GraphUAT: Uncertainty Attribution in Graph Neural Networks

Burouj Armgaan[1] **Karan Raj Singh**[2] **Abhirup Mondal**[3] **Sayan Ranu**[1,2]
[1]CSE, IIT Delhi [2]Yardi ScAI, IIT Delhi [3]Amazon, India
{csz228001, aiz247611, sayanranu}@iitd.ac.in mabhirup@amazon.com

Abstract

Models that recognize when their predictions are unreliable inspire greater trust than those that make confident but incorrect decisions. While recent advances enable uncertainty quantification in GNNs, understanding *why* a model is uncertain, whether due to noisy features, conflicting neighborhoods, or sparse training coverage, remains unsolved. We address this gap with GraphUAT, the first *uncertainty attribution* framework for GNNs, which identifies the sources of uncertainty rather than aiming to improve its quantification. GraphUAT employs a teacher-student paradigm, where the trained GNN's behavior is distilled into an uncertainty-aware student that disentangles aleatoric and epistemic uncertainties. We then use targeted probes to trace the student's uncertainty back to the responsible features and structural components. Experiments across benchmarks show that removing the identified elements consistently reduces the teacher's uncertainty, validating the faithfulness of our attributions.

1 Introduction and Related Works

Graph Neural Networks (GNNs) are the method of choice for learning from graph-structured data. They power applications in drug discovery [54], fraud detection [36], traffic forecasting [23], and recommendation systems [18], where relationships between entities are as important as the entities themselves. However, strong performance alone isn't enough. A well-performing, yet overconfident, model can be dangerous in high-stakes settings, leading to confident wrong decisions. In contrast, a model that knows when it does not know is inherently more trustworthy. This ability is captured by *uncertainty quantification (UQ)*, which estimates how confident a model is about its predictions [50, 12], allowing practitioners to decide when to trust model outputs and when to treat them with caution.

1.1 Related Works and Existing Gaps

Prior works on GNN uncertainty quantification have focused on building models that produce calibrated uncertainty estimates, e.g., Bayesian GNNs, ensemble methods, and techniques designed to improve robustness under distribution shift [58, 22, 53, 9, 31, 46, 13, 26, 57, 41, 38, 4, 51, 28, 29, 50, 12]. These approaches either integrate uncertainty estimation into the training process, enhance model robustness to out-of-distribution (OOD) data, or return predictive uncertainty alongside predictions (See App. D for more details).

Critically, these approaches do not provide insight into *why* the model is uncertain. This distinction is central to our work: we *do not* propose a new UQ method, nor do we aim to improve calibration or robustness. Instead, we ask a different question: ***where in the input does this uncertainty originate?*** Our work proposes a *post-hoc* technique that can be applied to any trained GNN to attribute its uncertainty to parts of the input graph. We focus on a post-hoc approach since most widely used GNN architectures (e.g., GCN [32], GAT [48], GRAPHSAGE [21]) are not designed to quantify uncertainty, and migrating them to uncertainty-aware counter-parts may be prohibitively disruptive in downstream dependencies. By working post-hoc, our method attributes uncertainty without imposing constraints on the underlying model or requiring access to its training process.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

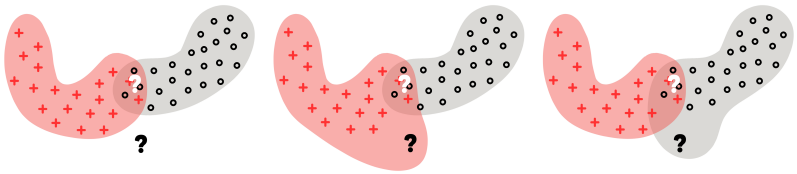


Figure 1: [27] The white query point represents aleatoric uncertainty. The prediction is ambiguous because of overlapping class distributions; adding more data won't change that. In contrast, epistemic uncertainty occurs in areas with insufficient training data or for out-of-distribution points, such as the black query. With more data, this could be classified as red (middle panel) or black (right panel).

Uncertainty attribution (UA) in GNNs presents unique challenges stemming from the interplay of feature and topology:

- **Entangled sources of uncertainty:** In contrast to images or text, a node's uncertainty in a graph stems from two intertwined sources: its own ambiguous features (*feature-level uncertainty*), and the influence of its neighbors through message passing (*topology-level uncertainty*). Since information propagates across connected nodes, uncertainty can diffuse through the graph, being amplified or attenuated in the process, making it difficult to isolate its true origin.
- **Post-hoc generalization:** Commonly used architectures such as GCN, GIN, and GAT are designed for deterministic prediction and provide no native mechanism for uncertainty estimation, let alone disentangled attribution. A practical solution must therefore operate *post-hoc*, augmenting a pretrained GNN with uncertainty interpretation without requiring retraining.

We address these challenges with the following contributions:

- **Novel problem formulation:** We introduce *uncertainty attribution* (UA) for GNNs, shifting the focus from quantifying how uncertain a model is to explaining *why* it is uncertain, thereby complementing rather than competing with uncertainty quantification methods.
- **Algorithmic innovation:** We propose **GraphUAT**, a post-hoc, model-agnostic framework that decomposes predictive uncertainty into aleatoric (inherent, data-driven) and epistemic (lack of evidence, model-driven) components. It further disentangles whether uncertainty originates from a node's own features or from its neighborhood, providing fine-grained, actionable explanations.
- **Evaluation framework:** We establish the first empirical protocol for assessing UA in GNNs that verify whether identified features and/or neighborhoods indeed drive model uncertainty. Experiments across benchmarks and a case study demonstrate that GraphUAT produces faithful and practically meaningful uncertainty attributions.

2 Preliminaries and Problem Formulation

Node classification: We focus on node classification in the main paper and later demonstrate that our framework naturally extends to graph classification (see App. J). Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{X})$ denote a graph, where $\mathcal{V}$ is the set of nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, and $\boldsymbol{X} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the node feature matrix containing d-dimensional node features $\mathbf{x}_v$, $\forall v \in \mathcal{V}$. Each node $v_i \in \mathcal{V}_{tr} \subset \mathcal{V}$ in the training subset has a ground-truth class $y_i \in \{\mathcal{Y}_1, \ldots, \mathcal{Y}_C\}$ from C possible classes. The goal is to learn a function $\Phi : (\mathcal{G}, v) \mapsto \mathcal{Y}$ that predicts a given node's class.

2.1 Evidential Deep Learning

Evidential Deep Learning (EDL) [47, 42, 39, 11] is a learning paradigm that unifies prediction and uncertainty quantification by modeling network outputs as higher-order distributions over class probabilities or regression targets. Rather than outputting simple point estimates, as in conventional softmax classifiers or regression models, EDL parameterizes distributions such as Dirichlet or Normal-Inverse-Gamma, whose parameters are interpreted as *evidence* in the sense of subjective logic and belief theory [42]. This structure allows a model to express both its support for particular outcomes and the associated uncertainty, including epistemic and aleatoric components, in a single forward pass, without the need for ensembling or Bayesian weight sampling.

Dirichlet Parameterization: For classification, EDL models the predictive probabilities as draws from a Dirichlet distribution over the probability simplex, exploiting the fact that the Dirichlet is the conjugate prior of the categorical distribution. The C-class categorical probability vector $\boldsymbol{p} = (p_1, \ldots, p_C)$ is treated as a random variable following:

$$\mathrm{Dir}(\boldsymbol{p} \mid \boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{c=1}^{C} p_c^{\alpha_c - 1}, \qquad B(\boldsymbol{\alpha}) = \prod_{c=1}^{C} \Gamma(\alpha_c) \Big/ \Gamma\left(\sum_{c=1}^{C} \alpha_c\right),$$

where Γ is the Gamma function, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_C)$ with $\alpha_c > 0$ are the concentration parameters. [1] Each α_c encodes the degree of belief that the input belongs to class k. In the absence of any prior knowledge, all α_c are initialized to one, yielding a uniform distribution over the simplex. As observations arrive, the prior is updated as $\alpha_c \leftarrow \alpha_c + \beta_c$, where β_c denotes the number of observations for class c; larger values of α_c therefore indicate stronger evidential support for class c.

Evidence Prediction via EDL: In classical Bayesian updating, evidence for class c accumulates by repeatedly observing data points labeled as such (think of ensembles, or training multiple models). This is computationally expensive and non-deterministic, making it impractical for standard deep learning pipelines. EDL resolves this by training the network to *predict* the evidence vector $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_C)$ directly from the input x. Denoting $\boldsymbol{\alpha}(x)$ as the concentration parameters predicted for input x, we get $\alpha_c(x) = 1 + \beta_c(x)$ so that the $\beta_c(x)$ act as pseudo-counts rather than real observation counts. Formally, when an input x is similar to training data from class-c, the network assigns large $\beta_c(x)$, inflating $\alpha_c(x)$ and concentrating the Dirichlet mass near $p_c(x) = 1$. Conversely, for an OOD input that resembles no training class, all $\beta_c(x)$ remain near zero, leaving $\boldsymbol{\alpha}(x) \approx \mathbf{1}$ and the Dirichlet distribution nearly uniform. This allows EDL to achieve deterministic, single-pass uncertainty quantification at a fraction of the Bayesian computational cost.

Uncertainty Decomposition: A key advantage of the Dirichlet parameterization is its natural quantification and decomposition of predictive uncertainty into two distinct components: aleatoric and epistemic (Fig. 1 illustrates both uncertainty types). For an input x, The predictive probability p_c for class c is given by the Dirichlet mean $\alpha_c(x)/\alpha_0(x)$, where $\alpha_0(x) = \sum_{c=1}^{C} \alpha_c(x)$ is the total concentration of the distribution. While this mean determines the predicted class probabilities, the full set of Dirichlet parameters encodes both the distribution of evidence across classes and the overall confidence of the model. Defining $\mathbf{y} \sim \text{Cat}(\mathbf{p})$ as the one-hot encoded prediction with $\mathbf{p} \sim \text{Dir}(\boldsymbol{\alpha}(x))$, and H, I as the Shannon entropy and mutual information, respectively, we can decompose the total predictive uncertainty as [39, 47]:

$$\underbrace{H[\mathbb{E}_{\mathbf{p}}[\mathbf{y} \mid \mathbf{p}]]}_{\text{total}} = \underbrace{\mathbb{E}_{\mathbf{p}}[H[\mathbf{y} \mid \mathbf{p}]]}_{\text{aleatoric}} + \underbrace{I[\mathbf{y}, \mathbf{p}]}_{\text{epistemic}}$$

- **Aleatoric Uncertainty (AU)** reflects situations where the outcome is inherently ambiguous and cannot be resolved even with unlimited data or a perfect model. A data point lying at the boundary between two classes is aleatorically uncertain: regardless of how well the model is trained, it can only assign comparable probabilities to each class without ever providing a definitive prediction. Defining the digamma function $\psi(x) = \frac{d}{dx} \log \Gamma(x)$, AU is defined as:

$$u_a(x) = \mathbb{E}_{\mathbf{p}}[H[\mathbf{y} \mid \mathbf{p}]] = -\sum_{c=1}^{C} \frac{\alpha_c(x)}{\alpha_0(x)} \left(\psi(\alpha_c(x) + 1) - \psi(\alpha_0(x) + 1)\right) \tag{1}$$

- **Epistemic Uncertainty (EU)** reflects uncertainty arising from a lack of knowledge about the true model parameters, such as out-of-distribution (OOD) inputs or regions of the feature space with sparse training coverage. EU can be calculated as the difference between the total uncertainty and aleatoric uncertainty:

$$u_e(x) = -\sum_{c=1}^{C} \frac{\alpha_c(x)}{\alpha_0(x)} \left(\log \frac{\alpha_c(x)}{\alpha_0(x)} - \psi(\alpha_c(x) + 1) + \psi(\alpha_0(x) + 1)\right) \tag{2}$$

2.2 Proposed Problem: Uncertainty Attribution

Problem 1 (Uncertainty Attribution in GNNs)**.** *We seek attribution functions*

$$\mathcal{A}_a^{ft}, \mathcal{A}_e^{ft} : (\mathcal{G}, v, \Phi) \longrightarrow \mathbb{R}^d, \quad \mathcal{A}_a^{net}, \mathcal{A}_e^{net} : (\mathcal{G}, v, \Phi) \longrightarrow \mathbb{R}^{|\mathcal{V}|} \tag{3}$$

such that, for $* \in \{a, e\}$*, the feature attribution vector* $\mathcal{A}_*^{\text{ft}}(v) = [a_{1,*}^{\text{ft}}(v), \ldots, a_{d,*}^{\text{ft}}(v)]$ *and the node attribution vector* $\mathcal{A}_*^{\text{net}}(v) = [a_{1,*}^{\text{net}}(v), \ldots, a_{|\mathcal{V}|,*}^{\text{net}}(v)]$ *measure the marginal contribution of each feature and node to the uncertainty, respectively. Concretely,* $a_{i,*}^{\text{ft}}(v) = u_*(v) - u_*^{\setminus i}(v)$ *captures the change in uncertainty upon perturbing feature* i *of* $\mathbf{x}_v$ *to an in-distribution value, and* $a_{n,*}^{\text{net}}(v) = u_*(v) - u_*^{\setminus n}(v)$ *captures the change after removing node* $n \in \mathcal{V}$*. The subscripts* $\{a, e\}$ *represent aleatoric and epistermic uncertainty respectively.*

[1] Table 8 in the Appendix summarizes all notations.

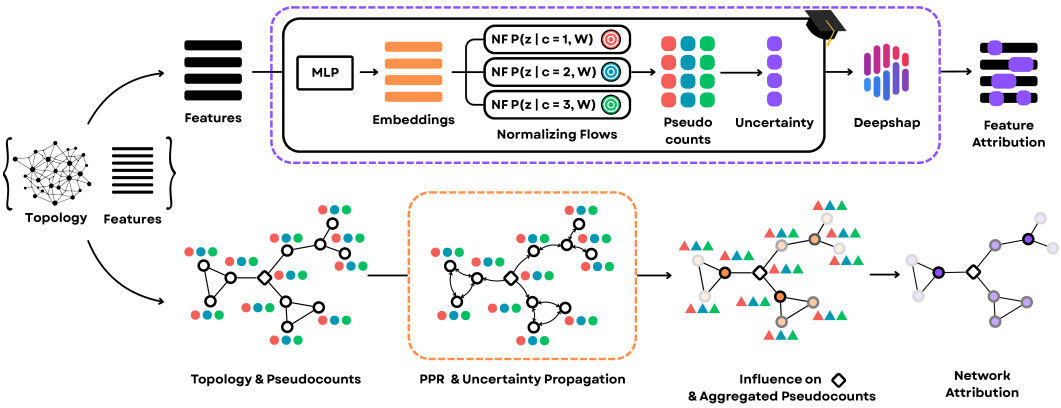


Figure 2: Overview of GraphUAT's pipeline: The top and bottom halves illustrate feature-level and network-level uncertainty attributions, respectively.

This formulation enforces three desiderata. First, attributions are separated by *type*: aleatoric and epistemic contributions are computed independently. Second, within each type, attributions are further divided by *source* into feature-level and network-level components, isolating the influence of the target node's own features from that of its neighbors. Third, the formulation is *faithful*: removing the components identified as most influential should produce a measurable reduction in uncertainty.

3 GraphUAT: Methodology

Designing an effective framework for uncertainty attribution in GNNs poses two central challenges.

Challenge 1: Quantifying uncertainty post-hoc. Typically, GNNs only output softmax scores, which does not disentangle aleatoric and epistemic components. Hence, the first challenge is to develop a *post-hoc* mechanism that enables this decomposition without requiring architectural changes or retraining of the original GNN.

Challenge 2: Attribution. The second challenge lies in explaining *where* each type of uncertainty originates including disentanglement of feature-level and topological contributions to uncertainty.

3.1 EDL Student GNN

We address the above challenges through a *teacher–student paradigm* (see Fig. 2). Given a trained GNN (the teacher), we train an *EDL student GNN* to mimic the teacher's uncertainty landscape. The student decouples aleatoric and epistemic uncertainty via its Dirichlet parameterization. Once trained, we probe each target node by comparing its responses against reference nodes (Sections 3.3 & 3.4), diagnosing which features and structural components are responsible for its uncertainty. The student separates uncertainty into two stages corresponding to its two sources (Fig. 2).

Stage 1: Feature-level Uncertainty. The first stage (Fig. 2, top) isolates uncertainty arising purely from a node's own features, without considering the graph structure. Each node v is assigned a Dirichlet prior parameterized by $\boldsymbol{\alpha}(v) = (\alpha_1(v), \ldots, \alpha_C(v)) \in \mathbb{R}^C_{>0}$, initialized uniformly ($\alpha_i(v) = 1$ for all i). The goal is to update this prior into a posterior that reflects how strongly the node's features support each class, via the additive update $\alpha'_i(v) = \alpha_i(v) + \beta_i(v)$, where $\beta_i(v)$ is the pseudo-count for class i.

As established in § 2.1, the Dirichlet–Categorical conjugacy yields the additive posterior update $\alpha_c \leftarrow \alpha_c + \beta_c$, where β_c counts observations of class c. Since a deterministic GNN produces a point estimate rather than a distribution, β_c cannot be observed directly. This can be addressed using *Normalizing Flows (NFs)* [47, 11, 10]: one NF per class is trained to model the conditional likelihood of observing a node's feature embedding under that class, giving $\beta_c(v) \propto \mathbb{P}(\mathbf{z}_v \mid c; \mathbf{W})$, where $\mathbf{z_v}$ is the embedding produced by an MLP encoder and $\mathbf{W}$ are the NF parameters. These likelihoods are scaled by a certainty budget N to yield the pseudo-count vector $\boldsymbol{\beta}(v) = (\beta_1(v), \ldots, \beta_C(v))$, which updates the Dirichlet prior.

The NF provides a continuous relaxation of the discrete observation count; the additive update structure is therefore not an analogy but inherited directly from the Dirichlet's conjugate update rule. A further advantage is that NFs naturally assign low likelihoods to embeddings far from the training distribution, yielding uniformly low pseudo-counts for OOD nodes and signalling high epistemic uncertainty without any additional mechanism.

Stage 2: Network-level Uncertainty. The second stage (Fig. 2, bottom) incorporates network context by aggregating the feature-level pseudo-counts, $\boldsymbol{\beta}^{\text{ft}}$, across the graph via a Personalized PageRank (PPR)-based message-passing scheme [19]. For each class c, the aggregated pseudo-count for node v is:

$$\beta_c^{\text{net}}(v) = \sum_{u \in \mathcal{V}} \Pi_{v,u}^{\text{PPR}} \, \beta_c^{\text{ft}}(u), \tag{4}$$

where $\Pi_{v,u}^{\text{PPR}}$ is the normalized PPR score between v and u. The update step is the standard Dirichlet update with $\boldsymbol{\alpha}^{net}$ initialized uniformly ($\alpha_c^{\text{net}}(v) = 1$ for all c). The PPR propagation is approximated using the power iteration method. We restrict the number of iteration steps to k, where k is the number of message passing layers in the teacher. This ensures that the student's receptive field is strictly restricted to the teacher's receptive field, rather than the complete set of graph nodes.

This aggregation admits a natural probabilistic interpretation. The PPR scores define a valid distribution over nodes, $\sum_u \Pi_{v,u}^{\text{PPR}} = 1$, corresponding to the stationary distribution of a PPR random walk, which can be viewed as a soft neighborhood for v accounting for all hop distances through infinitely many message-passing steps [19]. The NF output $\mathbb{P}(\mathbf{z}_u \mid c;\, \mathbf{W})$, on the other hand, defines a distribution over nodes using features alone. Writing $\mathbb{P}(v \mid u) = \Pi_{v,u}^{\text{PPR}}$ and $\mathbb{P}(u \mid c) = \mathbb{P}(\mathbf{z}_u \mid c;\, \mathbf{W})$, Eq. 4 can be recast as:

$$\beta_c^{\text{net}}(v) \;\propto\; \bar{\mathbb{P}}(v \mid c) = \sum_{u \in \mathcal{V}} \mathbb{P}(v \mid u)\, \mathbb{P}(u \mid c). \tag{5}$$

The quantity $\bar{\mathbb{P}}(v \mid c)$ is a valid distribution that normalizes jointly over all node features and accounts for the soft neighborhood, i.e. $\int \cdots \int \bar{\mathbb{P}}(v \mid c)\, dz^{(u_1)} \cdots dz^{(u_{|\mathcal{V}|})} = 1$. The PPR step thus transforms per-node feature distributions into a joint distribution over the soft-neighborhood features.

It is worth clarifying what this separation does and does not imply. The intent is not to claim that Stage 2 isolates structural uncertainty from feature uncertainty; rather, the two-stage design *factorizes* the sources of uncertainty. Stage 1 quantifies uncertainty arising purely from a node's own features; Stage 2 propagates and reshapes that uncertainty through the graph structure, modulating it based on local context. A node that is uncertain at the feature level may become more confident if its neighbors supply consistent, high-evidence signals during message passing. Conversely, uncertainty can increase when neighboring nodes convey conflicting or low-evidence information, even if each neighbor appears individually confident.

3.2 Teacher–Student Training

While standard GNNs (e.g., GCN, GIN, GAT) output only softmax probabilities without disentangling aleatoric and epistemic uncertainty, an EDL student GNN explicitly models predictive uncertainty via a Dirichlet posterior. This raises a key question: *how can we endow any pre-trained GNN with such uncertainty estimates post-hoc?*

Knowledge distillation as a bridge: We address this by distilling a trained GNN (the *teacher*) into an EDL GNN (the *student*) [24]. In our setting, however, simply matching predictions is insufficient: the student must also capture the teacher's *uncertainty landscape*, becoming uncertain in precisely the same regions where the teacher is uncertain. Once aligned, the student's uncertainty estimates serve as faithful proxies, enabling attribution.

Distillation objective: For each node v, we assume w.l.o.g. that the teacher produces a predictive distribution $\tilde{y}^v = \mathrm{softmax}(\mathbf{z}^v)$ over class labels, where $\mathbf{z}^v \in \mathbb{R}^C$. The student, in contrast, places a Dirichlet posterior over class probabilities, so its predictive distribution is the random variable $\mathbf{p}^{\text{net}}(v) \sim \mathrm{Dir}(\boldsymbol{\alpha}^{\text{net}}(v))$, and its point estimate is the Dirichlet mean $\bar{\mathbf{p}}^{\text{net}}(v) = \boldsymbol{\alpha}^{\text{net}}(v)/\alpha_0^{\text{net}}(v)$, where $\alpha_0^{\text{net}}(v) = \sum_c \alpha_c^{\text{net}}(v)$. In both cases the teacher's output $\tilde{y}^v$ is the target the student is fitted toward. The student's training loss is:

$$\mathcal{L}^v = (1-\omega)\, \underbrace{\mathbb{E}_{\mathbf{p}^{\text{net}}(v) \sim \mathrm{Dir}(\boldsymbol{\alpha}^{\text{net}}(v))}\big[-\log \Pr(\tilde{y}^v \mid \mathbf{p}^{\text{net}}(v))\big]}_{\text{Prediction matching}} - \lambda\, \underbrace{H\big(\mathrm{Dir}(\boldsymbol{\alpha}^{\text{net}}(v))\big)}_{\text{Posterior regularization}} + \omega\, \underbrace{\mathrm{KL}\big(\bar{\mathbf{p}}^{\text{net}}(v) \,\|\, \tilde{y}^v\big)}_{\text{Uncertainty alignment}}$$

Prediction matching ensures that the student's Dirichlet posterior yields predictive probabilities consistent with the teacher's. It integrates over the student's full posterior: it is the expected negative log-likelihood of the teacher's prediction under a categorical distribution whose parameters are drawn from $\mathrm{Dir}(\boldsymbol{\alpha}^{\text{net}}(v))$. *Posterior regularization* discourages low-entropy Dirichlet posteriors, preventing the student from collapsing into overconfident distributions. *Uncertainty alignment* explicitly minimizes the divergence between the teacher's softmax output and the student's mean

prediction $\bar{\mathbf{p}}^{\text{net}}(v)$, which is fixed given $\boldsymbol{\alpha}^{\text{net}}(v)$ rather than random, ensuring the student inherits the teacher's uncertainty landscape. The hyper-parameters $\omega \in [0, 1]$ and λ control the trade-off between alignment and regularization. Together, these terms encourage the student to not only reproduce the teacher's predictions but also learn *where and why* the teacher is uncertain. Our ablation study in App. M confirms that uncertainty alignment is the most influential of the three terms. Once distillation is complete, the problem reduces to performing uncertainty attribution over the trained EDL student, which now acts as a post-hoc explainer of the teacher's uncertainties.

Teacher uncertainty: Since the teacher is a standard softmax classifier with no native uncertainty decomposition, *teacher uncertainty* throughout the paper refers to the Shannon entropy of its predictive distribution:

$$u^{\text{teacher}}(v) = H[\tilde{y}^v] = -\sum_{c=1}^{C} \tilde{y}_c^v \log \tilde{y}_c^v. \tag{6}$$

This is the total, undecomposed predictive uncertainty exposed by the teacher, and its reduction is the quantity reported as Top-k and Random in Tables 2 and 3. App. S repeats this evaluation with two alternative teacher uncertainty measures.

3.3 Feature Uncertainty Attribution

Given a model's uncertainties, how do we distribute credit (or blame) among the input features? *Shapley values* [43] assign each feature a fair share of the output by averaging its marginal contribution across all possible subsets of features, making them a natural fit for attributing aleatoric and epistemic uncertainty in our setting. However, Shapley values have two practical issues: (i) dropping features is infeasible for GNNs whose parameter size depends on feature dimensionality, and (ii) exact computation requires 2^{d-1} model evaluations. DEEPSHAP [37] circumvents both. Instead of exhaustively evaluating all subsets, DEEPSHAP compares the model's activations for an input $\mathbf{x}_v$ against those for a set of *baseline* inputs $\mathcal{B} = \{\mathbf{b} \in \mathbb{R}^d\}$, chosen to represent "reference nodes". In our case, the baselines are the top-k nodes of the same predicted class as v with the lowest uncertainty. Intuitively, these baselines capture what the model considers highly certain inputs, so attributions quantify how much each feature of $\mathbf{x}_v$ deviates from a low-uncertainty reference. Thus, positive DEEPSHAP values indicate features that raise uncertainty, while negative values indicate features that contribute to more confident predictions. Applying this procedure separately to aleatoric and epistemic uncertainty yields:

$$\mathcal{A}_a^{\text{ft}}(v) = \left(\hat{\phi}_1(u_a^{\text{ft}}), \ldots, \hat{\phi}_d(u_a^{\text{ft}})\right), \quad \mathcal{A}_e^{\text{ft}}(v) = \left(\hat{\phi}_1(u_e^{\text{ft}}), \ldots, \hat{\phi}_d(u_e^{\text{ft}})\right),$$

where $\hat{\phi}_i(u_*^{\text{ft}})$ is feature i's DEEPSHAP value with respect to u_*^{ft}. Technical details, including the method's formulation and its connection to our attribution objectives, are provided in App. E. In terms of complexity, DEEPSHAP requires only $\mathcal{O}(|\mathcal{B}| \cdot d) = \mathcal{O}(k \cdot d)$ computations, in contrast to $\mathcal{O}(2^d)$ for exact Shapley values. Empirical validation is provided in App. K.

3.4 Network Uncertainty Attribution

In Stage 2 of the student, a node's aggregated Dirichlet parameters are computed via PPR (Eq. 4). Intuitively, if a neighbor u carries high feature-level uncertainty, or if neighbors are heterogeneous in their predicted classes, that uncertainty propagates to v in proportion to $\Pi_{v,u}^{\text{PPR}}$. Our objective is to attribute the network-level uncertainty of v back to all other nodes in the graph:

$$\mathcal{A}_a^{net}(v) = \left\{\text{Attr}_a^{marg}(v, u) \mid u \in \mathcal{N}^\ell(v)\right\}, \quad \mathcal{A}_e^{net}(v) = \left\{\text{Attr}_e^{marg}(v, u) \mid u \in \mathcal{N}^\ell(v)\right\},$$

where $\mathcal{N}^\ell(v) = \{u \in \mathcal{V} \mid sp(u, v) \leq \ell\}$ is the ℓ-hop neighborhood of v, and ℓ denotes the number of layers in the original GNN. Each $\text{Attr}_a^{marg}(v, u)$ and $\text{Attr}_e^{marg}(v, u)$ quantifies how much uncertainty from neighbor u contributes to v's network-level uncertainty, scaled by its PPR influence. We quantify these attribution maps via the *marginal reduction* in uncertainty upon the hypothetical removal of each neighbor. $\boldsymbol{\alpha}^{\text{net}\setminus u}(v) = \boldsymbol{\alpha}^{\text{net}}(v) - \Pi_{v,u}^{\text{PPR}} \boldsymbol{\beta}^{\text{ft}}(u)$ computes node v's concentration parameter after removing neighbor u's weighted pseudo-count from the aggregate.

AU attribution: The marginal contribution of u to v's aleatoric uncertainty is then $\text{Attr}_a^{\text{marg}}(v, u) = u_a^{\text{net}}(v) - u_a^{\text{net}\setminus u}(v)$, where $u_a(\cdot)$ is defined in Eq. (1) and $u_a^{\text{net}\setminus u}(v)$ is computed using $\boldsymbol{\alpha}^{\text{net}\setminus u}(v)$.

EU attribution: The epistemic attribution follows the same approach, with $\text{Attr}_e^{\text{marg}}(v, u) = u_e^{\text{net}}(v) - u_e^{\text{net}\setminus u}(v)$, where $u_e(\cdot)$ is defined in Eq. (2) and $u_e^{\text{net}\setminus u}(v)$ is computed using $\boldsymbol{\alpha}^{\text{net}\setminus u}(v)$.

Table 1: Dataset characteristics and student–teacher alignment validation.

(a) Characteristics of the datasets used.

Dataset	#Nodes	#Edges	#Features	#Classes
CoraML	2,995	16,316	2,879	7
CiteSeer	3,327	9,104	3,703	6
Amazon-P	7,650	238,162	745	8
Amazon-C	13,752	491,722	767	10
PubMed	19,717	88,648	500	3
Coauthor-CS	18,333	163,788	6,805	15
Coauthor-P	34,493	495,924	8,415	5
TAGCora	2,708	10,858	14,266	7
ogbn-arxiv	169,343	1,166,243	128	40

(b) Validation of student alignment with the teacher.

Dataset	Predictive Alignment		Uncertainty Alignment		
	Fidelity(↑)	κ (↑)	KL Div. (↓)	ρ (↑)	p-value (↓)
CoraML	0.95 ± 0.22	0.94	0.15	0.72	≈ 0
CiteSeer	0.83 ± 0.38	0.79	0.60	0.70	≈ 0
Amazon-P	0.96 ± 0.20	0.95	0.10	0.65	≈ 0
Amazon-C	0.92 ± 0.27	0.89	0.79	0.59	≈ 0
PubMed	0.92 ± 0.27	0.88	0.14	0.60	≈ 0
Coauthor-CS	0.92 ± 0.27	0.91	0.21	0.62	≈ 0
Coauthor-P	0.97 ± 0.18	0.95	0.08	0.54	≈ 0
TagCora	0.85 ± 0.36	0.81	0.38	0.64	≈ 0
ogbn-arxiv	0.78 ± 0.03	0.75	0.32	0.73	≈ 0

While prior works [30, 5, 45] have raised concerns about whether EDL-based uncertainty values are well-calibrated in absolute terms, GraphUAT is designed to be robust to this limitation. Our attribution mechanism depends solely on the *rank order* of uncertainties rather than their absolute magnitudes: network attribution measures the relative reduction in uncertainty upon neighbor ablation, and feature attribution compares activations against a reference set of low-uncertainty nodes. We prove in App. F that rank-faithfulness of EDL's uncertainty estimates is *sufficient* to guarantee rank-faithful attributions at both the network and feature levels. Critically, we validate this empirically not against EDL's internal estimates but against the teacher's uncertainty directly: the counterfactual experiments in Tables 2 and 3 show that removing GraphUAT-attributed features and neighbors produces significantly larger reductions in *teacher* uncertainty (Eq. (6)) than random removal, confirming that our attributions are faithful regardless of any EDL calibration concerns.

4 Experiments

In this section, we empirically establish **1) Faithfulness:** The student faithfully replicates the teacher's predictive distribution and its uncertainty profile. **2) Accurate uncertainty attributions:** GraphUAT produces meaningful and quantifiable attributions for both feature and network-level uncertainties. Our codebase is available at `https://github.com/idea-iitd/GraphUAT.git`

4.1 Experimental Setup

Hardware specs, hyper-parameters and training protocols are detailed in App. G and G.2, respectively. Unless mentioned, the default teacher is a GCN. We evaluate generalization to GAT and GRAPHSAGE in App. P. **Datasets:** Table 1a lists the datasets used to benchmark GraphUAT. App. G.1 presents further details. **Baselines:** To our knowledge, no prior method specifically targets uncertainty attribution on GNNs. For completeness, we adapt two natural baselines for comparison; full details and results appear in Section G.4. **Uncertainty quantification methods do not serve as a baseline** since our goal is not to improve the quality of uncertainty estimates but to *explain* their origins.

4.2 Faithfulness

Our objective is to verify that the student not only reproduces the teacher's predictive distribution but also faithfully matches its uncertainty profile.

Prediction distribution: We use two different metrics.

1. **Fidelity:** Fidelity measures how closely the student's class predictions match those of the teacher.
$$\text{Fidelity} = \frac{1}{|\mathcal{V}|} \sum_{v \in \mathcal{V}} \mathbf{1}\left[\hat{y}^{(v)}_{\text{student}} = \hat{y}^{(v)}_{\text{teacher}}\right],$$
2. **Cohen's Kappa(κ)** [14]: Cohen's Kappa goes a step further by adjusting for agreement that would occur purely by chance, making it a stricter measure of alignment than raw fidelity. $\kappa = 1$ indicates perfect agreement, while $\kappa = 0$ is equivalent to random guessing. In practice, $\kappa > 0.8$ is considered near-perfect alignment, while $0.6 < \kappa \leq 0.8$ indicates substantial agreement.

Table 1b reports **Fidelity** and κ between the student and the teacher. We observe high fidelity and κ across all datasets. These results demonstrate that GraphUAT effectively transfers the teacher's decision boundaries while maintaining predictive performance.

Uncertainty: To evaluate whether the student accurately captures the teacher's uncertainty, we report two complementary metrics. **(1)** The **KL divergence** between the teacher's distribution over class labels with the student's distribution (third term in Eq. 6). **(2)** The **Spearman rank correlation coefficient** (ρ) between the teacher's and student's total uncertainty rankings across nodes.

Table 2: Quantitative evaluation of feature-level uncertainty attribution ($k = 100$).

Dataset	Aleatoric Attribution						Epistemic Attribution					
	Top-k (↑)	**Random** (↑)	**Δ** (↑)	***t*-stat** (↑)	***p*-value** (↓)	**Cohen's** d_z (↑)	**Top-k** (↑)	**Random** (↑)	**Δ** (↑)	***t*-stat** (↑)	***p*-value** (↓)	**Cohen's** d_z (↑)
CoraML	$\mathbf{0.415 \pm 0.02}$	0.014 ± 0.00	0.401	26.76	≈ 0	0.98	$\mathbf{0.399 \pm 0.02}$	0.009 ± 0.00	0.390	26.11	≈ 0	0.95
CiteSeer	$\mathbf{0.323 \pm 0.01}$	0.005 ± 0.00	0.318	23.15	≈ 0	0.80	$\mathbf{0.172 \pm 0.02}$	0.009 ± 0.00	0.163	10.30	≈ 0	0.36
Amazon-P	$\mathbf{0.174 \pm 0.01}$	0.024 ± 0.00	0.150	24.90	≈ 0	0.57	$\mathbf{0.139 \pm 0.01}$	0.022 ± 0.00	0.116	23.30	≈ 0	0.53
Amazon-C	$\mathbf{0.177 \pm 0.01}$	0.015 ± 0.00	0.162	20.23	≈ 0	0.45	$\mathbf{0.132 \pm 0.01}$	0.013 ± 0.00	0.119	17.37	≈ 0	0.38
PubMed	$\mathbf{0.367 \pm 0.01}$	0.043 ± 0.00	0.324	53.91	≈ 0	0.99	$\mathbf{0.370 \pm 0.01}$	0.046 ± 0.00	0.324	52.29	≈ 0	0.96
Coauthor-CS	$\mathbf{0.348 \pm 0.01}$	0.004 ± 0.00	0.344	43.10	≈ 0	0.82	$\mathbf{0.360 \pm 0.01}$	0.004 ± 0.00	0.357	43.19	≈ 0	0.82
Coauthor-P	$\mathbf{0.100 \pm 0.00}$	0.002 ± 0.00	0.098	20.81	≈ 0	0.50	$\mathbf{0.067 \pm 0.00}$	0.001 ± 0.00	0.066	13.95	≈ 0	0.34
TAGCora	$\mathbf{0.302 \pm 0.02}$	0.005 ± 0.00	0.298	14.03	≈ 0	0.54	$\mathbf{0.319 \pm 0.02}$	0.000 ± 0.00	0.319	16.79	≈ 0	0.64
ogbn-arxiv	$\mathbf{0.458 \pm 0.06}$	0.077 ± 0.02	0.381	6.61	≈ 0	0.66	$\mathbf{0.419 \pm 0.06}$	0.074 ± 0.02	0.345	5.002	≈ 0	0.50

Table 3: Quantitative evaluation of network-level uncertainty attribution ($k = 50$).

Dataset	Aleatoric Attribution						Epistemic Attribution					
	Top-k (↑)	**Random** (↑)	**Δ** (↑)	***t*-stat** (↑)	***p*-value** (↓)	**Cohen's** d_z (↑)	**Top-k** (↑)	**Random** (↑)	**Δ** (↑)	***t*-stat** (↑)	***p*-value** (↓)	**Cohen's** d_z (↑)
CoraML	$\mathbf{0.975 \pm 0.02}$	0.031 ± 0.01	0.944	50.24	≈ 0	1.85	$\mathbf{0.779 \pm 0.02}$	0.028 ± 0.01	0.752	34.99	≈ 0	1.29
CiteSeer	$\mathbf{1.170 \pm 0.02}$	0.018 ± 0.01	1.152	64.58	≈ 0	2.25	$\mathbf{1.324 \pm 0.02}$	0.026 ± 0.01	1.300	104.93	≈ 0	3.67
Amazon-P	$\mathbf{0.886 \pm 0.01}$	0.008 ± 0.00	0.878	64.13	≈ 0	1.47	$\mathbf{0.205 \pm 0.01}$	0.006 ± 0.00	0.199	23.27	≈ 0	0.53
Amazon-C	$\mathbf{0.178 \pm 0.01}$	0.005 ± 0.00	0.172	20.63	≈ 0	0.35	$\mathbf{0.329 \pm 0.01}$	0.004 ± 0.00	0.325	32.84	≈ 0	0.56
PubMed	$\mathbf{0.714 \pm 0.01}$	0.001 ± 0.00	0.713	71.87	≈ 0	2.29	$\mathbf{0.684 \pm 0.01}$	0.004 ± 0.00	0.680	85.41	≈ 0	2.72
CoauthorCS	$\mathbf{0.490 \pm 0.03}$	0.004 ± 0.00	0.486	15.90	≈ 0	0.53	$\mathbf{1.016 \pm 0.02}$	0.004 ± 0.00	1.013	46.31	≈ 0	1.53
Coauthor-P	$\mathbf{0.462 \pm 0.02}$	0.000 ± 0.00	0.462	25.64	≈ 0	0.87	$\mathbf{0.643 \pm 0.01}$	0.000 ± 0.00	0.642	51.58	≈ 0	1.78
TAGCora	$\mathbf{1.103 \pm 0.02}$	0.027 ± 0.01	1.076	48.19	≈ 0	1.86	$\mathbf{1.101 \pm 0.02}$	0.018 ± 0.01	1.083	51.01	≈ 0	1.97
ogbn-arxiv	$\mathbf{2.581 \pm 0.02}$	0.001 ± 0.00	2.581	169.66	≈ 0	5.83	$\mathbf{2.535 \pm 0.01}$	0.000 ± 0.00	2.535	143.45	≈ 0	4.93

The uncertainty alignment results in Table 1b demonstrate that the student not only replicates the teacher's predictions but also closely matches its uncertainty estimates. The high **Spearman rank correlations** ($\rho = [0.54, 0.72]$) show that both models rank nodes by their total uncertainty in a consistent manner. This trend is also reflected in the low **KL divergence** (ranging 0.08 to 0.79). The p-values (≈ 0) confirm the statistical significance.

4.3 Quantifying Uncertainty Attribution

Our evaluation is organized into two components: feature-level attribution and network-level attribution. For both attribution tasks, our central hypothesis is that accurate attributions should pinpoint the true sources of uncertainty, and hence, replacing or removing these sources with more certain counterparts yields a significantly larger reduction in the model's overall uncertainty than performing the same operation on an equal number of randomly selected features or nodes. Importantly, the student model is used *only* to provide attribution signals, while uncertainty improvements are always computed on the teacher.

4.3.1 Feature Attribution

First, we compute the aleatoric (u_a^{ft}) and epistemic (u_e^{ft}) uncertainties resulting from features alone for each node in the test set. We sort the nodes based on these uncertainties to identify the groups of top-25% most *uncertain nodes* and the bottom 25% nodes corresponding to the most *certain nodes*. Next, we identify the top-k features contributing to the epistemic and aleatoric uncertainties of each node in the uncertain node set. To test if these top-k features attributions are faithful, we perform a counterfactual substitution experiment. For each uncertain node v, we identify the most similar certain node using the dot product between their embeddings. We then substitute the top-k attributing features of v with their counterparts from the most similar certain node. This substitution effectively makes the uncertain features less ambiguous while preserving the rest of the node's context. To establish statistical significance, we compare with a random baseline, where k features are chosen uniformly at random for substitution from the same most similar certain node. This setup allows us to isolate the impact of GraphUAT 's attributions.

Table 2 reports the improvements in uncertainty following the counterfactual substitutions[2]. The substantial gap between the top-k uncertain features with that of random selection, together with statistically significant t-statistics and p-values, strongly validates the effectiveness of GraphUAT's attributions. We also use the Cohen's d_z metric, where > 0.5 indicates that the improvement from removing top-ranked features or nodes exert a strong influence on model uncertainty. This pattern holds in 12 out of the 18 scenarios benchmarked in Table 2.

Impact of k: To observe how the difference in uncertainty reduction grows between GraphUAT's attributions and random, we refer to App. L in the Appendix.

4.3.2 Network Attribution Experiments

As in the case of features, we select the top 25% percentile of nodes based on their uncertainties and use GraphUAT to identify the top-K nodes from their ℓ-hop neighborhoods contributing to their epistemic and aleatoric uncertainties. To measure the faithfulness of these attributions, we

[2]We report ogbn-arxiv numbers for $k = 10$ instead of 100 as ogbn-arxiv only has 128 features.

Table 4: Alignment of the distilled student with the Bayesian teacher.

Dataset	Predictive Fidelity (↑)	Uncertainty ρ (↑)
CoraML	0.957	0.778
CiteSeer	0.813	0.812
Amazon-P	0.967	0.767
PubMed	0.857	0.641

Table 5: Decomposition of the teacher's total uncertainty (TU) into aleatoric (AU) and epistemic (EU) parts.

Dataset	Uncertainty			Share of TU	
	TU	AU	EU	AU/TU	EU/TU
CoraML	1.448	1.278	0.170	88.2%	11.8%
CiteSeer	1.517	1.454	0.063	95.9%	4.1%
Amazon-P	1.444	1.061	0.383	73.5%	26.5%
PubMed	0.976	0.848	0.129	86.8%	13.2%

Table 6: Quantitative evaluation of Bayesian GNN feature-level uncertainty attribution.

Dataset	Aleatoric Attribution				Epistemic Attribution			
	Ours (↑)	**Random** (↑)	Δ (↑)	p-**value** (↓)	**Ours** (↑)	**Random** (↑)	Δ (↑)	p-**value** (↓)
CoraML	$\mathbf{0.120 \pm 0.01}$	-0.026 ± 0.00	0.146	≈ 0	$\mathbf{0.377 \pm 0.02}$	0.234 ± 0.02	0.143	≈ 0
CiteSeer	$\mathbf{0.175 \pm 0.01}$	0.004 ± 0.00	0.171	≈ 0	$\mathbf{0.249 \pm 0.02}$	0.042 ± 0.01	0.207	≈ 0
Amazon-P	$\mathbf{0.132 \pm 0.00}$	0.047 ± 0.00	0.085	≈ 0	$\mathbf{0.140 \pm 0.01}$	0.078 ± 0.01	0.062	≈ 0
PubMed	$\mathbf{0.321 \pm 0.01}$	0.009 ± 0.01	0.312	≈ 0	$\mathbf{0.341 \pm 0.04}$	0.161 ± 0.04	0.180	≈ 0

exclude these top-k attributed neighbors while message passing and quantify the change in uncertainty. Further, we compare this change with the one imparted through removal of an equal number of neighbors selected uniformly at random from the ℓ-hop neighborhood.

Table 3 shows that the uncertainty reduction achieved by removing GraphUAT 's top-k neighbors is significantly higher than those obtained by removing random neighbors. The reported t-statistics, p-values and large Cohen's d_z values confirm the statistical robustness of this effect. In addition, we observe that the reduction in uncertainty achieved through topological attribution is substantially higher than that from feature attribution, suggesting that graph structure exerts a stronger influence on uncertainty propagation for these datasets.

4.4 Attribution faithfulness of Bayesian Teacher

In this section, we verify the correctness of the node attributions identified by the student by measuring the corresponding reductions in the Bayesian teacher's aleatoric and epistemic uncertainties. Our teacher is a Bayesian GCN, obtained by applying Bayes-by-Backprop [6] to the layers of a standard GCN. Following Depeweg et al. [17], we decompose the teacher's total predictive uncertainty into aleatoric and epistemic components via the mutual-information decomposition. This yields component-specific teacher signals against which we validate the student's attributions. Table 4 validates the student's alignment with the Bayesian GCN teacher.

Tables 6 and 7 report the feature-level and network-level results, respectively. These tables report relative reductions, $\left(u - u^{\setminus k}\right)/u$, averaged over the evaluated nodes. Normalization is required because the Bayesian teacher's aleatoric and epistemic uncertainties differ substantially in scale (Table 5), so absolute drops would not be comparable across the two. At the feature level, AU attributions reduce the teacher's aleatoric uncertainty and EU attributions reduce its epistemic uncertainty, both significantly on every dataset. At the network level, AU attributions remain statistically significant on every dataset. EU attributions outperform the random baseline on every dataset but reach statistical significance only on CiteSeer ($p = 0.021$). We attribute this to the composition of the teacher's uncertainty: the Bayesian teachers are predominantly aleatoric, with epistemic uncertainty accounting for only 4–27% of the total (Table 5). With little epistemic uncertainty to begin with, network-level perturbations have limited room to reduce it, making statistically significant effects harder to detect.

4.5 Case Study

In this section, we demonstrate the practical utility of UA for GNNs through concrete case studies on the Text-Attributed Cora (TAGCora) dataset [35]. TAGCora contains 7 class labels (`Rule Learning`, `Neural Networks`, `Case Based`, `Genetic Algorithms`, `Theory`, `Reinforcement Learning`, `Probabilistic Methods`). We choose TAGCora for the feature-level case study because, unlike CoraML, its features map explicitly to words, which makes the attributions human-readable. The teacher GNN, however, does not model textual semantics: each

Table 7: Quantitative evaluation of Bayesian GNN's network-level uncertainty attribution.

Dataset	Aleatoric Attribution				Epistemic Attribution			
	Ours (↑)	**Random** (↑)	Δ (↑)	p-**value** (↓)	**Ours** (↑)	**Random** (↑)	Δ (↑)	p-**value** (↓)
CoraML	$\mathbf{0.184 \pm 0.01}$	0.058 ± 0.01	0.126	≈ 0	0.153 ± 0.06	0.135 ± 0.04	0.018	0.396
CiteSeer	$\mathbf{0.059 \pm 0.00}$	0.036 ± 0.00	0.023	≈ 0	$\mathbf{0.098 \pm 0.02}$	0.060 ± 0.02	0.038	0.021
Amazon-P	$\mathbf{0.057 \pm 0.00}$	0.009 ± 0.00	0.048	≈ 0	0.155 ± 0.02	0.127 ± 0.02	0.028	0.108
PubMed	$\mathbf{0.205 \pm 0.01}$	0.092 ± 0.01	0.113	≈ 0	0.030 ± 0.07	0.019 ± 0.06	0.012	0.425

abstract enters the model only as a binary bag-of-words vector recording the presence or absence of vocabulary terms, with no access to word meaning or word order. The teacher draws its predictive power from co-occurrence patterns among these binary features together with topological homophily. We map features back to words and show the abstracts purely for the reader's convenience. The attributions below therefore explain which word-presence features drive the GNN's uncertainty, and should not be read as a claim of semantic-level uncertainty modeling comparable to NLP methods (see App. G.4). Two further case studies cover the remaining settings: uncertainty arising from graph structure and message passing (App. H), and a graph whose node features are tabular rather than textual (App. I).

Feature-level AU Attribution: Consider node 42's abstract from the "Neural Networks" class:

> Using Dirichlet mixture priors to derive hidden Markov models for protein families. A Bayesian method for estimating the amino acid distributions in the states of a hidden Markov model (HMM) for a protein family... [text truncated].

It lacks distinctive neural network terms and instead contains words like `Dirichlet, Bayesian, Markov`, which are more characteristic of "Probabilistic Methods" than "Neural Networks", making the sample inherently ambiguous and thus high in aleatoric uncertainty. Consistent with this, GraphUAT attributes increased aleatoric uncertainty to the presence of `Bayesian, Dirichlet, family, distributions, priors, estimating, Markov`, and to the absence of class-indicative terms such as `neural, network, nonlinear, neurons, learning`.

Feature-level EU Attribution: Consider node 1442's abstract from the "Neural Networks" class:

> Feature selection methods for classification. Intelligent Data Analysis: Feature selection is a problem of choosing a subset of relevant features. In general, an exhaustive search can bring about the optimal subset. With a monotonic measure, exhaustive search can be avoided without... [text truncated].

GraphUAT flagged it as epistemically uncertain and highlighted the **absence** of the following words as major contributors to its epistemic uncertainty: `network, neural, nonlinear, units, state, neurons, pattern, learning`. This attribution is revealing as a paper labeled as "Neural Networks" that omits such terms appears atypical relative to the training distribution. For an additional case study examining the uncertainty arising from network effects, refer to App. H.

4.6 Additional Experiments

Additional experiments appear in the appendix, including ablations (App. M), adapted baselines (App. G.3), robustness to under- and over-fitted teachers (App. O), and an extension to graph classification (App. J). A full listing is given in the appendix contents (p. 15).

5 Conclusion

We introduced GraphUAT, the first framework for *post-hoc uncertainty attribution* in GNNs. It provides fine-grained explanations that distinguish whether a node's uncertainty stems from its own features, its structural context, or both. Unlike prior methods that only quantify uncertainty, GraphUAT explains *why* the model is uncertain, complementing uncertainty quantification with attribution for the first time in graph learning. Being model-agnostic, it generalizes across architectures, providing uncertainty explanation as a plug-in capability. As the first effort in this direction, we also proposed systematic evaluation protocols for assessing uncertainty attribution in GNNs. Empirical results demonstrate that GraphUAT reliably identifies the uncertainty sources with statistically significant improvements over random baselines. We discuss the limitations of the framework in App. B.

Acknowledgments

Burouj Armgaan's research was supported by the Prime Minister's Research Fellowship, and the CSE Research Acceleration Fund of IIT Delhi provided partial support for this work. We thank the Yardi School of Artificial Intelligence, Indian Institute of Technology Delhi, for sponsoring Karan Raj Singh's travel and associated expenses for attending NeurIPS 2026.

References

[1] Javier Antorán, Umang Bhatt, Tameem Adel, Adrian Weller, and José Miguel Hernández-Lobato. Getting a clue: A method for explaining uncertainty estimates. In *ICLR*, 2021.

[2] Burouj Armgaan, Manthan Dalmia, Sourav Medya, and Sayan Ranu. Graphtrail: Translating gnn predictions into human-interpretable logical rules. In *NeurIPS*, 2024.

[3] Burouj Armgaan, Eshan Jain, Harsh Pandey, Mahesh Chandran, and Sayan Ranu. Gnnxemplar: Exemplars to explanations-natural language rules for global gnn interpretability. In *NeurIPS*, 2025.

[4] Gleb Bazhenov, Sergei Ivanov, Maxim Panov, Alexey Zaytsev, and Evgeny Burnaev. Towards ood detection in graph classification from uncertainty estimation perspective. *arXiv preprint arXiv:2206.10691*, 2022.

[5] Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. On second-order scoring rules for epistemic uncertainty quantification. In *ICML*, 2023.

[6] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. In *ICML*, 2015.

[7] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*, 2018.

[8] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. Active learning for graph embedding. *arXiv preprint arXiv:1705.05085*, 2017.

[9] Seohyeon Cha, Honggu Kang, and Joonhyuk Kang. On the temperature of bayesian graph neural networks for conformal prediction. In *NeurIPS*, 2023.

[10] Bertrand Charpentier, Oliver Borchert, Daniel Zügner, Simon Geisler, and Stephan Günnemann. Natural posterior network: Deep bayesian uncertainty for exponential family distributions. In *ICLR*, 2022.

[11] Bertrand Charpentier, Daniel Zügner, and Stephan Günnemann. Posterior network: Uncertainty estimation without ood samples via density-based pseudo-counts. In *NeurIPS*, 2020.

[12] Chao Chen, Chenghua Guo, Rui Xu, Xiangwen Liao, Xi Zhang, Sihong Xie, Hui Xiong, and Philip Yu. Uncertainty quantification on graph learning: A survey. *arXiv preprint arXiv:2404.14642*, 2024.

[13] Jase Clarkson. Distribution free prediction sets for node classification. In *ICML*, 2023.

[14] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 1960.

[15] Enyan Dai and Suhang Wang. Say no to the discrimination: Learning fair graph neural networks with limited sensitive attribute information. In *WSDM*, 2021.

[16] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 1991.

[17] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *ICML*, 2018.

[18] Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. Graph neural networks for recommender system. In *WSDM*, 2022.

[19] Johannes Gasteiger, Aleksandar Bojchevski, and Stephan Günnemann. Predict then propagate: Graph neural networks meet personalized pagerank. In *ICML*, 2019.

[20] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 1998.

[21] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017.

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

[23] Silu He, Qinyao Luo, Ronghua Du, Ling Zhao, Guangjun He, Han Fu, and Haifeng Li. Stgc-gnns: A gnn-based traffic prediction framework with a spatial–temporal granger causality graph. *Physica A: Statistical Mechanics and its Applications*, 2023.

[24] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015.

[25] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

[26] Kexin Huang, Ying Jin, Emmanuel Candes, and Jure Leskovec. Uncertainty quantification over graph with conformalized graph neural networks. In *NeurIPS*, 2023.

[27] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Machine Learning*, 2021.

[28] Wei Jin, Tong Zhao, Jiayuan Ding, Yozen Liu, Jiliang Tang, and Neil Shah. Empowering graph representation learning with test-time graph transformation. In *ICLR*, 2023.

[29] Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. Graphpatcher: mitigating degree bias for graph neural networks via test-time augmentation. In *NeurIPS*, 2023.

[30] Mira Juergens, Nis Meinert, Viktor Bengs, Eyke Hüllermeier, and Willem Waegeman. Is epistemic uncertainty faithfully represented by evidential deep learning methods? In *ICML*, 2024.

[31] Jian Kang, Qinghai Zhou, and Hanghang Tong. Jurygcn: quantifying jackknife uncertainty on graph convolutional networks. In *KDD*, 2022.

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

[33] Mert Kosan, Zexi Huang, Sourav Medya, Sayan Ranu, and Ambuj Singh. Gcfexplainer: Global counterfactual explainer for graph neural networks. *ACM Transactions on Intelligent Systems and Technology*, 2025.

[34] Mert Kosan, Samidha Verma, Burouj Armgaan, Khushbu Pahwa, Ambuj Singh, Sourav Medya, and Sayan Ranu. GNNX-BENCH: Unravelling the utility of perturbation-based GNN explainers through in-depth benchmarking. In *ICLR*, 2024.

[35] Hao Liu, Jiarui Feng, Lecheng Kong, Ningyue Liang, Dacheng Tao, Yixin Chen, and Muhan Zhang. One for all: Towards training one graph model for all classification tasks. In *ICLR*, 2024.

[36] Yajing Liu, Zhengya Sun, and Wensheng Zhang. Improving fraud detection via hierarchical attention-based graph neural network. *Journal of Information Security and Applications*, 2023.

[37] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.

[38] Robert Lunde, Elizaveta Levina, and Ji Zhu. Conformal prediction for network-assisted regression. *Journal of the American Statistical Association*, 2025.

[39] Andrey Malinin and Mark Gales. Predictive uncertainty estimation via prior networks. In *NeurIPS*, 2018.

[40] Galileo Mark Namata, Ben London, Lise Getoor, and Bert Huang. Query-driven active surveying for collective classification. In *International Workshop on Mining and Learning with Graphs*, Edinburgh, Scotland, 2012.

[41] Weizhu Qian, Dalin Zhang, Yan Zhao, Kai Zheng, and James JQ Yu. Uncertainty quantification for traffic forecasting: A unified approach. In *ICDE*, 2023.

[42] Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. In *NeurIPS*, 2018.

[43] L. Shapley. *7. A Value for n-Person Games. Contributions to the Theory of Games II (1953) 307-317.* Princeton University Press, 1997.

[44] Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *ArXiv*, 2018.

[45] Maohao Shen, Jongha Jon Ryu, Soumya Ghosh, Yuheng Bu, Prasanna Sattigeri, Subhro Das, and Gregory W. Wornell. Are uncertainty quantification capabilities of evidential deep learning a mirage? In *NeurIPS*, 2024.

[46] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. Graph posterior network: Bayesian predictive uncertainty for node classification. In *NeurIPS*, 2021.

[47] Dennis Ulmer, Christian Hardmeier, and Jes Frellsen. Prior and posterior networks: A survey on evidential deep learning methods for uncertainty estimation. *TMLR*, 2023.

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

[49] Samidha Verma, Burouj Armgaan, Sourav Medya, and Sayan Ranu. Induce: Inductive counterfactual explanations for graph neural networks. *TMLR*, 2024.

[50] Fangxin Wang, Yuqing Liu, Kay Liu, Yibo Wang, Sourav Medya, and Philip S. Yu. Uncertainty in graph neural networks: A survey. *TMLR*, 2024.

[51] Qingyi Wang, Shenhao Wang, Dingyi Zhuang, Haris Koutsopoulos, and Jinhua Zhao. Uncertainty quantification of spatiotemporal travel demand with probabilistic graph neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 2024.

[52] Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration. In *NeurIPS*, 2021.

[53] Dongxia Wu, Liyao Gao, Matteo Chinazzi, Xinyue Xiong, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Quantifying uncertainty in deep spatiotemporal forecasting. In *KDD*, 2021.

[54] Jiacheng Xiong, Zhaoping Xiong, Kaixian Chen, Hualiang Jiang, and Mingyue Zheng. Graph neural networks for automated de novo drug design. *Drug discovery today*, 2021.

[55] Hui Xu, Liyao Xiang, Junjie Ou, Yuting Weng, Xinbing Wang, and Chenghu Zhou. Open-world graph active learning for node classification. *ACM Transactions on Knowledge Discovery from Data*, 2023.

[56] Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *NeurIPS*, 2019.

[57] Soroush H Zargarbashi, Simone Antonelli, and Aleksandar Bojchevski. Conformal prediction sets for graph neural networks. In *ICML*, 2023.

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

[59] Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. Uncertainty aware semi-supervised learning on graph data. In *NeurIPS*, 2020.

Appendix Contents

A Notation **16**

B Limitations & Future Work **16**

C Background Primer **17**

D Uncertainty Quantification **17**

E DEEPSHAP **18**

F Proofs **19**
F.1 Network Attribution 19
F.2 Feature Attribution 20

G Experimental Setup **22**
G.1 Dataset details 22
G.2 Training details 23
G.3 Student classification performance 23
G.4 Baselines 23

H Continued Case Study: Network-Level Attribution **25**
H.1 AU Attribution 25
H.2 EU Attribution 25

I Case Study on a Non-Text-Attributed Graph **25**
I.1 Case 1: High AU — An Under-Median Player with Mixed Statistics 26
I.2 Case 2: High EU — An Atypical Over-Median Player 26

J Extension to Graph Classification **27**

K Runtime **27**

L Impact of k on Feature Attribution **29**

M Ablation Study on Loss Terms **30**

N Effect of Training Data Size on Student Fidelity **30**

O Robustness to sub-optimally trained teachers **31**

P Generalization to Other GNN Architectures **32**

Q Sensitivity of Uncertainty Attribution to GNN Architecture **32**

R Impact of distillation quality on uncertainty attribution **33**

S Framework agnostic validation of attributions **33**

T Post-intervention classification behavior **34**

A Notation

Table 8: Key symbols and notations used throughout the paper. AU and EU denote Aleatoric and Epistemic Uncertainty, respectively

Symbol	Interpretation
$\mathcal{G}$	Graph
$\mathcal{V}$	Vertex set
$\mathcal{E}$	Edge set
$\boldsymbol{X}$	Node features
$\mathcal{V}_{tr}$	Training nodes
$\mathcal{Y}$	Ground-truth classes
y_i	Ground-truth class of node i
Φ	GNN
u_t	Total uncertainty
u_a	Aleatoric uncertainty
u_e	Epistemic uncertainty
$\mathbf{E}[\cdot]$	Expectation
$H[\cdot]$	Entropy
$\Gamma(\cdot)$	Gamma function
$\Psi(\cdot)$	Digamma function
$\mathcal{A}_{a,e}^{\text{feat}}(v,j)$	Contribution of node v's j^{th} feature to its AU, EU
$\mathcal{A}_{a,e}^{\text{net}}(v,u)$	Contribution of node u to v's AU, EU
$\boldsymbol{\alpha}(v)$	Dirichlet parameter vector of node v's
$\boldsymbol{\beta}(v)$	Pseudocount vector of node v's
$\beta_c^{\text{ft}}(u)$	Node v's feature-level pseudocount for class c
$\beta_c^{\text{net}}(u)$	Node v's aggregated pseudocount for class c
$\Pi_{v,u}^{\text{PPR}}$	Normalized PPR score between nodes v and u
$\mathbf{p}(v)$	Predicted class probability distribution of node v's
$\alpha_0(v)$	Concentration parameter of node v
ω, λ	Weighting hyperparameters in loss
$f^v(\mathbf{x}_v)$	Function calculating feature-level AU/EU uncertainty
$\mathbf{b}$	DeepSHAP baseline
$\mathcal{B}$	Set of DeepSHAP baselines
$\hat{\phi}_j$	Shapley value of feature j
$C_j^{\text{DL}}(\mathbf{x}_v, \mathbf{b})$	DeepLIFT's feature level contribution
$C_j^{\text{DS}}(\mathbf{x}_v, \mathbf{b})$	DeepSHAP's feature level contribution

B Limitations & Future Work

Uncertainty management in graph learning can be viewed as a three-stage process: *quantification*, *attribution*, and *mitigation*. While prior work has focused on the first stage, this paper advances the second. We don't address mitigating the uncertainty after having found its source. It is an open problem and a natural direction for future work.

The use of PPR-based aggregation in Stage 2 of our pipeline implicitly assumes a degree of homophily. Our pipeline is most reliable on homophilous neighborhoods. On heterophilic graphs, conflicting neighborhood signals may be conflated with uncertainty. Extending the attribution framework to heterophilic settings is an interesting direction for future work.

Dependence on distillation. The faithfulness of the distilled student rests on empirical evidence; we do not provide theoretical guarantees for it. As with any distillation pipeline, users should verify a new teacher–student pair's fidelity and Spearman correlation before trusting its attributions. A poorly distilled student is not rank-faithful to its teacher, in which case the rank-faithfulness assumption underlying our attribution proofs (App. F) no longer holds, and the resulting attributions should not

be treated as reliable. This failure mode is observable rather than silent, since fidelity and ρ are reported alongside every attribution (App. R).

Scope of the guarantees. Even given rank-faithfulness, the feature-level guarantee holds unconditionally only under rank-averaged aggregation of DEEPSHAP; for the standard score-averaged estimator it is exact when EDL's uncertainty is an affine function of the true uncertainty, and approximate otherwise (App. F).

Evaluation signal. A standard softmax teacher exposes only total predictive uncertainty, so our main counterfactual evaluation measures the reduction in the teacher's entropy; a component-wise evaluation requires a Bayesian teacher (§ 4.4).

C Background Primer

Categorical distribution and the probability simplex. A C-class prediction is a probability vector $\mathbf{p} = (p_1, \ldots, p_C)$ with $p_c \geq 0$ and $\sum_c p_c = 1$. The set of all such vectors is the *probability simplex* Δ^{C-1}; for $C = 3$ it is a triangle whose corners are the three one-hot predictions and whose centre is the uniform prediction. Given $\mathbf{p}$, a label is drawn from the categorical distribution $\Pr(y = c \mid \mathbf{p}) = p_c$. A softmax classifier outputs a single point on this simplex.

Dirichlet distribution. The Dirichlet $\mathrm{Dir}(\boldsymbol{\alpha})$ is a distribution *over* the simplex, that is, a distribution over predictions rather than a single prediction. Its mean is $\mathbb{E}[p_c] = \alpha_c/\alpha_0$ with $\alpha_0 = \sum_c \alpha_c$, and its variance $\mathrm{Var}[p_c] = \alpha_c(\alpha_0 - \alpha_c)/\big(\alpha_0^2(\alpha_0 + 1)\big)$ shrinks as α_0 grows. The *ratios* of the α_c therefore say where the mass sits on the simplex, and the *total* α_0 says how concentrated it is. For $C = 3$:

- $\boldsymbol{\alpha} = (100, 1, 1)$: mass concentrated near one corner. Confident prediction; low AU and low EU.
- $\boldsymbol{\alpha} = (50, 50, 1)$: mass concentrated midway between two corners. The model has seen ample evidence, but the evidence is split between two classes. High AU, low EU.
- $\boldsymbol{\alpha} = (1, 1, 1)$: mass spread uniformly over the whole simplex. The model has seen no evidence at all. High EU.

Both the second and third cases have a near-uniform mean prediction, so a softmax output cannot tell them apart; the Dirichlet can, through α_0.

Conjugate prior. A prior is *conjugate* to a likelihood when the posterior stays in the same family as the prior. The Dirichlet is conjugate to the categorical distribution: starting from $\mathrm{Dir}(\boldsymbol{\alpha})$ and observing β_c labels of class c gives the posterior $\mathrm{Dir}(\boldsymbol{\alpha} + \boldsymbol{\beta})$. Updating beliefs therefore amounts to adding counts, which is why EDL can express evidence as additive pseudo-counts β_c (§ 2.1).

Normalizing flows. A normalizing flow is an invertible neural network T that maps a simple base density q_0 (e.g., a standard Gaussian) to a flexible density, with the exact likelihood given by the change-of-variables formula $q(\mathbf{z}) = q_0\big(T^{-1}(\mathbf{z})\big) \left| \det \partial T^{-1}(\mathbf{z})/\partial \mathbf{z} \right|$. GraphUAT trains one flow per class on node embeddings, so $q(\mathbf{z}_v \mid c)$ measures how typical node v looks for class c. Embeddings far from all training data receive low density under every class, and hence few pseudo-counts, which is how high EU arises for OOD nodes.

Personalized PageRank (PPR). PPR models a random walk that, at each step, moves to a random neighbor with probability $1 - \gamma$ and teleports back to the start node with probability γ. The resulting stationary distribution, $\boldsymbol{\Pi}^{\mathrm{PPR}} = \gamma\big(\mathbf{I} - (1 - \gamma)\hat{\mathbf{A}}\big)^{-1}$ for a normalized adjacency $\hat{\mathbf{A}}$, assigns each node u a weight $\Pi^{\mathrm{PPR}}_{v,u}$ that decays with its graph distance from v. Row v, normalized to sum to one, is a soft neighborhood of v. In practice it is approximated by a few steps of power iteration [19], which GraphUAT truncates at the teacher's number of layers.

D Uncertainty Quantification

Following the taxonomy proposed by [50], most approaches for graph UQ can be broadly categorized into Direct, Bayesian, Frequentist methods.

Direct estimation methods [52, 8, 55] are the most straightforward and rely solely on the model's output logits or softmax probabilities as a proxy for uncertainty. For example, one may use the

maximum softmax probability or predictive entropy to estimate how confident the model is. However, these approaches are inherently limited: they only provide an estimate of the total uncertainty and cannot disentangle aleatoric and epistemic components, as no information about model parameter variability is available.

Bayesian methods form a richer family of approaches, as they explicitly model uncertainty in a principled manner. Within this family, there are two broad paradigms:

- **Single Deterministic Model** [59, 46]: These approaches keep the model deterministic but place a distribution over its outputs, e.g., using a Dirichlet distribution to model predictive probabilities. This allows for capturing predictive variability without modifying the model's internal parameters. They allow for disentangling aleatoric and epistemic uncertainty without requiring multiple forward passes.
- **Single Model with Parameters Being Random Variables** [58, 22, 53, 9]: Here, the model parameters themselves are treated as random variables, enabling a full Bayesian treatment. In practice, exact Bayesian inference is intractable, so approximations such as Monte Carlo (MC) dropout or variational inference are employed. Performing multiple stochastic forward passes produces a distribution of predictions for each node, which enables the decomposition of total uncertainty into its aleatoric and epistemic components (via marginalization over the weight posterior).

Some of these methods are designed not to quantify uncertainty per se, but to make the model robust to uncertainty (e.g., distributionally robust GNNs).

Frequentist methods estimate uncertainty in a model-agnostic, distribution-free manner, typically through post-processing rather than modifying the model itself. These approaches are generally computationally efficient, as they avoid placing priors over model parameters or introducing additional learnable components. A common line of work leverages resampling techniques, such as jackknife (leave-one-out) estimation combined with influence functions, to capture variability in learned parameters without invoking Bayesian assumptions [31]. Another prominent direction is conformal prediction, which provides guaranteed coverage by constructing prediction sets calibrated on held-out data [13, 26, 57]. Recent refinements in this area focus on reducing set sizes while maintaining coverage guarantees, thereby improving efficiency. While these methods quantify total predictive uncertainty and are robust to model misspecification, they generally do not disentangle aleatoric and epistemic components [41]. Furthermore, these methods rely on the assumption that data samples are exchangeable, an assumption that is often approximated in practice through random splits or tailored sampling strategies for graphs [38].

The final set of methods, ensembles and test-time augmentation, do not neatly fit into this taxonomy. **Ensemble GNNs** [4, 51] combine predictions from multiple independently trained models, with predictive uncertainty emerging from the diversity among members. While effective, they remain computationally expensive and largely lack graph-specific adaptations. **Test-time augmentation** instead perturbs inputs at inference and measures prediction variability. Though applied in GNNs for OOD robustness [28, 29], its use for uncertainty quantification remains unexplored, yet promising due to its post hoc nature and efficiency.

Our objective is to identify the sources that make a GNN uncertain. This comes under the broader umbrella of GNN explainability [2, 34, 49, 33, 56, 3].

E DEEPSHAP

Starting with Shapley values, let the feature set of the target node v be $\mathcal{F} = \{1, \ldots, d\}$, with feature vector $\mathbf{x}_v \in \mathbb{R}^d$. We use $f^v(\mathbf{x}_v) \in \{u_a^{\text{ft}}(v),\, u_e^{\text{ft}}(v)\}$ to denote either aleatoric or epistemic feature-level uncertainty. The exact Shapley value for feature $j \in \mathcal{F}$ is:

$$\begin{aligned}\phi_j(f^v(\mathbf{x}_v)) = \sum_{S \subseteq \mathcal{F} \setminus \{j\}} & \frac{|S|!\,(d - |S| - 1)!}{d!} \\ & \times \left(f^v\left(\mathbf{x}_v^{S \cup \{j\}}\right) - f^v\left(\mathbf{x}_v^{S}\right) \right)\end{aligned} \tag{7}$$

where $\mathbf{x}_v^S$ is the feature vector with only features in S retained.

DEEPLIFT computes contributions by tracking differences in activations between $\mathbf{x}_v$ and a single baseline $\mathbf{b}$. Letting $\Delta y_k = y_k(\mathbf{x}_v) - y_k(\mathbf{b})$ be the activation difference at neuron k, DEEPLIFT redistributes these differences layer by layer back to the input features. The resulting feature-level contributions $C_j^{\mathrm{DL}}(\mathbf{x}_v, \mathbf{b})$ satisfy the conservation property:

$$\sum_{j=1}^{d} C_j^{\mathrm{DL}}(\mathbf{x}_v, \mathbf{b}) = f^v(\mathbf{x}_v) - f^v(\mathbf{b}). \tag{8}$$

Here, $C_j^{\mathrm{DL}}(\mathbf{x}_v, \mathbf{b})$ quantifies how much of the output difference $f^v(\mathbf{x}_v) - f^v(\mathbf{b})$ is traced back to the difference in feature j in $\mathbf{x}_v$ relative to a baseline $\mathbf{b}$, propagated through intermediate activations.

While DEEPLIFT provides a fast and locally consistent redistribution of output differences, it relies on a single baseline, which limits its faithfulness to the Shapley formulation that defines feature importance as an expectation over all possible reference contexts. DEEPSHAP approximates the Shapley value by averaging these contributions across a set of baselines $\mathcal{B}$:

$$\hat{\phi}_j\left(f^v(\mathbf{x}_v)\right) = C_j^{\mathrm{DS}}(\mathbf{x}_v, \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{b} \in \mathcal{B}} C_j^{\mathrm{DL}}(\mathbf{x}_v, \mathbf{b}), \tag{9}$$

$$\text{and} \quad \sum_{j=1}^{d} C_j^{\mathrm{DS}}(\mathbf{x}_v, \mathcal{B}) = f^{(v)}(\mathbf{x}_v) - \mathbb{E}_{\mathcal{B}}[f^{(v)}(\mathbf{b})]$$

F Proofs

The rank-faithfulness of EDL's uncertainty implies that for any data points v_1, v_2,

$$u(v_1) \geq u(v_2) \iff u_*(v_1) \geq u_*(v_2) \tag{10}$$

where $u(\cdot)$ is EDL's uncertainty estimate and $u_*(\cdot)$ is the true uncertainty. We prove that this is sufficient for GraphUAT to produce rank-faithful uncertainty attributions. The proof is broken into two parts

F.1 Network Attribution

Let A denote $\text{Attr}^{\text{marg}}$ for brevity, with subscripts a, e dropped. The proof covers both cases.

To prove: GraphUAT produces rank-faithful uncertainty attributions at the network level.

$$A(n, v_1) \geq A(n, v_2) \iff A_*(n, v_1) \geq A_*(n, v_2) \tag{11}$$

where $A_*(\cdot)$ is the correct attribution produced using $u_*(\cdot)$.

For a target node n with neighbors v_1, v_2, from § 3.4 we have

$$A(n, v_1) = u(n) - u^{\setminus v_1}(n), \tag{12}$$

$$A(n, v_2) = u(n) - u^{\setminus v_2}(n). \tag{13}$$

Subtracting (13) from (12) yields

$$A(n, v_1) - A(n, v_2) = u^{\setminus v_2}(n) - u^{\setminus v_1}(n). \tag{14}$$

Therefore,

$$A(n, v_1) \geq A(n, v_2) \iff u^{\setminus v_2}(n) \geq u^{\setminus v_1}(n). \tag{15}$$

By the same reasoning,

$$A_*(n, v_1) \geq A_*(n, v_2) \iff u_*^{\setminus v_2}(n) \geq u_*^{\setminus v_1}(n). \tag{16}$$

By EDL's rank-faithfulness (10), the ablated uncertainties satisfy

$$u^{\setminus v_2}(n) \geq u^{\setminus v_1}(n) \iff u_*^{\setminus v_2}(n) \geq u_*^{\setminus v_1}(n). \tag{17}$$

Combining (15), (16), and (17), we conclude

$$A(n, v_1) \geq A(n, v_2) \iff A_*(n, v_1) \geq A_*(n, v_2)$$

□

F.2 Feature Attribution

Feature attribution is performed using DEEPSHAP [37]. Unlike network attribution, feature attribution is independent of the graph structure: each node is considered in isolation, and its uncertainty stems solely from its own features. Consequently, the feature attribution of a target node n depends only on the baseline set $\mathcal{B}$; the rest of the graph is uninvolved.

To prove: GraphUAT produces rank-faithful uncertainty attributions at the feature level.

$$\mathcal{A}^{\text{ft}}(n)_j \geq \mathcal{A}^{\text{ft}}(n)_i \iff \mathcal{A}_*^{\text{ft}}(n)_j \geq \mathcal{A}_*^{\text{ft}}(n)_i \tag{18}$$

for features $i, j \in \{1, \ldots, d\}$, where $\mathcal{A}_*^{\text{ft}}(\cdot)$ is the correct attribution produced using $u_*(\cdot)$.

Scope of the guarantee. DEEPSHAP combines per-baseline DEEPLIFT contributions into a single feature ranking, and whether the guarantee holds depends on how this combination is done. We consider two aggregation schemes: (i) *score averaging*, the standard DEEPSHAP estimator in Eq. (9), which averages contributions across baselines and then ranks the averaged scores; and (ii) *rank averaging*, which ranks the features separately under each baseline and then averages these ranks. We show that rank averaging preserves the ranking for *any* strictly increasing transformation of the uncertainty, whereas score averaging preserves it exactly when the transformation is affine, and approximately when the baselines are tightly clustered in uncertainty. A toy example at the end of this section shows that the affine condition cannot be dropped for score averaging.

Baseline set invariance. For a target node n, GraphUAT selects the baseline set $\mathcal{B}$ as the top-k most confident nodes from the same predicted class. Given EDL's rank-faithfulness (10), the selection of $\mathcal{B}$ is unaffected: the top-k confident nodes chosen based on $u(\cdot)$ are identical to those chosen based on $u_*(\cdot)$, since only the ranking matters for this selection.

Monotonic transformation of uncertainty. EDL not producing exact uncertainty estimates but being rank-faithful implies that

$$u(x) = g(u_*(x)) \tag{19}$$

where $g(\cdot)$ is a strictly monotonic function. Since the equivalence in (10) preserves the direction of the ordering, g is strictly *increasing*. We write $f^{(n)} = g \circ f_*^{(n)}$, where $f_*^{(n)}$ computes the true feature-level uncertainty, so that $\mathcal{A}_*^{\text{ft}}$ is the DEEPSHAP attribution of $f_*^{(n)}$. We write $C_j^{\text{DL}}(\mathbf{x}_n, \mathbf{b};\, h)$ for the DEEPLIFT contribution of feature j when explaining a function h.

Effect on DEEPSHAP attributions. The baseline set $\mathcal{B}$ is chosen at the node level, not the feature level. All features of node n share the same baseline set. From Appendix E, DEEPSHAP computes the contribution of each feature j as

$$C_j^{\text{DS}}(\mathbf{x}_n, \mathcal{B}) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{b} \in \mathcal{B}} C_j^{\text{DL}}(\mathbf{x}_n, \mathbf{b}), \tag{20}$$

which satisfies the conservation property

$$\sum_{j=1}^{d} C_j^{\text{DS}}(\mathbf{x}_n, \mathcal{B}) = f^{(n)}(\mathbf{x}_n) - \mathbb{E}_{\mathcal{B}}[f^{(n)}(\mathbf{b})]. \tag{21}$$

Single baseline: invariance to any strictly increasing g. Fix a baseline $\mathbf{b} \in \mathcal{B}$. DEEPLIFT computes contributions through a chain-rule composition of layer-wise multipliers. Appending g after $f_*^{(n)}$ adds one final multiplier, the secant slope of g between the target's and the baseline's true uncertainty:

$$m_g(\mathbf{x}_n, \mathbf{b}) = \frac{g\big(f_*^{(n)}(\mathbf{x}_n)\big) - g\big(f_*^{(n)}(\mathbf{b})\big)}{f_*^{(n)}(\mathbf{x}_n) - f_*^{(n)}(\mathbf{b})}, \tag{22}$$

which is replaced by the derivative of g when the denominator vanishes, following the gradient fallback of the Rescale rule. Since g is strictly increasing, $m_g(\mathbf{x}_n, \mathbf{b}) > 0$. Crucially, m_g does not depend on the feature index j; it is shared by every feature's contribution:

$$C_j^{\mathrm{DL}}\big(\mathbf{x}_n, \mathbf{b};\, g \circ f_*^{(n)}\big) = m_g(\mathbf{x}_n, \mathbf{b}) \cdot C_j^{\mathrm{DL}}\big(\mathbf{x}_n, \mathbf{b};\, f_*^{(n)}\big) \qquad \forall j \in \{1, \dots, d\}. \tag{23}$$

Multiplying every feature's contribution by the same positive constant preserves their relative order. Hence, *for a single baseline*, the feature ranking is invariant to any strictly increasing g; no affine condition is needed.

Multiple baselines: what is averaged matters. With $|\mathcal{B}| > 1$, each baseline carries its own multiplier, and by (23),

$$C_j^{\mathrm{DS}}\big(\mathbf{x}_n, \mathcal{B};\, g \circ f_*^{(n)}\big) = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{b} \in \mathcal{B}} m_g(\mathbf{x}_n, \mathbf{b}) \cdot C_j^{\mathrm{DL}}\big(\mathbf{x}_n, \mathbf{b};\, f_*^{(n)}\big). \tag{24}$$

Unless $m_g(\mathbf{x}_n, \mathbf{b})$ is the same for every $\mathbf{b}$, this is a non-uniformly reweighted average of the true per-baseline contributions, which can reorder features. Whether the ranking survives therefore depends on the aggregation scheme.

Proposition (Rank averaging). For each $\mathbf{b} \in \mathcal{B}$, let $\mathrm{rank}_{\mathbf{b}}(j)$ denote the rank of feature j when features are sorted in decreasing order of $C_j^{\mathrm{DL}}(\mathbf{x}_n, \mathbf{b})$, with ties broken by a fixed rule, and define

$$\bar{R}_j = \frac{1}{|\mathcal{B}|} \sum_{\mathbf{b} \in \mathcal{B}} \mathrm{rank}_{\mathbf{b}}(j). \tag{25}$$

Then, for any strictly increasing g and all features i, j,

$$\bar{R}_j \le \bar{R}_i \iff \bar{R}_{j*} \le \bar{R}_{i*}, \tag{26}$$

where $\bar{R}_{j*}$ is computed from the contributions of $f_*^{(n)}$.

Proof. By (23), under every baseline $\mathbf{b}$ the contributions for $g \circ f_*^{(n)}$ are a positive multiple of those for $f_*^{(n)}$. The sorted order, ties included, is therefore unchanged, so $\mathrm{rank}_{\mathbf{b}}(j)$ is identical under u and u_* for every j and $\mathbf{b}$. By baseline set invariance, $\mathcal{B}$ is also identical. Averaging identical ranks over an identical set gives $\bar{R}_j = \bar{R}_{j*}$ for all j, and the claim follows. □

Proposition (Score averaging under affine g). If $g(z) = az + c$ with $a > 0$, then $C_j^{\mathrm{DS}}(\mathbf{x}_n, \mathcal{B};\, g \circ f_*^{(n)}) = a\, C_j^{\mathrm{DS}}(\mathbf{x}_n, \mathcal{B};\, f_*^{(n)})$ for all j, and hence

$$C_j^{\mathrm{DS}}(\mathbf{x}_n, \mathcal{B}) \ge C_i^{\mathrm{DS}}(\mathbf{x}_n, \mathcal{B}) \iff C_{j*}^{\mathrm{DS}}(\mathbf{x}_n, \mathcal{B}) \ge C_{i*}^{\mathrm{DS}}(\mathbf{x}_n, \mathcal{B}). \tag{27}$$

Proof. The secant slope of an affine function is constant, so $m_g(\mathbf{x}_n, \mathbf{b}) = a$ for every $\mathbf{b} \in \mathcal{B}$. Substituting into (24) gives a uniform positive scaling, which preserves the ordering. □

Remark (Score averaging under near-affine g). For general g, write $m_{\mathbf{b}} = m_g(\mathbf{x}_n, \mathbf{b})$ and let $\bar{m} > 0$ be its mean over $\mathcal{B}$. Decomposing (24),

$$C_j^{\mathrm{DS}} - C_i^{\mathrm{DS}} = \bar{m}\,\big(C_{j*}^{\mathrm{DS}} - C_{i*}^{\mathrm{DS}}\big) + \frac{1}{|\mathcal{B}|} \sum_{\mathbf{b} \in \mathcal{B}} (m_{\mathbf{b}} - \bar{m})\big(C_{j*}^{\mathrm{DL}}(\mathbf{b}) - C_{i*}^{\mathrm{DL}}(\mathbf{b})\big), \tag{28}$$

where the arguments $(\mathbf{x}_n, \cdot)$ are dropped for brevity. The strict order of any pair i, j with $C_{j*}^{\mathrm{DS}} \ne C_{i*}^{\mathrm{DS}}$ is therefore preserved whenever

$$\bar{m}\,\big|C_{j*}^{\mathrm{DS}} - C_{i*}^{\mathrm{DS}}\big| > \max_{\mathbf{b} \in \mathcal{B}} |m_{\mathbf{b}} - \bar{m}| \cdot \frac{1}{|\mathcal{B}|} \sum_{\mathbf{b} \in \mathcal{B}} \big|C_{j*}^{\mathrm{DL}}(\mathbf{b}) - C_{i*}^{\mathrm{DL}}(\mathbf{b})\big|. \tag{29}$$

If g is twice differentiable with $|g''| \le M$ on an interval containing $u_*(n)$ and $\{u_*(\mathbf{b})\}_{\mathbf{b} \in \mathcal{B}}$, then the secant slope in (22), viewed as a function of the baseline's uncertainty, is $(M/2)$-Lipschitz. This gives

$$\max_{\mathbf{b} \in \mathcal{B}} |m_{\mathbf{b}} - \bar{m}| \le \frac{M}{2}\, \delta_*, \qquad \delta_* = \max_{\mathbf{b}, \mathbf{b}' \in \mathcal{B}} \big|u_*(\mathbf{b}) - u_*(\mathbf{b}')\big|. \tag{30}$$

Because GraphUAT selects $\mathcal{B}$ as the top-k most confident nodes of a single predicted class, the baselines are tightly clustered in uncertainty and δ_* is small by construction. The right-hand side of (29) is then small, and only pairs of features whose true attributions are nearly tied can be reordered. This is a sufficient condition, not a worst-case guarantee.

Example (the affine condition is necessary for score averaging). Consider a target node with $u_*(\mathbf{x}) = 10$ and three baselines: $\mathbf{b}_1$ with $u_*(\mathbf{b}_1) = 9$, and $\mathbf{b}_2 = \mathbf{b}_3$ with $u_*(\mathbf{b}_2) = 1$. These baselines are deliberately *not* tightly clustered. Suppose the true per-baseline contributions of two features are $C(\mathbf{x}, \mathbf{b}_1) = (10, -9)$ and $C(\mathbf{x}, \mathbf{b}_2) = C(\mathbf{x}, \mathbf{b}_3) = (-1, 10)$, consistent with conservation. Let $g(z) = z^2$, which is strictly increasing on $z \geq 0$. Then $m_g(\mathbf{x}, \mathbf{b}) = u_*(\mathbf{x}) + u_*(\mathbf{b})$, giving $m_g(\mathbf{b}_1) = 19$ and $m_g(\mathbf{b}_2) = m_g(\mathbf{b}_3) = 11$. The transformed contributions are $(190, -171)$ and $(-11, 110)$, so each per-baseline ranking is unchanged. Table 9 shows that score averaging nevertheless flips the winning feature, whereas rank averaging does not.

Table 9: Toy example: score averaging (higher wins) versus rank averaging (lower wins), with and without the non-affine transformation $g(z) = z^2$.

	Score avg. (↑)		**Rank avg. (↓)**	
	u_*	$g(u_*)$	u_*	$g(u_*)$
Feature 1	2.67	**56.00**	1.67	1.67
Feature 2	**3.67**	16.33	**1.33**	**1.33**
Winner	Feature 2	Feature 1	Feature 2	Feature 2

Conclusion. Since the baseline set $\mathcal{B}$ is identical under $u(\cdot)$ and $u_*(\cdot)$, and each per-baseline ranking is invariant to any strictly increasing g, GraphUAT produces rank-faithful feature-level uncertainty attributions (i) *unconditionally* under rank averaging, and (ii) under the standard score-averaged DEEPSHAP estimator when g is affine, and approximately when the baselines are tightly clustered in uncertainty, as GraphUAT's baseline selection ensures. Rank averaging gains this unconditional invariance at the cost of the additive conservation property (21), which holds only for score averaging. □

G Experimental Setup

Experiments were conducted on a server with Intel(R) Xeon(R) Gold 6426Y: 64 cores, 126 GB RAM, 2 NVIDIA-L40S GPUs, 45 GiB each running on Ubuntu 20.04.6 LTS.

G.1 Dataset details

- **CoraML** [7] A citation network dataset where nodes represent machine learning papers, edges denote citations, and node features are bag-of-words representations of the paper's content. The task is to classify papers into one of seven machine learning categories.
- **CiteSeer** [20] A citation network dataset with nodes as scientific publications, edges as citation links, and bag-of-words node features. The goal is to classify publications into six research topics.
- **PubMed** [40] A biomedical citation network where nodes correspond to scientific papers on diabetes, edges represent citation links, and node features are term-frequency vectors from each paper's TF–IDF representation. The classification task is to categorize papers into three classes related to different types of diabetes research.
- **AmazonComputers** [44] A subset of Amazon co-purchasing network where nodes are products, edges indicate frequently co-purchased items, and node features come from product reviews (bag-of-words). The classification task is to predict product categories related to computers.
- **AmazonPhotos** [44] A subset of the Amazon co-purchasing network focused on photo and camera products. Similar to Amazon-Computers, nodes represent products with review-based features, and the goal is to classify products into relevant photo-related categories.
- **CoauthorCS** [44] A co-authorship graph derived from the Microsoft Academic Graph used in the KDD Cup 2016 challenge, with nodes representing authors in the computer science (CS)

Dataset	Teacher-acc	Student-acc
CoraML	0.906	0.886
CiteSeer	0.806	0.795
AmazonPhotos	0.924	0.905
AmazonComputers	0.833	0.823
PubMed	0.842	0.841

Table 10: Teacher and student accuracies across datasets.

Table 11: Feature attribution. Reduction in teacher predictive entropy when the top-k attributed features of uncertain nodes are replaced with those of certain nodes (higher is better). Best in bold.

Dataset	**GraphUAT**	**Grad**	**GnnEx**	**Random**
CoraML	$\mathbf{0.415 \pm 0.02}$	-0.093 ± 0.01	-0.007 ± 0.02	0.014 ± 0.00
CiteSeer	$\mathbf{0.323 \pm 0.01}$	-0.128 ± 0.01	-0.010 ± 0.01	0.005 ± 0.00
Amazon-P	$\mathbf{0.174 \pm 0.01}$	-0.105 ± 0.00	-0.003 ± 0.00	0.024 ± 0.00

community, edges indicating co-authorship links, and node features extracted from the authors' paper keywords. The classification task is to predict each author's main area of research in CS.

- **CoauthorPhysics** [44] A co-authorship graph derived from the Microsoft Academic Graph used in the KDD Cup 2016 challenge, with nodes representing authors in the physics community, edges indicating co-authorship links, and node features extracted from the authors' paper keywords. The classification task is to predict each author's main area of research in physics.
- **TAGCora** [35] A text-attributed variant of the Cora citation network where each node includes the raw abstract text of a research paper. Unlike CoraML, whose bag-of-words features are anonymized, TAGCora provides an explicit mapping between features and words, enabling direct interpretability of feature-level attributions. The task remains to classify papers into seven machine learning research topics.
- **ogbn-arxiv** [25] A directed citation network of Computer Science arXiv papers indexed by the Microsoft Academic Graph, where nodes represent papers and directed edges indicate citations. Each node is associated with a 128-dimensional feature vector obtained by averaging the skip-gram word embeddings of its title and abstract. The task is to classify each paper into one of 40 CS subject areas (e.g., cs.AI, cs.LG), as manually labeled by authors and arXiv moderators. The dataset is split by publication year, with training on papers up to 2017, validation on 2018, and testing on papers from 2019 onward.

G.2 Training details

- **Teacher GNN Architectures** All teacher models (GCN, GAT, GRAPHSAGE [32], [48], [21]) use 2 layers with 64 hidden dimensions.
- **Student GNN Configuration** All student models (GPN) are restricted to teacher GNN's neighborhood (2-hop) with entropy regularization $\lambda = 0.0001$ and $\omega = 0.5$.
- **Dataset-Specific Configuration** We have used a train/validation/test split of 3:3:4 labels per class across all datasets, except ogbn-arxiv which uses its public OGB split.
- **Training Protocol** The teacher is trained with cross entropy loss and early-stopping. The student uses the (6) and stops early on cross-entropy against the teacher. Both are optimized using Adam.

G.3 Student classification performance

Table 10 reports the classification accuracy of the trained student over the ground truth distribution.

G.4 Baselines

In the absence of prior methods designed for uncertainty attribution, we construct two natural baselines and evaluate them under the same counterfactual protocol used throughout § 4. The

Table 12: Network attribution. Reduction in teacher predictive entropy when the top-k attributed neighbors of uncertain nodes are removed (higher is better). Best in bold.

Dataset	GraphUAT	PPR	GnnEx	Random
CoraML	$\mathbf{0.975 \pm 0.02}$	0.488 ± 0.03	0.591 ± 0.10	0.031 ± 0.01
CiteSeer	$\mathbf{1.170 \pm 0.02}$	0.322 ± 0.02	0.319 ± 0.06	0.018 ± 0.01
Amazon-P	$\mathbf{0.886 \pm 0.01}$	0.143 ± 0.02	0.403 ± 0.12	0.008 ± 0.00

baselines operate on the teacher's predictive entropy and cannot differentiate between aleatoric and epistemic uncertainty since the teacher only provides the total uncertainty signal.

CLUE [1] This method explains a model's uncertainty by searching the latent space of a generative model for counterfactuals that reduce uncertainty, and takes the resulting feature changes as attributions. Although effective on tabular and image data, CLUE is ill-suited to the GNN setting for four reasons. First, it assumes i.i.d. inputs, whereas nodes in a graph are interdependent: a node's identity is defined not only by its own features but also by its position in the graph and by the features and labels of its neighbors, so uncertainty both propagates from a node to the network and is inherited from it. Second, graphs have no canonical node ordering, so attributions must be permutation equivariant; methods built for images instead rely on a fixed pixel grid. Third, graphs lack a coordinate system and fixed-size neighborhoods, which makes generative models designed for images difficult to transfer. Finally, message passing entangles feature and structural signals, so perturbing node features alone, as CLUE does, cannot capture uncertainty arising from graph structure.

Grad / PPR. A gradient-based attribution that ranks features by $|\nabla_{\mathbf{x}_i} H(p_\theta(y \mid \mathcal{G}))| \odot \mathbf{x}_i$. For network attribution we remove the top-k neighbors ranked by their raw PPR-weighted contribution to the teacher's softmax entropy. We then run our existing evaluation protocol against these. This is the most direct "why do you need a student?" challenge: it asks the teacher itself which inputs move its uncertainty.

GnnEx GNNEXPLAINER [56] identifies the minimal subgraph that maximizes the mutual information between the full graph's predictive distribution and the subgraph's predictive distribution, by minimizing their KL divergence subject to a subgraph-size regularizer. We repurpose it to explain the teacher's predictive entropy as the target signal rather than the predicted class, testing whether an existing GNN explainer, given the right target, can recover uncertainty sources without further modification.

Random. Uniformly sampled features and neighbors, as in our main experiments.

Tabular. Tabular attribution methods operate in the feature space. Applied to graphs, they score each node's raw feature vector directly and yield a feature-level uncertainty signal. This appears to work, but it explains the wrong model: our goal is to explain the uncertainty of the GNN, not that of a classifier trained on node features in isolation from the network. A natural alternative is to apply the method to the node embeddings produced by the GNN instead. This runs into an entanglement problem. The first message-passing layer already fuses a node's own features with those of its neighbors, so beyond the input there is no layer at which the node's features and its neighbors' features remain separately addressable. An attribution computed on the embeddings therefore cannot be cleanly credited back to the node's own features.

NLP. NLP-based attribution methods are the furthest removed from our setting, since they are designed to exploit textual semantics. GraphUAT explains an already-trained GNN, and a GNN neither models nor uses textual semantics in its decisions. NLP-based techniques are therefore a poor fit: attributing the teacher's uncertainty to semantic properties would credit the explanation to information the model never had access to. In our case study on Tag-Cora, node features are bag-of-words vectors over a fixed vocabulary; we map them back to words only for display, as a convenience for the reader. The teacher GNN draws its predictive power from the presence or absence of words together with topological homophily, not from lexical meaning.

Tables 11 and 12 report the change in teacher predictive entropy when the top-k attributed features (resp. neighbors) of uncertain nodes are replaced (resp. removed). GraphUAT outperforms all baselines by a substantial margin on both tasks and across all datasets. (Table 11), GraphUAT achieves

positive uncertainty reductions across all three datasets, while Grad and GnnEx fail to do so on any. On network attribution (Table 12), the baselines are competitive. GraphUAT nonetheless improves over the strongest baseline across datasets. This shows that while the baselines do identify informative neighbors, GraphUAT pinpoints them more effectively for a given budget.

H Continued Case Study: Network-Level Attribution

H.1 AU Attribution

To complement the feature-level analyses in the main paper, we now examine a case study of uncertainty arising from graph structure and message passing. Consider node `817` whose aleatoric uncertainty increases by 60.22% following message passing . GraphUAT attributes this rise in uncertainty to a set of confident yet differently labeled neighbors `[1028, 2564, 5, 661, 1686]`. Although each neighbor is individually certain, their class mismatch makes the neighborhood non-homophilic within an otherwise homophilic graph, introducing conflicting evidence that increases aleatoric uncertainty at node `817`.

Conversely, node `671` shows the opposite trend: it is feature-wise uncertain but becomes more confident after aggregation. Its aleatoric uncertainty drops by 58.58% post aggregation. GraphUAT reveals that its neighbors are largely from the same class with low aleatoric uncertainty, providing consistent evidence that stabilizes its prediction. Together, these examples illustrate how GraphUAT disentangles the direction and source of uncertainty propagation.

H.2 EU Attribution

Next, we examine how epistemic uncertainty propagates across the graph. Node `1121` is epistemically certain when considered in isolation but becomes uncertain once the network is integrated. Its epistemic uncertainty increases by 52.96% post aggregation. GraphUAT attributes this change to a set of neighbors exhibiting significantly higher epistemic uncertainty causing node `1121` to inherit their uncertainty during message passing.

Conversely, node `1299` shows the reverse trend: it is epistemically uncertain based on its own features but becomes more confident after aggregation. Its epistemic uncertainty decreases by 2.21 orders of magnitude post aggregation. GraphUAT reveals that its most influential neighbors have substantially lower epistemic uncertainty. Their stable representations provide reliable context, effectively grounding node `1299` within a well-learned neighborhood and reducing its epistemic uncertainty.

Overall, these examples demonstrate GraphUAT 's nuanced capacity to disentangle and localize uncertainty at the network level. It captures both how uncertainty flows through the network and why a node's confidence changes in context, showcasing interpretability that extends beyond individual nodes to the network.

I Case Study on a Non-Text-Attributed Graph

We present a qualitative case study on the NBA graph from FairGNN [15], whose node features are human-readable tabular attributes rather than bag-of-words text. The graph extends a Kaggle dataset of NBA players. Node features include 2016–17 performance statistics and related information, and edges are Twitter follow relationships collected via the official Twitter API. The task is to predict whether a player's salary is above the median. Table 13 summarizes the dataset.

Table 13: Statistics of the NBA dataset.

#Nodes	#Edges	#Features	#Classes
403	16,570	39	2

We examine one case of high aleatoric uncertainty (AU) and one of high epistemic uncertainty (EU).

I.1 Case 1: High AU — An Under-Median Player with Mixed Statistics

Aleatoric uncertainty is high when the evidence is inherently conflicting: even with perfect data, it is unclear which salary class the player belongs to.

GraphUAT ranks node 29 (`user_id=61288678`), an under-median player, in the top 1% by AU. It attributes this uncertainty to five features: minutes played (`MP`), personal fouls (`PF_x`), games played (`GP`), real plus-minus (`RPM`), and defensive RPM (`DRPM`). Table 14 compares these features against the class statistics; for each feature, the class the player more closely resembles is shown in bold.

Table 14: Attributed features of node 29 compared with class statistics. Bold marks the class each feature more closely resembles.

Feature	Node 29	Under-median	Over-median
MP	30.5	17.7 ± 8.2	$\mathbf{26.0 \pm 7.2}$
PF_x	2.5	1.59 ± 0.77	$\mathbf{2.02 \pm 0.55}$
GP	53	$\mathbf{51.7 \pm 25.0}$	65.0 ± 16.4
RPM	-1.57	$\mathbf{-1.48 \pm 1.95}$	0.23 ± 2.68
DRPM	0.06	-0.12 ± 1.54	0.14 ± 1.72

The attributed features disagree. `MP` and `PF_x` point toward over-median: the player logs starter-level minutes and fouls at a rate typical of higher-usage players. `GP` and `RPM` point toward under-median, landing almost exactly on that class's averages. `DRPM` (0.06) sits nearly midway between the two class means (-0.12 and 0.14) and favors neither. The features GraphUAT identifies thus split between the two classes rather than agreeing, which is precisely the kind of conflicting evidence that drives aleatoric uncertainty.

To test the attribution, we replace the attributed features with those of a confidently under-median player (node 168, `user_id=186027867`), who plays far fewer minutes (`MP`: 30.5 $\rightarrow$ 10.3) and commits fewer fouls (`PF_x`: 2.5 $\rightarrow$ 0.7). This reduces node 29's AU from 0.676 to 0.078 (-88%), whereas replacing a random set of features leaves it essentially unchanged ($\approx$0.681). Once the conflicting features are resolved toward a clear under-median profile, the model is no longer uncertain, consistent with genuine ambiguity in the data rather than a modeling failure.

I.2 Case 2: High EU — An Atypical Over-Median Player

Epistemic uncertainty is high when a node differs from the examples the model has seen in its class, for instance an extreme profile relative to peers with the same label.

GraphUAT ranks node 283 (`user_id=35982046`), an over-median player, in the top 1% by EU, and attributes this uncertainty to `points`, two-point attempts (`2PA`), `assists`, and offensive RPM (`ORPM`). As Table 15 shows, the player is not merely above average on these features but at the extreme tail of his class.

Table 15: Attributed features of node 283 compared with over-median class statistics.

Feature	Node 283	Over-median	Percentile
points	31.6	12.4 ± 6.7	**100th**
2PA	16.8	6.9 ± 3.7	**99th**
assists	10.4	2.7 ± 2.2	**99th**
ORPM	6.74	0.09 ± 2.30	**99th**

By contrast, over-median players with low EU average only about 8.1 points and 4.8 two-point attempts. Node 283 is clearly a high earner, but an extreme one: his scoring, usage, playmaking, and offensive impact far exceed those of typical over-median players. He therefore belongs to the correct class but lies at the edge of what that class looks like in the training graph.

Replacing the attributed features with those of a more typical over-median player (node 50, `user_id=272116860`), namely `points` 31.6 $\rightarrow$ 9.1, `2PA` 16.8 $\rightarrow$ 2.5, `assists` 10.4 $\rightarrow$ 0.6, and `ORPM` 6.74 $\rightarrow$ 1.62, reduces EU from 0.0365 to 0.00022 (-99%), compared with 0.018 (-51%) for

a random replacement. The predicted class remains over-median, but the profile no longer looks unusual.

The two uncertainty types have clearly distinct attribution signatures, both directly interpretable from raw statistics: aleatoric uncertainty arises from features that point to different classes, while epistemic uncertainty arises from features that consistently point to one class but take extreme values within it. In both cases, replacing the attributed features with those of a confident donor from the same class removes the corresponding uncertainty component, while random replacements have a much smaller effect, confirming the attributions. This demonstrates that GraphUAT's attributions remain interpretable beyond text-attributed graphs.

J Extension to Graph Classification

To evaluate whether GraphUAT generalizes beyond node classification, we apply it to a graph classification task using the Mutagenicity [16] dataset. Since GraphUAT operates on node-level uncertainty signals, we adapt the setting by introducing a virtual node to each graph and connecting it to all existing nodes, allowing the student to express graph-level uncertainty through this aggregation node.

The student GPN trained in this setting achieves a fidelity of 84% with respect to the teacher, indicating that the distillation process remains effective for graph-level tasks. Table 19 reports the quantitative evaluation of uncertainty attribution on network-level sources. GraphUAT yields substantial improvements over random baselines, with mean gains of 0.62 ± 0.01 for aleatoric uncertainty and 0.61 ± 0.04 for epistemic uncertainty. All improvements are statistically significant ($p < 0.001$), with large effect sizes (Cohen's $d_z = 4.53$ for AU and 6.90 for EU).

We do not analyze feature-level attribution for this dataset, as node features correspond only to atom types. Overall, these results demonstrate that GraphUAT extends naturally to graph classification and remains effective outside the node-level regime.

K Runtime

The runtime analysis in Fig. 3 demonstrates the computational efficiency of our pipeline. For most datasets (CoraML, AmazonPhotos, and CoauthorPhysics), distillation completes in under 100 seconds — comparable to or faster than teacher training — making the approach practical for real-world applications. The ogbn-arxiv dataset exhibits longer distillation time (~5.5 minutes) due to its 40 class labels, which requires training 40 normalizing flows (one for each class). Notably, even with this increased model complexity, the total runtime remains on par with teacher training (~4 minutes), demonstrating that our distillation overhead remains manageable even as the number of classes scales.

Fig. 4 reports the per-node DEEPSHAP attribution time across the same four datasets. Attribution remains fast across all scales, ranging from under 0.05 seconds per node on CoraML and AmazonPhotos to approximately 0.2 and 0.7 seconds on CoauthorPhysics and ogbn-arxiv, respectively. Consistent with DEEPSHAP's $\mathcal{O}(|\mathcal{B}| \cdot d)$ complexity, the runtime is governed by the size of the baseline set $\mathcal{B}$ which becomes larger with dataset size. The absolute times confirm that feature attribution adds negligible overhead to the overall pipeline even on the largest dataset tested.

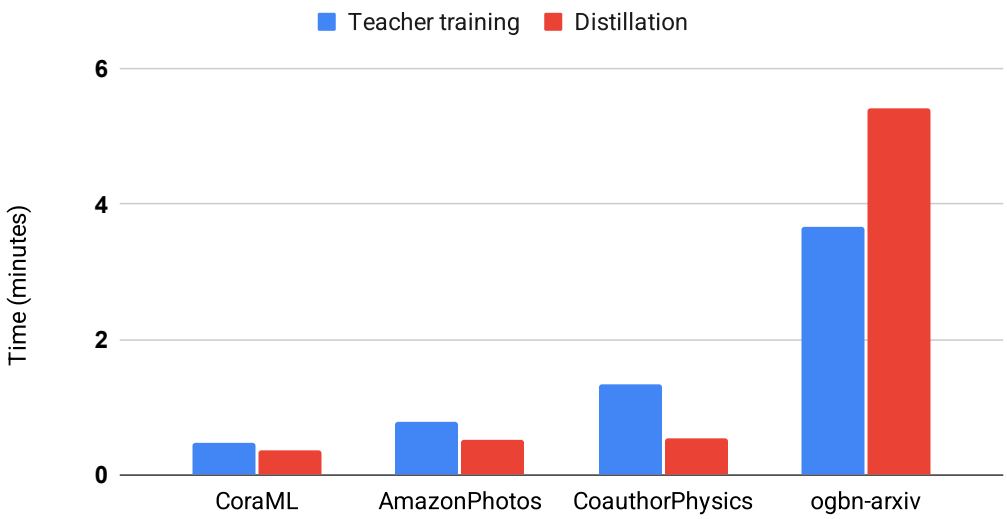


Figure 3: Runtime comparison *(in minutes)* between teacher training and distillation across four benchmark datasets of varying scale. Dataset scale (nodes/edges): CoraML (3K/16K), AmazonPhotos (8K/238K), CoauthorPhysics (34K/496K), ogbn-arxiv (169K/1.2M).

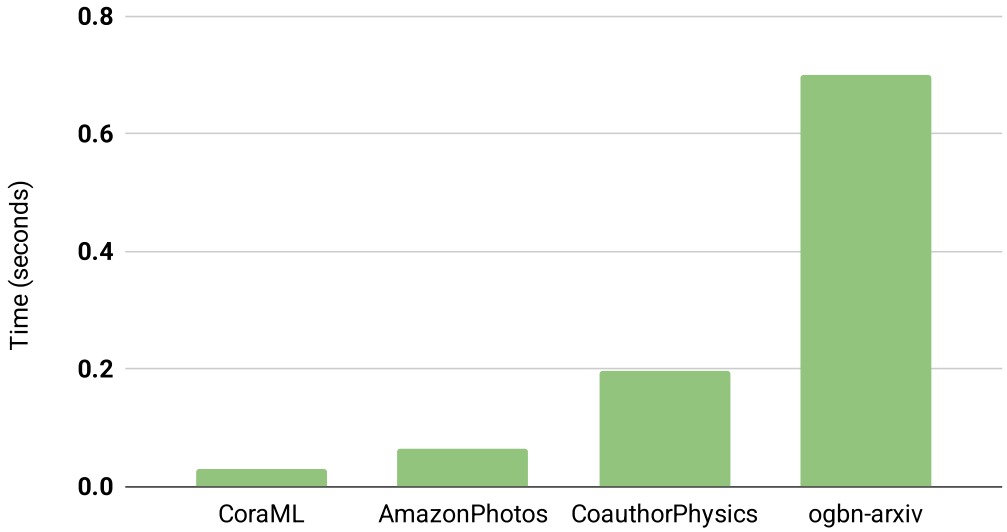


Figure 4: Per node DEEPSHAP attribution time *(in seconds)* across four benchmark datasets of varying scale. Dataset scale (nodes/edges): CoraML (3K/16K), AmazonPhotos (8K/238K), CoauthorPhysics (34K/496K), ogbn-arxiv (169K/1.2M).

Table 16: Student–teacher fidelity on CoraML across varying training data fractions and values of ω.

Fraction	**1.0**	**0.75**	**0.5**	**0.25**	**0.10**
$\omega = 0.25$	0.92	0.91	0.90	0.83	0.80
$\omega = 0.75$	0.94	0.91	0.90	0.88	0.80

Table 17: Quantitative evaluation of uncertainty attribution under sub-optimally trained teachers. Overfit-S stands for strong overfitting, where the fraction of training nodes is further reduced compared to standard overfitting.

	Feature		Network	
Model	**AU (↑)**	**EU (↑)**	**AU (↑)**	**EU (↑)**
Fit	0.415	0.399	0.916	0.824
Underfit	0.002	-0.011	1.380	1.060
Overfit	0.327	0.305	1.040	0.936
Overfit-S	0.282	0.194	1.180	0.976

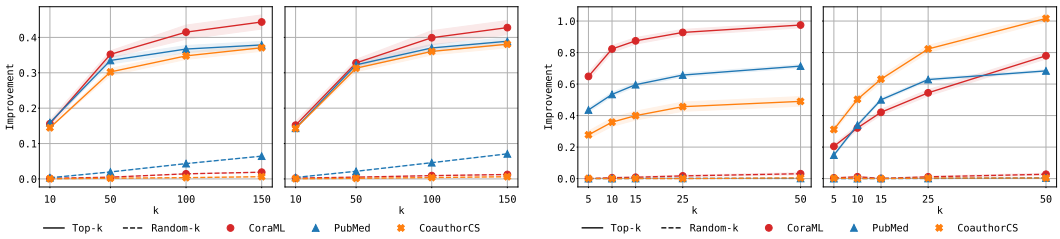


Figure 5: Improvement in the teacher's uncertainty when the top-k (solid) or random-k (dashed) node **features** are replaced (left) or **neighbors** are removed (right). The left panel of each plot shows aleatoric, and the right panel shows epistemic uncertainty improvements.

L Impact of k on Feature Attribution

k controls the top-k uncertain features (or nodes) that we replace (or delete) with those borrowed from certain nodes. Figs. 5- 9 study the impact of k [3]. As k increases, GraphUAT consistently yields higher improvements in uncertainty, while the random baseline remains nearly flat. Importantly, the gap between the two curves is large even at small k and widens as k grows, highlighting that GraphUAT reliably prioritizes the most uncertainty-contributing features.

Table 18: Spearman's rank correlation between the uncertainty attributions across GNN architectures on CoraML.

	Feature		Network	
Models	**AU (↑)**	**EU (↑)**	**AU (↑)**	**EU (↑)**
GCN-GAT	0.35 ± 0.19	0.33 ± 0.20	0.87 ± 0.26	0.64 ± 0.25
GCN-Sage	0.32 ± 0.20	0.30 ± 0.23	0.87 ± 0.26	0.60 ± 0.28
GAT-Sage	0.38 ± 0.18	0.34 ± 0.21	0.91 ± 0.22	0.72 ± 0.26

Table 19: Uncertainty attribution on the Mutagenicity graph classification dataset.

	Top-k (↑)	**Random (↑)**	**t-stat (↑)**	**p-value (↓)**	**Cohen's d_z (↑)**
AU	$\mathbf{0.62 \pm 0.01}$	0.00 ± 0.00	77.17	≈ 0	4.53
EU	$\mathbf{0.61 \pm 0.04}$	0.00 ± 0.00	13.80	≈ 0	6.90

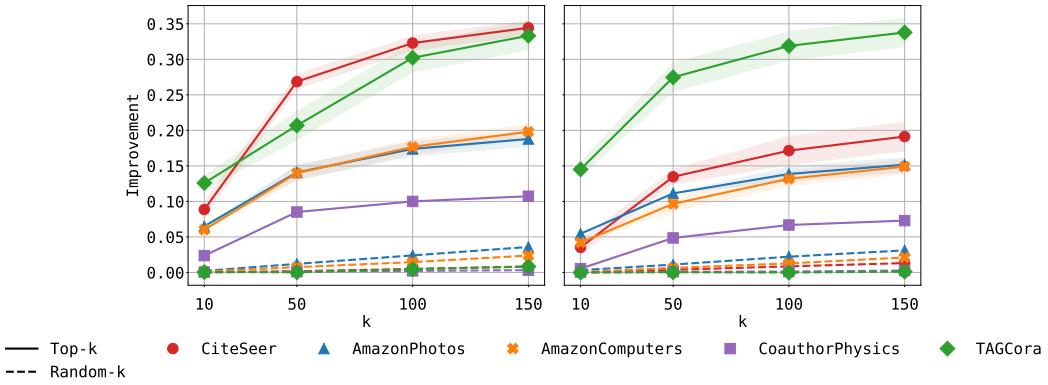


Figure 6: Improvement in the teacher model's uncertainty when the top-k (solid) or random-k (dashed) node **features** are replaced. The left panel shows aleatoric, and the right panel shows epistemic uncertainty improvements.

M Ablation Study on Loss Terms

The ablation study in Table 23 systematically examines the influence of the weighting coefficients λ and ω in the student's training objective (Eq. 6). Recall that λ controls the *posterior regularization* term, which enforces calibrated uncertainty, while ω balances the *prediction matching* and *uncertainty alignment* objectives. Setting $\lambda = 0$ effectively disables regularization, allowing us to assess its stabilizing role, whereas varying ω from 0 to 1 shifts the model's focus from matching the teacher's predictive probabilities to aligning with its uncertainty profile. The results show that intermediate values of ω (around 0.75) yield the best trade-off across metrics—achieving high Fidelity and Cohen's κ, low KL divergence, and strong rank correlation—indicating that both prediction and uncertainty alignment are necessary for faithful distillation. When ω is near the extremes (0 or 1), performance degrades, highlighting that overemphasizing either component leads to underfitting one aspect of the teacher's behavior. Similarly, small but non-zero values of λ improve consistency and prevent overconfident posteriors. Overall, these trends confirm that all three loss components contribute synergistically to effective knowledge transfer and calibrated uncertainty modeling.

N Effect of Training Data Size on Student Fidelity

Table 16 reports the student's fidelity to the teacher across different fractions of the available training data and for two values of ω. We observe that the student retains high fidelity even when trained on

[3]We plot ogbn-arxiv separately because its feature count is much smaller than the other datasets, so the probes have a larger relative impact on the uncertainty. Plotting all curves on the same axes would compress the others and make them difficult to distinguish.

Table 20: Quantitative evaluation of feature-level uncertainty attribution ($k = 100$) on different GNN architectures. Top-k and Random indicate the improvement observed in uncertainty following the feature replacements. ↑ and ↓ denote higher and lower is better, respectively.

Dataset	**Aleatoric Attribution**						**Epistemic Attribution**					
	Top-k (↑)	**Random (↑)**	**Δ (↑)**	***t*-stat (↑)**	***p*-value (↓)**	**Cohen's d_z (↑)**	**Top-k (↑)**	**Random (↑)**	**Δ (↑)**	***t*-stat (↑)**	***p*-value (↓)**	**Cohen's d_z (↑)**
CoraML-GAT	$\mathbf{0.243 \pm 0.01}$	0.002 ± 0.00	0.240	18.30	≈ 0	0.67	$\mathbf{0.214 \pm 0.01}$	0.006 ± 0.00	0.207	18.08	≈ 0	0.66
CoraML-GRAPHSAGE	$\mathbf{0.364 \pm 0.02}$	0.013 ± 0.00	0.351	19.17	≈ 0	0.70	$\mathbf{0.278 \pm 0.02}$	0.007 ± 0.00	0.271	17.13	≈ 0	0.63
CiteSeer-GAT	$\mathbf{0.168 \pm 0.01}$	0.002 ± 0.00	0.166	14.92	≈ 0	0.52	$\mathbf{0.148 \pm 0.01}$	0.001 ± 0.00	0.147	12.17	≈ 0	0.42
CiteSeer-GRAPHSAGE	$\mathbf{0.362 \pm 0.02}$	0.003 ± 0.00	0.359	22.07	≈ 0	0.77	$\mathbf{0.219 \pm 0.01}$	0.005 ± 0.00	0.214	14.88	≈ 0	0.52

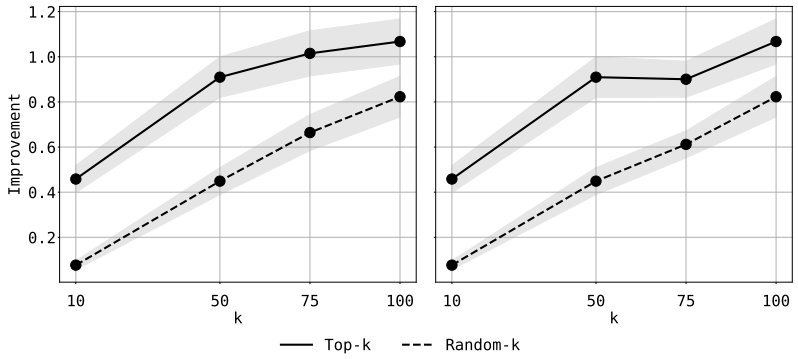


Figure 7: **ogbn-arxiv:** Improvement in the teacher model's uncertainty when the top-k (solid) or random-k (dashed) node **features** are replaced. The left panel shows aleatoric, and the right panel shows epistemic uncertainty improvements.

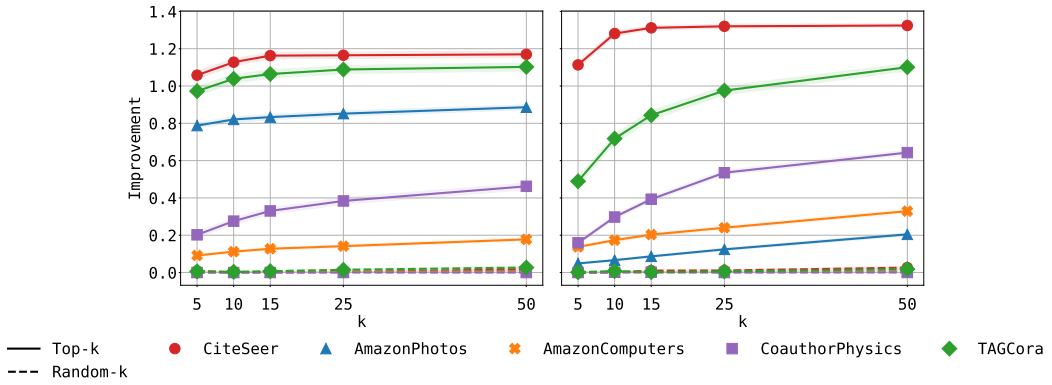


Figure 8: Improvement in the teacher model's uncertainty when the top-k (solid) or random-k (dashed) **neighbors** are removed. The left panel shows aleatoric, and the right panel shows epistemic uncertainty improvements.

substantially fewer samples. For instance, with only 50% of the training data, the fidelity remains at 0.90 for both ω settings, and even with 25% of the data it stays above 0.83. This indicates that the student does not require access to the full training set to effectively capture the teacher's behavior. These results suggest that the proposed distillation procedure is data-efficient, which is desirable in practical settings.

O Robustness to sub-optimally trained teachers

Table 17 evaluates uncertainty attribution when the teacher is not well-trained. Our objective is to remain faithful to the teacher's behavior rather than to the underlying data distribution, even when the teacher underfits or overfits. To simulate these regimes, we train the teacher to *underfit* by limiting training to 50 epochs, and to *overfit* by training for more epochs using only 10% of the available data.

Feature attribution: When the teacher underfits, feature-level attribution quality collapses to near-zero. This behavior is expected: an underfit model has not learned meaningful feature uncertainty relationships, leaving no coherent signal for GraphUAT to recover. Under overfitting, attribution quality improves but remains below that of a well-trained teacher (e.g., AU/EU of 0.28/0.19 versus 0.42/0.40 for a well-trained teacher), indicating that the teacher has learned some decision logic and GraphUAT can identify where that logic becomes uncertain.

Network attribution: In contrast, network-level attribution remains relatively strong under both underfitting and overfitting. Compared to the well-trained setting, underfitting yields improvements

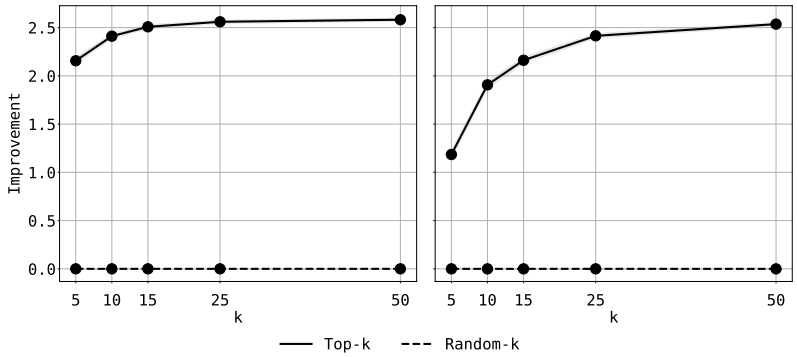


Figure 9: **ogbn-arxiv:** Improvement in the teacher model's uncertainty when the top-k (solid) or random-k (dashed) **neighbors** are removed. The left panel shows aleatoric, and the right panel shows epistemic uncertainty improvements.

Table 21: Validation of student's predictive performance against GAT and GraphSage

Dataset	Fidelity	Cohen's Kappa
CoraML-GAT	0.91 ± 0.28	0.89
CoraML-GRAPHSAGE	0.92 ± 0.26	0.91
CiteSeer-GAT	0.92 ± 0.28	0.90
CiteSeer-GRAPHSAGE	0.89 ± 0.32	0.86

of 50.6% (AU) and 28.6% (EU), while overfitting yields gains of 13.5% (AU) and 13.6% (EU). Under stronger overfitting, these gains further increase to 28.8% (AU) and 18.4% (EU). This difference from feature attribution arises because network-driven uncertainty diffusion is dictated primarily by graph topology. Well-connected neighbors dominate the uncertainty signal regardless of the teacher's training quality.

Overall, these results confirm that GraphUAT faithfully reflects the teacher's behavior: when the teacher lacks meaningful feature-level structure, attribution degrades accordingly, while more stable structural uncertainty patterns can still be recovered.

P Generalization to Other GNN Architectures

We evaluate the generalization capability of GraphUAT to GAT and GRAPHSAGE. Table 24 presents the results for uncertainty attribution arising from network effects, while additional experiments on student–teacher alignment and feature-level attribution are reported in Tables 20–21 in the Appendix.

Across both architectures, the results exhibit trends consistent with those observed for GCN. The uncertainty attributions generated by GraphUAT outperform random baselines, as reflected by statistically significant improvements in the evaluation metrics. Furthermore, the student models demonstrate strong alignment with their respective teachers, confirming that GraphUAT generalizes reliably beyond a single GNN architecture.

Q Sensitivity of Uncertainty Attribution to GNN Architecture

Table 18 reports the Spearman rank correlation between uncertainty attributions obtained from different GNN backbones on CoraML. We consider three commonly used architectures: GCN, GAT, and GRAPHSAGE, and measure agreement between their feature and network-level attributions.

We observe that agreement across architectures is only moderate for feature-level attribution (AU/EU correlations around 0.30-0.38), indicating that different backbones often identify different input features as primary sources of uncertainty. This suggests that feature-driven uncertainty is model-dependent and must be attributed separately for each trained architecture.

Table 22: Validation of student's modeling of uncertainty against GAT and GraphSage

Dataset	KL Divergence	Rank Correlation	p-value
CoraML-GAT	0.34	0.58	≈ 0
CoraML-GRAPHSAGE	0.28	0.56	≈ 0
CiteSeer-GAT	0.55	0.62	≈ 0
CiteSeer-GRAPHSAGE	0.47	0.65	≈ 0

Table 23: Ablation Study: Effect of λ and ω on various evaluation metrics. Best values (per dataset and metric) are in **bold**.

Dataset	λ	ω	CoraML					CiteSeer				
			Fidelity ↑	Cohen's ↑	KL Div. ↓	ρ ↑	p-value ↓	Fidelity ↑	Cohen's ↑	KL Div. ↓	ρ ↑	p-value ↓
Both	0.0	0.00	0.92	0.90	0.31	0.62	≈ 0	0.91	0.88	0.75	0.35	≈ 0
	0.0	0.25	0.94	0.93	0.19	0.69	≈ 0	0.92	0.90	0.43	0.67	≈ 0
	0.0	0.50	0.95	0.94	0.16	0.71	≈ 0	0.92	0.90	0.22	0.80	≈ 0
	0.0	0.75	0.94	0.93	0.12	0.75	≈ 0	0.92	0.90	0.14	0.81	≈ 0
	0.0	1.00	**0.95**	**0.94**	**0.10**	**0.77**	≈ 0	0.89	0.86	0.12	0.85	≈ 0
	10^{-4}	0.00	0.93	0.92	0.24	0.64	≈ 0	0.90	0.87	0.67	0.48	≈ 0
	10^{-4}	0.25	0.94	0.92	0.19	0.69	≈ 0	0.92	0.91	0.45	0.70	≈ 0
	10^{-4}	0.50	**0.95**	**0.94**	0.15	0.73	≈ 0	0.92	0.91	0.23	0.79	≈ 0
	10^{-4}	0.75	**0.95**	**0.94**	**0.10**	**0.77**	≈ 0	**0.93**	**0.91**	0.12	0.84	≈ 0
	10^{-4}	1.00	0.93	0.92	**0.10**	0.76	≈ 0	0.91	0.89	**0.10**	**0.86**	≈ 0

In contrast, network-level attributions exhibit higher agreement (correlations of 0.87–0.91 for AU and 0.60-0.72 for EU). This aligns with the intuition that topology-induced uncertainty is strongly governed by the underlying graph structure and therefore remains relatively stable across models.

These results support the need for uncertainty attribution as a model-specific procedure. Since different architectures can rely on distinct features and neighborhood aggregation behaviors, their sources of uncertainty also differ. Consequently, uncertainty attribution cannot be treated as a one-time step but must be performed separately for each deployed GNN model.

R Impact of distillation quality on uncertainty attribution

The quality of distillation affects the downstream attribution performance. A poorly distilled student fails to capture the target distribution, resulting in less trustworthy uncertainty estimates. However, this degradation is not silent, as the *fidelity* and ρ are generated alongside the attribution outputs, highlighting the trustworthiness of these numbers.

To illustrate this, Table 25 reports the attribution results for students checkpointed for three separate epochs of training, so that fidelity and ρ vary in a controlled way against the fully-trained result reported in the paper. For each block, the first two rows correspond to an unconverged student at an early and intermediate epoch, while the final row is the epoch at which our student's training actually stops.

S Framework agnostic validation of attributions

In addition to Shannon entropy, we evaluate the attributions using two alternative uncertainty metrics: the maximum logit and the Mahalanobis distance.

- **Max probability.** The complement of the largest softmax probability, so that lower confidence yields higher uncertainty:

$$u^{\text{teacher}}_{\text{maxprob}}(v) \;=\; 1 - \max_{c\in\{1,\dots,C\}} \tilde{y}^v_c. \tag{31}$$

- **Mahalanobis distance.** Using the teacher's penultimate-layer embedding $\mathbf{h}_v$, we fit class-conditional Gaussians with per-class means $\hat{\boldsymbol{\mu}}_c$ and a shared covariance $\hat{\boldsymbol{\Sigma}}$ on the training set. Each node is scored by its distance to the nearest class centroid:

$$u^{\text{teacher}}_{\text{Maha}}(v) \;=\; \min_{c\in\{1,\dots,C\}} \left(\mathbf{h}_v - \hat{\boldsymbol{\mu}}_c\right)^\top \hat{\boldsymbol{\Sigma}}^{-1} \left(\mathbf{h}_v - \hat{\boldsymbol{\mu}}_c\right). \tag{32}$$

Table 24: Quantitative evaluation of network-level uncertainty attribution ($k = 50$) on GAT and GRAPHSAGE.

Dataset	Aleatoric Attribution						Epistemic Attribution					
	Top-k (↑)	**Random** (↑)	**Δ** (↑)	***t*-stat** (↑)	***p*-value** (↓)	**Cohen's d_z** (↑)	**Top-k** (↑)	**Random** (↑)	**Δ** (↑)	***t*-stat** (↑)	***p*-value** (↓)	**Cohen's d_z** (↑)
CoraML-GAT	$\mathbf{0.832 \pm 0.02}$	0.020 ± 0.01	0.811	32.58	≈ 0	1.20	$\mathbf{0.778 \pm 0.02}$	0.004 ± 0.00	0.774	32.71	≈ 0	1.20
CoraML-GRAPHSAGE	$\mathbf{0.494 \pm 0.03}$	0.015 ± 0.01	0.479	16.95	≈ 0	0.62	$\mathbf{0.761 \pm 0.02}$	0.020 ± 0.01	0.738	30.52	≈ 0	1.12
CiteSeer-GAT	$\mathbf{1.208 \pm 0.02}$	0.023 ± 0.01	1.185	54.34	≈ 0	1.90	$\mathbf{1.437 \pm 0.01}$	0.010 ± 0.00	1.423	126.00	≈ 0	4.40
CiteSeer-GRAPHSAGE	$\mathbf{0.947 \pm 0.02}$	0.021 ± 0.01	0.926	36.45	≈ 0	1.27	$\mathbf{1.297 \pm 0.01}$	0.016 ± 0.01	1.280	90.70	≈ 0	3.16

Table 25: Evaluation of uncertainty attribution for under-distilled students.

Dataset	Fidelity	Δ-AU	Δ-EU	ρ (p-value)
CoraML	0.29	0.28 ± 0.02	0.51 ± 0.02	0.24 ($\sim$0)
CoraML	0.65	0.49 ± 0.02	0.52 ± 0.02	0.38 ($\sim$0)
CoraML	0.95	0.98 ± 0.02	0.78 ± 0.02	0.72 ($\sim$0)
CiteSeer	0.53	0.64 ± 0.02	0.81 ± 0.02	0.22 ($\sim$0)
CiteSeer	0.70	0.66 ± 0.02	0.86 ± 0.01	0.27 ($\sim$0)
CiteSeer	0.83	1.17 ± 0.02	1.32 ± 0.01	0.70 ($\sim$0)
Amazon-P	0.45	0.23 ± 0.01	0.14 ± 0.01	0.01 ($\sim$0)
Amazon-P	0.68	0.42 ± 0.01	0.17 ± 0.01	0.35 ($\sim$0)
Amazon-P	0.96	0.89 ± 0.01	0.20 ± 0.01	0.65 ($\sim$0)

Larger distances indicate that v lies farther from all class-conditional distributions, reflecting higher, predominantly epistemic, uncertainty.

Tables 26 and 27 show that GraphUAT performs consistently across these alternative uncertainty metrics, with significant improvements over the random baseline for both aleatoric and epistemic attributions on every dataset.

T Post-intervention classification behavior

The counterfactual protocol measures how much the teacher's predictive entropy falls after a top-k intervention. Replacing features or removing neighbors could, in principle, reduce entropy by pushing the teacher toward a different class. In that case the drop would reflect a changed decision rather than a resolved source of uncertainty. To rule this out, we record for every node subjected to a top-k intervention whether the teacher's predicted class after the intervention matches its original prediction. Table 28 reports the fraction of predictions that remain unchanged.

Across all 32 dataset-uncertainty profile combinations, an average of 98.8% of predictions remain unchanged, and the lowest value in any combination is 94.7% (CiteSeer, feature-level EU). The interventions therefore leave the teacher's decision essentially intact while substantially reducing its predictive entropy. This indicates that the uncertainty reductions reported throughout the paper come from resolving the sources GraphUAT identifies, not from flipping the teacher's prediction to another class.

Table 26: Network attribution evaluated with the max-probability uncertainty.

Dataset	Aleatoric Attribution						Epistemic Attribution					
	Top-k (↑)	**Random** (↑)	Δ (↑)	t-**stat** (↑)	p-**value** (↓)	**Cohen's** d_z (↑)	**Top-k** (↑)	**Random** (↑)	Δ (↑)	t-**stat** (↑)	p-**value** (↓)	**Cohen's** d_z (↑)
CoraML	**0.345 ± 0.01**	0.008 ± 0.01	0.336	43.22	≈ 0	1.59	**0.256 ± 0.01**	0.009 ± 0.00	0.247	29.39	≈ 0	1.08
CiteSeer	**0.386 ± 0.01**	0.006 ± 0.00	0.380	54.68	≈ 0	1.90	**0.409 ± 0.01**	0.005 ± 0.00	0.403	62.28	≈ 0	2.16
Amazon-P	**0.286 ± 0.01**	0.002 ± 0.00	0.283	51.47	≈ 0	1.17	**0.069 ± 0.00**	0.003 ± 0.00	0.065	21.58	≈ 0	0.49

Table 27: Network attribution evaluated with the Mahalanobis-distance uncertainty.

Dataset	Aleatoric Attribution						Epistemic Attribution					
	Top-k (↑)	**Random** (↑)	Δ (↑)	t-**stat** (↑)	p-**value** (↓)	**Cohen's** d_z (↑)	**Top-k** (↑)	**Random** (↑)	Δ (↑)	t-**stat** (↑)	p-**value** (↓)	**Cohen's** d_z (↑)
CoraML	**4.220 ± 0.31**	0.154 ± 0.07	4.065	12.57	≈ 0	0.46	**5.367 ± 0.28**	0.017 ± 0.10	5.350	17.42	≈ 0	0.64
CiteSeer	**8.114 ± 0.25**	0.114 ± 0.06	7.999	30.96	≈ 0	1.08	**8.376 ± 0.18**	0.079 ± 0.04	8.298	45.27	≈ 0	1.57
Amazon-P	**5.865 ± 0.27**	0.071 ± 0.02	5.794	21.01	≈ 0	0.48	**3.163 ± 0.24**	0.067 ± 0.02	3.095	13.01	≈ 0	0.30

Table 28: Fraction of teacher predictions left unchanged by top-k interventions: feature replacement (Feature) and neighbor removal (Network).

	Feature		**Network**	
Dataset	**AU**	**EU**	**AU**	**EU**
CoraML	1.000	0.998	0.983	0.991
CiteSeer	0.992	0.947	0.970	0.973
Amazon-P	0.998	0.999	0.995	0.998
Amazon-C	0.992	0.997	0.987	0.993
PubMed	0.998	0.999	0.971	0.993
Coauthor-CS	0.999	0.999	0.991	0.994
Coauthor-P	1.000	0.996	0.983	0.996
TAGCora	0.956	0.994	0.974	0.975
Mean	0.992	0.991	0.982	0.989

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: §3, 4, Appendices.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: §5, App. B

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: App. F

   Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: App. G, G.1, G.2

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: §4 provides the link to our anonymized GitHub repository.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.

- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: App. G.2, G.1, M

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Tables 16, 2, 3, 5, 6, 8

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: App. G, G.2

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: 1) We do not anticipate significant negative societal impacts specific to this work beyond those generally associated with machine learning systems. 2) Our study does not involve human subjects. 3) For dataset documentation, please see App. G.1

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Positive impacts are discussed in section 1. We do not anticipate significant negative societal impacts specific to this work beyond those generally associated with machine learning systems.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This work relies exclusively on publicly available datasets App. G.1 and open-source models Tab. 19. We do not release any new datasets, scraped data, or pretrained models, and thus pose no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: App. G.1, G.2

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We release our code as an anonymized repository §4. The repository includes documentation and instructions for reproducing the experiments described in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This study does not involve crowdsourcing experiments or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This study does not involve human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Our LLM usage was limited to paper writing and help with standard code. The LLM does not play a core methodological part in our research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.