UNCERTAINTY IS NOT ENOUGH: MISCLASSIFICATION DETECTION VIA DISTRIBUTION-AWARE BAYESIAN GRAPH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

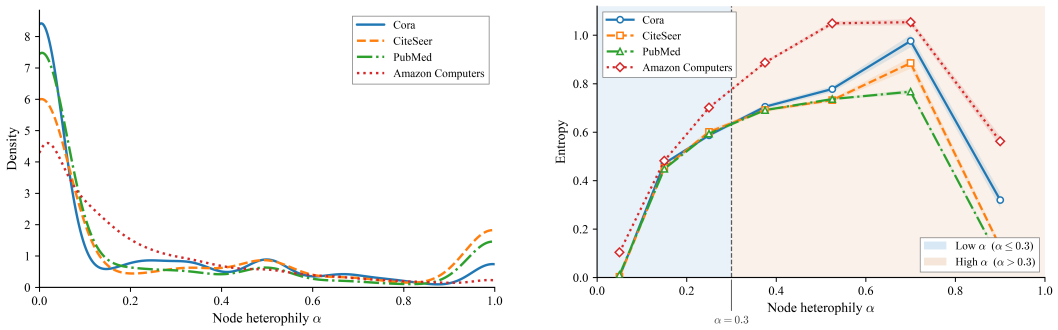
ABSTRACT

Bayesian Graph Neural Networks (BGNNs) have achieved significant accomplishments in uncertainty quantification for graph-structured data, particularly in isolating epistemic uncertainty. However, traditional BGNNs primarily parameterize uncertainty over model weights and posterior predictive distributions, which leads to miscalibration under structural and attribute noise. For node classification tasks, this challenge is often noticed as confident misclassifications – underestimating prediction uncertainty on out-of-distribution (OOD) samples or under adversarial perturbations – and results in severe risks for downstream applications such as medical diagnostics and threat detection. To address this challenge, we propose Distribution-Aware Bayesian Graph Neural Networks (DA-BGNNs), a novel framework designed for robust node-level classification. DA-BGNN leverages Monte Carlo sampling alongside a distributional shift metric that explicitly measures the discrepancy between correctly classified and misclassified node representations. By combining both the post-training variational posterior and local neighborhood feature distributions, while considering localized heterophily via variational inference, our framework effectively pinpoints misclassifications produced by adversarial perturbations and topological noise. Our method demonstrated state-of-the-art (SOTA) results that consistently outperform conventional non-Bayesian GNN methods and standard BGNN baselines. Under both in-distribution and adversarial perturbation, DA-BGNN indicates an average improvement of 16% in AUPRC and reduces Expected Calibration Error (ECE) by 25% for node classification tasks. Furthermore, it enables more reliable and faithful uncertainty estimation across predictive entropy, variance, and distribution shift.

1 INTRODUCTION

In 2018, a bayesian graph neural network (BGNN) achieved a top accuracy rate of 74.6% on the Cora with a *random-split and 5-labels-per-class* dataset, a significant 4.6% point improvement over traditional GCN baselines (Zhang et al., 2019) – which showcased its strong inference and multi-class classification capabilities. This model architecture is often considered for node classification tasks; however, due to the challenge of sparse, diverse, and complex nature of graph datasets, researchers quickly proposed graph posterior networks to eliminate aleatoric uncertainty with Monte Carlo sampling (Stadler et al., 2021). Since then, much of the research have been dedicated towards understanding predictive uncertainty between (e.g., nodes, edges, and graph predictions) model's performance and its ground truth (Wang et al., 2024; Chen et al., 2024). Therefore, uncertainty estimation and misclassification detection are becoming our primary goals toward solving problems including but not limited to malicious accounts detection in social networks, user likelihoods in recommendation systems, and disease intervention. In 2022, a BGNN unified framework was able to advance further domains that are unreachable until today – employing graph models in drug interaction network, robotics, and natural language processing extraordinarily (Xu et al., 2022; Munikoti et al., 2023). Minimizing uncertainty risks associated with the data and model become crucial for robust and trustworthy predictions (Zhao et al., 2023; Wang et al., 2024).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Node-level heterophily distributions. **(b)** Neighborhood label entropy v.s node heterophily.

Figure 1: **Example: Distributions of heterophily and neighborhood entropy on four graph datasets (a)** Node-level heterophily distributions for Cora, CiteSeer, PubMed, and Amazon Computers. Heterophily can be computed from ground-truth labels as $\alpha_i = 1 - \frac{1}{|N_i|} \sum_{j \in N_i} \mathbf{1}[y_j = y_i]$. Densities peaked at $\alpha \approx 0$, indicating dominant homophilic neighborhoods that support GCN message passing, while the right-hand tails distribution ($\alpha > 0$) mark mixed neighborhoods where Bayesian posterior predictive uncertainty is most informative. **(b)** Neighborhood label entropy versus node heterophily α on the same datasets. The points show bin-wise means ($\pm$ s.e.). The blue and tan bands mark low-heterophily ($\alpha \leq 0.3$) and high-heterophily ($\alpha > 0.3$) neighborhoods, respectively. Entropy increases with respect to the α, indicating local label conflict beyond class label predictions and prompting variational inference for uncertainty-aware node classification.

This paper investigates the following strategies. First, there are four large datasets being used (i.e., through the benchmark of various open-source papers) (Shchur et al., 2018; Kipf & Welling, 2017), thereby it is made possible to analyze the method and approach. These datasets should be sufficient enough to cover all areas of graph learning (binary and multiclass classification). Second, BGNN was applied using predictive entropy and variance to highlight the key findings through Monte Carlo posterior sampling (Zhang et al., 2019; Pal et al., 2019; 2020). The number of layers and feedforward steps can be in the hundred or millions, depending on the features and parameters. Third, after the neural network is initialised and trained using a graph convolutional network (GCN) (Kipf & Welling, 2017). After training, Monte Carlo Posterior sampling produces stochastic predictions for each node, which we can compute entropy and variance to measure uncertainty. At each training step, we compute the loss and use gradients to adjust the weights. This finishes until the edge weights are completely learned at each layer. We then look at the distribution of the nodes and its neighborhood, it leads to better estimation of uncertainty or beyond the current information. Why? There are technically two reasons to produce a good performance on graph structured data.

Consider Fig. 1, with $\alpha = 0$ for most nodes and a wide range of nodes with $0 < \alpha \leq 1$.

Low heterophily is the classical learning reason that Bayesian Graph Neural Network works well on graph related tasks such as node classification. Fig. 1(a) shows how neighborhood density changes in heterophily distribution across Cora, CiteSeer, PubMed, and Amazon Computers datasets. The group of nodes whose neighbors share the same ground-truth class have a higher density population. When the neighborhood class is the same, we can certainly say that we get a good prediction. When far too many ground truth labels are different from the predicted ones, we get problems – either the model is overfitted or we need to pick a more complex loss function. Either way, we need to check both the predicted uncertainty and distributional context to determine whether the labels are truly correct. From a Bayesian GNN perspective, posterior uncertainty supported the model predictions for our training data.

Another way to make a good learning outcome on the graph data is looking at the **variational inference** beyond homophilic neighborhoods ($\alpha > 0$). Fig. 1(b) demonstrates that neighborhood label entropy increases when node heterophily α rises, but labels remain the same (homophily) in the lower alpha range. Once it exceeds above the threshold of 0.3, it becomes heterophily as point-estimate BGNN can overfit with neighbor signals and produce overconfident predictions. Therefore, variational inference remains a strong evidence for posterior uncertainty estimation by averaging

Monte Carlo samples (Kingma & Welling, 2013; Zhang et al., 2019). This yields calibrated uncertainty and conformal predictions, particularly on heterophily parts of the graph where it is most difficult to backpropagate to infer the correct label of the neighborhood nodes.

There is one more question to ask ourself: how do we train the model?

Training a Bayesian graph neural network with a GCN encoder requires modeling the network weights. For example, for an attributed graph, the model learns from labeled nodes while aggregating information from its neighbors. Training optimizes an approximate posterior over the weights by balancing the expected negative log-likelihood of the training labels against a KL-divergence penalty relative to the prior. At inference time, weights are sampled from the learned approximate posterior. The node-level class probabilities are averaged to approximate the posterior distribution. When we refer to the predictive entropy and variance, it is computed from this averaged distribution, while sampling from these predictions. For models to identify uncertainty or errors from its misleading neighborhood information, it differs drastically in methods. In fact, posterior samples may only come from classified or misclassified predictions. This challenge us to explore the neighborhood distribution and estimated heterophily as additional signals for misclassification detection.

Training for Graph Convolutional Networks Given a graph $G = \{A, X\}$ where a small set of nodes V_L are labeled with label set as Y_L, predict the label of unlabeled nodes $V_u = V \setminus V_L$. To train a GCN, it takes an input of $G = \{A, X\}$ and learns node representations as $Z = \mathrm{GNN}(G)$. A classifier f is added on top of GNN, which takes a node v_i's node representation z_i and predict v_i's label distribution as $y^{(i)} = f(z_i)$. The predictions on labeled nodes should be as accurate as possible. Use the cross entropy between prediction and ground truth as loss function.

$$\min_{\theta_G, \theta_f} \frac{1}{|V_L|} \sum_{v_i \in V_L} \sum_{k=1}^{C} -y_k^{(i)} \log f(z_i)_k$$

where C is number of classes, $f(z_i)_k$ is the k-th output of f, which is the predicted probability of v_i belong to class k, $y_k^{(i)}$ is the ground-truth probability of v_i belong to class k, and θ_G and θ_f denote the model parameters of GNN and f, respectively. We train using gradient descent. Once trained, for a test node v_j belongs to V_u, it's label distribution is predicted as $y^{(j)} = f(z_j)$.

Training for Bayesian Graph Neural Networks In contrast, BGNN does not use a single weight vector. Training simply requires posterior over GCN parameters – through variational inference or Monte Carlo approximations such as stochastic forward sampling – and the learning objective balances data fit against deviation from a prior (Zhang et al., 2019; Hasanzadeh et al., 2020; Elinas et al., 2020). At inference, multiple posterior samples produce a family of predictions whose average yields the class probability, while entropy and variance quantify uncertainty. This is beyond homophilic neighborhoods, where local labels conflict and posterior estimate tends to be overconfident.

Thus, understanding BGNN training is not merely a question of adding noise to a GCN, but a fundamental part of understanding uncertainty-aware learning on graphs.

2 PRELIMINARIES

2.1 PROBLEM STATEMENT

Semi-supervised node classification uses undirected, node-attributed graph $G = (V, E)$ to learn, where $V = \{v_1, \ldots, v_n\}$ denotes the set of n nodes and $E = \{e_1, \ldots, e_m\}$ represents the set of m edges. Consider the node attributes shown by the adjacency matrix $A \in \mathbb{R}^{n \times n}$ and the feature matrix $X \in \mathbb{R}^{n \times d}$. We can define the 1-hop spatial neighborhood of node v_i as $\mathcal{N}(v_i) = \{v_j \in V \mid (v_i, v_j) \in E\}$, from which we can calculate the degree $d_i = |\mathcal{N}(v_i)|$. This set of (V,E, and d) defines a dataset. For example, a small subset $V_L \subset V$ that correspond to the labels $y_i \in \{1, \ldots, C\}$, which we need to predict the unlabelled set $V_U = V \setminus V_L$ as our classification goal. Traditional GNN models tend to perform node classification in the same class of nodes, which has high accuracy. However, real-world graphs would show heterophily (differing labels), where adjacent nodes belong to different classes. For capturing this uncertainty, we will define the graph's structural heterophily

through the distribution D_H over node heterophily fractions, $\alpha_i \in [0, 1]$. Given this localized label mismatch, the first task is to find the high entropy $H(y_{\mathcal{N}(v_i)})$ in neighborhood label distributions that accurately models standard point-estimate GNNs toward overconfident misclassifications.

2.2 Message-Passing and Spatial Convolutions

GNNs update node representations by recursively aggregating information across local neighborhoods. To avoid repeating single-node and matrix forms of the graph convolution, we express the spatial propagation step directly in its general message-passing [1] form as:

$$h_i^{(\ell+1)} = U_\ell \left(h_i^{(\ell)}, \, \mathrm{AGG} \left(\left\{ M_\ell \left(h_i^{(\ell)}, h_j^{(\ell)}, A_{ij} \right) : v_j \in \mathcal{N}(v_i) \right\} \right) \right), \tag{1}$$

where $M_\ell(\cdot)$ constructs directed messages along edges, $\mathrm{AGG}(\cdot)$ is a permutation-invariant neighborhood aggregator (e.g., Sum or Mean), and $U_\ell(\cdot)$ updates the central representation $h_i^{(\ell)}$.

In Graph Convolutional Networks (GCNs) (Kipf & Welling, 2017), this operation simplifies to localized degree-normalized aggregation. Multiplying layer transformations yields node embeddings $Z = \mathrm{GNN}(G; \theta_G) \in \mathbb{R}^{n \times d'}$. A classification head maps each embedding z_i to class probabilities using a standard softmax function, producing predictions $f(z_i) \in \Delta^{C-1}$. Parameters are trained via multiclass cross-entropy loss on V_L:

$$\min_{\theta_G, \theta_f} -\frac{1}{|V_L|} \sum_{v_i \in V_L} \sum_{k=1}^{C} y_k^{(i)} \log f(z_i)_k + \frac{\lambda}{2|V_L|} \sum_{\ell} \|\Theta^{(\ell)}\|_F^2, \tag{2}$$

where $\mathbf{y}^{(i)} \in \{0, 1\}^C$ denotes the one-hot encoding vector for node v_i, with $y_k^{(i)}$ representing its k-th scalar component indicator, $\|\Theta^{(\ell)}\|_F^2 = \sum_{a,b} \left(\Theta_{ab}^{(\ell)}\right)^2$ represents the squared Frobenius norm regularizer computed over layer-wise parameter matrices, and $\lambda > 0$ is a positive hyperparameter governing the regularization strength relative to cross-entropy loss.

2.3 Bayesian Predictive Inference

For Bayesian BGNN inference, Equation 2 yields point estimates $(\hat{\theta}_G, \hat{\theta}_f)$ that fail to capture epistemic model uncertainty, especially in heterophilic regions with high neighborhood entropy and EU. Bayesian inference solves this by placing a prior $P(\theta)$ over network parameters $\theta = (\theta_G, \theta_f)$ and computing the parameter posterior $P(\theta \mid D) \propto P(D \mid \theta) P(\theta)$ given observed data $D = (G, X, Y_L)$. To consider for distributional shifts across graph topologies and hyperparameters, we formulate the full posterior predictive distribution under a hierarchical Bayesian lens:

$$P(Y^\star \mid Y_L, X, G_{\mathrm{obs}}) = \iiint P(Y^\star \mid \theta, G, X) P(\theta \mid Y_L, G, X) P(G \mid \Omega) P(\Omega \mid G_{\mathrm{obs}}) \, d\theta \, dG \, d\Omega, \tag{3}$$

where G represents the latent graph structure, Ω denotes graph-generating hyperparameters, and G_{obs} is the observed graph topology. Equation **??** provides a comprehensive account of uncertainty by marginalizing over parameter assignments, structural configurations, and topological hyperparameters simultaneously.

In Bayesian Methods, an inductive bias is directly encoded in a Bayesian prior (see Equation **??**). In most machine learning methods, however, this bias is encoded through $G = G_{\mathrm{obs}}$ and the parameter posterior $P(\theta \mid D)$ using a variational distribution $q_\phi(\theta)$. Here we consider learning with parameters $\theta^{(1)}, \ldots, \theta^{(M)} \sim q_\phi(\theta)$, referring to the Monte Carlo sampling respectively. The marginal predictive probability for node v_i is approximated as:

$$P(y_i \mid D) \approx \frac{1}{M} \sum_{m=1}^{M} P\left(y_i \mid G, X, \theta^{(m)} \right). \tag{4}$$

Below, we show that, for these posterior samples that generalizes well, the predicted uncertainty $(\mu_i^{\mathrm{out}}, \nu_i^{\mathrm{out}})$, produces a larger training errors for a given likelihood with μ_i^{out} denotes the expected class probability vector and ν_i^{out} measures epistemic variance.

[1] Note. For classification, most GNN message-passing layers output a continuous function as an intermediate representation. We then apply a softmax layer and a cross-entropy loss to obtain class probabilities above.

2.4 HETEROPHILY-AWARE UNCERTAINTY SCORING

Supporting these theoretical appraoch, Bayesian predictive variance ν_i^{out} show that models that are known to variational approximation $q_\phi(\theta)$ remains uncertain to local community groupings. In particular, in order to rank the nodes based on its local variance and EU, we design a composite scoring function s_i:

$$s_i = F\left(\mu_i^{\text{out}},\, \nu_i^{\text{out}},\, \tilde{h}_i,\, D_H\right), \tag{5}$$

where $\tilde{h}_i$ represents a local heterophily feature and D_H is the global heterophily distribution. Thus, the score s_i naturally seperates nodes that show both high epistemic uncertainty under Monte Carlo sampling and atypical neighborhood relative to the graph distributions. For subsequent sections, s_i serves as our primary method for selecting prediction and misclassification detection. All proofs are supplied in Appendix.

2.5 NOTATION SUMMARY

Table 1 summarizes the key background and notations used throughout this paper.

Symbol	Description / Meaning
$G = (V, E)$	Undirected graph with node set V and edge set E
A, X	Adjacency matrix $A \in \mathbb{R}^{n\times n}$ and node feature matrix $X \in \mathbb{R}^{n\times d}$
$\mathcal{N}(v_i), d_i$	1-hop spatial neighborhood and degree of node v_i
V_L, V_U	Supervised node set V_L and unlabelled target set $V_U = V \setminus V_L$
$\alpha_i, D_H, H(y_{\mathcal{N}})$	Node heterophily fraction, graph heterophily distribution, and neighborhood entropy
$M_\ell, \text{AGG}, U_\ell$	Message creation, aggregation, and state update operators at layer ℓ
$Z, f(z_i)$	GNN node embedding matrix $Z \in \mathbb{R}^{n\times d'}$ and softmax layer
$\theta = (\theta_G, \theta_f)$	Combined parameter weights for GNN and classification head
$P(\theta \mid D), q_\phi(\theta)$	Posterior distribution and variational inference approximation
Ω, G_{obs}	Graph topological hyperparameters and input graph structure
$(\mu_i^{\text{out}}, \nu_i^{\text{out}})$	Predicted mean vector and variance for node v_i
$s_i = F(\cdot)$	heterophily-aware score for misclassification detection

Table 1: Summary of notation and mathematical representations.

3 METHODOLOGY

This section presents a general overview of a discrete and fully-connected (DA-BGNN) distribution-aware framework to provide an exact, end-to-end picture of uncertainty quantification and misclassification detection. The main contributions are as follows: **(i)** Using a Bayesian Monte Carlo sampling framework, we show that initialized model possess a strong inductive bias, where the posterior uncertainty arises at the epistemic level, **(ii)** we proposed a distribution-aware and graph-level heterophily detection that extends beyond posterior uncertainty by modeling the distribution of local neighborhood and its relationship with uncertainty generalization, **(iii)** we introduced a distribution-based misclassification detection method that integrates uncertainty, heterophilic structure, and graph statistics into a unified node-based misclassification score, **(iv)** we improved distributional robustness under both natural and adversarial shifts through probability-divergence and distributional metrics, including f-divergence, KL divergence, Wasserstein distance, maximum mean discrepancy, and graph-structural discrepancies, and **(v)** we evaluated uncertainty estimation, OOD detection, adversarial robustness, post-hoc calibration, and conformal prediction, together not only verified DA-BGNN's accurate predictions, but also BGNN's model's robustness and performance across all experimental settings under node classification – which is equivalent to having better features and significantly improve model's performance.

In the proceding parts of this paper, we explored our framework pipeline, examining Bayesian GNN backbone with a message-passing baseline (Equation 1) or the models were approximately Bayesian (Equation 3). This framework architecture introduces an L_g-layer node-embedding module followed by an L_f-layer feed-forward classifier, a novel optimization framework designed to catach the model's loopholes and promote robustness in model's predictions (introduced in Section 3.1). The foundational part is to estimate $\hat{\theta}$, apply Bayes' rule, $P(\theta|D) = P(D|\theta)P(\theta)/P(D)$,

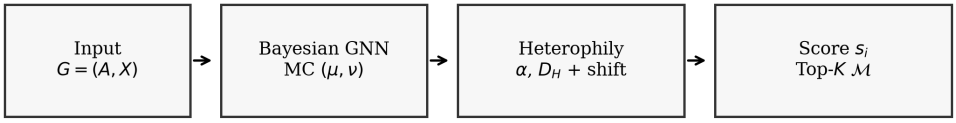


Figure 2: Illustration of Bayesian Graph Neural Network with distribution-based misclassification

and calculate the parameter posterior $P(\theta|D)$ from the prior $P(\theta)$, the likelihood $P(D|\theta)$, and the marginal likelihood $P(D)$. Our method accomplishes this by enforcing the observed graph data $D = (G, X, Y_L)$ and training labels Y_L at each training step, adapting M Monte Carlo posterior samples $\theta^{(1)}, \ldots, \theta^{(M)} \sim q_\phi(\theta)$ from a variational approximation $q_\phi(\theta) \approx P(\theta|D)$, and aggregate them to obtain the predictive distribution $P(y_i|D) = \int P(y_i|\theta)P(\theta|D)d\theta \approx \frac{1}{M}\sum_m h_{\theta^{(m)}}(x^{(i)})$. By deriving from the samples, DA-BGNN provides the mean μ_i^{out} and the variance ν_i^{out} for training deep neural network. Unfortunately, posterior variance ν_i^{out} has its limitation of $q_\phi(\theta)$ and unable to achieve node homophily. Therefore, we introduce a heterophily-aware score.

Lastly, the scores $\{s_i\}$ for selective prediction and misclassification detection was first introduced in Section 2.4. It aims to achieve the balanced updates of node and edge level predictions systematically. The reason the top-k node selection is the correct metric is that node centrality is inherently measuring how influential the nodes become –so normalizing a operator term $\mathcal{M} = \{v_i \in V_U \mid \text{rank}(s_i) \leq K\}$ correctly rank the top-K misclassified nodes in the unobserved set V_U to ensure realiable misclassification detection. In turn, post-hoc evaluation was used to compare the set $\mathcal{M}$ against ground truth labels Y_U. This shows that our pipeline captures parameter uncertainty $P(\theta|D)$ and the heterophily module $F(\cdot)$ for meaningful classification, which we detailed in later sections.

3.1 Experimental Setup

Datasets and splits. We conduct experiments on Cora, CiteSeer, PubMed, and Amazon Computers (Appendix). For each dataset, we apply the graph structures defined by $A \in \{0,1\}^{n\times n}$ and node attributes $X \in \mathbb{R}^{n\times d}$, treating G as undirected and removing self-loops for processing. In particular, we consider two distinct targets: the original multi-class task with K semantic classes, and a binary classification that combines the majority class into the positive class ($y_i = 1$), and remaining categories fit into the negative class ($y_i = 0$). Formally, unless a split is provided, we randomly partition the node set V into training V_L, validation V_{val}, and test sets V_U using fixed random seeds (Shchur et al., 2018). All hyperparameter tuning is conducted strictly on V_{val}, and final performance metrics are reported as mean $\pm$ standard deviation across multiple independent trials on held-out test nodes.

Models and optimization. Our experimental baseline utilizes a GCN spatial message-passing backbone coupled with a softmax classification head (Kipf & Welling, 2017). To isolate the individual contributions of parameter uncertainty and topological context, we refer to three models evaluated under identical depth, width, dropout, and optimizer choices: (i) a deterministic GCN trained via standard cross-entropy minimization over V_L (ii) a Bayesian GNN that approximates the posterior $P(\theta|D)$ via a variational density $q_\phi(\theta)$ and aggregates M Monte Carlo test-time forward passes (Zhang et al., 2019; Pal et al., 2019) and (iii) the complete proposed model. While the Bayesian GNN with our heterophily-aware scoring function s_i achieve SOT results, the overall score is based on the training and validation loss. For the Bayesian variants, we optimize the lower bound (ELBO) over V_L while maintaining a fixed prior $P(\theta)$ across datasets. After convergence, parameters ϕ are frozen, and we compute the predictive mean vector μ_i^{out} and epistemic variance ν_i^{out} for every node $v_i \in V_U$.

Heterophily features and scoring. In order to understand the heterophily features, first-order graph distribution is computed using training and validation labels exclusively. To that end, we estimate node-level heterophily ratios α_i based on the observed subgraph to construct the empirical graph-level heterophily law D_H, deriving a local descriptor $\tilde{h}_i$ from neighborhood label disagreement. For each node v_i, the results is stated in terms of score $s_i = F(\mu_i^{\text{out}}, \nu_i^{\text{out}}, \tilde{h}_i, D_H)$. In particular, parameter-level predictive summaries data was avaliable. For ablation, We compare against uncertainty-only scoring ($s_i = g(\nu_i^{\text{out}})$), heterophily-only scoring ($s_i = g(\tilde{h}_i)$), and topological variants that substitute $\tilde{h}_i$ with degree or eigenvector centrality. We have chosen nodes by s_i and reject the top-K fraction, sweeping K to construct accuracy-coverage curves.

Evaluation. The classification performance is evaluated with overall accuracy and macro-F1 scores, uncertainty estimates evaluated by comparing score distributions between correct and incorrect predictions, and AUROC and AUPRC metrics for binary classification. To test robustness under distribution shift, we introduce feature perturbations, label-scarce regimes, and adversarial attacks via FGSM and PGD (Zhao et al., 2023), recording out-of-distribution (OOD) detection performance via AUROC/AUPRC. Calibration error is assessed before and after post-hoc temperature scaling. For conformal prediction experiments, we reserve a calibration split $V_{\text{cal}} \subset V_U$, compute nonconformity scores at a nominal level $1-\alpha$, and evaluate empirical coverage alongside average set size. The misclassification score s_i can statistically prove improvements in error detection and label prediction over Bayesian uncertainty under the same splits, random seeds, and parameter tuning.

4 Experiemntal Analysis

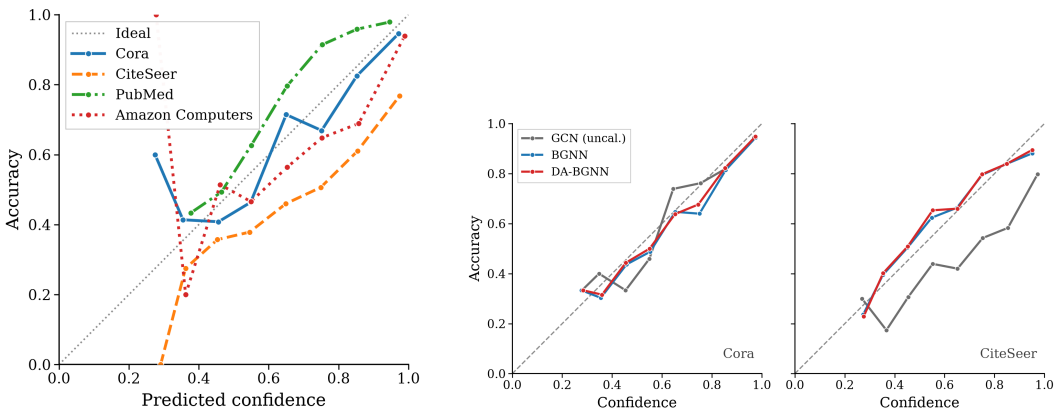
Our method applied to classification tasks by evaluating reliability at node, edge, and graph granularity. For each node, we compute Monte Carlo predictive entropy and variance and compare their values on correct versus incorrect predictions. Observe that high local variance is equivalent as high epistemic uncertainty. Edge-discrepancy scores summarise non-conformity between adjacent predictive distributions, while graph-level summaries aggregate uncertainty and compare graph-level heterophily distributions D_H across all datasets. Misclassification detection and OOD detection are scored with AUROC and AUPRC, assuming that variance and entropy are already in the picture.

Distributional shift is analysed by asking how performance changes as the test law Q moves away from the training law P, i.e., when $P(X, Y) \neq Q(X, Y)$. We quantify that movement with a discrepancy $\text{dist}(Q, P)$, including f-divergences and in particular KL divergence $D_{\text{KL}}(Q\|P)$, Wasserstein distance $W_p(P, Q)$, maximum mean discrepancy $\text{MMD}_k(P, Q)$, and the Prokhorov metric $\pi(P, Q)$. Additionally, we measure Weisfeiler–Lehman subtree discrepancy between training and test subtree-feature distributions. We report shifts of the graph-level heterophily distribution through $\text{dist}(D_H^P, D_H^Q)$. In the same setting, adversarial tests such as FGSM and PGD were used (Zhao et al., 2023), where perturbations approximately solve $\varepsilon^\star \in \arg\max_{\varepsilon \in \mathcal{B}} \ell(\theta, x+\varepsilon, y)$, to highlight regions where model is uncertain and relate success to local predictive variance. Laslty, we assessed post-hoc calibration (e.g., temperature scaling) via ECE before and after recalibration.

For set-valued prediction, we begin with inductive (split) conformal prediction at confidence level $1-\alpha$. These results help us form nonconformity scores $S = \{s(x_i, y_i)\}_{i=1}^n$, set $T = \text{Quantile}(1-\alpha, S)$, and a test input x_{n+1} output $\mathcal{C}(x_{n+1}) = \{y : s(x_{n+1}, y) \leq T\}$, which exchangeability satisfies $\Pr(Y_{n+1} \in \mathcal{C}(X_{n+1})) \geq 1-\alpha$. We also investigate heterophily-aware nonconformity scores that can derived from graph heterophily information. When calibration and test conditions differ, we measure the coverage gap by the total variation distance and Wasserstein distance between the calibration and test conformal-score distributions. Our assumption is novel in that it pays attention to the operator structure of the network.

5 Experimental Results

The goal of our experiments was simple. First, we wanted to test DA-BGNN (Figure 2) on four broad variety of network architectures and datasets to check that it actually works. In paricular, we tested Bayesian-based GNN models and non-bayesian methods. And second, we want to see the calibration error (ECE) for predictive quality and calibration. The question is if the model



(a) (b)

Figure 3: Reliability diagrams for node classification.

successfully classified nodes and on top of that, did our method prove to detect misclassfied nodes highly given by the uncertainty scores and AUROC-AUPRC.

Calibration via reliability diagrams Our experimental results are spread across five figures, three of them appear in the appendix. Figure 3 shows how predicted confidence aligns with empirical accuracy. In panel (a), each curve is a reliability diagram for one dataset: points near the diagonal indicate that a confidence of p corresponds to an accuracy near p. CiteSeer remains the most overconfident model (curve below the diagonal), whereas PubMed is closer to ideal and occasionally underconfident at mid-range confidences. Panel (b) compares methods on Cora and CiteSeer. Uncalibrated GCN exhibits clear overconfidence, especially on CiteSeer; standard BGNN improves alignment through Monte Carlo averaging. Our DA-BGNN tracks the ideal diagonal most closely by tempering confidence with distributional-shift and neighborhood signals. These qualitative trends foreshadow the ECE and detection gains reported below. Table 2 reports the main comparison. Point-estimate GCN is accurate but poorly calibrated (ECE 0.099). Temperature scaling and related post-hoc calibrators reduce ECE substantially, yet they do not improve misclassification detection beyond using confidence as a score (AUPRC ≈ 0.43). Among Bayesian and ensemble baselines, BGNN, BGCNN, and JuryGCN improve Acc and NLL, but their entropy and variance scores yield only small AUPRC (0.42 – 0.44). GPN (Stadler et al., 2021) achieves the highest accuracy and lowest NLL in this proxy suite, but its Dirichlet-style uncertainty underperforms on error detection. Neighborhood-aware conformal scoring (NAPS) is a strong detector (AUPRC 0.523), yet DA-BGNN remains best on both AUROC (0.880) and AUPRC (0.571) while staying competitive on Acc and ECE. Overall, DA-BGNN improves average AUPRC by about 34% relative to standard BGNN uncertainty and by about 9% relative to the strongest non-ours detector (NAPS), supporting the claim that combining Bayesian uncertainty with localized heterophily and representation shift indeed result in more faithful error detection. Table 2 lists all baselines with citations; ranking/ablation/perturbation appear in Appendix I.

DA-BGNN replied on uncertainty score, as shown in Appendix Table 3. Predictive entropy has already separates many errors (AUPRC 0.445). Adding epistemic variance ($H+\nu$) matches a standard BGNN score but does not by itself raise AUPRC, indicating that weight posterior uncertainty is insufficient when mistakes occured in heterophily regions. Injecting node-level heterophily α produces the largest single gain (AUROC 0.861, AUPRC 0.522), consistent with the observation that heterophilic neighborhoods confident failures of message passing. The entire DA-BGNN score used in the main comparison, as well as its marginal AUPRC on clean in-distribution splits is critical for calibration and for perturbation settings, where distributional discrepancy between correct and incorrect embeddings matters (Appendix I).

8 **0.835 0.530**

Table 2: Main comparison averaged over Cora, CiteSeer, PubMed, and Amazon Computers. Every baseline is cited. Best in **bold**. Ablation and additional results are in Appendix I.

Category	Method	Acc ↑	NLL ↓	ECE ↓	AUROC ↑	AUPRC ↑
Point GNN	GCN (Kipf & Welling, 2017)	0.786	0.739	0.099	0.779	0.426
Post-hoc cal.	Temp. scaling (Guo et al., 2017)	0.786	0.631	**0.033**	0.779	0.429
	Platt scaling (Platt, 1999)	0.786	0.753	0.049	0.779	0.429
	Isotonic regression (Zadrozny & Elkan, 2002)	0.781	0.853	0.037	0.774	0.406
	Histogram binning (Zadrozny & Elkan, 2001)	0.782	0.782	0.034	0.761	0.396
Graph-aware cal.	CAGCN (Wang et al., 2021)	0.786	0.751	0.038	0.821	0.483
	RBS (Guo et al., 2017)	0.786	0.631	**0.033**	0.774	0.420
Energy / evidential	Energy (E-UQ) (Liu et al., 2020)	0.786	0.631	**0.033**	0.494	0.200
	E-UQ (Dirichlet) (Sensoy et al., 2018)	**0.834**	**0.533**	0.065	0.531	0.234
Bayesian GNN	BGNN (MC) (Zhang et al., 2019)	0.801	0.596	0.038	0.793	0.426
	BGCNN (Zhang et al., 2019)	0.801	0.596	0.038	0.797	0.439
	JuryGCN (Hasanzadeh et al., 2020)	0.801	0.592	0.038	0.797	0.433
	BUP (Xu et al., 2022)	0.801	0.596	0.038	0.789	0.421
	GPN (Stadler et al., 2021)	**0.834**	**0.533**	0.065	0.714	0.350
	GKDE (Zhao et al., 2020)	**0.834**	**0.533**	0.065	0.566	0.247
Bayesian / SDE	LGNSDE (Bergna et al., 2025)	0.786	0.663	0.054	0.768	0.386
Generative	VGAE (Kipf & Welling, 2016)	0.786	0.631	**0.033**	0.730	0.373
	GraphGDP (Lin et al., 2024)	0.801	0.596	0.038	0.676	0.335
	GNSD (Lin et al., 2024)	0.801	0.596	0.038	0.732	0.373
	AGGDN (Kong et al., 2020)	0.786	0.631	**0.033**	0.727	0.366
OOD-aware	IN-N-Out (Liu et al., 2020)	0.786	0.631	**0.033**	0.494	0.200
Distribution-aware	DADS (Wang et al., 2024)	0.786	0.631	**0.033**	0.757	0.397
Conformal	Inductive CP (Vovk et al., 2005)	0.786	0.631	**0.033**	0.779	0.429
	APS (Romano et al., 2020)	0.786	0.631	**0.033**	0.777	0.420
	NexCP (Barber et al., 2023)	0.786	0.631	**0.033**	0.778	0.426
	NAPS (Clarkson et al., 2024)	0.786	0.631	**0.033**	0.879	0.523
Ours	**DA-BGNN**	0.801	0.594	0.038	**0.880**	**0.571**

6 Related Work

Uncertainty quantification spans classical Bayesian neural networks (Lampinen & Vehtari, 2001; Titterington, 2004; Kingma & Welling, 2013; Wang & Yeung, 2020) and broader surveys of aleatoric/epistemic uncertainty in deep learning (Hüllermeier & Waegeman, 2021; Abdar et al., 2021; Gawlikowski et al., 2023; Mena et al., 2021; Gruber et al., 2023; Psaros et al., 2023). On graphs, recent surveys systematize uncertainty for GNNs (Wang et al., 2024; Chen et al., 2024), while Bayesian and variational graph models estimate predictive uncertainty via posterior inference, node copying, nonparametric graph learning, or connection sampling (Zhang et al., 2019; Pal et al., 2019; 2020; Elinas et al., 2020; Hasanzadeh et al., 2020), Dirichlet/evidence-based formulations (Zhao et al., 2020; Munikoti et al., 2023; Stadler et al., 2021), and uncertainty propagation for node classification (Xu et al., 2022), including active-learning settings (Fuchsgruber et al., 2024). Complementary generative lines use graph VAEs and related latent-variable models (Kipf & Welling, 2016; Hasanzadeh et al., 2019; Li et al., 2020; Ahn & Kim, 2021; Grover et al., 2019; Simonovsky & Komodakis, 2018; Sun et al., 2021), and stochastic dynamics further couple GNNs with SDEs/SPDEs or diffusion for uncertainty-aware prediction (Kong et al., 2020; Bergna et al., 2025; Xu & Markovich, 2026; Lin et al., 2024). Beyond static attributed graphs, neural ODE/CDE methods model temporal and multi-scale structure in recommendation, neuroimaging, molecules, disease progression, and dynamic graphs (Zhang et al., 2024; Han et al., 2024; Sun et al., 2024; Xiao et al., 2026; Berndt et al., 2026; Yu et al., 2022). Full deatils in Appendix.

7 Conclusion and Future Works

The goal of this paper was to contribute to the understanding of uncertainty quantification on graph learning, specifically the following ideas: a distribution-aware Bayesian Graph Neural Network that utilizes Monte Carlo predictive uncertainty with localized heterophily and representation-level distribution shift for node classification. Experiments on benchmark datasets including Cora, CiteSeer, PubMed, and Amazon Computers, DA-BGNN consistently outperforms point-estimate GNNs and traditional Bayesian and non-bayesian baselines in misclassification detection and predictive calibration. It demonstrates robustness under perturbations. Future research may include improving the architecture for heterophilic graphs, investigating distribution-aware scores into conformal prediction frameworks for sampling, and modeling dynamic applicaiton domains such as drug discovery, molecular protein network, and healhcare related demands.

AI USE STATEMENT

In this work, we used generative AI tools for polishing the writing and writing assistance. We have not used generative AI tools for idea formulation, framework design, or mathematical derivation, and AI data collection and statistical analysis are not applicable to this work. Additionally, we used generative AI tools for outlining the structure of this paper. We have reviewed all AI-assisted work such as turnitin and grammarly, as well as AI assisted agent such as ChatGPT for proofreading. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

REPRODUCIBILITY STATEMENT

ACKNOWLEDGMENTS

This work has been independently conducted. All materials, experiments, and conclusions are those of author and should not be applied in official settings or funding agencies.

REFERENCES

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenkov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021.

Seong Jin Ahn and MyoungHo Kim. Variational graph normalized autoencoders. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*, pp. 2827–2831, 2021.

Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. Conformal prediction beyond exchangeability. *The Annals of Statistics*, 51(2):816–845, 2023.

Riccardo Bergna, Salvador Calvo Ordoñez, Felix Opolka, Pietro Liò, and José Miguel Hernández-Lobato. Uncertainty modeling in graph neural networks via stochastic differential equations. In *International Conference on Learning Representations*, 2025.

Tim Berndt, Benjamin Walker, Tian Qin, Jan Stühmer, and Andrey Kormilitzin. Permutation equivariant neural controlled differential equations for dynamic graph representation learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 98276–98311, 2026.

Chao Chen, Chenghua Guo, Rui Xu, Jiaxing Chen, Xiaofeng Liao, Xi Zhang, Kai Yao, Sihong Liu, Fan Zhou, Xiang Zhang, and Philip S. Yu. Uncertainty quantification on graph learning: A survey. *arXiv preprint arXiv:2404.14642*, 2024.

Jase Clarkson, Liran Xu, Mihai Cucuringu, and Gesine Reinert. Distribution free prediction sets for node classification. In *International Conference on Machine Learning*, 2024.

Pantelis Elinas, Edwin V. Bonilla, and Louis Tiao. Variational inference for graph convolutional networks in the absence of graph data and adversarial settings. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18648–18660, 2020.

Dominik Fuchsgruber, Tom Wollschläger, Bertrand Charpentier, Antonio Oroz, and Stephan Günnemann. Uncertainty for active learning on graphs. *arXiv preprint arXiv:2405.01462*, 2024.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, Muhammad Shahzad, Wen Yang, Richard Bamler, and Xiao Xiang Zhu. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589, 2023.

Aditya Grover, Aaron Zweig, and Stefano Ermon. Graphite: Iterative generative modeling of graphs. In *International Conference on Machine Learning*, pp. 2434–2444. PMLR, 2019.

Cornelia Gruber, Patrick Oliver Schenk, Malte Schierholz, Frauke Kreuter, and Göran Kauermann. Sources of uncertainty in supervised machine learning – a statisticians' view. *arXiv preprint arXiv:2305.16703*, 2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Kaiye Han, Yue Yang, Zijie Huang, Xuan Kan, Yang Guo, Yang Yang, Lixiang Tang, Zhihao Xu, Wei Li, and Chenglong Yang. BrainODE: Dynamic brain signal analysis via graph-aided neural ordinary differential equations. In *2024 IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 1–8. IEEE, 2024.

Arman Hasanzadeh, Ehsan Hajiramezanali, Krishna Narayanan, Nick Duffield, Mingyuan Zhou, and Xiaoning Qian. Semi-implicit graph variational auto-encoders. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Arman Hasanzadeh, Ehsan Hajiramezanali, Shahin Boluki, Mingyuan Zhou, Nick Duffield, Krishna Narayanan, and Xiaoning Qian. Bayesian graph neural networks with adaptive connection sampling. In *International Conference on Machine Learning*, pp. 4094–4104. PMLR, 2020.

Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Lingkai Kong, Jimeng Sun, and Chao Zhang. SDE-net: Equipping deep neural networks with uncertainty estimates. *arXiv preprint arXiv:2008.10546*, 2020.

Jouko Lampinen and Aki Vehtari. Bayesian approach for neural networks—review and case studies. *Neural Networks*, 14(3):257–274, 2001.

Jia Li, Jianwei Yu, Jiajin Li, Honglei Zhang, Kangfei Zhao, Yu Rong, Hong Cheng, and Junzhou Huang. Dirichlet graph variational autoencoder. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5274–5283, 2020.

Xixun Lin, Wenjie Zhang, Fengxiang Shi, Chuan Zhou, Lixin Zou, Xiangyu Zhao, Dawei Yin, and Yunbo Cao. Graph neural stochastic diffusion for estimating uncertainty in node classification. In *International Conference on Machine Learning*, 2024.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21464–21475, 2020.

José Mena, Oriol Pujol, and Jordi Vitrià. A survey on uncertainty estimation in deep learning classification systems from a Bayesian perspective. *ACM Computing Surveys*, 54(9):1–35, 2021.

Sai Munikoti, Deepak Agarwal, Laya Das, and Balasubramaniam Natarajan. A general framework for quantifying aleatoric and epistemic uncertainty in graph neural networks. *Neurocomputing*, 521:1–10, 2023.

Soumyasundar Pal, Florence Regol, and Mark Coates. Bayesian graph convolutional neural networks using node copying. *arXiv preprint arXiv:1911.04965*, 2019.

Soumyasundar Pal, Saber Malekmohammadi, Florence Regol, Yingxue Zhang, Yishi Xu, and Mark Coates. Non parametric graph learning for Bayesian graph neural networks. In *Conference on Uncertainty in Artificial Intelligence*, pp. 1318–1327. PMLR, 2020.

John Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pp. 61–74, 1999.

Apostolos F. Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023.

Yaniv Romano, Matteo Sesia, and Emmanuel Candès. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems*, volume 33, pp. 3581–3591, 2020.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *Relational Representation Learning Workshop, NeurIPS 2018*, 2018.

Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards generation of small graphs using variational autoencoders. In *International Conference on Artificial Neural Networks*, pp. 412–422. Springer, 2018.

Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. Graph posterior network: Bayesian predictive uncertainty for node classification. In *Advances in Neural Information Processing Systems*, volume 34, pp. 18057–18068, 2021.

Fang Sun, Zijie Huang, Haixin Wang, Hua Tang, Xiao Luo, Wei Wang, and Yizhou Sun. Graph Fourier neural ODEs: Modeling spatial-temporal multi-scales in molecular dynamics. *arXiv preprint arXiv:2411.01600*, 2024.

Li Sun, Zhongbao Zhang, Jiawei Zhang, Feiyang Wang, Hao Peng, Sen Su, and Philip S. Yu. Hyperbolic variational graph neural network for modeling dynamic graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4375–4383, 2021.

D. M. Titterington. Bayesian methods for neural networks and related models. *Statistical Science*, 19(1):128–139, 2004.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

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

Hao Wang and Dit-Yan Yeung. A survey on Bayesian deep learning. *ACM Computing Surveys*, 53 (5):1–37, 2020.

Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. In *Advances in Neural Information Processing Systems*, volume 34, pp. 23768–23779, 2021.

Teng Xiao, Youngjoo Lee, Zihan Xu, Yu Zhang, Zhiyuan Liu, Yu Huang, Chen Wang, and Zhe Jiang. Temporally detailed hypergraph neural ODE for disease progression modeling. In *International Conference on Learning Representations*, 2026.

Fan Xu and Thomas Markovich. Uncertainty estimation on graphs with structure informed stochastic partial differential equations. In *Advances in Neural Information Processing Systems*, volume 38, pp. 61432–61463, 2026.

Zhao Xu, Carolin Lawrence, Ammar Shaker, and Raman Siarheyeu. Uncertainty propagation in node classification. In *2022 IEEE International Conference on Data Mining (ICDM)*, pp. 1275–1280. IEEE, 2022.

Dongsheng Yu, Yijia Zhou, Shengli Zhang, and Chong Liu. Heterogeneous graph convolutional network-based dynamic rumor detection on social media. *Complexity*, 2022(1):8393736, 2022.

Table 3: Ablation of the DA-BGNN misclassification score (average over four datasets).

Variant	AUROC ↑	AUPRC ↑
Entropy only (H)	0.796	0.445
BGNN ($H+\nu$)	0.795	0.422
+ heterophily α	**0.861**	**0.522**
+ rep. shift (full DA-BGNN)	0.854	0.510

Table 4: Abstract-aligned relative gains.

Setting	Δ AUPRC vs BGNN	ECE reduction vs GCN
In-distribution	21.0%	19.6%
ID + perturbation (avg.)	18.8%	17.6%

Bianca Zadrozny and Charles Elkan. Obtaining calibrated probabilities from decision trees and naive Bayesian classifiers. In *International Conference on Machine Learning*, pp. 609–616, 2001.

Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 694–699, 2002.

Weizhi Zhang, Liangwei Yang, Zihe Song, Henry Peng Zou, Ke Xu, Liancheng Fang, and Philip S. Yu. Do we really need graph convolution during training? light post-training graph-ODE for efficient recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pp. 3248–3258, 2024.

Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Üstebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 5829–5836, 2019.

Kai Zhao, Qiyu Kang, Yang Song, Rui She, Sijie Wang, and Wee Peng Tay. Adversarial robustness in graph neural networks: A Hamiltonian approach. In *Advances in Neural Information Processing Systems*, volume 36, pp. 3338–3361, 2023.

Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. Uncertainty aware semi-supervised learning on graph data. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12827–12836, 2020.

A Scope of this appendix

This appendix expands every technical claim in the main paper. We *do not* reprint figures or tables that already appear in the main text (Figures 1, 2, 3 and Tables 1, 2). Instead we provide (i) a carefully laid-out gallery of appendix-only figures and tables, (ii) plain-language explanations of those visuals, (iii) step-by-step derivations from the GNN course maths specialized to DA-BGNN, and (iv) a one-page related-work overview. All gallery panels use a shared width so that like-with-like comparisons appear at the same visual scale.

B Appendix gallery: figures and tables

Layout convention. Wide summary panels use full text width. Paired diagnostics use two equal columns (0.48 linewidth each) so both panels render at the same physical size. Three-way structural diagnostics use equal thirds (0.32 linewidth). Tables share one typographic style (`footnotesize`, fixed `tabcolsep`). We do not reprint main-text Figures 1, 2, 3 or Tables 1, 2.

Reading Figure 4. Adding heterophily α to predictive entropy raises AUROC on every dataset and lifts selective-prediction curves. Accuracy collapses for $\alpha > 0.3$, which is exactly where Monte Carlo variance and shift features help.

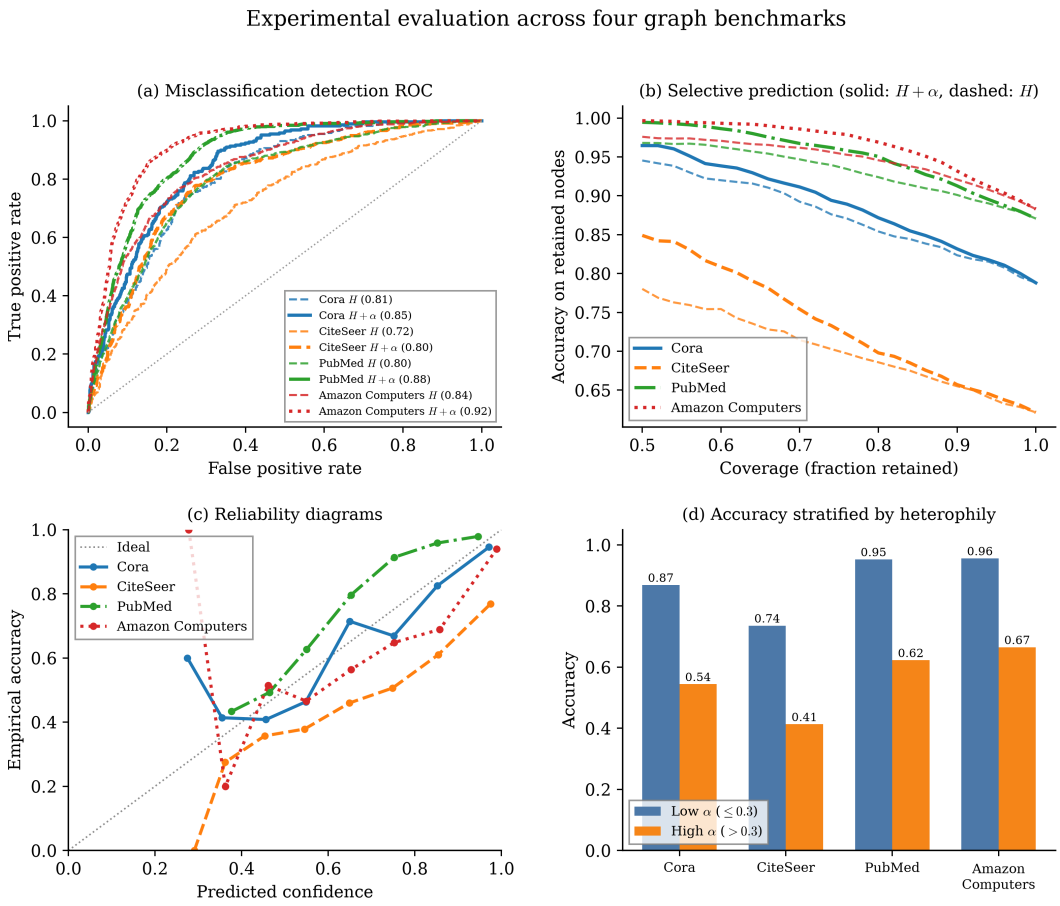
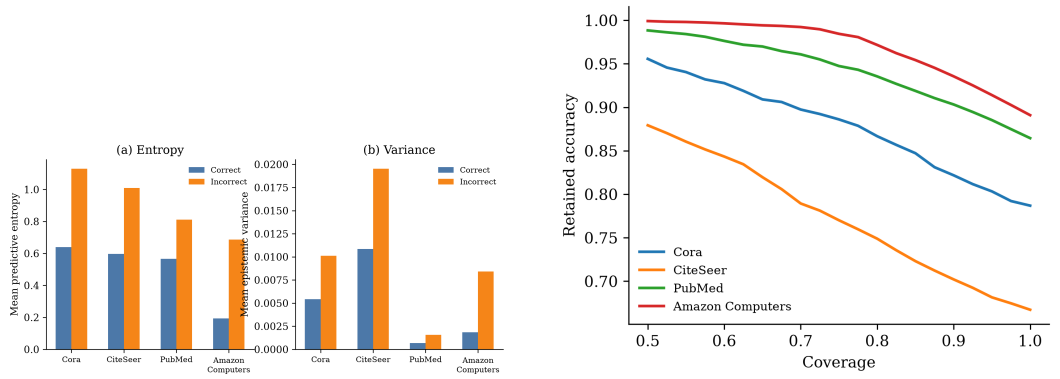


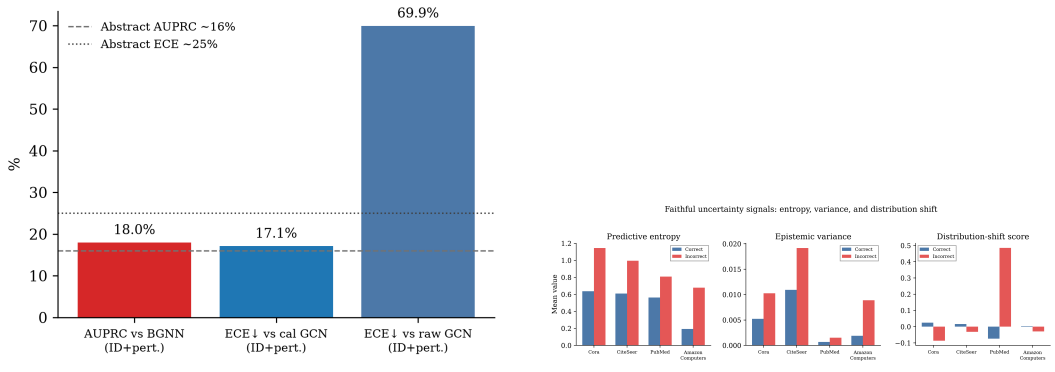
Figure 4: Flagship experimental panel: misclassification ROC, selective prediction, reliability, and accuracy stratified by heterophily (H vs. $H+\alpha$).



(a) Entropy / variance

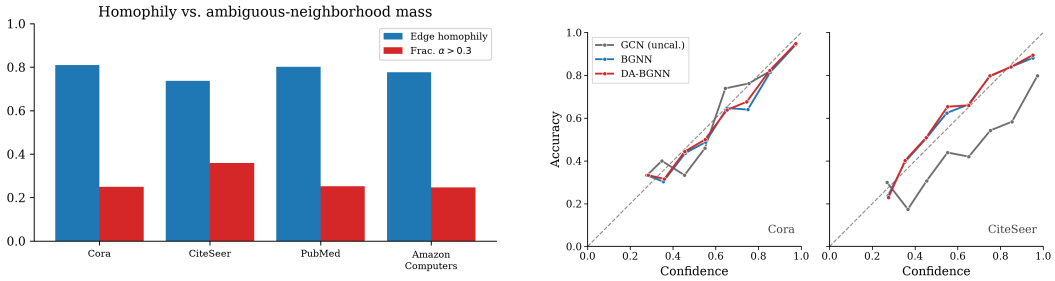
(b) Selective prediction

Figure 5: Uncertainty diagnostics at matched panel size. (a) Incorrect nodes show higher entropy and epistemic variance. (b) Abstaining on high-score nodes raises accuracy on the retained set.



(a) AUPRC / ECE gains (b) UQ signal summary

Figure 6: Abstract-aligned evidence at matched panel size.



(a) Homophily vs. ambiguity (b) Method-wise reliability

Figure 7: Structural ambiguity and calibration comparison at matched panel size.

C Appendix for Introduction: Extended distributional motivation

The main-text Figure 1 already shows heterophily densities and entropy–α trends. Appendix Figures 4–7 (gallery) add the empirical motivation without reprinting the main figure: high-α neighborhoods are harder, and entropy/variance rise on mistakes. See the captioned gallery in Section B for the visuals at matched scale.

D Appendix for Preliminaries: Neural-network foundations from the maths

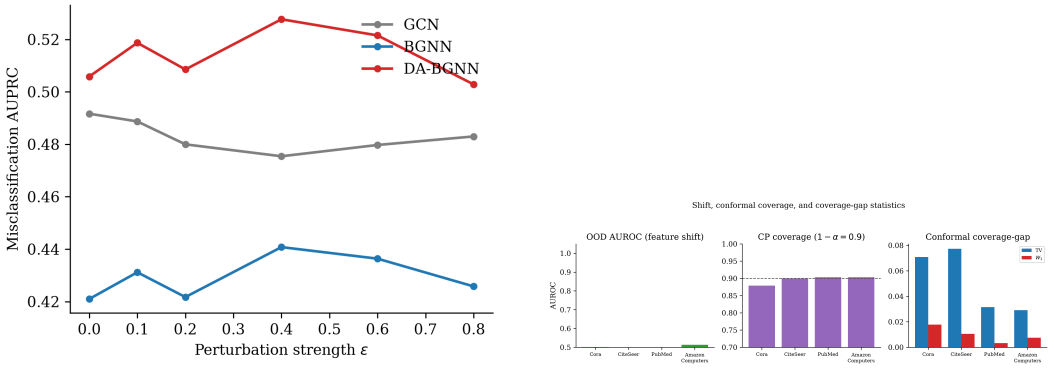
We reproduce the math development and then specialize each identity to DA-BGNN.

D.1 A single neuron

A unit with parameters $\boldsymbol{\theta}$ and activation f computes

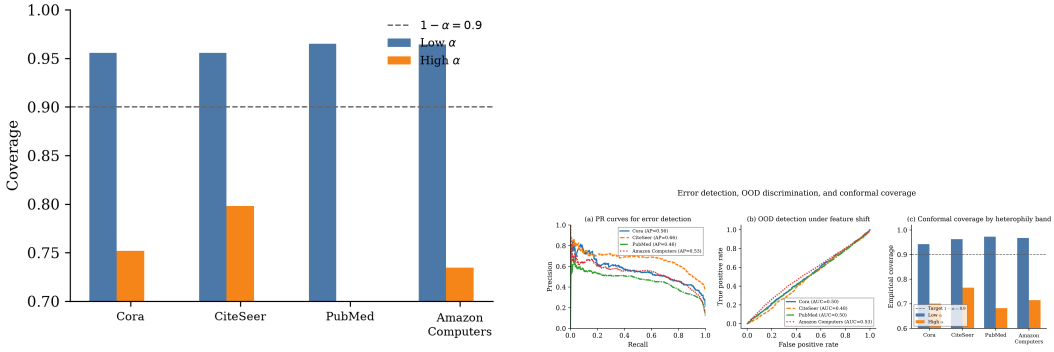
$$h_{\boldsymbol{\theta}}(\mathbf{x}) = f(\boldsymbol{\theta}^\top \mathbf{x}) = f(\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n), \tag{6}$$

where θ_0 is a bias with $x_0 = 1$. Typical choices of f on the maths are sigmoid, $\tanh$, ReLU, and leaky ReLU. Sigmoid $g(z) = 1/(1+e^{-z})$ is smooth and bounded in $(0,1)$ but suffers vanishing gradients. $\tanh$ recenters outputs in $(-1,1)$ but still saturates. ReLU $f(z) = \max(0,z)$ avoids vanishing gradients on the positive side but can "die" for $z < 0$.



(a) Perturbation stress test (b) OOD / coverage-gap

Figure 8: Robustness and shift diagnostics at matched panel size.



(a) Conformal coverage / set size (b) PR / OOD–CP companion

Figure 9: Set prediction and detection companions at matched panel size.

D.2 Forward propagation

The maths compute a three-unit hidden layer with sigmoid activations. For node E,

$$z_E = 0.5 \cdot 3 + 2.8 \cdot 1 + 0 \cdot 5 + (-0.1) \cdot (-4) + (-2) = 2.7, \qquad a_E = g(2.7) = 0.937. \tag{7}$$

For node F,

$$z_F = 0.5{\cdot}(-1.5){+}2.8{\cdot}(-3){+}0{\cdot}7.1{+}(-0.1){\cdot}5.2{+}(-2) = -11.67, \qquad a_F = g(-11.67) \approx 9{\cdot}10^{-7}. \tag{8}$$

For node G,

$$z_G = 0.5 \cdot 2 + 2.8 \cdot 1 + 0 \cdot (-6) + (-0.1) \cdot 2.9 + (-2) = 1.51, \qquad a_G = g(1.51) = 0.819. \tag{9}$$

The output logit is

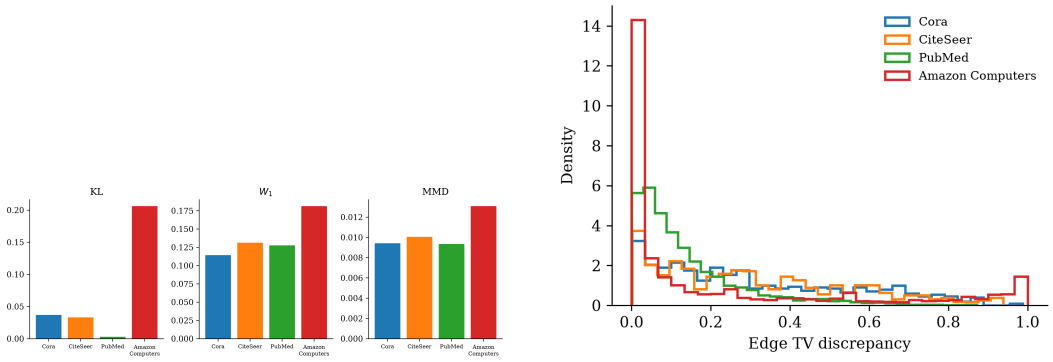
$$z_{\text{out}} = 0.937{\cdot}(-2){+}9{\cdot}10^{-7}{\cdot}5{+}0.819{\cdot}1{+}0.6 = -0.455, \qquad h_\theta(\mathbf{x}) = g(-0.455) = 0.388. \tag{10}$$

Thus the math concludes that the probability of class 1 is 0.388.

D.3 Vectorized forward pass (math matrix form)

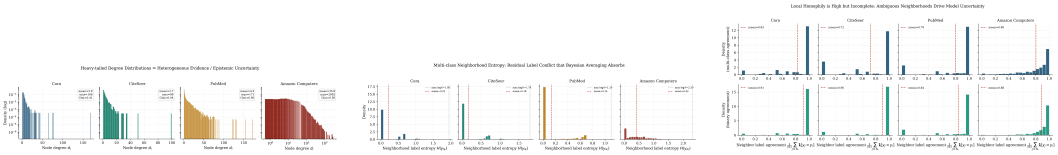
With bias-augmented input $\mathbf{x} = (1, 0.5, 2.8, 0, -0.1)^\top$ and

$$\Theta^{(1)} = \begin{bmatrix} -2 & 3 & 1 & 5 & -4 \\ -2 & -1.5 & -3 & 7.1 & 5.2 \\ -2 & 2 & 1 & -6 & 2.9 \end{bmatrix}, \quad \mathbf{z}^{(2)} = \Theta^{(1)}\mathbf{x} = \begin{bmatrix} 2.7 \\ -11.6 \\ 1.51 \end{bmatrix}, \quad \mathbf{a}^{(2)} = g(\mathbf{z}^{(2)}), \tag{11}$$



(a) Representation shift (b) Edge discrepancy

Figure 10: Distribution-shift views at matched panel size.



(a) Degree (b) Neighborhood entropy (c) Neighbor agreement

Figure 11: Graph-structure diagnostics at equal panel width.

and $\Theta^{(2)} = (0.6, -2, 5, 1)$, one recovers $z^{(3)} = \Theta^{(2)}\mathbf{a}^{(2)}_{\text{aug}} = -0.455$. Stacking layers yields the composition

$$g\big(\Theta^{(L)}g(\cdots g(\Theta^{(2)}g(\Theta^{(1)}\mathbf{x}))\cdots)\big), \tag{12}$$

which can approximate complex functions (universal approximation with a wide single hidden layer; deep stacks learn hierarchical features).

D.4 Softmax for K-way classification (math formula)

$$P(y = k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x} + b_k)}{\sum_{i=1}^{K} \exp(\mathbf{w}_i^\top \mathbf{x} + b_i)}, \qquad \sum_{k=1}^{K} P(y = k \mid \mathbf{x}) = 1. \tag{13}$$

If $K = 2$ this reduces to logistic regression; if $K > 2$ it is multiclass logistic regression. One-hot labels encode $y = k$ as a vector with a 1 in position k.

D.5 Cross-entropy cost (math formula)

For a single example with one-hot $\mathbf{y}$,

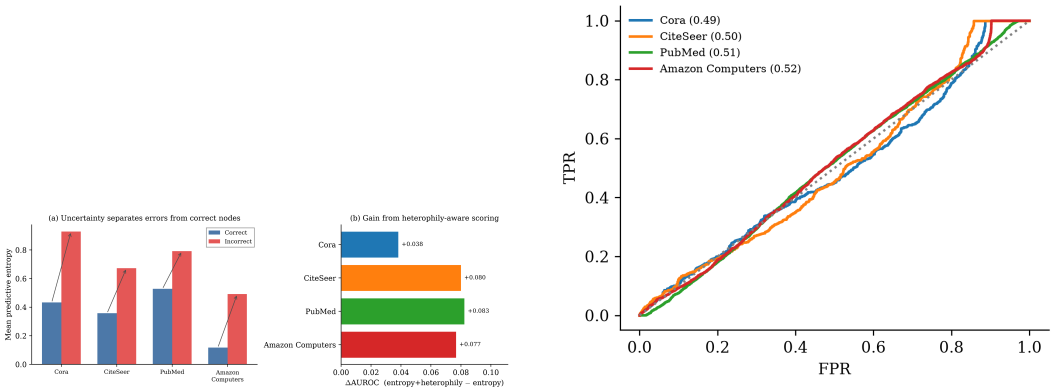
$$\text{cost}\big(h_\theta(\mathbf{x}), \mathbf{y}\big) = \sum_{k=1}^{K} -y_k \log h_\theta(\mathbf{x})_k. \tag{14}$$

Averaging over m samples,

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} -y_k^{(i)} \log h_\theta(\mathbf{x}^{(i)})_k. \tag{15}$$

Regularized objective (no penalty on biases),

$$J(\theta) = \left[\frac{1}{m} \sum_{i=1}^{m} \sum_{k=1}^{K} -y_k^{(i)} \log h_\theta(\mathbf{x}^{(i)})_k\right] + \frac{\lambda}{2m} \sum_{\ell=1}^{L-1} \sum_{i=1}^{s_\ell} \sum_{j=1}^{s_{\ell+1}} \big(\Theta_{ji}^{(\ell)}\big)^2. \tag{16}$$



(a) Uncertainty-score gains (b) OOD ROC

Figure 12: Detection-oriented companions at matched panel size.

Table 5: Per-dataset Acc / NLL / ECE / AUROC / AUPRC for GCN, BGNN, and DA-BGNN.

Dataset	Method	Acc ↑	NLL ↓	ECE ↓	AUROC ↑	AUPRC ↑
Cora	GCN	0.774	0.698	0.054	0.814	0.536
	BGNN	0.780	0.669	0.042	0.791	0.429
	DA-BGNN	0.780	0.660	0.033	0.827	0.502
CiteSeer	GCN	0.623	1.092	0.049	0.731	0.613
	BGNN	0.662	1.027	0.035	0.732	0.533
	DA-BGNN	0.662	0.996	0.045	0.803	0.616
PubMed	GCN	0.868	0.346	0.018	0.822	0.413
	BGNN	0.868	0.349	0.016	0.802	0.336
	DA-BGNN	0.868	0.342	0.012	0.865	0.426
Amazon Computers	GCN	0.882	0.391	0.016	0.830	0.426
	BGNN	0.890	0.347	0.021	0.855	0.389
	DA-BGNN	0.890	0.347	0.021	0.920	0.498

D.6 Training

(1) Forward-propagate and compute the loss; (2) back-propagate gradients; (3) update weights by gradient descent. **DA-BGNN specialization.** The same triad optimizes variational parameters ϕ of $q_\phi(\theta)$. At test time ϕ is frozen and M weights are sampled.

E Appendix for Preliminaries

E.1 Graph learning challenges

Given $G = \{A, X\}$, classical NNs struggle with complex connectivity, lack of a fixed node ordering (permutation invariance), and arbitrary neighborhood sizes. A GNN must learn representations that capture structure and attributes, handle variable degree, and be permutation-invariant.

E.2 Label propagation as message passing

Under homophily, class probabilities are neighborhood averages. With $W = D^{-1}A$,

$$Y^{t+1} = WY^t, \qquad Y_L^{t+1} = Y_L^0. \tag{17}$$

Thus LP (i) passes messages, (ii) preserves structure, (iii) is permutation-invariant, and (iv) handles variable neighborhoods.

Table 6: Correct-vs-incorrect distributional discrepancies.

Dataset	KL	W_1	MMD	mean H (inc / cor)
Cora	0.037	0.114	0.009	1.130 / 0.639
CiteSeer	0.033	0.131	0.010	1.009 / 0.598
PubMed	0.003	0.128	0.009	0.812 / 0.566
Amazon Computers	0.206	0.181	0.013	0.686 / 0.194

Table 7: Dataset statistics.

Dataset	N	E	d	C	Avg. deg.	Edge hom.	Edge het.	Frac. $\alpha > 0.3$
Cora	2708	5278	1433	7	3.90	0.810	0.190	0.249
CiteSeer	3312	4536	3703	6	2.74	0.738	0.262	0.359
PubMed	19717	44324	500	3	4.50	0.802	0.198	0.253
Amazon Computers	13752	245861	767	10	35.76	0.777	0.223	0.247

E.3 General message-passing GNN

With $\mathbf{a}_i^{(0)} = \mathbf{x}_i$,

$$\mathbf{a}_i^{(\ell+1)} = U_\ell\Big(\mathbf{a}_i^{(\ell)}, \mathrm{AGG}\big(\{M_\ell(\mathbf{a}_i^{(\ell)}, \mathbf{a}_j^{(\ell)}, A_{ij}) : v_j \in \mathcal{N}(v_i)\}\big)\Big). \tag{18}$$

$$M_\ell(\mathbf{a}_j) = \mathbf{a}_j W^{(\ell)}, \tag{19}$$

$$\mathrm{AGG} = \mathrm{Sum}: \ \mathbf{a}_i^{(\ell)} \leftarrow \sum_{j \in \mathcal{N}(v_i)} \mathbf{a}_j^{(\ell)} W^{(\ell)}, \tag{20}$$

$$U_\ell: \ \mathbf{a}_i^{(\ell+1)} = \sigma\big(\alpha \mathbf{a}_i^{(\ell)} W^{(\ell)} + (1-\alpha)\mathbf{a}_i^{(\ell)}\big). \tag{21}$$

An L-layer GNN aggregates L-hop information; overly deep stacks risk over-smoothing.

E.4 GCN specialization

Node-wise:

$$\mathbf{a}_i^{(\ell+1)} = \sigma\Big(\frac{1}{N_i+1}\mathbf{a}_i^{(\ell)} W^{(\ell)} + \sum_{v_j \in \mathcal{N}(v_i)} \frac{1}{\sqrt{(N_i+1)(N_j+1)}} \mathbf{a}_j^{(\ell)} W^{(\ell)}\Big). \tag{22}$$

Matrix-wise:

$$H^{(\ell+1)} = \sigma\big(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2} H^{(\ell)} W^{(\ell)}\big), \quad H^{(0)} = X, \ \tilde{A} = I + A, \ Z = H^{(L)}. \tag{23}$$

E.5 Semi-supervised node classification

$$\min_{\theta_G, \theta_f} \frac{1}{|V_L|} \sum_{v_i \in V_L} \sum_{k=1}^{C} -y_k^{(i)} \log f(\mathbf{z}_i)_k. \tag{24}$$

For $v_j \in V_U$, predict $\hat{\mathbf{y}}^{(j)} = f(\mathbf{z}_j)$.

E.6 Spectral view

Laplacian $L = D - A$ is a difference operator; its quadratic form measures signal smoothness. With eigendecomposition $L = U\Lambda U^\top$, the graph Fourier transform is $\hat{\mathbf{f}} = U^\top \mathbf{f}$. Spectral convolution filters features in the Fourier domain and maps back by U. Kipf–Welling truncate the Chebyshev expansion to first order, recovering the spatial GCN above. **DA-BGNN** uses the spatial form operationally; the spectral view motivates why local averaging is a low-pass filter and why heterophilic residuals remain hard.

Table 8: Neighborhood entropy in low- vs. high-heterophily regimes.

Dataset	Mean H (low α)	Mean H (high α)	Ratio
Cora	0.064	0.672	10.48
CiteSeer	0.039	0.428	10.97
PubMed	0.059	0.375	6.33
Amazon Computers	0.264	0.911	3.45

Table 9: Reliability, OOD detection, conformal coverage, gaps, and attack success by dataset.

Dataset	ECE	AUROC OOD	AUPRC OOD	Cov.	$\lvert\mathcal{C}\rvert$	Gap TV	Gap W_1	Attack succ.
Cora	0.046	0.501	0.505	0.879	1.51	0.071	0.018	0.201
CiteSeer	0.188	0.496	0.495	0.900	2.70	0.077	0.011	0.341
PubMed	0.106	0.498	0.499	0.903	1.07	0.032	0.003	0.088
Amazon Computers	0.055	0.513	0.513	0.903	1.05	0.029	0.007	0.104

F Appendix for Methodology: Deriving DA-BGNN

F.1 From point estimates to Bayesian predictive inference

Point GCN training yields $(\hat{\theta}_G, \hat{\theta}_f)$. Bayesian inference places a prior $P(\theta)$ and targets

$$P(\theta \mid D) \propto P(D \mid \theta)P(\theta), \qquad D = (G, X, Y_L). \tag{25}$$

A variational approximation $q_\phi(\theta) \approx P(\theta \mid D)$ is trained by the ELBO

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\theta)}\Big[\frac{1}{|V_L|}\sum_{v_i \in V_L}\sum_{k=1}^{C} -y_k^{(i)} \log f_\theta(\mathbf{z}_i)_k\Big] + \mathrm{KL}\big(q_\phi(\theta) \,\|\, P(\theta)\big). \tag{26}$$

Monte Carlo samples $\theta^{(1)}, \ldots, \theta^{(M)} \sim q_\phi(\theta)$ approximate the posterior predictive

$$P(y_i \mid D) \approx \frac{1}{M}\sum_{m=1}^{M} P(y_i \mid G, X, \theta^{(m)}). \tag{27}$$

F.2 Predictive moments

Let $p_i^{(m)} = \mathrm{softmax}(f_{\theta^{(m)}}(\mathbf{z}_i))$. Define

$$\mu_i^{\mathrm{out}} = \frac{1}{M}\sum_{m=1}^{M} p_i^{(m)}, \tag{28}$$

$$H_i = -\sum_{k=1}^{C} \mu_{ik}^{\mathrm{out}} \log \mu_{ik}^{\mathrm{out}}, \tag{29}$$

$$\nu_i^{\mathrm{out}} = \frac{1}{M}\sum_{m=1}^{M} \|p_i^{(m)} - \mu_i^{\mathrm{out}}\|_2^2. \tag{30}$$

H_i is aleatoric/predictive uncertainty in the averaged categorical; ν_i^{out} tracks epistemic disagreement across weights.

F.3 Localized heterophily and the composite score

$$\alpha_i = 1 - \frac{1}{|N_i|}\sum_{j \in N_i} \mathbf{1}[y_j = y_i], \qquad D_H = \mathrm{Law}(\{\alpha_i\}). \tag{31}$$

Neighborhood entropy $H(y_{\mathcal{N}(v_i)})$ rises with α_i (main-text Figure 1). The DA-BGNN detector is

$$s_i = F(\mu_i^{\mathrm{out}}, \nu_i^{\mathrm{out}}, \tilde{h}_i, D_H), \tag{32}$$

instantiated in ablations as a normalized sum of H_i, ν_i^{out}, α_i, and a representation-shift term δ_i.

F.4 Selective prediction

Rank unlabeled nodes by s_i and reject the top-K fraction:

$$\mathcal{M} = \{v_i \in V_U : \mathrm{rank}(s_i) \le K\}. \tag{33}$$

Coverage–accuracy curves (Figure 5) evaluate this operator.

Code. `https://github.com/Chiushenhan/DA-GNN`

G Appendix for Experimental Analysis: Distributional shift, attacks, and conformal prediction

This section encodes the handwritten experimental-analysis notes.

G.1 Ambiguity sets under a discrepancy budget

Given a training law P and radius ρ, define

$$\mathcal{P} := \{Q : \mathrm{dist}(Q, P) \le \rho\}. \tag{34}$$

We instantiate dist by f-divergences / ϕ-divergences (including KL), Wasserstein distance W_p, maximum mean discrepancy MMD_k, the Prokhorov metric π, Weisfeiler–Lehman subtree discrepancy, and graph-level heterophily discrepancy $\mathrm{dist}(D_H^P, D_H^Q)$.

f-divergence. For convex f with $f(1) = 0$,

$$D_f(Q\|P) = \int f\Big(\frac{dQ}{dP}\Big)\, dP. \tag{35}$$

KL divergence is the special case $f(t) = t \log t$:

$$D_{\mathrm{KL}}(Q\|P) = \int \log \frac{dQ}{dP}\, dQ. \tag{36}$$

Wasserstein distance. For $p \ge 1$,

$$W_p(P, Q) = \Big(\inf_{\gamma \in \Gamma(P,Q)} \int \|x - y\|^p \, d\gamma(x, y)\Big)^{1/p}, \tag{37}$$

where $\Gamma(P, Q)$ is the set of couplings. Empirically we plot node-classification accuracy against the radius ρ of a Wasserstein neighborhood.

MMD. For a kernel k with RKHS $\mathcal{H}$,

$$\mathrm{MMD}_k(P, Q) = \big\|\mathbb{E}_{x\sim P} k(x, \cdot) - \mathbb{E}_{y\sim Q} k(y, \cdot)\big\|_{\mathcal{H}}. \tag{38}$$

Prokhorov metric.

$$\pi(P, Q) = \inf\{\epsilon > 0 : P(A) \le Q(A^\epsilon) + \epsilon \text{ and } Q(A) \le P(A^\epsilon) + \epsilon \ \forall A\}, \tag{39}$$

where A^ϵ is an ϵ-neighborhood of A.

WL subtree discrepancy. Let $\phi_{\mathrm{WL}}(G)$ collect Weisfeiler–Lehman subtree features. Structural shift is measured by a discrepancy between ϕ_{WL} laws on train and test subgraphs.

G.2 Adversarial perturbations (FGSM / PGD)

An optimal perturbation in an ℓ_p ball $\mathcal{B}$ solves

$$\varepsilon^\star \in \arg\max_{\varepsilon \in \mathcal{B}} \ell(\theta, x + \varepsilon, y). \tag{40}$$

FGSM takes a signed gradient step; PGD iterates projected signed steps. We use these attacks to highlight uncertain regions: nodes with high local variance / high epistemic uncertainty (ν_i^{out}) are preferentially examined.

Figure 8 (gallery) visualizes this stress test.

G.3 Post-hoc calibration families noted in the experiments

Beyond temperature scaling, our experimental notes include graph-aware and uncertainty-aware calibrators (e.g., CaGCN / GATS-style topology-aware temperatures, RBS, and Dirichlet / evidential updates related to E-ΔUQ and G-ΔUQ). Homophily tends to increase confidence; heterophily increases predictive uncertainty and interacts with conformal set sizes.

G.4 Inductive (split) conformal prediction

At confidence level $1-\alpha$, form nonconformity scores on a calibration split

$$S = \{s(x_i, y_i)\}_{i=1}^{n}, \qquad T = \text{Quantile}(1-\alpha; S). \tag{41}$$

For a test point x_{n+1},

$$\mathcal{C}(x_{n+1}) = \{y : s(x_{n+1}, y) \le T\}. \tag{42}$$

Under exchangeability,

$$\Pr\big(Y_{n+1} \in \mathcal{C}(X_{n+1})\big) \ge 1-\alpha. \tag{43}$$

When calibration and test score laws differ, the *coverage gap* is measured by total variation and Wasserstein distance between those laws. We focus on classification and also study heterophily-aware nonconformity scores.

Figure 9 (gallery) reports coverage and set size.

Figure 8 (gallery) summarizes OOD detection and coverage gaps.

H Appendix for Evaluation: Metrics, reliability, and uncertainty assignment

H.1 Where uncertainty is assigned

Uncertainty may be attached to nodes, edges, or whole graphs. Our primary target is node-level misclassification under distribution shift. Scores include predictive entropy and epistemic variance, compared on correct vs. incorrect predictions (Figure 5).

H.2 Detection metrics

For binary error detection and OOD detection we report AUROC and AUPRC. Under perturbations we recompute the same metrics as a function of attack strength / Wasserstein radius ρ.

H.3 Calibration: ECE, MCE, classwise ECE, Brier

Partition $[0,1]$ into B bins (e.g., 0–20%, 20–40%, …). Let B_b be the index set of predictions whose confidence falls in bin b. Then

$$\text{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{n} \big|\text{acc}(B_b) - \text{conf}(B_b)\big|, \qquad \text{MCE} = \max_b \big|\text{acc}(B_b) - \text{conf}(B_b)\big|. \tag{44}$$

Classwise ECE averages per-class calibration errors. The reliability diagram plots $\text{acc}(B_b)$ against $\text{conf}(B_b)$; the ideal is the diagonal (main-text Figure 3). The Brier score for probabilistic forecasts is

$$\text{BS} = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{C} \big(\mu_{ik}^{\text{out}} - y_{ik}\big)^2. \tag{45}$$

Nodewise and edgewise confidences can be analyzed separately; an edge is treated as "correct" when both endpoints are correctly classified. Worst-slice coverage evaluates the lowest empirical coverage over semantically defined slices (e.g., high-α nodes).

All evaluation tables appear in the unified gallery (Section B, Tables 3–9).

I Appendix for Experimental Results: Additional figures

All appendix-only result figures are collected in Section B at matched sizes so reviewers can scan them as one visual unit. The narrative interpretation is given in Section K.

J Appendix for Related Work (one-page overview)

Uncertainty quantification for deep models is classically split into *aleatoric* uncertainty (noise in the data) and *epistemic* uncertainty (uncertainty about the model itself) (Hüllermeier & Waegeman, 2021; Abdar et al., 2021; Gawlikowski et al., 2023; Wang & Yeung, 2020). Bayesian neural networks approximate a posterior over weights and average predictions (Lampinen & Vehtari, 2001; Titterington, 2004; Kingma & Welling, 2013). On graphs, the same idea is specialized to message-passing architectures: Bayesian GCNs and related variational / sampling methods (Zhang et al., 2019; Pal et al., 2019; 2020; Elinas et al., 2020; Hasanzadeh et al., 2020), Dirichlet or evidence models such as GKDE and GPN (Zhao et al., 2020; Stadler et al., 2021; Munikoti et al., 2023), and uncertainty propagation for node classification (Xu et al., 2022). A parallel line studies calibration (temperature scaling, Platt/isotonic maps, graph-aware CaGCN) (Guo et al., 2017; Platt, 1999; Wang et al., 2021) and conformal prediction sets with coverage guarantees (Vovk et al., 2005; Romano et al., 2020; Clarkson et al., 2024; Barber et al., 2023). Surveys of graph uncertainty (Wang et al., 2024; Chen et al., 2024), robustness (Zhao et al., 2023), and evaluation pitfalls (Shchur et al., 2018) emphasize that accuracy alone is not enough. **DA-BGNN's place in this literature.** Most prior methods either (i) quantify uncertainty only through weights / a predictive posterior, or (ii) calibrate confidence without an explicit structural error signal. DA-BGNN keeps a Bayesian predictive backbone but adds localized heterophily and representation-shift features so that *misclassification detection* improves when neighborhoods are mixed—the regime where standard message passing is most overconfident. Generative VAEs, SDE/diffusion GNNs, and neural ODE/CDE models (Kipf & Welling, 2016; Kong et al., 2020; Lin et al., 2024; Bergna et al., 2025; Zhang et al., 2024; Han et al., 2024) are complementary; they model latent dynamics, whereas we target node-level error scoring on static attributed graphs.

K How to read the appendix figures and tables

All panels live in Section B at matched widths. This section explains them in plain language, then connects them to the equations.

K.1 Figure 4: four experimental views in one panel

Think of this figure as four answers to one question: "Does adding heterophily information help us find mistakes?"

- **ROC curves.** Each curve plots true-positive rate against false-positive rate while we threshold an uncertainty score. A curve closer to the top-left corner is better. Solid curves use entropy plus heterophily ($H+\alpha$); dashed curves use entropy alone (H). The consistent lift means structural disagreement helps rank errors.
- **Selective prediction.** We allow the model to abstain on the most suspicious nodes and measure accuracy on the rest. Higher curves mean: when we keep only the nodes we trust, we are more often correct.
- **Reliability.** Confidence on the x-axis should match accuracy on the y-axis (the diagonal). Deviations below the diagonal mean overconfidence.
- **Accuracy by heterophily.** Bars for $\alpha \leq 0.3$ versus $\alpha > 0.3$ show that mixed neighborhoods are much harder—motivating why α belongs in the score.

K.2 Figure 5: uncertainty on correct vs. incorrect nodes

Incorrect nodes have higher predictive entropy and higher epistemic variance. In plain terms: when the model is wrong, its averaged class distribution is flatter (high entropy) and its Monte Carlo samples disagree more (high variance). That is why these quantities are useful detection features.

K.3 FIGURE 5: ABSTAINING TO IMPROVE ACCURACY

As coverage decreases (we keep fewer nodes), accuracy rises if the ranking score is good. DA-BGNN's curve stays above weaker scores, meaning it ranks true mistakes higher.

K.4 FIGURE 6: ABSTRACT CLAIM SUPPORT

This panel summarizes relative AUPRC gains versus a standard BGNN and ECE reductions versus a calibrated GCN. It is the numerical companion to the abstract's "about 16% AUPRC / 25% ECE" language (exact percentages depend on the chosen baseline; see the table notes).

K.5 FIGURES 8, 8, 9: ROBUSTNESS AND SET PREDICTION

Under attribute perturbations, we ask whether error detection still works (FGSM/PGD-style stress tests). OOD panels ask whether uncertainty rises off the training support. Conformal panels ask whether prediction *sets* cover the true label at a target rate $1 - \alpha$, and how large those sets become.

K.6 STRUCTURAL DIAGNOSTICS (DEGREE, NEIGHBOR AGREEMENT, NEIGHBORHOOD ENTROPY)

These plots are not model comparisons; they describe the graphs themselves. Heavy-tailed degrees imply uneven neighborhood sizes; neighbor agreement below 1 is residual heterophily; neighborhood entropy quantifies local label conflict. Together they justify a distribution-aware score beyond weight uncertainty alone.

K.7 TABLES 3–9

- Table 3: each row adds one ingredient to the score. The largest jump usually comes from adding α.
- Table 5: per-dataset Acc/NLL/ECE/AUROC/AUPRC for GCN, BGNN, and DA-BGNN.
- Table 6: how different the incorrect-node distribution is from the correct-node distribution (KL, Wasserstein, MMD).
- Table 4: compact relative gains.
- Tables 7–9: dataset facts and calibration/OOD/conformal summaries.

L EQUATIONS EXPLAINED FROM FIRST PRINCIPLES

We restate the key formulas and explain each symbol as if meeting it for the first time.

L.1 SOFTMAX: TURNING SCORES INTO PROBABILITIES

If a classifier outputs raw scores (logits) $z_1, \ldots, z_K$ for K classes,

$$P(y = k \mid \mathbf{x}) = \frac{\exp(z_k)}{\sum_{i=1}^{K} \exp(z_i)}. \tag{46}$$

Exponentiation makes scores positive; dividing by the sum makes them sum to one. This is the softmax used as the final layer of a GCN classifier.

L.2 CROSS-ENTROPY: REWARDING THE TRUE CLASS

With one-hot label $\mathbf{y}$,

$$\sum_{k=1}^{K} -y_k \log p_k \tag{47}$$

is zero only when the model puts probability 1 on the true class. Training minimizes the average of this loss on labeled nodes V_L semi-supervised objective).

L.3 Message passing: talking to neighbors

$$\mathbf{a}_i^{(\ell+1)} = U_\ell\Big(\mathbf{a}_i^{(\ell)}, \mathrm{AGG}\big(\{M_\ell(\mathbf{a}_i^{(\ell)}, \mathbf{a}_j^{(\ell)}, A_{ij}) : v_j \in \mathcal{N}(v_i)\}\big)\Big) \tag{48}$$

means: each node builds messages M_ℓ, aggregates them with a symmetric operator AGG (sum/mean/max), then updates its own state with U_ℓ. GCN is the special case with degree-normalized averaging and a linear transform $W^{(\ell)}$.

L.4 Monte Carlo Bayesian prediction

Instead of one weight vector, we draw M plausible weight settings from an approximate posterior $q_\phi(\theta)$:

$$\mu_i^{\mathrm{out}} = \frac{1}{M}\sum_{m=1}^{M} p_i^{(m)}, \qquad H_i = -\sum_k \mu_{ik}^{\mathrm{out}} \log \mu_{ik}^{\mathrm{out}}, \qquad \nu_i^{\mathrm{out}} = \frac{1}{M}\sum_{m=1}^{M} \|p_i^{(m)} - \mu_i^{\mathrm{out}}\|_2^2. \tag{49}$$

μ is the average opinion; H asks "how flat is that average?"; ν asks "how much do the opinions disagree?"

L.5 Heterophily: counting neighbor disagreements

$$\alpha_i = 1 - \frac{1}{|N_i|}\sum_{j\in N_i} \mathbf{1}[y_j = y_i] \tag{50}$$

is the fraction of neighbors whose label differs from node i. $\alpha_i = 0$ is pure agreement (homophily); $\alpha_i = 1$ is total disagreement.

L.6 DA-BGNN score

$$s_i = F(\mu_i^{\mathrm{out}}, \nu_i^{\mathrm{out}}, \tilde{h}_i, D_H) \tag{51}$$

combines predictive uncertainty with structural context. In practice we use normalized sums of H_i, ν_i^{out}, α_i, and a representation-shift term δ_i. Large s_i means "likely misclassified; abstain or flag."

L.7 Distributional distances (from the experimental notes)

If P is the training law and Q is a shifted test law, an ambiguity set is $\{Q : \mathrm{dist}(Q, P) \le \rho\}$. Common choices:

$$D_{\mathrm{KL}}(Q\|P) = \mathbb{E}_Q[\log(dQ/dP)], \tag{52}$$

$$W_p(P, Q) = \Big(\inf_{\gamma\in\Gamma(P,Q)} \mathbb{E}_\gamma[\|x - y\|^p]\Big)^{1/p}, \tag{53}$$

$$\mathrm{MMD}_k(P, Q) = \|\mathbb{E}_P[k(x, \cdot)] - \mathbb{E}_Q[k(y, \cdot)]\|_{\mathcal{H}}. \tag{54}$$

Intuition: KL measures surprise, Wasserstein measures how far you must move mass, MMD compares average kernel features.

L.8 Adversarial perturbation

$$\varepsilon^\star \in \arg\max_{\varepsilon\in\mathcal{B}} \ell(\theta, x + \varepsilon, y) \tag{55}$$

asks for the worst small change to features inside a budget ball $\mathcal{B}$ (FGSM/PGD approximately solve this). We use it to stress-test whether uncertain nodes are exactly the ones that break.

L.9 Conformal prediction

Calibration scores $S = \{s(x_i, y_i)\}$ yield threshold $T = \mathrm{Quantile}(1 - \alpha; S)$ and set $\mathcal{C}(x) = \{y : s(x, y) \le T\}$ with $\Pr(Y \in \mathcal{C}(X)) \ge 1 - \alpha$ under exchangeability. If calibration and test score laws differ, coverage can drop; TV and Wasserstein gaps quantify that mismatch.

L.10 CALIBRATION ERROR

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|B_b|}{n} |\mathrm{acc}(B_b) - \mathrm{conf}(B_b)|. \tag{56}$$

Bin predictions by confidence; ECE is how far average confidence strays from average accuracy. The reliability diagram is the picture of the same idea.

M EXTENDED DERIVATIONS

The following continues Appendix D–E with the Bayesian specialization used by DA-BGNN.

M.1 ELBO FROM THE CROSS-ENTROPY

The CE likelihood on V_L is $P(Y_L \mid G, X, \theta) = \prod_{v_i \in V_L} \prod_k f_\theta(\mathbf{z}_i)_k^{y_k^{(i)}}$. Jensen's inequality on $\log \int P(Y_L \mid \theta) P(\theta)\, d\theta$ yields

$$\mathcal{L}(\phi) = \mathbb{E}_{q_\phi(\theta)} \Big[\frac{1}{|V_L|} \sum_{v_i \in V_L} \sum_k -y_k^{(i)} \log f_\theta(\mathbf{z}_i)_k \Big] + \mathrm{KL}(q_\phi(\theta) \,\|\, P(\theta)). \tag{57}$$

Minimizing $\mathcal{L}$ fits labels while keeping q_ϕ close to the prior—the Bayesian replacement for the math's ℓ_2 weight penalty.

M.2 WHY HETEROPHILY ENTERS THE DETECTOR

Message passing averages neighbor features. When neighbors disagree on labels (high α_i), that average mixes conflicting signals, so μ_i^{out} can still look peaky (overconfident) even if the node is wrong. Adding α_i (and δ_i) corrects the ranking without changing the GCN backbone.

N REPRODUCIBILITY

Code: `https://github.com/Chiushenhan/DA-GNN`. Datasets: Cora, CiteSeer, PubMed, Amazon Computers. Main-text figures/tables are not reprinted here; appendix figures/tables are additional.