DO DEEP ENSEMBLES ACTUALLY CAPTURE UNCERTAINTY IN GRAPH NEURAL NETWORKS?

Anonymous authors
Paper under double-blind review

ABSTRACT

While deep ensembles are widely considered to be the default method for uncertainty quantification in deep learning, their effectiveness for graph-structured data is often simply assumed based on successes in domains like computer vision. We investigate standard deep ensembles specifically for message-passing graph neural networks. Benchmarking across eight datasets representing varied tasks and complexities, we reveal that ensembles provide only marginal improvement over a single model. These marginal gains stem primarily from averaging out optimization noise in point predictions rather than from meaningfully better uncertainty estimates. Through an aleatoric-epistemic decomposition, we identify *epistemic collapse*: independently trained networks consistently converge to overly similar predictions. Because disagreement is the fundamental mechanism through which ensembles capture epistemic uncertainty, this lack of diversity neutralizes their key advantage. Towards explaining this phenomenon, we derive a bound on epistemic variance in terms of the graph topology and Dirichlet energy budget, linking epistemic collapse to the smoothing bias of message passing. Our results suggest that deep ensemble success does not seamlessly transfer to graph machine learning.

1 INTRODUCTION

Uncertainty quantification is essential in high-stakes settings or automated decision-making applications [22, 54, 77]. Among practical approaches to uncertainty quantification, deep ensembles (DEs) [40] have emerged as a widely used method. Their appeal lies in simplicity: building a standard deep ensemble requires only the repeated training of a single architecture. The disagreement among these independently trained models captures uncertainty that a single model cannot. This mechanism has proven highly effective, establishing deep ensembles as the default for uncertainty quantification in deep learning, provided the added computational cost is not prohibitive [56, 64].

When processing graph-structured data—such as molecules or road networks—uncertainty quantification is much less studied. Here, message-passing neural networks, like graph convolutional (GCN) [36] or attention (GAT) [67] networks, have emerged as one of the dominant architecture classes. While other classes like graph transformers [16, 58] exist, we focus exclusively on message-passing architectures, which we refer to simply as graph neural networks (GNNs) throughout this work. Unlike standard deep learning architectures, GNN computation flow is defined by the structure of the input graph, a property that fundamentally alters the hypothesis space and how optimization trajectories explore it [60, 75]. This structural difference can significantly affect how ensembles of such models behave, for better or worse.

Surveys and application papers often present DEs as a successful, out-of-the-box solution for uncertainty quantification in graph learning tasks [28, 69]. However, they tend not to compare the uncertainty of DEs with that of even simpler baselines: the strength of DEs in classical deep learning settings like computer vision is largely *assumed* to carry over to graphs. Works that do evaluate the uncertainty of DEs paint a mixed picture. DEs range from hard to beat [65] to barely distinguishable from a single model [63], and even underperform a trivial baseline on the PEMS road network benchmark [7, 51]—a failure we revisit in Section 3. In these works, DEs themselves are a baseline

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.

rather than the object of study, and, to our knowledge, no work systematically studies how good this simple and practical baseline is: how well standard DEs of GNNs capture uncertainty, and why.[1]

Motivated by these mixed signals, we provide such a study. To this end, we systematically benchmark DEs of GNNs across multiple tasks, focusing explicitly on the quality of their uncertainty estimates. Our results demonstrate that their behavior in graph settings falls far short of expectations inherited from classic deep learning settings, most notably computer vision, where DEs are most widely tested. Our central findings are an empirical phenomenon we term *epistemic collapse*—individual GNNs within a DE exhibit surprisingly limited diversity—and a theoretical bound on epistemic variance that links this collapse to the smoothing bias of message passing.

Our contributions are:

1. We systematically benchmark standard DEs of GNNs across multiple classic and modern graph machine learning datasets.
2. We empirically demonstrate the phenomenon of epistemic collapse, showing that GNNs tend to converge to highly similar functions regardless of initialization.
3. We demonstrate that the minor improvements in the negative log-likelihood (NLL) score provided by DEs are primarily attributable to moderately improved prediction accuracy rather than better uncertainty estimates.
4. We derive a bound on epistemic variance in terms of the graph topology and Dirichlet energy budget. It links epistemic collapse to the smoothing bias of message passing—the same bias that helps on homophilic graphs and, in excess, causes oversmoothing—and suggests weaker collapse under heterophily, aligning with our experiments.

2 Background

Graph neural networks. As mentioned in Section 1, we focus on *message-passing GNNs* [23]. Given a graph $\mathcal{G}$ with node features $\boldsymbol{x}_i$, each layer computes

$$\boldsymbol{h}_i^{(\ell+1)} = \gamma_{\boldsymbol{\Theta}_1}\left(\boldsymbol{h}_i^{(\ell)}, \bigoplus_{j\in\mathcal{N}(i)} \psi_{\boldsymbol{\Theta}_2}(\boldsymbol{h}_i^{(\ell)}, \boldsymbol{h}_j^{(\ell)})\right), \qquad \boldsymbol{h}_i^{(0)} = \boldsymbol{x}_i, \tag{1}$$

where $\mathcal{N}(i)$ denotes the neighbors of i, $\psi_{\boldsymbol{\Theta}_2}$ computes messages, $\bigoplus$ is a permutation-invariant aggregation, and $\gamma_{\boldsymbol{\Theta}_1}$ post-processes the result. Architectures such as GCN [36], GAT [67], and GIN [75] vary this template, but always follow the graph topology.

Uncertainty quantification and deep ensembles. Uncertainty quantification aims to measure not just what a model predicts, but how reliable that prediction is. Predictive uncertainty is conceptually divided into *aleatoric* uncertainty, which reflects irreducible noise in the data, and *epistemic* uncertainty, which reflects the model's lack of knowledge [33].

Originally proposed by Lakshminarayanan et al. [40], deep ensembles (DEs) provide a straightforward, non-Bayesian approach to uncertainty quantification. The method aggregates predictions from M copies of the same architecture trained from different random weight initializations. In a deep ensemble, aleatoric and epistemic uncertainty can be explicitly separated.

For regression, the m-th member of a DE predicts the mean $\mu_m(x)$ and variance $\sigma_m^2(x)$ of a Gaussian distribution. The ensemble prediction is modeled as a Gaussian mixture, yielding an aggregated point prediction $\mu(x) = \frac{1}{M}\sum_{m=1}^{M}\mu_m(x)$. The total predictive uncertainty is the variance of this mixture, which elegantly decomposes into the average variance and the variance of the means:

$$\underbrace{\sigma^2(x)}_{\text{Total}} = \underbrace{\frac{1}{M}\sum\nolimits_{m=1}^{M}\sigma_m^2(x)}_{\text{Aleatoric } \sigma^2_{\text{ale}}(x)} + \underbrace{\frac{1}{M}\sum\nolimits_{m=1}^{M}\left(\mu_m(x)-\mu(x)\right)^2}_{\text{Epistemic } \sigma^2_{\text{epi}}(x)}. \tag{2}$$

[1] Modified ensemble methods exist for specific graph tasks (discussed in Section 2.1), but we focus strictly on the standard DE mechanism of repeatedly training the same architecture from different initializations.

For classification, the m-th member of a DE outputs a probability distribution $p_m(y \mid x)$ over classes. The ensemble outputs a single distribution formed by averaging the member probabilities, $p(y \mid x) = \frac{1}{M}\sum_{m=1}^{M} p_m(y \mid x)$. Here, total uncertainty is measured by the predictive entropy $\mathcal{H}$, which similarly decomposes into expected data uncertainty and mutual information [13]:

$$\underbrace{\mathcal{H}[p(y \mid x)]}_{\text{Total}} = \underbrace{\frac{1}{M}\sum_{m=1}^{M} \mathcal{H}[p_m(y \mid x)]}_{\text{Aleatoric } \mathcal{H}_{\text{ale}}(x)} + \underbrace{\left(\mathcal{H}[p(y \mid x)] - \frac{1}{M}\sum_{m=1}^{M} \mathcal{H}[p_m(y \mid x)]\right)}_{\text{Epistemic } \mathcal{H}_{\text{epi}}(x)}. \tag{3}$$

2.1 Related Work

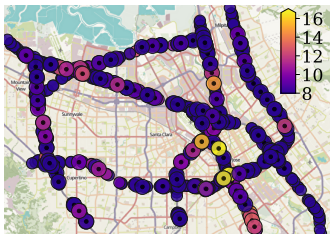
Much of the prior work on GNN ensembles departs from the standard DE recipe. Examples include ensembling models trained on different graph views [52], employing varied GNN architectures [71], or relying on bootstrapped training data [38]. Other approaches approximate Bayesian inference by sampling from a single optimization trajectory [57] or utilizing stochastic message passing [42]. In specialized domains like chemistry and material sciences, ensemble methods often use customized edge-centric architectures, shallow weight sharing, or post-hoc recalibration [10, 29, 66, 68].

Such modifications give up the simplicity of the standard recipe. If they were required for ensemble disagreement on graphs, this would mark a fundamental departure from classical deep learning, where random initialization alone suffices [40, 56]—a departure that was never systematically established. Standard DEs of GNNs appear mainly as baselines, both for graph-specific uncertainty methods [19, 20, 63, 65] and in benchmarks [3, 4, 61]. Their results form the mixed picture described in Section 1, but analyzing DEs is not their aim. We make standard DEs the object of study: we systematically evaluate what they gain in uncertainty estimation over a single GNN, which costs a fraction as much, identify epistemic collapse, and link it to the smoothing bias of message passing.

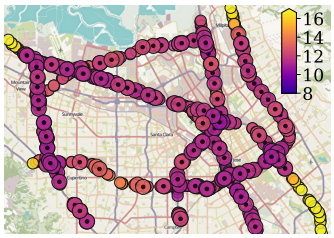
3 A Motivating Example: The PEMS Benchmark

We start by revisiting the PEMS road network benchmark [7]. Originally introduced to evaluate graph Gaussian processes [8], this transductive node regression dataset tasks models with interpolating traffic speeds and quantifying the associated predictive uncertainty. PEMS has no informative node features, so the GNNs receive the training labels as inputs, as in Shi et al. [62].

We expand upon the prior evaluation of Borovitskiy [7] by following the DE recipe of Lakshminarayanan et al. [40]. Specifically, we equip the base GCN models with a variance head trained with the Gaussian NLL (the original benchmark uses an ensemble of prediction-only models). We compare a single GCN, a five-model deep ensemble (DE), a Bayesian GCN (B-GCN) trained via stochastic variational inference [44, 70], and a Matérn Gaussian process on the graph (Graph GP) [8]. As a sanity check, we include a *Trivial* baseline that predicts a constant Gaussian distribution $\mathrm{N}(\mu_{\text{train}}, \sigma^2_{\text{train}})$ fitted to the training labels. For more details on the setup, see Appendix A.



(a) Uncertainty Map: GCN DE



(b) Uncertainty Map: Graph GP

Method	NLL (↓)	RMSE (↓)
Trivial	1.37 ± 0.00	16.4 ± 0.0
GCN	2.51 ± 0.39	13.5 ± 0.1
DE	2.39 ± 0.11	13.6 ± 0.1
B-GCN	1.32 ± 0.09	13.4 ± 0.1
Graph GP	0.98 ± 0.02	11.1 ± 0.2

(c) Performance Metrics

Figure 1: PEMS road network benchmark. Nodes with a black dot in the middle are train nodes. DE produces highly overconfident and uninformative uncertainty estimates (a) compared to the geometrically grounded Graph GP (b). Table (c) confirms that DE fails at uncertainty quantification (NLL), performing significantly worse than Trivial, and only marginally improving over GCN. Red denotes scores worse than trivial and Green better than trivial.

The results, shown in Figure 1, reveal a failure of the deep ensemble. It improves RMSE over the Trivial baseline, but its NLL is much worse. The single GCN is already worse than Trivial, because its variance head is overconfident. A DE could repair this, but it does not: its NLL barely improves. Visually (Figures 1a and 1b), the Graph GP produces an uncertainty map that sensibly increases away from training data. In contrast, the DE's uncertainty is uniformly low and sometimes spikes at training nodes. The Bayesian GCN, whose hidden dimension we had to reduce to train it (Appendix A), beats the Trivial baseline, so the failure here is specific to deep ensembles of GNNs, not to GNNs alone. Finally, the Graph GP performs best, as expected: the signal is a smooth function on the graph, the setting graph GPs were designed for, and there are no rich node features, whose processing is where GNNs normally have the edge.

This motivating example raises a critical question: *Is the failure on PEMS a symptom of a systematic weakness of deep ensembles of graph neural networks, and if so, what causes this weakness?*

4 Benchmarking Ensembles of Graph Neural Networks

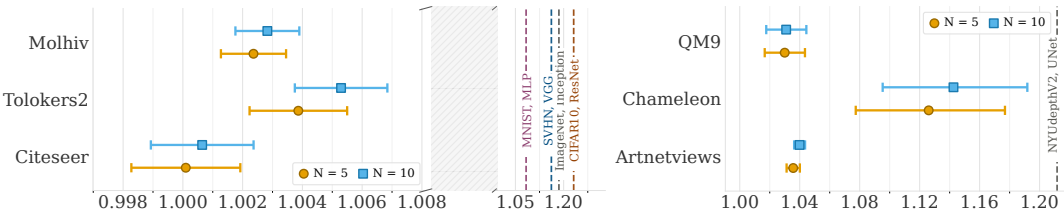
To test whether the failure on `PEMS` points to a systematic weakness, we conduct a comprehensive evaluation of deep ensembles of GNNs across a diverse range of graph learning tasks.

Datasets. In total, we evaluate on eight diverse graph datasets spanning node- and graph-level classification and regression. Having established the failure on the PEMS (`PEMS`) road network in Section 3 and deferring the simplest Cora (`CORA`) [76] to Appendix E, we focus here on the remaining six datasets. For node classification, we include the classic CiteSeer (`CTSR`) [76] and a newer, larger-scale Tolokers2 (`TLK2`) [5]. For node regression, we use Artnetviews (`ARTNV`) [5] and Chameleon (`CHAM`) [59], which both offer modern, structurally challenging evaluation settings. `CHAM`, specifically, exhibits heterophily [47] in a benign form still amenable to message passing. Finally, for graph-level classification we use MolHIV (`MHIV`) from the Open Graph Benchmark [31], and for graph-level regression, we use the classic QM9 dataset [74]. `MHIV` is highly imbalanced (about 3.5% positives), so accuracy barely separates any model from the Trivial baseline (NLL remains informative). We subsample QM9 to 5% of its original size (denoted `QM9-5%`) and restrict to the HOMO-LUMO gap prediction task, one of the most difficult in the dataset. We stratify this subsampling by the target variable to preserve its overall distribution. We deliberately implement this subsampling to test a data-scarce regime, a setting where the quality of uncertainty estimates is especially critical. Together, these datasets capture diverse graph learning settings.

Models. We select architectures that represent standard practice for each task. For `CTSR`, `ARTNV`, and `QM9-5%`, we employ standard GCNs [36]. On `ARTNV`, the GCN is the established state-of-the-art [5]. While attention-based models can yield marginal improvements on `CTSR`, GCNs are often preferred for their superior training stability and simplicity, hence our choice. For `QM9-5%`, although top-performing methods utilize 3D spatial coordinates, GCNs perform comparably to other

Table 1: Absolute performance metrics for a DE of five models. Values represent NLL, ECE (for regression stands for miscalibration area), and point estimation metric (PEM: RMSE for regression and accuracy for classification), showing the mean and standard deviation over 10 independent runs. The ECE of the trivial baseline is not taken into account when highlighting the best score. Trivial baseline has low ECE by construction but is uninformative: it predicts the same regardless of input.

		Classification			Regression		
		CTSR	**TLK2**	**MHIV**	**ARTNV**	**CHAM**	**QM9-5%**
Trivial	NLL	1.792 ± 0.000	0.525 ± 0.000	0.141 ± 0.000	1.425 ± 0.000	1.415 ± 0.000	1.404 ± 0.000
	ECE	0.069 ± 0.000	0.000 ± 0.000	0.006 ± 0.000	0.015 ± 0.000	0.072 ± 0.000	0.070 ± 0.000
	PEM	0.077 ± 0.000	0.782 ± 0.000	0.968 ± 0.000	1.715 ± 0.000	2.160 ± 0.000	1.268 ± 0.000
GNN	NLL	$\mathbf{0.929 \pm 0.003}$	0.371 ± 0.002	0.115 ± 0.001	0.993 ± 0.005	0.868 ± 0.066	0.833 ± 0.021
	ECE	0.142 ± 0.002	0.079 ± 0.000	0.057 ± 0.004	0.033 ± 0.017	$\mathbf{0.137 \pm 0.054}$	$\mathbf{0.568 \pm 0.014}$
	PEM	0.710 ± 0.004	0.854 ± 0.005	0.970 ± 0.001	1.113 ± 0.006	1.215 ± 0.044	0.384 ± 0.014
DE	NLL	$\mathbf{0.929 \pm 0.001}$	$\mathbf{0.367 \pm 0.001}$	$\mathbf{0.112 \pm 0.001}$	$\mathbf{0.958 \pm 0.004}$	$\mathbf{0.749 \pm 0.026}$	$\mathbf{0.803 \pm 0.003}$
	ECE	$\mathbf{0.141 \pm 0.002}$	$\mathbf{0.077 \pm 0.005}$	$\mathbf{0.054 \pm 0.003}$	$\mathbf{0.009 \pm 0.002}$	0.266 ± 0.027	0.595 ± 0.004
	PEM	$\mathbf{0.711 \pm 0.000}$	$\mathbf{0.856 \pm 0.002}$	$\mathbf{0.972 \pm 0.001}$	$\mathbf{1.086 \pm 0.004}$	$\mathbf{1.132 \pm 0.022}$	$\mathbf{0.362 \pm 0.005}$



(a) Classification datasets (b) Regression datasets

Figure 2: Likelihood ratios $\exp(\text{NLL}_{\text{GNN}} - \text{NLL}_{\text{DE}})$ and 95% Gaussian confidence intervals. This quantifies the relative predictive likelihood improvement of DE over GNN for both classification and regression datasets. Baselines indicate typical improvements observed in foundational DE literature.

purely topological GNNs [21, 74] and provide a clean, representative baseline. Conversely, we use GATv2 [9] for `CHAM` and `TLK2`, where evidence favors attention; on `TLK2`, it is the state-of-the-art [5]. For `MHIV`, we use GIN, the strongest standard message-passing baseline on the OGB leaderboard [31]. For regression tasks, we equip each architecture with an additional variance head to capture aleatoric uncertainty, jointly optimizing the predictive mean and variance by maximizing the likelihood. All models are optimized to achieve performance levels commensurate with the established literature. Appendix A provides more details.

Metrics. Our primary metric for jointly evaluating predictive performance and uncertainty quality is negative log-likelihood (NLL). It is a strictly proper scoring rule [25]: in expectation it is minimized only by the true predictive distribution. It is also the standard metric in the DE literature [40, 56]. For finer-grained insights, we evaluate separate metrics that isolate predictive accuracy from uncertainty calibration. Specifically, we report the root mean squared error (RMSE) for regression and accuracy for classification. To assess calibration, we compute the expected calibration error (ECE) [27] for classification and its regression analog, expected coverage error (ECE; termed *miscalibration area* in Tran et al. [64], which we multiply by 2 to match the expected calibration error's range). We report the mean and standard deviation of these metrics across ten independent runs.

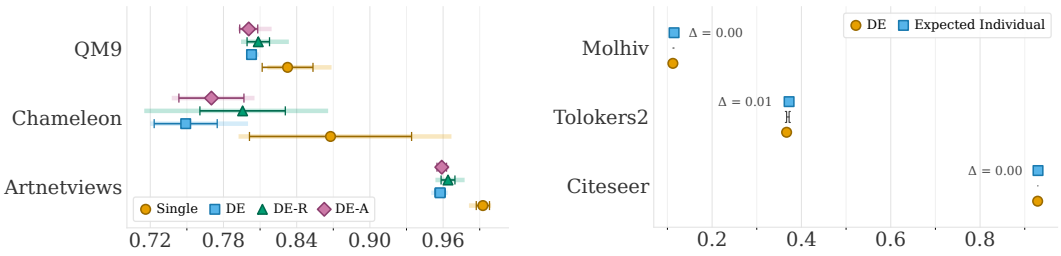
Results. In contrast to the pathological failure on `PEMS`, Table 1 shows that deep ensembles generally outperform the Trivial baseline across all other datasets. They improve both point estimation and NLL compared to a single GNN in most cases, though the resulting gains appear modest and calibration (ECE) sometimes degrades, as seen on `CHAM` and `QM9-5%`. To properly analyze this behavior, we must ask: how do these improvements actually compare to the established literature?

To answer this question and quantify the performance gap, we compute the likelihood ratios: $\exp(\text{NLL}_{\text{GNN}} - \text{NLL}_{\text{DE}})$. They provide an interpretable measure of the relative gain of an ensemble over a single model, representing the multiplicative factor by which the ensemble increases the likelihood of the test labels.[2] Figure 2 compares these ratios against historical improvements derived from foundational deep ensemble literature predominantly anchored in computer vision [40, 43, 50].

For classification tasks, the contrast is drastic: while classic computer vision benchmarks report ensembles assigning roughly 20% more likelihood to test points than a single model, our graph ensembles yield improvements below 1%. In regression, the reference computer vision baseline similarly shows a $\sim$20% improvement, whereas our graph ensembles typically assign only about 5% more likelihood to the test data. Even the most favorable case, `CHAM`, peaks at roughly 14% improvement, which is not only well below the standard computer vision baseline but also exhibits a wide confidence interval, meaning the gains of a DE are highly unstable.

Crucially, Table 1 reveals that improvements in NLL are consistently coupled with improvements in accuracy or RMSE. Because NLL is a composite metric, these likelihood gains might be driven entirely by slightly more accurate point estimates rather than genuinely improved uncertainty calibration. To determine whether deep ensembles on graphs actually fulfill their primary purpose—improving uncertainty quantification—we must further analyze these mechanisms in isolation.

[2]We avoid reporting relative percentage changes in NLL. Since log-likelihood lacks an absolute zero and can be negative, especially in continuous settings, direct ratios are mathematically flawed. We instead report the exponentiated difference, providing a sound and intuitive likelihood ratio.



(a) Regression: Comparing the NLL

(b) Classification: Jensen's Gap

Figure 3: Deconstructing the source of NLL improvements. (a) compares the NLL of a GNN, a DE, and two additional baselines (DE-R and DE-A) that isolate the point prediction improvement effect. (b) visualizes the nonnegative gap between the expected individual NLL and the ensemble's NLL (Jensen's gap). The marker represents mean, thin bars represent $\pm\sigma$ and transparent ones min–max.

Larger ensembles and distribution shift. While we report results for ensembles of five models here, we attest that these limitations are persistent. As detailed in Appendix E, scaling the ensembles to ten models yields virtually identical trends. Additionally, evaluating deep ensembles under covariate shift on purpose-built splits for `ARTNV` and `TLK2` reveals that the relative improvements over a single GNN remain just as marginal, if not worse, than in the in-distribution setting (Appendix B).

5 How Deep Ensembles Improve Uncertainty

While Section 4 shows that deep ensembles somewhat reduce NLL, we must disentangle whether these gains reflect genuinely better uncertainty quantification or merely more accurate point predictions. To this end, in Section 5.1, we deconstruct the improvements of deep ensembles, revealing that the modest gains in NLL are driven almost entirely by variance reduction and predictive accuracy rather than better uncertainty estimates. Further, in Section 5.2, we investigate *why* this occurs by analyzing ensemble diversity. We identify a systemic phenomenon we term *epistemic collapse*, wherein individual GNNs in an ensemble fail to explore diverse functional hypotheses.

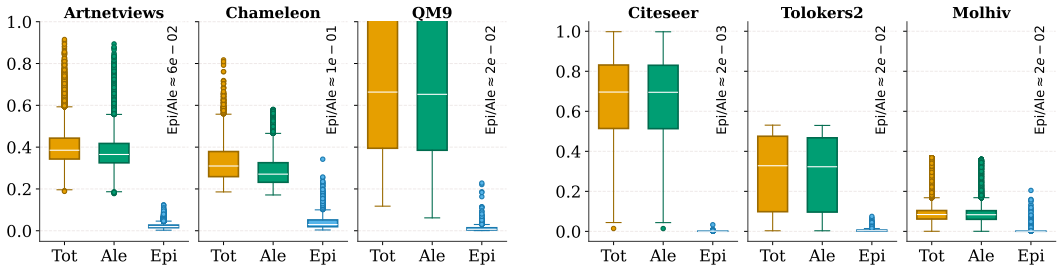
5.1 Deconstructing the NLL Gains

Because regression and classification parameterize uncertainty differently, we require distinct methodologies to trace the source of the NLL improvements. For regression, we can explicitly decouple the predicted mean from the predicted variance to isolate point prediction gains. For classification, where point estimates and uncertainty are inherently entangled in the predicted probability distribution, we instead analyze the distributional benefit of ensembling via Jensen's inequality gap.

Regression. To isolate the contribution of the ensemble's point prediction, we evaluate two constructed baselines alongside a standard single GNN and a full DE. The first baseline pairs the DE's mean prediction—the average of the individual means—with the variance of a randomly selected single model (DE-R). The second baseline pairs the DE's mean prediction with only the aleatoric variance (DE-A), which is the average of the individual variances (cf. Equation (2)).

Figure 3a displays the NLL performance for the GNN, DE, DE-R, and DE-A models. On `ARTNV` and `QM9-5%`, all models utilizing the DE mean (DE, DE-R, DE-A) achieve nearly identical NLL improvements over the single model. This is somewhat less so for `CHAM`, as Section 6 later on suggests is plausible under heterophily. Thus, the already modest NLL gains are driven mainly by the ensembled mean providing a smoothed, slightly more accurate point prediction, rather than the ensemble supplying better-calibrated uncertainty. This explains how in Table 1 DEs can improve NLL on datasets like `CHAM` and `QM9-5%` while simultaneously worsening the calibration error (ECE). Furthermore, the distribution of NLL scores in Figure 3a reveals another key insight. We observe that the best-performing single models can sometimes reach NLL scores comparable to those of the full ensemble or get extremely close.[3] This suggests that ensembling here acts primarily as a variance reduction technique over optimization noise, stabilizing the performance of a single model.

[3]Because data splits are fixed across runs, the variance stems entirely from random initialization/batching.



(a) Uncertainty breakdown for regression.

(b) Uncertainty breakdown for classification.

Figure 4: Comparison of epistemic, aleatoric, and total uncertainty across datasets, demonstrating the *epistemic collapse*. For both regression and classification, the epistemic part of uncertainty collapses to near-zero values, while aleatoric uncertainty remains relatively large.

Classification. Unlike regression, classification inherently entangles point estimates and confidence scores in a single predicted probability distribution. To understand if the ensemble genuinely improves this distribution—for instance, by tempering overconfident predictions—we analyze the gap between the expected NLL of an individual model and the NLL of the ensemble. The ensemble averages the members' class probabilities, so by Jensen's inequality its NLL is at most the average NLL of its members. The difference (Jensen's gap) is zero exactly when all members assign the same probability to the true class, and it grows with their disagreement.

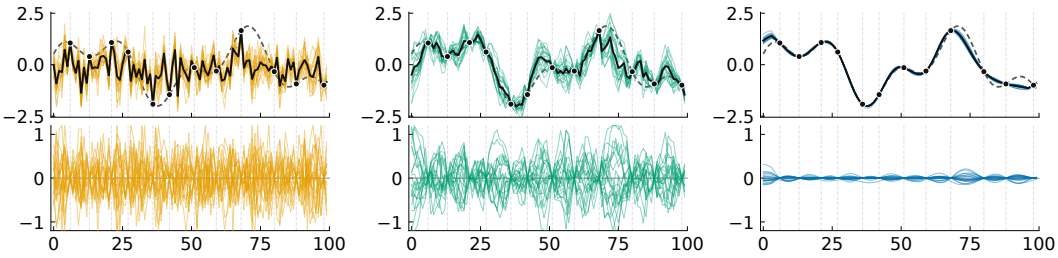
As shown in Figure 3b, this gap is remarkably small across all tested classification datasets. This near-zero gap implies that the individual models are outputting practically identical probability distributions. Because the ensembled distributions are virtually indistinguishable from the individual ones, the ensemble provides no fundamental enhancement to the predictive capabilities of the model. Instead we observe that ensembling acts as a variance reduction mechanism over optimization noise. The variance in performance across random initializations heavily dominates the actual ensembling effect, meaning the best-performing individual models in our runs sometimes nearly match or even improve the NLL of the full ensemble.

5.2 Diversity and Epistemic Collapse

In Section 5.1, we established that DEs of GNNs act primarily as variance-reducing smoothers. To understand why this happens, we must consider the fundamental mechanism deep ensembles use to quantify uncertainty: the disagreement among their constituent models [1, 14, 40]. This disagreement is the epistemic component of uncertainty (Equations (2) and (3)), the main contribution of a DE over a single model. If the members barely disagree, the ensemble reduces to a smoother. We therefore investigate whether GNN ensembles suffer from what we term *epistemic collapse*: a systemic failure of independently trained models to explore diverse functional hypotheses.

Figure 4 breaks the predictive uncertainty into its aleatoric and epistemic parts. Across all tasks, the aleatoric part dominates. In regression, the epistemic variance is dwarfed by the aleatoric variance; in classification, the epistemic entropy normalized by $\log C$ is close to zero, rarely exceeding 0.1. This deviates sharply from classic deep learning settings. For instance, Kendall and Gal [33] report an epistemic-to-aleatoric ratio of roughly 5 for in-distribution regression in computer vision, while our ratios are well below 1. In classification, epistemic collapse has been reported mainly for highly over-parameterized models or massive ensembles [18, 37]; with similar normalization, we observe even smaller values across standard architectures and datasets. Importantly, the members agree even where more epistemic uncertainty is warranted: under covariate shift, where errors grow, the DE's NLL gain stays below 3% (Appendix B), and on `PEMS`, the Graph GP shows that much more epistemic uncertainty pays off in NLL. Ultimately, because GNNs converge to highly similar functions, they neutralize the exact mechanism deep ensembles rely on to improve uncertainty quantification.

The results thus point towards epistemic collapse: despite independent initializations, the members converge to very similar predictions. Do they also converge to similar weights? If so, averaging them into a single network (a model soup, [72]) would perform at least as well as a single member. Instead, it performs generally worse (Appendix C), as expected from the permutation symmetries of



(a) 2-layer MLP (b) 1-layer GCN (c) 4-layer GCN

Figure 5: Smoothing limits the disagreement. Ensembles trained on a path graph with 100 nodes; the x-axis is the position along the path. Top: target (dashed), anchors (dots), member predictions (colored) and ensemble mean (solid black). Bottom: member deviations from the ensemble mean.

neural networks [2, 17]. This suggests the collapse is thus functional: the members differ in weights yet implement nearly the same function.

6 Why Ensemble Members Agree: A Smoothness Bound

Section 5.2 showed that the members of DEs of GNNs barely disagree. We now take a first step towards explaining why, concentrating specifically on node-level tasks. We show that if the members' predictions are smooth on the graph and fit the training nodes, their disagreement is bounded by this smoothness and by how well the graph connects the other nodes to the training nodes.

Intuition. Consider the deviation of a DE member's predictions from the ensemble mean as a signal on the nodes of the graph. At the training nodes, which we call *anchors*, this deviation is close to zero, as members fit the labels. If the members' predictions change little between neighboring nodes, so do their deviations. A signal that is zero at the anchors and changes little between neighboring nodes cannot grow large, unless some nodes are poorly connected to the anchors. Figure 5 illustrates this on a path graph, for an MLP and GCNs of different depth. All models fit the anchors perfectly, and the deviations vanish at them. An MLP is oblivious to the graph structure, so between the anchors its members disagree freely from node to node. The deviations of GCN members are smooth, and the deeper GCN, which smooths more, barely disagrees. We now make this argument precise, measuring how much predictions change between neighboring nodes by the *Dirichlet energy* [60].

Setup. Let G be a graph with n nodes and a graph Laplacian $\mathbf{L}$, and let S be the set of anchors (training nodes with labels y_s), such that every connected component of G contains at least one anchor. Let $\boldsymbol{f}_w \in \mathbb{R}^n$ be the predictions $f_w(v)$ of a model with trained weights w at all nodes v. Randomness in initialization and training makes w, and hence $\boldsymbol{f}_w$, random: DE members are merely independent draws of $\boldsymbol{f}_w$. Denoting the Dirichlet energy of $\boldsymbol{z} \in \mathbb{R}^n$ by $\mathcal{E}(\boldsymbol{z}) = \boldsymbol{z}^\top \mathbf{L}\, \boldsymbol{z}$, we make two assumptions. **Assumption 1:** predictions are smooth in expectation, $\mathbb{E}_w[\mathcal{E}(\boldsymbol{f}_w)] \le \rho$, where we call ρ the *energy budget*. This reflects the smoothing bias of message passing [46, 49], which we discuss below. **Assumption 2:** predictions fit the anchors, $\mathbb{E}_w[(f_w(s) - y_s)^2] \le \varepsilon^2$ for all $s \in S$.

Theorem. *Under Assumptions 1 & 2, the total epistemic variance* $V_{epi} = \sum_v \mathrm{Var}_w(f_w(v))$ *satisfies*

$$V_{epi} \le \inf_{\alpha > 0} \frac{\rho - \mathcal{E}_{\min} + \alpha |S| \varepsilon^2}{\lambda_\alpha}, \tag{4}$$

where $\mathcal{E}_{\min} = \min\{\mathcal{E}(\boldsymbol{z}) : |z_s - y_s| \le \varepsilon \text{ for all } s \in S\}$ *is the* minimal fit energy*, the least energy needed to fit the anchors within* ε *error, and* $\lambda_\alpha = \min_{\|\boldsymbol{x}\|=1} \big(\mathcal{E}(\boldsymbol{x}) + \alpha \sum_{s \in S} x_s^2\big)$ *is the* anchored spectral gap*, which is small when some nodes are poorly connected to the anchors.*

The proof (Appendix D) roughly follows the intuition above, combining Assumptions 1 and 2 with a Poincaré-type inequality. If predictions fit the anchors exactly ($\varepsilon = 0$), letting $\alpha \to \infty$ gives $V_{\text{epi}} \le (\rho - \mathcal{E}_{\min})/\lambda_U$, where λ_U is the smallest eigenvalue of $\mathbf{L}$ restricted to the non-anchor nodes. The bound also covers classification, where a model predicts a vector of class probabilities at each node. The theorem holds for such vector-valued predictions, with squares replaced by squared

norms and one-hot labels as targets. Assumption 2 then bounds the model's Brier score at each anchor. The bound thus limits how much the DE members' class probabilities can disagree, and therefore epistemic uncertainty, since close probability vectors have close entropies.

The *spare energy* $\rho - \mathcal{E}_{\min}$ is the part of the budget left for disagreement, so DE members can disagree much only if it is large or some nodes are poorly connected to the anchors. The model enters the bound only through ρ and ε, which depend on both its architecture and its training. For example, an MLP ignores the graph, so its predictions need not be smooth and its energy budget can be large (Figure 5). For graph transformers it can go either way: their ρ depends on whether they *learn* a smoothness bias. For message passing, we expect ρ to be small, as the next result suggests.

Proposition 1 (Corollary 3.4.1 in Cai and Wang [11])**.** *For an l-layer GCN with ReLU activations and no biases, $\mathcal{E}(\mathbf{X}^{(l)}) = \mathcal{O}((s\bar{\lambda})^l)$, where $\mathbf{X}^{(l)}$ are the node representations after l layers, $\mathcal{E}$ is the energy for the augmented normalized Laplacian, s bounds the squared spectral norms of the weight matrices, and $\bar{\lambda} = \max_i(1-\lambda_i)^2 < 1$ over the nonzero eigenvalues λ_i of this Laplacian.*

Thus, the energy shrinks with depth when $s\bar{\lambda} < 1$, as in Figure 5. Such results, also available for attention-based GNNs [73], were developed to study oversmoothing, where the representations of different nodes within one network become indistinguishable after many layers [55, 60].[4] Our bound puts them to use in a much milder regime: it needs no oversmoothing, only a moderate budget ρ at realistic depth, where smoothing is known to be beneficial [35]. This supports Assumption 1 for message-passing architectures, though it does not guarantee a small ρ, since s depends on training. However, recent results [34] seem to suggest s remains generally small on message-passing models.

Discussion and limitations. Together, the theorem and the proposition above theoretically support the epistemic collapse: message passing tends to keep ρ small, leaving little spare energy for disagreement. Heterophily may leave more: a model that captures it must change its predictions between neighbors throughout the graph, which can raise ρ well above $\mathcal{E}_{\min}$. This is consistent with the heterophilous `CHAM` having the largest NLL gain and epistemic-to-aleatoric ratio among regression tasks. The bound does not account for training dynamics, so it cannot by itself explain the gap between a DE and a B-GCN of the same family on `PEMS` (Section 3). Still, combined with our empirical findings, it offers an explanation: DE members each converge to different optima that implement nearly the same function (Section 5.2), whereas the B-GCN samples weights around an optimum, spreading its predictions within the limit set by the bound.

7 Conclusion

We can now answer the question posed in Section 3. The failure on `PEMS` is indeed a symptom of a systematic weakness of DEs of GNNs: epistemic collapse. Empirically, the members of a DE barely disagree on every dataset we study. The NLL gains of the DE are far smaller than in computer vision and come almost entirely from averaging point predictions. Theoretically, we link this collapse to the smoothing bias of message passing (not *over*smoothing!): it makes predictions similar for neighboring nodes. For node-level tasks, our bound shows that such smoothness, together with training fit, limits how much the members can disagree. Together, our empirical and theoretical results answer the question in our title: *Do deep ensembles actually capture uncertainty in graph neural networks?* Largely, they do not, at least not beyond what a single member already captures.

In practice, standard DEs of GNNs multiply training and inference costs, which is hard to justify given that they mostly average out optimization noise. Since the best of several seeds can come close to the ensemble, selecting it on validation might retain most of this benefit at single-model inference cost. Better uncertainty on graphs is nonetheless attainable: on `PEMS`, a Bayesian GCN beats the Trivial baseline, so the smoothing bias limits, but does not altogether preclude, useful epistemic uncertainty in GNNs. We studied standard DEs of message-passing GNNs because their appeal lies in simplicity: they only require training the same model several times. Graph-specific diversity techniques or architectures beyond message passing may fare better at the cost of this simplicity, and we hope our analysis guides the search for methods that deliver better uncertainty on graphs.

[4] Beyond oversmoothing, prior work relates message passing to the variance of predictions, but only for infinitely wide networks (NTK) [15, 30, 39, 53] or at initialization [32, 41], not for trained, finite networks.

AI USE STATEMENT

In this work, we used generative AI tools to review and give feedback on the theoretical results and their proofs, and for brainstorming. We have not used generative AI tools for the other tasks requiring disclosure, such as generating data, implementing methods, or running experiments. Additionally, we used generative AI tools to edit the text, to aid in creating figures, to aid the literature review, and for occasional code verification. We have reviewed all AI-assisted work by at least two authors, and checked cited results against the original papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Complete hyperparameter configurations, baseline details, and experimental setups are provided in Appendix A. We also cite relevant works that describe the setups we adopted in more detail. Appendix D states all assumptions and gives detailed proofs of our theoretical results. We provide the code used to run the experiments as supplemental material. It includes instructions on how to run the experiments and anonymous links to intermediary results for inspection. All code uses standard rootless containerized environments to ensure ease of use and reproducibility.

REFERENCES

[1] T. Abe, E. K. Buchanan, G. Pleiss, R. Zemel, and J. P. Cunningham. Deep ensembles work, but are they necessary? In *Advances in Neural Information Processing Systems*, 2022. Cited on page 7.

[2] S. Ainsworth, J. Hayase, and S. Srinivasa. Git Re-Basin: Merging Models modulo Permutation Symmetries. In *International Conference on Learning Representations*, 2023. Cited on pages 8, 16.

[3] G. Bazhenov, S. Ivanov, M. Panov, A. Zaytsev, and E. Burnaev. Towards OOD Detection in Graph Classification from Uncertainty Estimation Perspective. *arXiv preprint arXiv:2206.10691*, 2022. Cited on page 3.

[4] G. Bazhenov, D. Kuznedelev, A. Malinin, A. Babenko, and L. Prokhorenkova. Evaluating Robustness and Uncertainty of Graph Models Under Structural Distributional Shifts. In *Advances in Neural Information Processing Systems*, 2023. Cited on pages 3, 15.

[5] G. Bazhenov, O. Platonov, and L. Prokhorenkova. GraphLand: Evaluating Graph Machine Learning Models on Diverse Industrial Data. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. Cited on pages 4, 5, 16, 21.

[6] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra. Weight uncertainty in neural networks. In *International Conference on International Conference on Machine Learning*, 2015. Cited on page 15.

[7] V. Borovitskiy. PeMS Regression: A Benchmark Suite for Node Regression with Uncertainty, 2025. URL: https://github.com/vabor112/pems-regression. Cited on pages 1, 3, 15.

[8] V. Borovitskiy, I. Azangulov, A. Terenin, P. Mostowsky, M. Deisenroth, and N. Durrande. Matérn Gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, 2021. Cited on page 3.

[9] S. Brody, U. Alon, and E. Yahav. How Attentive are Graph Attention Networks? In *International Conference on Learning Representations*, 2022. Cited on page 5.

[10] J. Busk, P. Bjørn Jørgensen, A. Bhowmik, M. N. Schmidt, O. Winther, and T. Vegge. Calibrated uncertainty for molecular property prediction using ensembles of message passing neural networks. *Machine Learning: Science and Technology*, 3, 2021. Cited on page 3.

[11] C. Cai and Y. Wang. A Note on Over-Smoothing for Graph Neural Networks. In *Graph Representation Learning and Beyond (GRL+) Workshop at ICML 2020*, 2020. Cited on page 9.

[12] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. Cited on page 18.

[13] S. Depeweg, J.-M. Hernandez-Lobato, F. Doshi-Velez, and S. Udluft. Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and Risk-sensitive Learning. In *International Conference on Machine Learning*, 2018. Cited on page 3.

[14] T. G. Dietterich. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*, 2000. Cited on page 7.

[15] S. Du, K. Hou, R. R. Salakhutdinov, B. Poczos, R. Wang, and K. Xu. Graph Neural Tangent Kernel: Fusing Graph Neural Networks with Graph Kernels. In *Advances in Neural Information Processing Systems*, 2019. Cited on page 9.

[16] V. P. Dwivedi and X. Bresson. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020. Cited on page 1.

[17] R. Entezari, H. Sedghi, O. Saukh, and B. Neyshabur. The Role of Permutation Invariance in Linear Mode Connectivity of Neural Networks. In *International Conference on Learning Representations*, 2022. Cited on pages 8, 16.

[18] M. Fellaji and F. Pennerath. The Epistemic Uncertainty Hole: an issue of Bayesian Neural Networks, 2024. Cited on page 7.

[19] D. Fuchsgruber, T. Wollschläger, B. Charpentier, A. Oroz, and S. Günnemann. Uncertainty for active learning on graphs. In *International Conference on Machine Learning*, 2024. Cited on page 3.

[20] D. Fuchsgruber, T. Wollschläger, and S. Günnemann. Energy-based Epistemic Uncertainty for Graph Neural Networks. In *Advances in Neural Information Processing Systems*, 2024. Cited on page 3.

[21] V. Fung, J. Zhang, E. Juarez, and B. G. Sumpter. Benchmarking graph neural networks for materials chemistry. *npj Computational Materials*, 2021. Cited on page 5.

[22] R. Garnett. *Bayesian optimization*. Cambridge University Press, 2023. Cited on page 1.

[23] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for Quantum chemistry. In *International Conference on Machine Learning*, 2017. Cited on page 2.

[24] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2010. Cited on page 15.

[25] T. Gneiting and A. E. Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 2007. Cited on page 5.

[26] A. Graves. Practical Variational Inference for Neural Networks. In *Advances in Neural Information Processing Systems*, 2011. Cited on page 15.

[27] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. Cited on page 5.

[28] E. Heid, C. J. McGill, F. H. Vermeire, and W. H. Green. Characterizing Uncertainty in Machine Learning for Chemistry. *Journal of Chemical Information and Modeling*, 2023. Cited on page 1.

[29] L. Hirschfeld, K. Swanson, K. Yang, R. Barzilay, and C. W. Coley. Uncertainty Quantification Using Neural Networks for Molecular Property Prediction. *Journal of Chemical Information and Modeling*, 60, 2020. Cited on page 3.

[30] J. Hu, J. Shen, B. Yang, and L. Shao. Infinitely Wide Graph Convolutional Networks: Semi-supervised Learning via Gaussian Processes. *arXiv preprint arXiv:2002.12168*, 2020. Cited on page 9.

[31] W. Hu, M. Fey, M. Zitnik, Y. Dong, H. Ren, B. Liu, M. Catasta, and J. Leskovec. Open Graph Benchmark: Datasets for Machine Learning on Graphs. *arXiv preprint arXiv:2005.00687*, 2020. Cited on pages 4, 5.

[32] D. Kelesis, D. Fotakis, and G. Paliouras. Reducing oversmoothing through informed weight initialization in graph neural networks. *Applied Intelligence*, 2025. Cited on page 9.

[33] A. Kendall and Y. Gal. What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, 2017. Cited on pages 2, 7.

[34] N. Keriven. Backward Oversmoothing: why is it hard to train deep Graph Neural Networks? *arXiv preprint arXiv:2505.16736*, 2026. Cited on page 9.

[35] N. Keriven. Not Too Little, Not Too Much: A Theoretical Analysis of Graph (Over)Smoothing. In *Advances in Neural Information Processing Systems*, 2022. Cited on page 9.

[36] T. N. Kipf and M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*, 2017. Cited on pages 1, 2, 4.

[37] A. Kirsch. (Implicit) Ensembles of Ensembles: Epistemic Uncertainty Collapse in Large Models. *Transactions on Machine Learning Research*, 2025. Cited on page 7.

[38] S. Krieg, W. Burgis, P. Soga, and N. Chawla. Deep Ensembles for Graphs with Higher-order Dependencies. In *International Conference on Learning Representations*, 2023. Cited on page 3.

[39] S. Krishnagopal and L. Ruiz. Graph Neural Tangent Kernel: Convergence on Large Graphs. In *International Conference on Machine Learning*, 2023. Cited on page 9.

[40] B. Lakshminarayanan, A. Pritzel, and C. Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, 2017. Cited on pages 1–3, 5, 7.

[41] J. Li, Y. Song, X. Song, and D. Wipf. On the Initialization of Graph Neural Networks. In *International Conference on Machine Learning*, 2023. Cited on page 9.

[42] Q. Lin, S. Yu, K. Sun, W. Zhao, O. Alfarraj, A. Tolba, and F. Xia. Robust Graph Neural Networks via Ensemble Learning. *Mathematics*, 10, 2022. Cited on page 3.

[43] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In *Advances in Neural Information Processing Systems*, 2020. Cited on page 5.

[44] C. Louizos and M. Welling. Multiplicative normalizing flows for variational Bayesian neural networks. In *International Conference on Machine Learning*, 2017. Cited on pages 3, 15.

[45] C. Louizos and M. Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *International Conference on International Conference on Machine Learning*, 2016. Cited on page 15.

[46] S. Luan, C. Hua, Q. Lu, L. Ma, L. Wu, X. Wang, M. Xu, X.-W. Chang, D. Precup, R. Ying, S. Z. Li, J. Tang, G. Wolf, and S. Jegelka. The Heterophilic Graph Learning Handbook: Benchmarks, Models, Theoretical Analysis, Applications and Challenges. *arXiv preprint arXiv:2407.09618*, 2024. Cited on pages 8, 21.

[47] S. Luan, Q. Lu, C. Hua, X. Wang, J. Zhu, and X.-W. Chang. Re-evaluating the Advancements of Heterophilic Graph Learning. *arXiv preprint arXiv:2409.05755*, 2024. Cited on page 4.

[48] S. Luan, Q. Lu, C. Hua, X. Wang, J. Zhu, and X.-W. Chang. Revealing the Pitfalls and Re-Evaluating the Advancement of Heterophilic Graph Learning. In *International Conference on Artificial Neural Networks*, 2026. Cited on page 21.

[49] Y. Ma, X. Liu, N. Shah, and J. Tang. Is Homophily a Necessity for Graph Neural Networks? In *International Conference on Learning Representations*, 2022. Cited on pages 8, 21.

[50] H. Manh Bui and A. Liu. Density-Regression: Efficient and Distance-aware Deep Regressor for Uncertainty Estimation under Distribution Shifts. In *International Conference on Artificial Intelligence and Statistics*, 2024. Cited on page 5.

[51] P. Mostowsky, V. Dutordoir, I. Azangulov, N. Jaquier, M. J. Hutchinson, A. Ravuri, L. Rozo, A. Terenin, and V. Borovitskiy. The GeometricKernels Package: Heat and Matérn Kernels for Geometric Learning on Manifolds, Meshes, and Graphs. *Journal of Machine Learning Research*, 2025. Cited on page 1.

[52] V. T. Nguyen, D. A. Pham, A. T. Le, J. Peter, and G. Gust. Persistent Homology-induced Graph Ensembles for Time Series Regressions. *arXiv preprint arXiv:2503.14240*, 2025. Cited on page 3.

[53] Z. Niu, M. Anitescu, and J. Chen. Graph Neural Network-Inspired Kernels for Gaussian Processes in Semi-Supervised Learning. In *International Conference on Learning Representations*, 2023. Cited on page 9.

[54] J. Ojha, O. Presacan, P. G. Lind, E. Monteiro, and A. Yazidi. Navigating Uncertainty: A User-Perspective Survey of Trustworthiness of AI in Healthcare. *ACM Trans. Comput. Healthcare*, 6, 2025. Cited on page 1.

[55] K. Oono and T. Suzuki. Graph Neural Networks Exponentially Lose Expressive Power for Node Classification. In *International Conference on Learning Representations*, 2020. Cited on page 9.

[56] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, 2019. Cited on pages 1, 3, 5, 15.

[57] C. M. A. Rahman, G. Bhandari, N. M. Nasrabadi, A. H. Romero, and P. K. Gyawali. Enhancing material property prediction with ensemble deep graph convolutional networks. *Frontiers in Materials*, 11, 2024. Cited on page 3.

[58] L. Rampášek, M. Galkin, V. P. Dwivedi, A. T. Luu, G. Wolf, and D. Beaini. Recipe for a general, powerful, scalable graph transformer. In *Advances in Neural Information Processing Systems*, 2022. Cited on page 1.

[59] B. Rozemberczki, C. Allen, and R. Sarkar. Multi-Scale attributed node embedding. *Journal of Complex Networks*, 9, 2021. Cited on page 4.

[60] T. K. Rusch, M. M. Bronstein, and S. Mishra. A Survey on Oversmoothing in Graph Neural Networks. *arXiv preprint arXiv:2303.10993*, 2023. Cited on pages 1, 8, 9.

[61] G. Scalia, C. A. Grambow, B. Pernici, Y.-P. Li, and W. H. Green. Evaluating Scalable Uncertainty Estimation Methods for Deep Learning-Based Molecular Property Prediction. *Journal of Chemical Information and Modeling*, 60, 2020. Cited on page 3.

[62] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun. Masked Label Prediction: Unified Message Passing Model for Semi-Supervised Classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. Cited on page 3.

[63] M. Stadler, B. Charpentier, S. Geisler, D. Zügner, and S. Günnemann. Graph Posterior Network: Bayesian Predictive Uncertainty for Node Classification. In *Advances in Neural Information Processing Systems*, 2021. Cited on pages 1, 3.

[64] K. Tran, W. Neiswanger, J. Yoon, Q. Zhang, E. Xing, and Z. W. Ulissi. Methods for comparing uncertainty quantifications for material property predictions. *Machine Learning: Science and Technology*, 1, 2020. Cited on pages 1, 5.

[65] P. Trivedi, M. Heimann, R. Anirudh, D. Koutra, and J. J. Thiagarajan. Accurate and Scalable Estimation of Epistemic Uncertainty for Graph Neural Networks. In *International Conference on Learning Representations*, 2024. Cited on pages 1, 3.

[66] D. Varivoda, R. Dong, S. S. Omee, and J. Hu. Materials Property Prediction with Uncertainty Quantification: A Benchmark Study. *arXiv preprint arXiv:2211.02235*, 2022. Cited on page 3.

[67] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph Attention Networks. In *International Conference on Learning Representations*, 2018. Cited on pages 1, 2.

[68] T. Vinchurkar, K. Abdelmaqsoud, and J. R. Kitchin. Uncertainty quantification in graph neural networks with shallow ensembles. *Machine Learning: Science and Technology*, 6, 2025. Cited on page 3.

[69] F. Wang, Y. Liu, K. Liu, Y. Wang, S. Medya, and P. S. Yu. Uncertainty in Graph Neural Networks: A Survey. *Transactions on Machine Learning Research*, 2024. Cited on page 1.

[70] Y. Wen, P. Vicol, J. Ba, D. Tran, and R. Grosse. Flipout: Efficient Pseudo-Independent Weight Perturbations on Mini-Batches. *arXiv preprint arXiv:1803.04386*, 2018. Cited on pages 3, 15.

[71] Z. H. Wong, L. Yue, and Q. Yao. Ensemble Learning for Graph Neural Networks. *arXiv preprint arXiv:2310.14166*, 2023. Cited on page 3.

[72] M. Wortsman, G. Ilharco, S. Y. Gadre, R. Roelofs, R. Gontijo-Lopes, A. S. Morcos, H. Namkoong, A. Farhadi, Y. Carmon, S. Kornblith, and L. Schmidt. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, 2022. Cited on pages 7, 16.

[73] X. Wu, A. Ajorlou, Z. Wu, and A. Jadbabaie. Demystifying Oversmoothing in Attention-Based Graph Neural Networks. In *Advances in Neural Information Processing Systems*, 2023. Cited on page 9.

[74] Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing, and V. Pande. MoleculeNet: a benchmark for molecular machine learning. *Chemical Science*, 9, 2018. Cited on pages 4, 5.

[75] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How Powerful are Graph Neural Networks? In *International Conference on Learning Representations*, 2019. Cited on pages 1, 2.

[76] Z. Yang, W. W. Cohen, and R. Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on International Conference on Machine Learning*, 2016. Cited on pages 4, 21.

[77] Q. Zhu, W. Li, H. Kim, Y. Xiang, K. Wardega, Z. Wang, Y. Wang, H. Liang, C. Huang, J. Fan, and H. Choi. Know the unknowns: addressing disturbances and uncertainties in autonomous systems. In *International Conference on Computer-Aided Design*, 2020. Cited on page 1.

A Extra Details on the Experiments Conducted

Like in the main text each experiment is repeated ten times to obtain the required statistics. Furthermore, for all plots in these sections, the marker indicates the mean improvement. The thick whisker denotes $\pm\sigma$, and the thin semi-transparent lines represent the minimum and maximum values.

The selected hyperparameters used throughout the experiments in the main text are displayed in Table 2. For the GAT models, the number of attention heads used was four. For Tolokers2, Artnetviews and Chameleon, besides the GNN we used a multi-layer perceptron (MLP) module to pre-process the input. We also apply an MLP in between the GNN layers and after the output of the final GNN layer. The MLP used has depth two. The models for QM9 and MolHIV also include a pooling mechanism to create the graph level representations. This mechanism concatenates the representations obtained from the basic pooling mechanisms: mean, maximum and standard deviation. This representation is then used as input for a two layer MLP with 256 hidden dimension that will produce the final graph level representation. Additionally, MolHIV uses a step LR scheduler with step size 10 and $\gamma = 0.707$, and the standard OGB scaffold split.

Table 2: Hyperparameter configurations used the used datasets.

Dataset	Model	Hidden Dim.	Layers	Learning Rate	Weight Decay	Epochs	Dropout
Tolokers2	GATv2	256	3	0.0003	0.0000	1000	0.1
Citeseer	GCN	20	2	0.0100	0.0005	200	0.0
MolHIV	GIN	64	2	0.0010	6e-6	60	0.525
Cora (Appendix E)	GCN	20	2	0.0100	0.0005	500	0.0
Artnetviews	GCN	512	3	0.0003	0.0000	1000	0.1
Chameleon	GATv2	512	3	0.0003	0.0000	250	0.3
QM9	GCN	128	4	0.0010	0.0000	500	0.1
PEMS	GCN	100	3	0.0100	0.0005	500	0.0

The parameters for Section 3, GCN and GCN DE follow Table 2. The Bayesian GCN was trained using SVI [6, 26, 44, 45, 70] with a Normal surrogate distribution. The prior was initialized with standard Glorot initialization [24] scaled by 177.7. The hidden dimension was reduced to 50 to speed convergence. The Matérn GP followed Borovitskiy [7][5].

B Dataset Shift

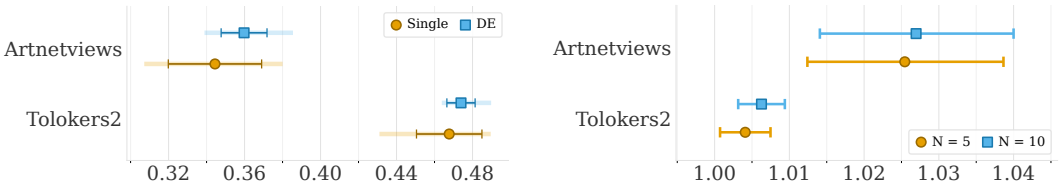
Besides general uncertainty quantification in in-distribution (ID) settings, uncertainty quantification is popular in out-of-distribution (OOD) settings [56]. Like in ID settings it is important that a model can signal if it is uncertain about the prediction it makes. However, in OOD settings, the model is acting on data that is likely no longer supported by the training distribution. Thus evaluating uncertainty quantification in OOD scenarios is also very important since we expect cases where uncertainty increases to appear more often than in ID. To extend our findings to OOD we selected two datasets, one for classification (Tolokers2) and one for regression (Artnetviews), and split them into OOD and ID. Since all results held across datasets for ID, we selected the two largest node datasets as representatives of the others.

To evaluate model robustness under distribution shift, we adopt the density-based split proposed by Bazhenov et al. [4]. This method partitions data based on local graph topology rather than feature distribution. However, it still introduces meaningful partitions in graph datasets, often being able to replicate splits based on features [4].

Importantly, the number of training, validation, and test examples is kept identical to the original fully ID split. We first rank all nodes by their local clustering coefficient in descending order. The first N_{test} examples, where N_{test} is the number of test examples in ID, are assigned for test, followed by N_{val} nodes for validation and N_{train} nodes for training. Consequently, test nodes correspond to those with the highest local clustering coefficients, while training nodes have the lowest. This split based on the connectedness of a node's neighbouhood assigns denser neibhourhoods, the ones that can capture richer information, almost exclusively to the test set.

[5] All experiments were performed using a NVIDIA RTX 5090 or equivalent.

As expected, the performance in point estimation drops between ID and OOD. Specifically, for a five model ensemble for Artnetviews we register a drop in R^2 from 0.599 ± 0.003 to 0.360 ± 0.012 and for Tolokers2 a drop in average precision from 0.617 ± 0.003 to 0.474 ± 0.007[6]. However, like the ID case, deep ensembles provide little gains over a single model for uncertainty quantification. Figure 6 shows that the gains of an ensemble for Tolokers2 are $< 1\%$ and for Artnetviews $< 3\%$—an even worse performance than ID for Artnetviews.



(a) Point Estimation (b) Likelihood Ratio

Figure 6: Comparison of point estimation performance, R^2, for Artnetviews and average precision for Tolokers2, between a DE (5 models) and a single model. For both metrics higher score is better (a). Likelihood ratio $\exp(\text{NLL}_{\text{GNN}} - \text{NLL}_{\text{DE}})$ and 95% CI assuming normally distributed errors, quantifying the relative predictive likelihood improvement of the DE (5 models) over a single model in structural OOD shift (b).

C Model Soup

As noted in Section 5.2, we construct a "model soup" [72] by averaging the weights of the individual ensemble members. If the models resided within a simple convex basin in the weight space, averaging their weights would act as a variance reduction mechanism over a noisy approximation of the global optimum, yielding a point closer to the true global optimum and thus improving performance. However, as shown in Figure 7, Figure 8, the model soup generally underperforms a single model.

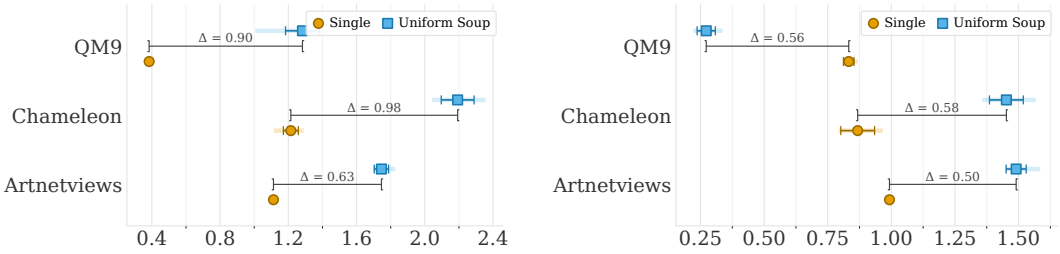
For regression, specifically for point prediction, its performance degrades to essentially match a Trivial baseline. In the only case where the aggregate NLL metric technically improves (`QM9-5%`), the soup remains practically uninformative for uncertainty quantification, as it outputs near-identical uncertainties for all predictions. This degraded behavior confirms that the weight space itself remains highly non-convex—a natural consequence once one considers the permutation symmetries inherent in deep neural networks [2, 17].

Like the results for regression, model soup deteriorates performance for both point prediction and uncertainty[7]. Despite the inherent limitations of accuracy, for example, the insensitivity to confidence and the threshold rigidity even the best model soup cannot reach the worst single model. Furthermore, the gaps in NLL are more pronounced than those observed for accuracy, indicating that model soup performs particularly poorly in terms of uncertainty estimation. In conclusion, even if model soup consistently matched the performance of a single model, this would still be insufficient evidence for weight-space convexity. To support that hypothesis, model soup would need to clearly outperform the single model and at least match ensemble performance [72]. The closer the gap from the model soup to the ensemble performance the stronger the weight-space convexity hypothesis holds (assuming standard optimization procedures without perfect convergence to the global optimum). In our results, model soup fails to match even the single-model baseline. Thus we argue against weight-space convexity.

Therefore, we are left with a compelling hypothesis: the weight space is non-convex, yet epistemic collapse strongly suggests that GNNs experience a form of convexity in the *functional* space. That is, independently trained models find different sets of weights that ultimately implement the same (or similar) predictive function.

[6] We use R^2 and average precision to enable easy comparison with the original work by Bazhenov et al. [5] if necessary.

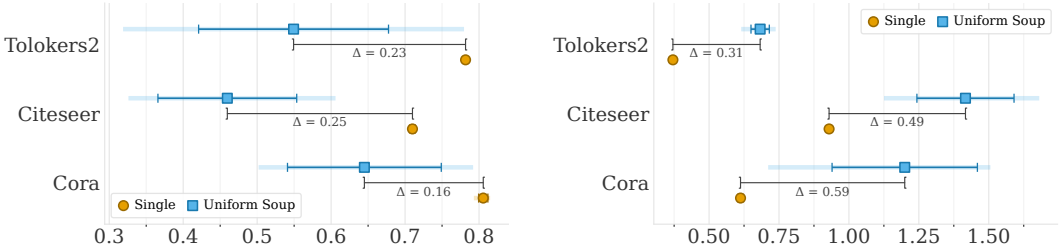
[7] For classification, the model soup uses Cora (Appendix E) in place of `MHIV`.



(a) Model Soup — Point Estimation (RMSE)

(b) Model Soup — NLL

Figure 7: Weight-space convexity analysis. (a) compares the point prediction performance and (b) compares the NLL of a single model against a model soup formed by averaging the weights of the ensemble members. The degraded performance of the model soup confirms that the models reside in different regions of a highly non-convex weight space. Markers represent the mean, thin bars represent $\pm\sigma$ and transparent bars represent min–max.



(a) Model Soup – Point Estimation

(b) Model Soup – NLL

Figure 8: Weight-space convexity analysis. Figure 8a compares the point prediction performance and Figure 8b the NLL of a single model against a "model soup" formed by averaging the weights of the ensemble members. Lower values are better for NLL and higher are better for point prediction. The degraded performance of the model soup confirms that the models reside in different regions of a highly non-convex parameter space.

D Proof of Theorem in Section 6

In this section we prove the main theorem from Section 6 and extend it to models that achieve perfect training performance and for graphs with multiple connected components.

D.1 Setup and assumptions

Let $G = (V, E)$ be a finite, connected, undirected, unweighted graph with $|V| = n$ and combinatorial Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The results derived extend to the other Laplacian, for example, symmetric normalized. We chose the combinatorial for convenience. Let $\varnothing \neq S \subsetneq V$ be the labeled nodes, with fixed labels $\boldsymbol{y}_S$, and let $U = V \setminus S$ be unlabeled (test) nodes. For $z \in \mathbb{R}^n$, the Dirichlet energy is[8]

$$\mathcal{E}(\boldsymbol{z}) = \boldsymbol{z}^\top \mathbf{L} \boldsymbol{z} = \sum_{\{i,j\} \in E} (\boldsymbol{z}_i - \boldsymbol{z}_j)^2 \tag{5}$$

Let $\boldsymbol{f}_w$ be the prediction vector obtained from a randomly initialized (e.g., Glorot) trained model. We assume a distribution over trained weights $w \sim \mu$, $\mathbb{E}_w[\|\boldsymbol{f}_w\|_2^2] < \infty$, and define

$$\bar{\boldsymbol{f}} = \mathop{\mathbb{E}}_w[\boldsymbol{f}_w], \qquad \boldsymbol{e} = \boldsymbol{f}_w - \bar{\boldsymbol{f}}, \qquad V_{\text{epi}} = \sum_{v \in V} \mathrm{Var}_w((\boldsymbol{f}_w)_v) = \mathop{\mathbb{E}}_w[\|\boldsymbol{e}\|_2^2] \tag{6}$$

[8]Contrarily to the main text, Section 6, for visual clarity, we will distinguish between minimum Dirichlet Energy for non-perfect training error, $\mathcal{E}^{\epsilon}_{\text{min}}$, and perfect training error $\mathcal{E}_{\text{min}}$. The main text refers to the former.

Assumption 1 (Expected prediction energy). For some finite ρ,

$$\mathop{\mathbb{E}}_{w}[\mathcal{E}(\boldsymbol{f}_w)] \le \rho \tag{7}$$

Assumption 2 (Bounded squared error). For every $s \in S$, there are $\varepsilon_s \ge 0$ such that

$$\mathop{\mathbb{E}}_{w}[(\boldsymbol{f}_w(s) - \boldsymbol{y}_s)^2] \le \varepsilon_s^2 \tag{8}$$

Because $\boldsymbol{y}_s$ is fixed, we note that the equation above can be interpreted as bounding the prediction variance by ε_s^2 and expected deviation from the prediction to the label by ε_s.

Why prediction energy alone is insufficient. Take any fixed $\boldsymbol{h}$ with $\boldsymbol{h}_S = \boldsymbol{y}_S$ and let $\boldsymbol{f}_w = \boldsymbol{h} + Z\mathbf{1}$, where Z is a scalar random variable with finite but arbitrarily large variance. Since $\mathbf{L}\mathbf{1} = 0$, if we assume $\mathbb{E}[Z] = \mu_Z = 0$, taking $\bar{\boldsymbol{f}} = \mathbb{E}_w[\boldsymbol{f}_w] = h + \mu_Z\mathbf{1}$, yields

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{f}_w - \bar{\boldsymbol{f}} && (9)\\ &= \boldsymbol{h} + Z\mathbf{1} - (\boldsymbol{h} + \mu_Z\mathbf{1}) && (10)\\ &= (Z - \mu_Z)\mathbf{1} && (11) \end{aligned}$$

leading to $V_{\text{epi}} = \mathbb{E}_w[\|\boldsymbol{e}\|_2^2] = n\,\mathrm{Var}(Z)$ while $\mathcal{E}(\boldsymbol{f}_w) = \mathcal{E}(\boldsymbol{h})$ almost surely. Thus an energy bound cannot, by itself, bound epistemic variance. Assumption 2 controls this example because $\mathrm{Var}_w((\boldsymbol{f}_w)_s) = \mathrm{Var}(Z)$ at every labeled node.

D.2 Spectral and minimum-energy lemmas

For $A, B \subseteq V$, let $\mathbf{L}_{AB} = (\mathbf{L}_{ij})_{i\in A, j\in B}$; in particular, $\mathbf{L}_{UU}$ is a principal submatrix of the *full* Laplacian. Its diagonal entry at $u \in U$ is the degree of u in G, including edges from u to S. Equivalently, if $\mathbf{L}_U^{\text{ind}}$ is the Laplacian of the subgraph induced by U and $a_{us} = \mathbf{1}_{\{\{u,s\}\in E\}}$, then

$$\mathbf{L}_{UU} = \mathbf{L}_U^{\text{ind}} + \mathrm{diag}\left(\sum_{s\in S} a_{us}\right)_{u\in U} \tag{12}$$

We refer to this principal submatrix as the grounded Laplacian.

Let $\mathbf{P}_S$ be the diagonal projector onto the coordinates in S, so $\boldsymbol{x}^\top\mathbf{P}_S\boldsymbol{x} = \|\boldsymbol{x}_S\|_2^2$. For $\alpha > 0$, define

$$\mathbf{M}_\alpha = \mathbf{L} + \alpha\mathbf{P}_S, \qquad \lambda_\alpha = \lambda_{\min}(\mathbf{M}_\alpha) \tag{13}$$

Lemma 1 (Anchored and grounded spectral inequalities)**.** *Both $\mathbf{M}_\alpha$ and $\mathbf{L}_{UU}$ are positive definite. Consequently, for every $\boldsymbol{x} \in \mathbb{R}^n$,*

$$\|\boldsymbol{x}\|_2^2 \le \frac{\boldsymbol{x}^\top\mathbf{L}\boldsymbol{x} + \alpha\|\boldsymbol{x}_S\|_2^2}{\lambda_\alpha}. \tag{14}$$

If $\boldsymbol{x}_S = 0$, then also

$$\|\boldsymbol{x}\|_2^2 \le \frac{\boldsymbol{x}^\top\mathbf{L}\boldsymbol{x}}{\lambda_{\min}(\mathbf{L}_{UU})} \tag{15}$$

Proof. 1st Inequality: Both terms in $\boldsymbol{x}^\top\mathbf{M}_\alpha\boldsymbol{x} = \boldsymbol{x}^\top\mathbf{L}\boldsymbol{x} + \alpha\|\boldsymbol{x}_S\|_2^2$ are nonnegative. If their sum is zero, connectedness implies $\boldsymbol{x} \in \ker\mathbf{L} = \mathrm{span}\{\mathbf{1}\}$; the nonempty set S then forces $\boldsymbol{x} = 0$. Thus $\mathbf{M}_\alpha \succ 0$, and its Rayleigh quotient gives the first Poincaré inequality [12].

2nd Inequality: If $\boldsymbol{z} \in \mathbb{R}^{|U|}$, extend it to $\boldsymbol{x} \in \mathbb{R}^n$ by setting $\boldsymbol{x}_S = 0$ and $\boldsymbol{x}_U = \boldsymbol{z}$. Then $\boldsymbol{z}^\top\mathbf{L}_{UU}\boldsymbol{z} = \boldsymbol{x}^\top\mathbf{L}\boldsymbol{x}$. This is zero only if $\boldsymbol{x}$ is constant, hence, because $\boldsymbol{x}_S = 0$, only if $\boldsymbol{z} = 0$. Therefore $\mathbf{L}_{UU} \succ 0$. For any $\boldsymbol{x}$ with $\boldsymbol{x}_S = 0$, we have $\boldsymbol{x}^\top\mathbf{L}\boldsymbol{x} = \boldsymbol{x}_U^\top\mathbf{L}_{UU}\boldsymbol{x}_U$ and $\|\boldsymbol{x}\|_2 = \|\boldsymbol{x}_U\|_2$; the second Poincaré inequality [12] follows from the Rayleigh quotient of $\mathbf{L}_{UU}$. □

For fixed labeled values $\boldsymbol{b} \in \mathbb{R}^{|S|}$, define

$$\mathcal{E}_{\min}(\boldsymbol{b}) = \min\{\boldsymbol{z}^\top \mathbf{L}\boldsymbol{z} : \boldsymbol{z}_S = \boldsymbol{b}\} \tag{16}$$

For the interpolation-error intervals, define

$$\mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S) = \min\left\{\boldsymbol{z}^\top \mathbf{L}\boldsymbol{z} : |\boldsymbol{z}_s - \boldsymbol{y}_s| \le \varepsilon_s \text{ for every } s \in S\right\}. \tag{17}$$

Lemma 2 (Minimum energy with prescribed labeled values)**.** *Let*

$$\mathbf{R} = \mathbf{L}_{SS} - \mathbf{L}_{SU}\mathbf{L}_{UU}^{-1}\mathbf{L}_{US} \tag{18}$$

Then

$$\mathcal{E}_{\min}(\boldsymbol{b}) = \boldsymbol{b}^\top \mathbf{R}\boldsymbol{b}, \qquad \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S) = \min_{\substack{\boldsymbol{b}\in\mathbb{R}^{|S|} \\ |\boldsymbol{b}_s - \boldsymbol{y}_s| \le \varepsilon_s \ \forall s \in S}} \boldsymbol{b}^\top \mathbf{R}\boldsymbol{b} \tag{19}$$

For fixed $\boldsymbol{b}$, the minimum is attained by the harmonic extension

$$\boldsymbol{z}_S = \boldsymbol{b}, \qquad \boldsymbol{z}_U = -\mathbf{L}_{UU}^{-1}\mathbf{L}_{US}\boldsymbol{b} \tag{20}$$

Proof. By Lemma 1, $\mathbf{L}_{UU}$ is invertible. Completing the square, with $\boldsymbol{z}_S = \boldsymbol{b}$, gives

$$\boldsymbol{z}^\top \mathbf{L}\boldsymbol{z} = \left(\boldsymbol{z}_U + \mathbf{L}_{UU}^{-1}\mathbf{L}_{US}\boldsymbol{b}\right)^\top \mathbf{L}_{UU}\left(\boldsymbol{z}_U + \mathbf{L}_{UU}^{-1}\mathbf{L}_{US}\boldsymbol{b}\right) + \boldsymbol{b}^\top \mathbf{R}\boldsymbol{b} \tag{21}$$

Minimizing first over $\boldsymbol{z}_U$ proves the formula for $\mathcal{E}_{\min}(\boldsymbol{b})$. Minimizing subsequently over the compact box of admissible $\boldsymbol{b}$ proves the second formula and attainment of the minimum. □

Lemma 3 (Centered-energy identity)**.** *The centered predictions satisfy*

$$\mathop{\mathbb{E}}_{w}[\boldsymbol{e}^\top \mathbf{L}\boldsymbol{e}] = \mathop{\mathbb{E}}_{w}[\boldsymbol{f}_w^\top \mathbf{L}\boldsymbol{f}_w] - \bar{\boldsymbol{f}}^\top \mathbf{L}\bar{\boldsymbol{f}}. \tag{22}$$

Under Assumptions 1 and 2, it follows that

$$\mathop{\mathbb{E}}_{w}[\boldsymbol{e}^\top \mathbf{L}\boldsymbol{e}] \le \rho - \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S) \tag{23}$$

Proof. Expand $\boldsymbol{f}_w = \bar{\boldsymbol{f}} + \boldsymbol{e}$ and use $\mathbb{E}_w[\boldsymbol{e}] = 0$ to obtain the identity. Assumption 2 places $\bar{\boldsymbol{f}}_S$ in the prescribed bounded squared-error interval (e.g., by Jensen's inequality $|\bar{\boldsymbol{f}}_s - \boldsymbol{y}_s|^2 \le \mathbb{E}_w(\boldsymbol{f}_w(s) - \boldsymbol{y}_s)^2 \le \varepsilon_s^2$). Hence $\bar{\boldsymbol{f}}^\top \mathbf{L}\bar{\boldsymbol{f}} \ge \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S)$, while Assumption 1 bounds the first term in the identity by ρ. □

D.3 Variance bounds on a connected graph

Theorem 1 (Soft-Pinning Epistemic Variance Bound)**.** *Under Assumptions 1 and 2,*

$$\boxed{V_{epi} \le \inf_{\alpha>0} \frac{\rho - \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S) + \alpha \sum_{s\in S} \varepsilon_s^2}{\lambda_\alpha}} \tag{24}$$

Proof. Apply the first inequality of Lemma 1 to $\boldsymbol{e}$ and take expectations:

$$V_{\text{epi}} \le \frac{\mathbb{E}_w[\boldsymbol{e}^\top \mathbf{L}\boldsymbol{e}] + \alpha\, \mathbb{E}_w[\|\boldsymbol{e}_S\|_2^2]}{\lambda_\alpha}. \tag{25}$$

By Lemma 3, the first term in the numerator is at most $\rho - \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S)$. By Assumption 2,

$$\mathop{\mathbb{E}}_{w}[\|\boldsymbol{e}_S\|_2^2] = \sum_{s\in S} \mathrm{Var}_w((\boldsymbol{f}_w)_s) \le \sum_{s\in S} \varepsilon_s^2 \tag{26}$$

The resulting bound holds for every $\alpha > 0$; take the infimum. □

Corollary 1 (Hard-Pinning Variance Bound)**.** *Suppose $\boldsymbol{e}_S = 0$ almost surely, so all models have the same labeled predictions $\boldsymbol{b}_S = \bar{\boldsymbol{f}}_S$, not necessarily equal to $\boldsymbol{y}_S$. Under Assumption 1,*

$$\boxed{V_{epi} \le \frac{\rho - \mathcal{E}_{\min}(\boldsymbol{b}_S)}{\lambda_{\min}(\mathbf{L}_{UU})}} \tag{27}$$

In particular, exact interpolation corresponds to $\boldsymbol{b}_S = \boldsymbol{y}_S$.

Proof. Since $\boldsymbol{e}_S = 0$ almost surely, the grounded inequality in Lemma 1 gives

$$V_{\text{epi}} \le \frac{\mathbb{E}_w[\boldsymbol{e}^\top \mathbf{L}\boldsymbol{e}]}{\lambda_{\min}(\mathbf{L}_{UU})} \tag{28}$$

The centered-energy identity holds independently of Assumption 2. As $\bar{\boldsymbol{f}}_S = \boldsymbol{b}_S$, we have $\bar{\boldsymbol{f}}^\top \mathbf{L}\bar{\boldsymbol{f}} \ge \mathcal{E}_{\min}(\boldsymbol{b}_S)$, and Assumption 1 therefore gives $\mathbb{E}_w[\boldsymbol{e}^\top \mathbf{L}\boldsymbol{e}] \le \rho - \mathcal{E}_{\min}(\boldsymbol{b}_S)$. □

D.4 Extension to disconnected graphs

Now let G have connected components $C_1, \dots, C_K$, and set $S_k = S \cap C_k$ and $U_k = U \cap C_k$. A component with no labeled node has an uncontrolled constant mode, so an additional condition is necessary.

Assumption 3 (A label in every component). $S_k \neq \varnothing$ for every k.

Let $\mathbf{L}^{(k)}$ be the Laplacian of the component C_k, and let P_{S_k} be its labeled-coordinate projector. Define

$$\mathbf{M}_{\alpha,k} = \mathbf{L}^{(k)} + \alpha \mathbf{P}_{S_k}, \qquad \lambda_{\alpha,k} = \lambda_{\min}(\mathbf{M}_{\alpha,k}) \tag{29}$$

and define $\mathcal{E}^{\epsilon}_{\min,k}((\boldsymbol{y}_S)_k)$ using the edges and labeled nodes of C_k. Choose any constants ρ_k satisfying

$$\mathop{\mathbb{E}}_{w}\big[(\boldsymbol{f}_w)_{C_k}^\top \mathbf{L}^{(k)} (\boldsymbol{f}_w)_{C_k}\big] \le \rho_k. \tag{30}$$

For example, each ρ_k may be taken equal to the expectation on the left; their sum is then at most ρ under Assumption 1.

Corollary 2 (Disconnected-Graph Soft-Pinning Variance Bound)**.** *Under Assumptions 1–3,*

$$\boxed{V_{epi} \le \sum_{k=1}^{K} \inf_{\alpha_k > 0} \frac{\rho_k - \mathcal{E}^{\epsilon}_{\min,k}((\boldsymbol{y}_S)_k) + \alpha_k \sum_{s\in S_k} \varepsilon_s^2}{\lambda_{\alpha_k,k}}} \tag{31}$$

The formula also covers components with $U_k = \varnothing$.

Proof. Each C_k is connected and contains a labeled node. Thus the positive-definiteness, centered-energy, and soft-pinning arguments apply to that component, yielding the k-th summand as a bound on $\mathbb{E}_w[\|\boldsymbol{e}_{C_k}\|_2^2]$. Sum these component-wise bounds and use $V_{\text{epi}} = \sum_k \mathbb{E}_w[\|\boldsymbol{e}_{C_k}\|_2^2]$. The penalty strengths may be optimized separately because the component-wise inequalities are independent. □

In particular, a component with few labeled nodes may have fewer "anchors", meaning lower minimum energy and a looser bound than components with a large amount of train nodes. Similarly, a component with low expansion coefficient or when unlabeled nodes are poorly connected to the training nodes, for instance when they are far from them or reachable only through bottlenecks, lead to a looser bound due to smaller $\lambda_{\alpha,k}$.

D.5 Interpretation of the expected-energy assumption

The quantity ρ is an upper bound on Dirichlet energy, or equivalently, on the *expected squared differences between predictions at adjacent nodes*:

$$\mathop{\mathbb{E}}_{w}[\mathcal{E}(\boldsymbol{f}_w)] = \sum_{\{i,j\}\in E} \mathop{\mathbb{E}}_{w}\left[\left((\boldsymbol{f}_w)_i - (\boldsymbol{f}_w)_j\right)^2\right] \tag{32}$$

Representation-collapse results such as those discussed by Ma et al. [49] can motivate expecting small prediction differences on some same-class edges, but feature-space closeness or a high-probability collapse statement does not alone provide an upperbound on ρ. Such a deduction would also require control of the readout from representations to predictions and of the squared differences on the remaining edges. We select an upperbound in Proposition 1.

Good heterophily can support accurate message passing without graph homophily [46, 48, 49]. By Lemma 3, however, epistemic variance is controlled not by ρ itself but by the fluctuation energy

$$\mathop{\mathbb{E}}_{w}[\boldsymbol{e}^\top \mathbf{L}\boldsymbol{e}] = \mathop{\mathbb{E}}_{w}[\mathcal{E}(\boldsymbol{f}_w)] - \bar{\boldsymbol{f}}^\top \mathbf{L}\bar{\boldsymbol{f}}, \tag{33}$$

which our bounds estimate from above by the spare energy $\rho - \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S)$. If label changes across edges are concentrated among unlabeled nodes (e.g., heterophily between test nodes but not between training nodes), a model that follows this structure may require a larger ρ, while $\mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_S)$, which depends only on the training labels, is unchanged. The spare energy, and hence the bound, then grows.

Two caveats limit what this implies. First, a looser upper bound permits, but does not force, larger epistemic variance. Second, the bound lower-estimates $\bar{\boldsymbol{f}}^\top \mathbf{L}\bar{\boldsymbol{f}}$ using the training labels only. If the ensemble mean is also accurate on unlabeled nodes, the same argument yields the spare energy $\rho - \mathcal{E}^{\epsilon}_{\min}(\boldsymbol{y}_V)$, and heterophily may raise both terms. Room for disagreement therefore arises only insofar as individual members are rougher than their mean, something that should be observed in practice for heterophilic graphs. That is, disagreement arises when ensemble members fit high-energy structure inconsistently rather than all capturing it.

Consequently, our framework does not predict that heterophily by itself causes ensemble diversity. It predicts that the constraint behind epistemic collapse binds most tightly when the task requires low-energy predictions, as on strongly homophilous graphs, and can be relaxed when the task requires high-energy predictions that message passing represents inconsistently across initializations. We regard the latter as a plausible mechanism for the weaker collapse observed on CHAM, not as a direct consequence of the theorem.

E Other Plots and Results

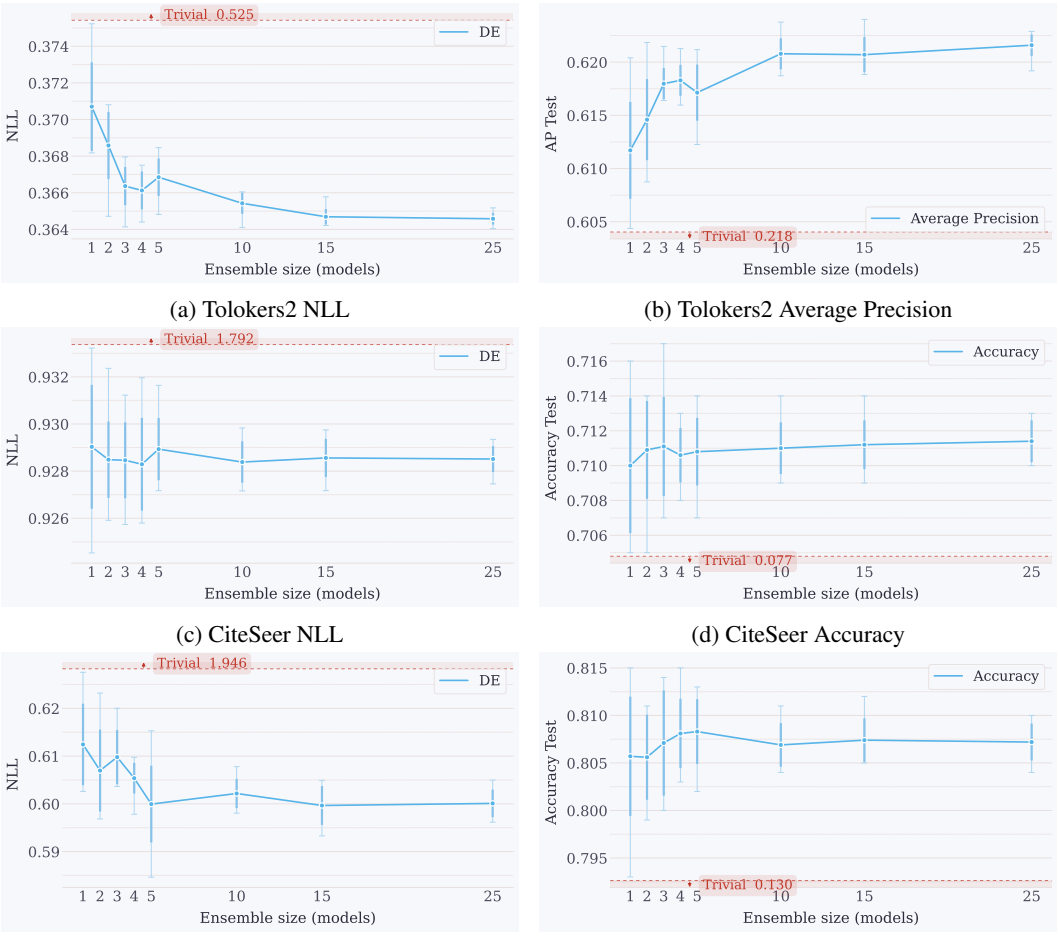
More Datasets: Cora. In addition to the datasets presented in the main text, we also tested the classic Cora (CORA) dataset [76]. We followed the same reasoning as CiteSeer to pick the architecture. Thus, we picked GCN for the better training stability it offers compared to GAT, despite the slightly worse point estimation performance. Epistemic/aleatoric ratio is comparable to the ones shown in Figure 4b; the same applies for Jensen's gap. The likelihood ratios are slightly above the ones from Figure 2a with mean around 1% and confidence interval between 0.05% and 2%.

Dynamics. Figure 10 and Figure 9 show the dynamics of the point estimation metric and NLL for an increasing ensemble size. For Artnetviews and Tolokers2 we revert back to the original metrics used in Bazhenov et al. [5]. This should allow one to more easily compare results if necessary.

Table 3: Absolute performance metrics for a DE of five models on Cora. Values represent NLL, ECE, and PEM (accuracy for classification), showing the mean and standard deviation over 10 independent runs. The ECE of the trivial baseline is not taken into account when highlighting the best score. Trivial baseline has low ECE by construction but is uninformative: it predicts the same regardless of input.

Dataset	Metric	Trivial	GNN	DE
CORA	NLL	1.946 ± 0.000	0.612 ± 0.009	$\mathbf{0.600 \pm 0.008}$
	ECE	0.116 ± 0.000	0.123 ± 0.001	$\mathbf{0.122 \pm 0.002}$
	PEM	0.130 ± 0.000	0.806 ± 0.006	$\mathbf{0.808 \pm 0.003}$

Generally, the dynamics hold when changing metrics. An added bonus is that average precision, compared with accuracy, makes the dynamics easier to see.



(a) Tolokers2 NLL

(b) Tolokers2 Average Precision

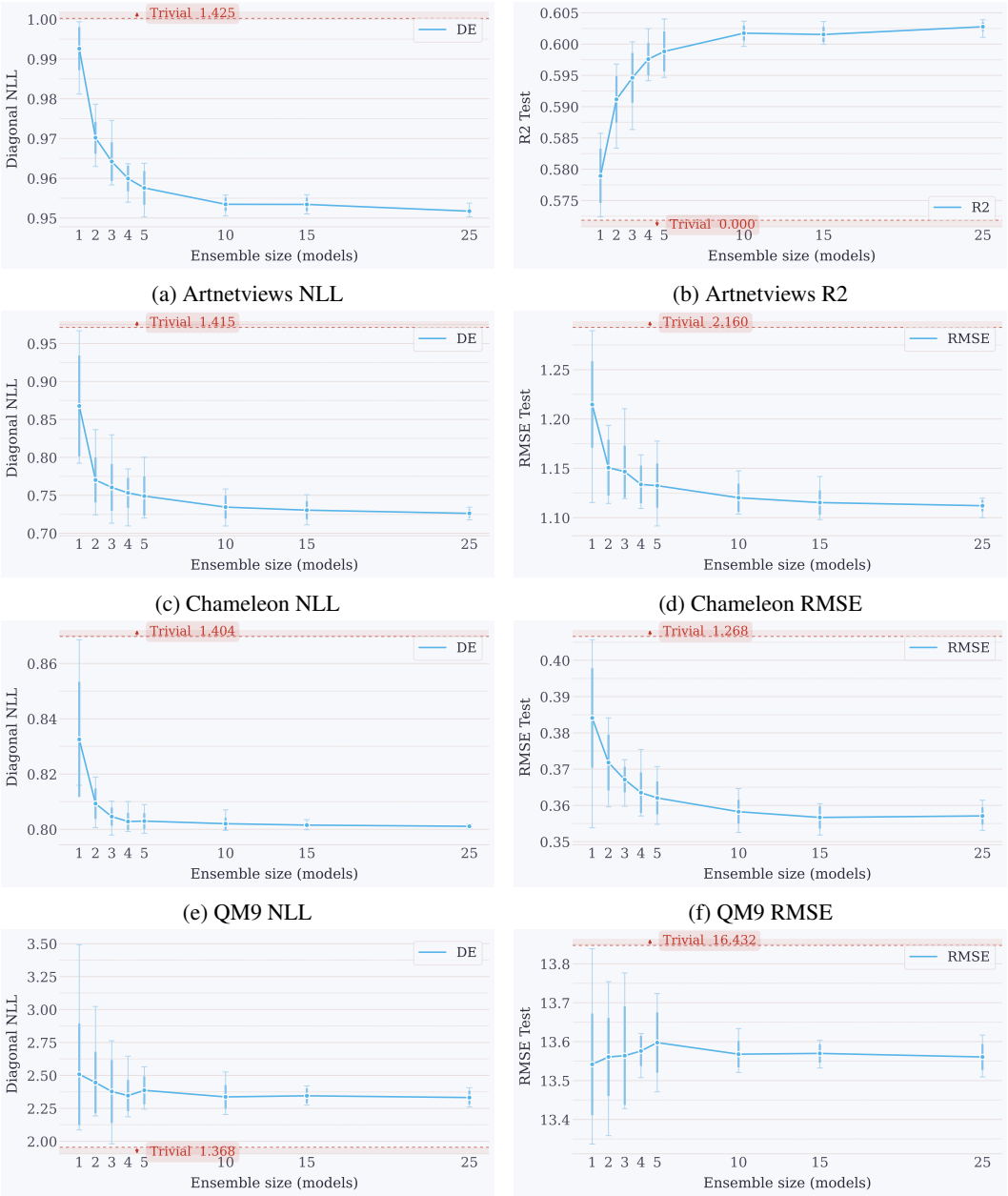
(c) CiteSeer NLL

(d) CiteSeer Accuracy

(e) Cora NLL

(f) Cora Accuracy

Figure 9: Evolution of NLL (a, c and e) and point estimation metric (b, d and f) as the number of base models in the ensemble increases. Red dashed line denotes a trivial baseline.



(a) Artnetviews NLL

(b) Artnetviews R2

(c) Chameleon NLL

(d) Chameleon RMSE

(e) QM9 NLL

(f) QM9 RMSE

(g) PEMS NLL

(h) PEMS RMSE

Figure 10: Evolution of NLL (a, c, e and g) and point estimation metric (b, d, f and h) as the number of base models in the ensemble increases. Red dashed line denotes a trivial baseline.