0.311±0.059
		Ensembling	0.932±0.002	**0.943**±0.006	0.086±0.016	**0.047**±0.003	**0.714**±0.012	0.699±0.009	**0.298**±0.383	0.321±0.196
GOODSST2	Length	×	0.694±0.002	0.693±0.001	0.288±0.017	0.277±0.011	0.826±0.002	0.828±0.004	0.159±0.027	0.154±0.039
		Dirichlet	0.686±0.02	0.683±0.001	0.15±0.021	**0.138**±0.015	0.793±0.005	0.8±0.012	0.15±0.02	**0.131**±0.007
		ETS	0.685±0.02	0.683±0.001	0.21±0.009	0.211±0.003	0.794±0.005	0.8±0.011	0.287±0.007	0.296±0.014
		IRM	0.685±0.019	0.682±0.002	0.239±0.002	0.231±0.006	0.796±0.006	0.801±0.011	0.26±0.005	0.265±0.011
		Orderinvariant	0.685±0.02	0.683±0.001	0.225±0.002	0.222±0.003	0.794±0.005	0.8±0.011	0.226±0.003	0.224±0.007
		Spline	0.684±0.02	0.683±0.002	0.233±0.005	0.23±0.005	0.79±0.004	0.794±0.016	0.259±0.005	0.263±0.012
		VS	0.685±0.019	0.683±0	0.334±0.044	0.374±0.002	0.787±0.008	0.8±0.013	0.307±0.116	0.32±0.011
		Ensembling	0.705±0.002	**0.709**±0.004	0.276±0.038	0.248±0.022	0.838±0.001	**0.842**±0.006	0.154±0.032	0.132±0.019

5.2 Calibration under Concept and Covariate Shifts

Next, we assess the ability of G-ΔUQ to produce well-calibrated models under covariate and concept shift in graph classification tasks. We find that G-ΔUQ not only provides better calibration out of the box, its performance is further improved when combined with post-hoc calibration techniques.

Experimental Setup. We use four different datasets (GOODCMNIST, GOODMotif-basis, GOODMotif-size, GOODSST2) with their corresponding splits and shifts from the recently proposed Graph Out-Of Distribution (GOOD) benchmark (Gui et al., 2022). The architectures and hyperparameters suggested by the benchmark are used for training. G-ΔUQ uses `READOUT` anchoring and 10 random anchors (see App. A.6 for more details). We report accuracy and expected calibration error for the OOD test dataset, taken over three seeds.

Results. As shown in Table 1, we observe that G-ΔUQ leads to inherently better calibrated models, as the ECE from G-ΔUQ without additional post-hoc calibration (✗) is better than the vanilla ("No G-ΔUQ") counterparts on 5/6 datasets. Moreover, we find that the performance of post-hoc calibration methods is further improved when applied to stochastically centered models. Indeed, on 5/6 datasets, the best calibration is obtained by a G-ΔUQ temperature scaled variant. When directly comparing performance for a fixed post-hoc calibration strategy, G-ΔUQ improves the calibration, while maintaining comparable if not better accuracy on the vast majority of the methods and datasets. Our results clearly indicate that, unlike images, partially stochastic GNNs are sufficient for providing meaningful uncertainity estimates under challenging distribution shifts with minimal cost. In Sec. 6, we build upon this observation to demonstrate that G-ΔUQ is effective at improving the calibration of pretrained models as well.

5.3 Using Confidence Estimates in Safety-Critical Tasks

While post-hoc calibration strategies rely upon an additional calibration dataset to provide meaningful uncertainty estimates, such calibration datasets are not always available and may not necessarily improve OOD performance (Ovadia et al., 2019). Thus, we also evaluate the quality of the uncertainty estimates directly provided by G-ΔUQ on two additional UQ-based, safety-critical tasks (Hendrycks et al., 2022b, 2021; Trivedi et al., 2023b): (i) generalization error prediction (GEP) (Jiang et al., 2019), which attempts to predict the generalization on unlabeled test datasets (to the best of our knowledge, we are the first to study GEP of graph classifiers), and (ii) OOD detection (Hendrycks et al., 2019), which attempts to classify samples as in- or out-of-distribution.

GEP Experimental Setup. GEPs (Garg et al., 2022; Ng et al., 2022; Jiang et al., 2019; Trivedi et al., 2023a; Guillory et al., 2021) aggregate sample-level scores capturing a model's uncertainty about the correctness of a prediction into dataset-level error estimates. Here, we use maximum softmax probability for scores and a thresholding mechanism as the GEP. (See Appendix A.8 for more details.) We consider `READOUT` anchoring with both pretrained and end-to-end training, and report the mean absolute error between the predicted and true target dataset accuracy on the OOD test split.

GEP Results. As shown in Table 2a, both pretrained and end-to-end G-ΔUQ outperform the vanilla model on 7/8 datasets. Notably, we see that pretrained G-ΔUQ is particularly effective as it obtains the best performance across 6/8 datasets. This not only highlights its utility as a flexible, light-weight strategy for improving uncertainty estimates without sacrificing accuracy, but also emphasizes that importance of structure, in lieu of full stochasticity, when estimating GNN uncertainties.

OOD Detection Experimental Setup. By reliably detecting OOD samples and abstaining from making predictions on them, models can avoid over-extrapolating to irrelevant distributions. 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), popular scores, such as maximum softmax probability and predictive entropy (Hendrycks & Gimpel, 2017), are derived from uncertainty estimates. Here, we report the AUROC for the binary classification task of detecting OOD samples using the maximum softmax probability as the score (Kirchheim et al., 2022).

OOD Detection Results. As shown in Table 2b, we observe that G-ΔUQ variants improve OOD detection performance over the vanilla baseline on 6/8 datasets, where pretrained G-ΔUQ obtains the best overall performance on 6/8 datasets. G-ΔUQ performs comparably on GOODSST2(concept shift), but does lose some performance on GOODMotif(Covariate). We note that vanilla models provided by the original benchmark generalized poorly on this particular dataset (increased training

(a) **GOOD-Datasets, Generalization Error Prediction Performance**. The MAE between the predicted and true test error on the OOD test split is reported. G-ΔUQ variants outperform vanilla models on 7/8 datasets.

Method	CMNIST (Color) Concept(↓)	CMNIST (Color) Covariate (↓)	MotifLPE (Basis) Concept(↓)	MotifLPE (Basis) Covariate(↓)	MotifLPE (Size) Concept(↓)	MotifLPE (Size) Covariate(↓)	SST2 Concept(↓)	SST2 Covariate(↓)
Vanilla	0.200 ± 0.009	0.510 ± 0.089	0.045 ± 0.003	$\mathbf{0.570 \pm 0.012}$	0.324 ± 0.018	0.537 ± 0.146	0.117 ± 0.006	0.056 ± 0.044
G-ΔUQ	$\mathbf{0.190 \pm 0.010}$	0.493 ± 0.072	0.023 ± 0.003	0.572 ± 0.019	0.317 ± 0.007	0.528 ± 0.189	0.124 ± 0.016	0.054 ± 0.043
Pretr. G-ΔUQ	0.192 ± 0.005	$\mathbf{0.387 \pm 0.048}$	$\mathbf{0.018 \pm 0.012}$	0.573 ± 0.004	$\mathbf{0.307 \pm 0.016}$	$\mathbf{0.356 \pm 0.143}$	$\mathbf{0.114 \pm 0.004}$	$\mathbf{0.030 \pm 0.026}$

(b) **GOOD-Datasets, OOD Detection Performance.** The AUROC of the binary classification tasks of classifying OOD samples is reported. G-ΔUQ outperforms vanilla models on 6/8 datasets.

Method	CMNIST (Color) Concept(↑)	CMNIST (Color) Covariate(↑)	MotifLPE (Basis) Concept(↑)	MotifLPE (Basis) Covariate(↑)	MotifLPE (Size) Concept(↑)	MotifLPE (Size) Covariate(↑)	SST2 Concept(↑)	SST2 Covariate(↑)
Vanilla	0.759 ± 0.006	0.468 ± 0.092	0.736 ± 0.021	$\mathbf{0.466 \pm 0.001}$	0.680 ± 0.003	0.755 ± 0.074	$\mathbf{0.350 \pm 0.014}$	0.345 ± 0.066
G-ΔUQ	0.771 ± 0.002	0.470 ± 0.043	0.758 ± 0.006	0.328 ± 0.022	0.677 ± 0.005	0.691 ± 0.067	0.338 ± 0.023	0.351 ± 0.042
Pretr. G-ΔUQ	$\mathbf{0.774 \pm 0.016}$	$\mathbf{0.543 \pm 0.152}$	$\mathbf{0.769 \pm 0.029}$	0.272 ± 0.025	$\mathbf{0.686 \pm 0.004}$	$\mathbf{0.829 \pm 0.113}$	0.324 ± 0.055	$\mathbf{0.446 \pm 0.049}$

Table 3: **RotMNIST-Calibration.** Here, we report expanded results (calibration) on the Rotated MNIST dataset, including a variant that combines G-ΔUQ with Deep Ens. Notably, we see that anchored ensembles outperform basic ensembles in both accuracy and calibration.

Architecture	LPE?	G-ΔUQ	Calibration	Avg.ECE (↓)	ECE (10) (↓)	ECE (15) (↓)	ECE (25) (↓)	ECE (35) (↓)	ECE (40) (↓)
GatedGCN	✗	✗	✗	0.038 ±0.001	0.059 ±0.001	0.068 ±0.340	0.126 ±0.008	0.195 ±0.012	0.245 ±0.011
	✗	✓	✗	**0.018** ±0.008	0.029 ±0.013	**0.033** ±0.164	0.069 ±0.033	0.117 ±0.048	0.162 ±0.067
	✗	✗	Ensembling	0.026 ±0.000	0.038 ±0.001	0.042 ±0.001	0.084 ±0.002	0.135 ±0.001	0.185 ±0.003
	✗	✓	Ensembling	**0.014** ±0.003	**0.018** ±0.005	**0.021** ±0.005	0.036 ±0.012	0.069 ±0.032	0.114 ±0.056
GatedGCN	✓	✗	✗	0.036 ±0.003	0.059 ±0.002	0.068 ±0.340	0.125 ±0.006	0.191 ±0.007	0.240 ±0.008
	✓	✓	✗	0.022 ±0.007	**0.028** ±0.014	0.034 ±0.169	**0.062** ±0.022	**0.109** ±0.019	**0.141** ±0.019
	✓	✗	Ensembling	0.024 ±0.001	0.038 ±0.001	0.043 ±0.002	0.083 ±0.001	0.139 ±0.004	0.181 ±0.002
	✓	✓	Ensembling	0.017 ±0.002	0.024 ±0.005	0.027 ±0.008	**0.030** ±0.004	**0.036** ±0.012	**0.059** ±0.033
GPS	✓	✗	✗	0.026 ±0.001	0.044 ±0.001	0.052 ±0.156	0.108 ±0.006	0.197 ±0.012	0.273 ±0.008
	✓	✓	✗	0.022 ±0.001	0.037 ±0.005	0.044 ±0.133	0.091 ±0.008	0.165 ±0.018	0.239 ±0.018
	✓	✗	Ensembling	0.016 ±0.001	0.026 ±0.002	0.030 ±0.000	0.066 ±0.000	0.123 ±0.000	0.195 ±0.000
	✓	✓	Ensembling	**0.014** ±0.000	0.023 ±0.002	0.027 ±0.003	0.055 ±0.004	0.103 ±0.006	0.164 ±0.006

time/accuracy did not improve performance), and this behavior was reflected in our experiments. We suspect that poor generalization coupled with stochasticity may explain G-ΔUQ's performance here.

6 Fine Grained Analysis of G-ΔUQ

Given that the previous sections extensively verified the effectiveness of G-ΔUQ on a variety of covariate and concept shifts across several tasks, we seek a more fine-grained understanding of G-ΔUQ's behavior with respect to different architectures and training strategies. In particular, we demonstrate that G-ΔUQ continues to improve calibration with expressive graph transformer architectures, and that using `READOUT` anchoring with pretrained GNNs is an effective lightweight strategy for improving calibration of frozen GNN models.

6.1 Calibration under Controlled Shifts

Recently, it was shown that modern, non-convolutional architectures (Minderer et al., 2021) are not only more performant but also more calibrated than older, convolutional architectures (Guo et al., 2017) under vision distribution shifts. Here, we study an analogous question: are more expressive GNN architectures better calibrated under distribution shift, and how does G-ΔUQ impact their calibration? Surprisingly, we find that more expressive architectures are not considerably better calibrated than their MPNN counterparts, and ensembles of MPNNs outperform ensembles of GTrans. Notably, G-ΔUQ continues to improve calibration with respect to these architectures as well.

Experimental Setup. *(1) Models.* While improving the expressivity of GNNs is an active area of research, positional encodings (PEs) and graph-transformer (GTran) architectures (Müller et al., 2023) are popular strategies due to their effectiveness and flexibility. GTrans not only help mitigate over-smoothing and over-squashing (Alon & Yahav, 2021; Topping et al., 2022) but they also better capture long-range dependencies (Dwivedi et al., 2022b). Meanwhile, graph PEs help improve expressivity by differentiating isomorphic nodes, and capturing structural vs. proximity information (Dwivedi et al., 2022a). Here, we ask if these enhancements translate to improved calibration under distribution shift by comparing architectures with/without PEs and transformer vs. MPNN models. We use equivariant and stable PEs (Wang et al., 2022b), the state-of-the-art, “general, powerful, scalable" (GPS) framework with a GatedGCN backbone for the GTran, GatedGCN for the vanilla MPNN, and perform `READOUT` anchoring with 10 random anchors. *(2) Data.* In order to understand calibration behavior as distribution shifts become progressively more severe, we create structurally distorted but valid graphs by rotating MNIST images by a fixed number of degrees (Ding et al., 2021) and then creating the corresponding super-pixel graphs (Dwivedi et al., 2020; Knyazev et al., 2019; Velickovic et al., 2018). (See Appendix, Fig. 4.) Since superpixel segmentation on these rotated images will yield different superpixel k-nn graphs but leave class information unharmed, we can emulate different severities of label-preserving structural distortion shifts. We note that models are trained only using the original (0° rotation) graphs. Accuracy (see appendix) and ECE over 3 seeds are reported for the rotated graphs.

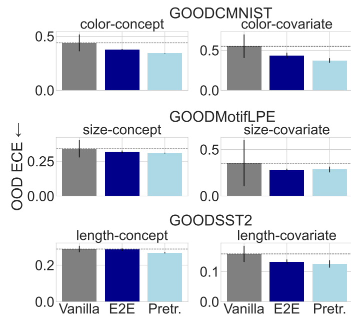

Figure 3: Out-of-distribution calibration error from applying G-ΔUQ in end-to-end training vs. to a pretrained model, which is a simple yet effective way to use stochastic anchoring.

Results. In Table 3, we present the OOD calibration results, with results of more variants and metrics in the supplementary Table 5 and 6. First, we observe that PEs have minimal effects on both calibration and accuracy by comparing GatedGCN with and without LPEs. This suggests that while PEs may enhance theoretical and empirical expressivity, they do not directly induce better calibration. Next, we find that while vanilla GPS is better calibrated when the distribution shift is not severe (10, 15, 25 degrees), it is less calibrated (but more performant) than GatedGCN at more severe distribution shifts (35, 40 degrees). This is in contrast to known findings about vision transformers, where such a tradeoff is not observed. Lastly, we see that G-ΔUQ continues to improve calibration across all considered architectural variants, with minimal accuracy loss. Surprisingly, however, we observe that *ensembles* of G-ΔUQ models not only effectively resolve any performance drops, they also cause MPNNs to be better calibrated than their GTran counterparts. Overall, our results indicate the interaction between increased expressivity and GNN calibration remains under-explored, though G-ΔUQ improves uncertainty estimates.

6.2 How does G-ΔUQ perform with pretrained models?

As large-scale pretrained models become increasingly more common, it is beneficial if practitioners are able to perform lightweight training that leads to more calibrated or safer models. Here, we investigate if `READOUT` anchoring is such a viable strategy when working with pretrained GNN backbones, as it only requires training a stochastically centered classifier on top of a frozen backbone.

Indeed, in Fig. 3, we observe that across datasets, pretraining yields competitive (often superior) OOD calibration with respect to end-to-end G-ΔUQ. Given that G-ΔUQ already outperformed other techniques (Sec. 3), this suggests that `READOUT` anchoring is a plausible solution for improving uncertainty estimation with pretrained backbones (we show results for additional performance metrics in the supplementary Fig. 6).

7 Conclusion

In this work, we propose G-ΔUQ, a novel training approach that adapts stochastic data centering for GNNs through newly introduced graph-specific anchoring strategies. Our extensive experiments demonstrate G-ΔUQ’s effectiveness for improving calibration and uncertainty estimates of GNNs under distribution shifts. Furthermore, we demonstrate that partially stochastic GNNs are sufficient for obtaining reliable uncertainty estimates and show that G-ΔUQ can be used as a lightweight strategy for improving the calibration of pretrained GNNs. Overall, G-ΔUQ is an effective strategy for improving the intrinsic quality of GNN uncertainty estimates.

References

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

Rushil Anirudh and Jayaraman J. Thiagarajan. Out of distribution detection via neural network anchoring. In *Asian Conference on Machine Learning, ACML 2022, 12-14 December 2022, Hyderabad, India*, 2022.

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

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2015.

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

Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Velickovic. Principal neighbourhood aggregation for graph nets. In *NeurIPS*, 2020.

Nicki Skafte Detlefsen, Jiri Borovec, Justus Schock, Ananya Harsh, Teddy Koker, Luca Di Liello, Daniel Stancl, Changsheng Quan, Maxim Grechkin, and William Falcon. Torchmetrics - measuring reproducibility in pytorch, 2022. URL https://github.com/Lightning-AI/torchmetrics.

Mucong Ding, Kezhi Kong, Jiuhai Chen, John Kirchenbauer, Micah Goldblum, David Wipf, Furong Huang, and Tom Goldstein. 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.

Vijay Prakash Dwivedi, Chaitanya K. Joshi, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Benchmarking graph neural networks. *CoRR*, 2020.

Vijay Prakash Dwivedi, Anh Tuan Luu, Thomas Laurent, Yoshua Bengio, and Xavier Bresson. Graph neural networks with learnable structural and positional representations. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022a.

Vijay Prakash Dwivedi, Ladislav Rampásek, Michael Galkin, Ali Parviz, Guy Wolf, Anh Tuan Luu, and Dominique Beaini. Long range graph benchmark. In *Proc. Adv. in Neural Information Processing Systems NeurIPS, Datasets and Benchmark Track*, 2022b.

Saurabh Garg, Sivaraman Balakrishnan, Zachary C. Lipton, Behnam Neyshabur, and Hanie Sedghi. Leveraging unlabeled data to predict out-of-distribution performance. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Thomas Gaudelet, Ben Day, Arian R. Jamasb, Jyothish Soman, Cristian Regep, Gertrude Liu, Jeremy B. R. Hayter, Richard Vickers, Charles Roberts, Jian Tang, David Roblin, Tom L. Blundell, Michael M. Bronstein, and Jake P. Taylor-King. Utilising graph machine learning within drug discovery and development. *CoRR*, abs/2012.05716, 2020.

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

Devin Guillory, Vaishaal Shankar, Sayna Ebrahimi, Trevor Darrell, and Ludwig Schmidt. Predicting with confidence on unseen distributions. In *ICCV*, 2021.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proc. of the Int. Conf. on Machine Learning, (ICML)*, 2017.

Kartik Gupta, Amir Rahimi, Thalaiyasingam Ajanthan, Thomas Mensink, Cristian Sminchisescu, and Richard Hartley. Calibration of neural networks using splines. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2021.

Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *ICML*, 2020.

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

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

Dan Hendrycks, Nicholas Carlini, John Schulman, and Jacob Steinhardt. Unsolved problems in ML safety. *CoRR*, abs/2109.13916, 2021.

Dan Hendrycks, Steven Basart, Mantas Mazeika, Andy Zou, Joseph Kwon, Mohammadreza Mostajabi, Jacob Steinhardt, and Dawn Song. Scaling out-of-distribution detection for real-world settings. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2022a.

Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *Proc. Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022b.

José Miguel Hernández-Lobato and Ryan P. Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2015.

Hans Hao-Hsun Hsu, Yuesong Shen, Christian Tomani, and Daniel Cremers. What makes graph neural networks miscalibrated? In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2022.

Yiding Jiang, Dilip Krishnan, Hossein Mobahi, and Samy Bengio. 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.

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

Konstantin Kirchheim, Marco Filax, and Frank Ortmeier. 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.

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

Meelis Kull, Miquel Perelló-Nieto, Markus Kängsepp, Telmo de Menezes e Silva Filho, Hao Song, and Peter A. Flach. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2019.

Ananya Kumar, Percy Liang, and Tengyu Ma. Verified uncertainty calibration. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2019.

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

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

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2020.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. Revisiting the calibration of modern neural networks. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2021.

Christopher Morris, Nils M. Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. 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.

Luis Müller, Mikhail Galkin, Christopher Morris, and Ladislav Rampásek. Attending to graph transformers. *CoRR*, abs/2302.04181, 2023.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proc. Conf. on Adv. of Artificial Intelligence (AAAI)*, 2015.

Aviv Netanyahu, Abhishek Gupta, Max Simchowitz, Kaiqing Zhang, and Pulkit Agrawal. Learning to extrapolate: A transductive approach. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2023.

Nathan Ng, Neha Hulkund, Kyunghyun Cho, and Marzyeh Ghassemi. Predicting out-of-domain generalization with local manifold smoothness. *CoRR*, abs/2207.02093, 2022.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2019.

Amir Rahimi, Amirreza Shaban, Ching-An Cheng, Richard Hartley, and Byron Boots. Intra order-preserving functions for calibration of multi-class neural networks. *Advances in Neural Information Processing Systems*, 33:13456–13467, 2020.

Mrinank Sharma, Sebastian Farquhar, Eric Nalisnick, and Tom Rainforth. Do bayesian neural networks need to be fully stochastic? In *AISTATS*, 2023.

Jayaraman J. Thiagarajan, Rushil Anirudh, Vivek Narayanaswamy, and Peer-Timo Bremer. Single model uncertainty estimation via stochastic data centering. In *Proc. Adv. in Neural Information Processing Systems (NeurIPS)*, 2022.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M. Bronstein. Understanding over-squashing and bottlenecks on graphs via curvature. In *Proc. Int. Conf. on Learning Representations ICLR*, 2022.

Puja Trivedi, Danai Koutra, and Jayaraman J Thiagarajan. 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.

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

Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

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

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

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2021.

Olivia Wiles, Sven Gowal, Florian Stimberg, Sylvestre-Alvise Rebuffi, Ira Ktena, Krishnamurthy Dj Dvijotham, and Ali Taylan Cemgil. A Fine-Grained Analysis on Distribution Shift. In *Proc. Int. Conf. on Learning Representations (ICLR)*, 2022.

Andrew Gordon Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In *Proc. Adv. in Neural Information Processing Systems NeurIPS*, 2020.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *ICLR*, 2019.

Yujun Yan, Jiong Zhu, Marlena Duda, Eric Solarz, Chandra Sekhar Sripada, and Danai Koutra. Groupinn: Grouping-based interpretable neural network for classification of limited, noisy brain data. In *Proc. Int. Conf. on Knowledge Discovery & Data Mining, KDD*, 2019.

Jianwei Yang, Jiasen Lu, Stefan Lee, Dhruv Batra, and Devi Parikh. Graph R-CNN for scene graph generation. In *Proc. Euro. Conf. on Computer Vision (ECCV)*, 2018.

Gilad Yehudai, Ethan Fetaya, Eli Meirom, Gal Chechik, and Haggai Maron. From local structures to size generalization in graph neural networks. In *International Conference on Machine Learning*, pp. 11975–11986. PMLR, 2021.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 694–699, 2002.

Jize Zhang, Bhavya Kailkhura, and Thomas Yong-Jin Han. Mix-n-match : Ensemble and compositional methods for uncertainty calibration in deep learning. In *Proc. Int. Conf. on Machine Learning (ICML)*, 2020.

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

Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *AAAI*, 2019.

Yanqiao Zhu, Yuanqi Du, Yinkai Wang, Yichen Xu, Jieyu Zhang, Qiang Liu, and Shu Wu. A survey on deep graph generation: Methods and applications. In *Learning on Graphs Conference (LoG)*, 2022.

A Appendix

- **Ethics** (Sec. A.1)
- **Reproducibility** (Sec. A.2)
- **Details and Expanded Results for Super-pixel Graph Experiments**(Sec. A.3)
- **Stochastic Centering on the Empirical NTK of Graph Neural Networks** (Sec. A.4)
- **Size-Generalization Dataset Statistics** (Sec. A.5)
- **GOOD Dataset Statistics and Expanded Results** (Sec. A.6)
- **Discussion of Post-hoc Calibration Strategies** (Sec. A.7)
- **Details of Generalization Gap Experiments** (Sec. A.8)
- **Expanded Pretraining Results** (Sec. A.9)

A.1 Ethics Statement

This work proposes a method to improve uncertainty estimation in graph neural networks, which has potential broader societal impacts. As graph learning models are increasingly deployed in real-world applications like healthcare, finance, and transportation, it becomes crucial to ensure these models make reliable predictions and know when they may be wrong. Unreliable models can lead to harmful outcomes if deployed carelessly. By improving uncertainty quantification, our work contributes towards trustworthy graph AI systems.

We also consider several additional safety-critical tasks, including generalization gap prediction for graph classification (to the best of our knowledge, we are the first to report results on this task) and OOD detection. We hope our work will encourage further study in these important areas.

However, there are some limitations. Our method requires (modest) additional computation during training and inference, which increases resource usage. Although G-ΔUQ, unlike post-hoc methods, does not need to be fit on a validation dataset, evaluation of its benefits also also relies on having some out-of-distribution or shifted data available, which may not always be feasible. Finally, there are open questions around how much enhancement in uncertainty calibration translates to real-world safety and performance gains.

Looking ahead, we believe improving uncertainty estimates is an important direction for graph neural networks and deep learning more broadly. This will enable the development safe, reliable AI that benefits society. We hope our work inspires more research in the graph domain that focuses on uncertainty quantification and techniques that provide guarantees about model behavior, especially for safety-critical applications. Continued progress will require interdisciplinary collaboration between graph machine learning researchers and domain experts in areas where models are deployed.

A.2 Reproducibility

For reproducing our experiments, we have made our code available at this anonymous repository. In the remainder of this appendix (specifically App. A.5, A.6), and A.8), we also provide additional details about the benchmarks and experimental setup.

A.3 Details on Super-pixel Experiments

We provide an example of the rotated images and corresponding super-pixel graphs in Fig. 4. (Note that classes "6" and "9" may be confused under severe distribution shift, i.e. 90 degrees rotation or more. Hence, to avoid harming class information, our experiments only consider distribution shift from rotation up to 40 degrees.)

Tables 4 and 5 provided expanded results on the rotated image super-pixel graph classification task, discussed in Sec. 6.1.

In addition to the structural distribution shifts we get by rotating the images before constructing super-pixel graphs, we also simulate feature distribution shifts by adding Gaussian noise with different standard deviations to the pixel value node features in the super-pixel graphs. In Table 6, we report accuracy and calibration results for varying levels of distribution shift (represented by the size of the

Table 4: **RotMNIST-Accuracy.** Here, we report expanded results (accuracy) on the Rotated MNIST dataset, including a variant that combines G-ΔUQ with Deep Ens. Notably, we see that anchored ensembles outperform basic ensembles in both accuracy and calibration.

MODEL	G-ΔUQ?	LPE?	Avg. Test (↑)	Acc. (10) (↑)	Acc. (15) (↑)	Acc. (25) (↑)	Acc. (35) (↑)	Acc. (40) (↑)
GatedGCN	×	×	0.947 ±0.002	0.918 ±0.002	0.904 ±0.005	0.828 ±0.009	0.738 ±0.009	0.679 ±0.007
	✓	×	0.933 ±0.015	0.894 ±0.019	0.878 ±0.020	0.794 ±0.032	0.698 ±0.036	0.636 ±0.048
	×	✓	0.949 ±0.002	0.917 ±0.004	0.904 ±0.005	0.829 ±0.007	0.744 ±0.007	0.685 ±0.006
	✓	✓	0.915 ±0.032	0.872 ±0.038	0.852 ±0.0414	0.776 ±0.039	0.680 ±0.037	0.631 ±0.033
GPS	×	✓	**0.970** ±0.001	**0.948** ±0.001	**0.938** ±0.001	**0.873** ±0.006	**0.770** ±0.013	**0.688** ±0.009
	✓	✓	**0.969** ±0.001	0.946 ±0.003	**0.937** ±0.003	0.869 ±0.003	**0.769** ±0.012	0.679 ±0.014
GPS (Pretrained)	✓	✓	0.967 ±0.002	0.945 ±0.004	0.934 ±0.005	0.864 ±0.009	0.759 ±0.010	0.674 ±0.002
GatedGCN-DENS	×	×	0.963 ±0.0002	0.943 ±0.001	0.933 ±0.001	0.874 ±0.002	0.794 ±0.002	0.731 ±0.002
	✓	×	0.949 ±0.008	0.922 ±0.008	0.907 ±0.011	0.828 ±0.020	0.733 ±0.032	0.662 ±0.046
	×	✓	0.965 ±0.001	0.943 ±0.001	0.933 ±0.001	0.873 ±0.001	0.792 ±0.004	0.736 ±0.003
	✓	✓	0.954 ±0.005	0.930 ±0.010	0.917 ±0.011	0.850 ±0.023	0.759 ±0.025	0.696 ±0.032
GPS-DENS	×	✓	**0.980** ±0.000	**0.969** ±0.000	**0.961** ±0.000	**0.913** ±0.000	**0.834** ±0.000	**0.750** ±0.000
	✓	✓	0.978 ±0.001	0.963 ±0.000	0.953 ±0.001	0.905 ±0.000	0.822 ±0.002	0.736 ±0.003

Table 5: **RotMNIST-Calibration.** Here, we report expanded results (calibration) on the Rotated MNIST dataset, including a variant that combines G-ΔUQ with Deep Ens. Notably, we see that anchored ensembles outperform basic ensembles in both accuracy and calibration.

MODEL	G-ΔUQ	LPE?	Avg.ECE (↓)	ECE (10) (↓)	ECE (15) (↓)	ECE (25) (↓)	ECE (35) (↓)	ECE (40) (↓)
GatedGCN-TS	×	×	0.035 ±0.001	0.054 ±0.002	0.062 ±0.003	0.118 ±0.007	0.185 ±0.006	0.233 ±0.008
	×	✓	0.033 ±0.002	0.053 ±0.002	0.061 ±0.004	0.116 ±0.005	0.179 ±0.006	0.225 ±0.005
GatedGCN	×	×	0.038 ±0.001	0.059 ±0.001	0.068 ±0.340	0.126 ±0.008	0.195 ±0.012	0.245 ±0.011
	✓	×	**0.018** ±0.008	0.029 ±0.013	**0.033** ±0.164	0.069 ±0.033	0.117 ±0.048	0.162 ±0.067
	×	✓	0.036 ±0.003	0.059 ±0.002	0.068 ±0.340	0.125 ±0.006	0.191 ±0.007	0.240 ±0.008
	✓	✓	0.022 ±0.007	**0.028** ±0.014	0.034 ±0.169	**0.062** ±0.022	**0.109** ±0.019	**0.141** ±0.019
GPS-TS	×	✓	0.024 ±0.001	0.041 ±0.001	0.049 ±0.001	0.102 ±0.006	0.188 ±0.012	0.261 ±0.008
GPS	×	✓	0.026 ±0.001	0.044 ±0.001	0.052 ±0.156	0.108 ±0.006	0.197 ±0.012	0.273 ±0.008
	✓	✓	0.022 ±0.001	0.037 ±0.005	0.044 ±0.133	0.091 ±0.008	0.165 ±0.018	0.239 ±0.018
GPS (Pretrained)	✓	✓	0.021 ±0.001	0.032 ±0.003	0.039 ±0.116	0.083 ±0.002	0.153 ±0.007	0.217 ±0.012
GatedGCN-DENS	×	×	0.026 ±0.000	0.038 ±0.001	0.042 ±0.001	0.084 ±0.002	0.135 ±0.001	0.185 ±0.003
	✓	×	**0.014** ±0.003	**0.018** ±0.005	**0.021** ±0.005	0.036 ±0.012	0.069 ±0.032	0.114 ±0.056
	×	✓	0.024 ±0.001	0.038 ±0.001	0.043 ±0.002	0.083 ±0.001	0.139 ±0.004	0.181 ±0.002
	✓	✓	0.017 ±0.002	0.024 ±0.005	0.027 ±0.008	**0.030** ±0.004	**0.036** ±0.012	**0.059** ±0.033
GPS-DENS	×	✓	0.016 ±0.001	0.026 ±0.002	0.030 ±0.000	0.066 ±0.000	0.123 ±0.000	0.195 ±0.000
	✓	✓	**0.014** ±0.000	0.023 ±0.002	0.027 ±0.003	0.055 ±0.004	0.103 ±0.006	0.164 ±0.006

Table 6: **MNIST Feature Shifts**. G-ΔUQ improves calibration and maintains competitive or even improved accuracy across varying levels of feature distribution shift.

				STD = 0.1		STD = 0.2		STD = 0.3		STD = 0.4	
MODEL	LPE?	G-ΔUQ?	Calibration	Accuracy (↑)	ECE (↓)	Accuracy (↑)	ECE (↓)	Accuracy (↑)	ECE (↓)	Accuracy (↑)	ECE (↓)
GatedGCN	×	×	×	0.742±0.005	0.186±0.018	0.481±0.015	0.414±0.092	0.293±0.074	0.606±0.147	0.197±0.092	0.71±0.178
	×	✓	×	**0.773**±0.053	**0.075**±0.032	0.536±0.010	**0.160**±0.087	**0.356**±0.101	0.422±0.083	**0.249**±0.074	**0.529**±0.047
	✓	×	×	0.751±0.02	0.176±0.014	0.519±0.004	0.348±0.03	0.345±0.032	0.485±0.096	0.233±0.043	0.581±0.142
	✓	✓	×	0.745±0.026	0.100±0.036	**0.541**±0.040	0.235±0.067	**0.355**±0.062	**0.408**±0.116	0.242±0.063	0.539±0.139

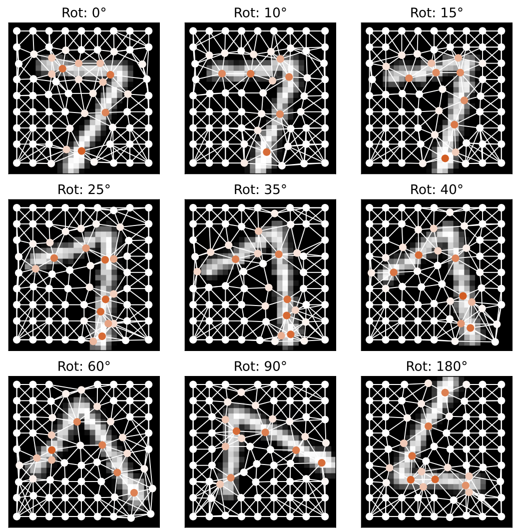

Figure 4: **Rotated Super-pixel MNIST.** Rotating images prior to creating super-pixels to leads to some structural distortion (Ding et al., 2021). However, we can see that the class-discriminative information is preserved, despite rotation. This allows for simulating different levels of graph structure distribution shifts, while still ensuring that samples are valid.

standard deviation of the Gaussian noise). Across different levels of feature distribution shift, we also see that G-ΔUQ results in superior calibration, while maintaining competitive or in many cases superior accuracy.

A.4 Stochastic Centering on the Empirical NTK of Graph Neural Networks

Using a simple grid-graph dataset and 4 layer GIN model, we compute the Fourier spectrum of the NTK. As shown in Fig. 5, we find that shifts to the node features can induce systematic changes to the spectrum.

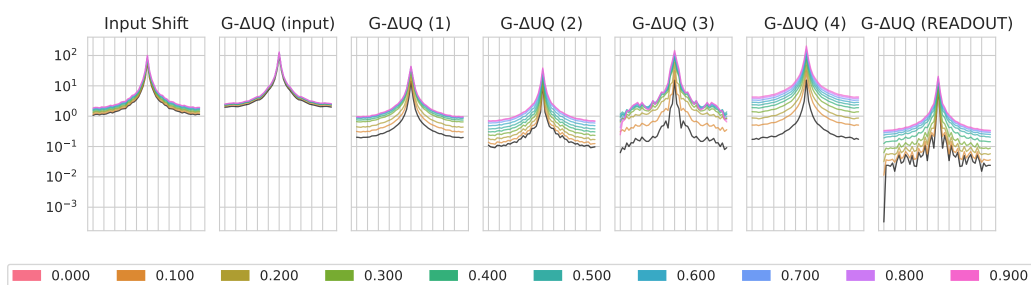

Figure 5: **Stochastic Centering with the empirical GNN NTK.** We find that performing constant shifts at intermediate layers introduces changes to a GNN's NTK. We include a vanilla GNN NTK in black for reference. Further, note the shape of the spectrum should not be compared across subplots as each subplot was created with a different random initialization.

A.5 Size-Generalization Dataset Statistics

The statistics for the size generalization experiments (see Sec. 5.1) are provided below in Table 7.

A.6 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

Table 7: **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%
# 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%
# OF GRAPHS	1113	567	112	1178	592	118
AVG GRAPH SIZE	39	15	138	284	144	746

Dataset	Shift	Train	ID validation	ID test	OOD validation	OOD test	Train	OOD validation	ID validation	ID test	OOD test
		Length									
GOOD-SST2	covariate	24744	5301	5301	17206	17490					
	concept	27270	5843	5843	15142	15944					
		Color									
GOOD-CMNIST	covariate	42000	7000	7000	7000	7000					
	concept	29400	6300	6300	14000	14000					
	no shift	42000	14000	14000	-	-					
		Base					Size				
GOOD-Motif	covariate	18000	3000	3000	3000	3000	18000	3000	3000	3000	3000
	concept	12600	2700	2700	6000	6000	12600	2700	2700	6000	6000
		Word					Degree				
GOOD-Cora	covariate	9378	1979	1979	3003	3454	8213	1979	1979	3841	3781
	concept	7273	1558	1558	3807	5597	7281	1560	1560	3706	5686
		University									
GOOD-WebKB	covariate	244	61	61	125	126					
	concept	282	60	60	106	109					
		Color									
GOOD-CBAS	covariate	420	70	70	70	70					
	concept	140	140	140	140	140					

Table 8: Number of Graphs/Nodes per dataset.

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.

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 repository for the full details.

We also include results on additional node classification benchmarks featuring distribution shift in Table 10.

A.7 Post-hoc Calibration Strategies

Several post hoc strategies have been developed for calibrating the predictions of a model. These have the advantage of flexibility, as they operate only on the outputs of a model and do not require that any changes be made to the model itself. Some methods include:

- **Temperature scaling (TS)** (Guo et al., 2017) simply scales the logits by a temperature parameter $T > 1$ to smooth the predictions. The scaling parameter T can be tuned on a validation set.
- **Ensemble temperature scaling (ETS)** (Zhang et al., 2020) learns an ensemble of temperature-scaled predictions with uncalibrated predictions ($T = 1$) and uniform probabilistic outputs ($T = \infty$).
- **Vector scaling** (VS) Guo et al. (2017) scales the entire output vector of class probabilities, rather than just the logits.

Dataset	model	# model layers	batch size	# max epochs	# iterations per epoch	initial learning rate
GOOD-SST2	GIN-Virtual	3	32	200/100	–	1e-3
GOOD-CMNIST	GIN-Virtual	5	128	500	–	1e-3
GOOD-Motif	GIN	3	32	200	–	1e-3
GOOD-Cora	GCN	3	4096	100	10	1e-3
GOOD-WebKB	GCN	3	4096	100	10	1e-3/5e-3
GOOD-CBAS	GCN	3	1000	200	10	3e-3

Table 9: Model and hyperparameters for GOOD datasets.

Table 10: **Additional Node Classification Benchmarks.** For more datasets with different kinds of distribution shifts, we find that G-ΔUQ improves model calibration and pairs well with post-hoc calibration methods for even better results.

			Shift: Concept				Shift: Covariate			
			Accuracy (↑)		ECE (↓)		Accuracy (↑)		ECE (↓)	
Dataset	Domain	Calibration	No G-Δ UQ	G-Δ UQ	No G-Δ UQ	G-Δ UQ	No G-Δ UQ	G-Δ UQ	No G-Δ UQ	G-Δ UQ
WebKB	University	×	0.253±0.003	**0.281**±0.009	0.67±0.061	0.593±0.025	0.122±0.029	0.115±0.041	0.599±0.091	0.525±0.033
		CAGCN	0.253±0.005	0.268±0.008	0.452±0.14	0.473±0.12	0.122±0.018	0.092±0.161	0.355±0.227	0.396±0.161
		Dirichlet	0.229±0.018	0.22±0.022	0.472±0.06	0.472±0.03	0.244±0.105	**0.295**±0.044	**0.299**±0.092	0.328±0.044
		ETS	0.253±0.005	0.273±0.012	0.64±0.06	0.575±0.019	0.121±0.021	0.084±0.027	0.539±0.112	0.499±0.027
		GATS	0.253±0.005	0.273±0.01	0.608±0.008	0.485±0.02	0.122±0.018	0.079±0.029	0.455±0.057	0.376±0.029
		IRM	0.251±0.005	0.266±0.011	**0.342**±0.017	0.349±0.006	0.097±0.04	0.046±0.013	0.352±0.037	0.422±0.013
		Orderinvariant	0.253±0.005	0.27±0.01	0.628±0.026	0.564±0.024	0.122±0.018	0.106±0.065	0.545±0.079	0.47±0.065
		Spline	0.237±0.012	0.257±0.023	0.436±0.029	0.386±0.034	0.122±0.013	0.171±0.056	0.472±0.031	0.39±0.056
		VS	0.253±0.005	0.275±0.011	0.67±0.009	0.588±0.011	0.122±0.018	0.095±0.014	0.602±0.044	0.507±0.014
Cora	Degree	×	0.581±0.003	0.595±0.003	0.307±0.009	0.13±0.011	0.47±0.002	0.518±0.014	0.348±0.032	0.141±0.008
		CAGCN	0.581±0.003	**0.597**±0.002	0.135±0.009	0.128±0.025	0.47±0.002	0.522±0.025	0.256±0.08	0.231±0.025
		Dirichlet	0.534±0.007	0.551±0.004	0.12±0.004	0.196±0.003	0.414±0.007	0.449±0.01	0.163±0.002	0.356±0.01
		ETS	0.581±0.003	0.596±0.004	0.301±0.009	0.116±0.018	0.47±0.002	**0.523**±0.003	0.31±0.077	0.141±0.003
		GATS	0.581±0.003	0.596±0.004	0.185±0.018	0.229±0.039	0.47±0.002	0.521±0.011	0.211±0.004	0.308±0.011
		IRM	0.582±0.002	0.597±0.002	0.125±0.001	0.102±0.002	0.469±0.001	0.522±0.004	0.194±0.005	0.13±0.004
		Orderinvariant	0.581±0.003	0.592±0.002	0.226±0.024	0.213±0.049	0.47±0.002	0.498±0.027	0.318±0.042	0.196±0.027
		Spline	0.571±0.003	0.595±0.003	0.080±0.004	**0.068**±0.004	0.459±0.003	0.52±0.004	0.158±0.01	**0.098**±0.004
		VS	0.581±0.003	0.596±0.004	0.306±0.004	0.127±0.002	0.47±0.001	0.522±0.005	0.345±0.005	0.146±0.005
Cora	Word	×	0.607±0.003	0.628±0.001	0.284±0.009	0.111±0.013	0.603±0.004	0.633±0.031	0.263±0.004	0.118±0.019
		CAGCN	0.607±0.002	0.628±0.002	0.138±0.011	0.236±0.019	0.603±0.004	0.634±0.035	0.129±0.009	0.253±0.035
		Dirichlet	0.579±0.007	0.588±0.006	0.105±0.011	0.168±0.005	0.562±0.007	0.578±0.007	0.095±0.006	0.269±0.007
		ETS	0.607±0.002	0.628±0.002	0.282±0.002	0.11±0.003	0.603±0.004	0.634±0.013	0.243±0.023	0.106±0.013
		GATS	0.607±0.002	0.628±0.002	0.166±0.009	0.261±0.028	0.603±0.004	0.635±0.037	0.16±0.015	0.293±0.037
		IRM	0.608±0.001	0.63±0.002	0.115±0.002	0.088±0.003	0.602±0.003	0.635±0.004	0.106±0.002	0.098±0.004
		Orderinvariant	0.607±0.002	0.624±0.002	0.174±0.024	0.201±0.061	0.603±0.004	0.621±0.076	0.154±0.022	0.202±0.076
		Spline	0.598±0.005	0.629±0.002	0.073±0.002	**0.062**±0.005	0.591±0.002	0.635±0.004	0.063±0.006	**0.053**±0.004
		VS	0.607±0.001	**0.63**±0.002	0.283±0.003	0.111±0.003	0.603±0.004	**0.636**±0.003	0.261±0.005	0.119±0.003
CBAS	Color	×	0.83±0.014	0.829±0.011	0.169±0.013	0.151±0.014	0.703±0.015	0.746±0.027	0.266±0.02	0.169±0.018
		CAGCN	0.83±0.013	0.83±0.013	0.137±0.011	0.143±0.022	0.703±0.019	0.749±0.033	0.25±0.021	0.186±0.017
		Dirichlet	0.801±0.02	0.806±0.008	0.161±0.012	0.17±0.01	0.671±0.018	0.771±0.03	0.241±0.029	0.217±0.017
		ETS	0.83±0.013	0.827±0.014	0.146±0.013	0.164±0.007	0.703±0.019	0.76±0.037	0.28±0.023	0.176±0.019
		GATS	0.83±0.013	0.83±0.021	0.16±0.009	0.173±0.021	0.703±0.019	0.751±0.016	0.236±0.039	0.16±0.015
		IRM	0.829±0.013	0.839±0.015	0.142±0.009	**0.133**±0.006	0.72±0.019	0.803±0.04	0.207±0.035	**0.158**±0.017
		Orderinvariant	0.83±0.013	0.803±0.008	0.174±0.006	0.173±0.009	0.703±0.019	0.766±0.045	0.261±0.017	0.194±0.031
		Spline	0.82±0.016	0.824±0.011	0.159±0.009	0.16±0.014	0.683±0.019	0.786±0.038	0.225±0.034	0.179±0.035
		VS	0.829±0.012	**0.840**±0.011	0.166±0.011	0.146±0.012	0.717±0.019	**0.809**±0.008	0.242±0.019	0.182±0.014

- **Multi-class isotonic regression (IRM)** (Zhang et al., 2020) is a multiclass generalization of the famous isotonic regression method (Zadrozny & Elkan, 2002)): it ensembles predictions and labels, then learns a monotonically increasing function to map transformed predictions to labels.
- **Order-invariant calibration** (Rahimi et al., 2020) uses a neural network to learn an intra-order-preserving calibration function that can preserve a model's top-k predictions.
- **Spline** calibration instead uses splines to fit the calibration function (Gupta et al., 2021).
- **Dirichlet calibration** (Kull et al., 2019) models the distribution of outputs using a Dirichlet distribution, using simple log-transformation of the uncalibrated probabilities which are then passed to a regularized fully connected neural network layer with softmax activation.

For node classification, some graph-specific post-hoc calibration methods have been proposed. **CaGCN** (Wang et al., 2021) uses the graph structure and an additional GCN to produce node-wise temperatures. GATS (Hsu et al., 2022) extends this idea by using graph attention to model the influence of neighbors' temperatures when learning node-wise temperatures. We use the post hoc calibration baselines provided by Hsu et al. in our experiments.

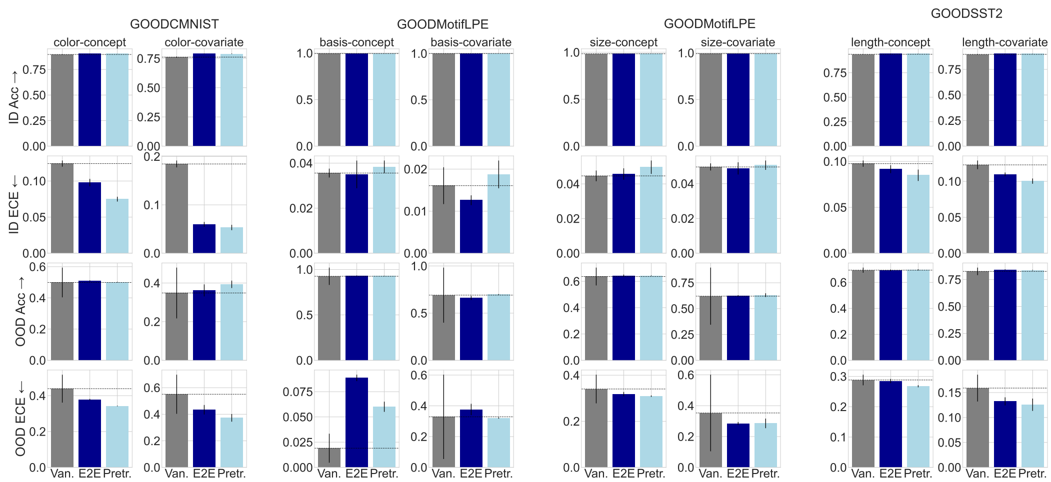

Figure 6: Results of applying G-ΔUQ to pretrained models vs. in training, on in-distribution and out-of-distribution accuracy and calibration error. Pretraining is a competitive strategy by all metrics.

All of the above methods, and others, may be applied to the output of any model including one using G-ΔUQ. As we have shown, applying such post hoc methods to the outputs of the calibrated models may improve uncertainty estimates even more. Notably, calibrated models are expected to produce confidence estimates that match the true probabilities of the classes being predicted (Naeini et al., 2015; Guo et al., 2017; Ovadia et al., 2019). While poorly calibrated CIs are over/under confident in their predictions, calibrated CIs are more trustworthy and can also improve performance on other safety-critical tasks which implicitly require reliable prediction probabilities (see Sec. 5). We report the top-1 label expected calibration error (ECE) (Kumar et al., 2019; Detlefsen et al., 2022). Formally, let p_i be the top-1 probability, c_i be the predicted confidence, b_i a uniformly sized bin in $[0, 1]$. Then, $ECE := \sum_i^N b_i \|(p_i - c_i)\|$.

A.8 Details on Generalization Gap 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: $GEPError := ||\mathrm{Acc}_{target} - \frac{1}{|X|}\sum_i \mathbb{I}(\mathrm{S}(\bar{\mathrm{x}}_i; \mathrm{F}) > \tau)||$ where τ is tuned by minimizing GEP error on the validation dataset. 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. The MAE between the estimated error and true error is reported on both in- and out-of -distribution test splits provided by the GOOD benchmark.

A.9 Additional Study on Pretrained G-ΔUQ

For the datasets and data shifts on which we reported out-of-distribution calibration error of pretrained vs. in-training G-ΔUQ earlier in Fig. 3, we now report additional results for in-distribution and out-of distribution accuracy as well as calibration error. We also include results for the additional GOODMotif-basis benchmark for completeness, noting that the methods provided by the original benchmark Gui et al. (2022) generalized poorly to this split (which may be related to why G-ΔUQ methods offer little improvement over the vanilla model.) Fig. 6 shows these extended results. By these additional metrics, we again see the competitiveness of applying G-ΔUQ to a pretrained model versus using it in end-to-end training.