Conformal Prediction Sets for Graph Neural Networks

Soroush H. Zargarbashi [1] **Simone Antonelli** [1] **Aleksandar Bojchevski** [2]

Abstract

Despite the widespread use of graph neural networks (GNNs) we lack methods to reliably quantify their uncertainty. We propose a conformal procedure to equip GNNs with prediction *sets* that come with distribution-free guarantees – the output set contains the true label with arbitrarily high probability. Our post-processing procedure can wrap around any (pretrained) GNN, and unlike existing methods, results in meaningful sets even when the model provides only the top class. The key idea is to diffuse the node-wise conformity scores to incorporate neighborhood information. By leveraging the network homophily we construct sets with comparable or better efficiency (average size) and significantly improved singleton hit ratio (correct sets of size one). In addition to an extensive empirical evaluation, we investigate the theoretical conditions under which smoothing provably improves efficiency.

1. Introduction

From health to traffic forecasting, graph neural networks (GNNs) have become a fundamental building block in a variety of applications. Even though their versatility places them in the spotlight among other machine learning topics, they seldom provide reliable uncertainty estimates. Since test accuracy is not necessarily a trustworthy indicator of performance, it is essential to explicitly quantify the model uncertainty for different inputs, especially in safety-critical domains. Naively considering the predicted distribution over labels (e.g. from the softmax) does not produce a good estimate of the true conditional probability $p(y \mid \boldsymbol{x})$ since models are often overconfident and uncalibrated (Guo et al., 2017; Hein et al., 2019). Most uncertainty quantification methods are computationally expensive, and/or require modifications to the model architecture or at least retraining of the model (Hüllermeier & Waegeman, 2021; Abdar et al., 2021). Moreover, most techniques rely on the i.i.d. assumption, which is clearly violated for node classification due to the interdependence between nodes. Hence, methodological contributions in this direction are often incompatible with graph-based models such as GNNs (Stadler et al., 2021).

Conformal prediction (CP) is a promising paradigm for constructing prediction sets (or intervals in the case of regression) with a statistically sound coverage guarantee – the output set covers the true label with any user-specified probability. CP is distribution free and it relies on exchangeability, i.e. the only assumption is that every permutation of the instances (in our case the nodes) is equally likely. In other words, we assume that the indexing of the random variables is immaterial. This makes CP a prime candidate for uncertainty quantification on graphs since exchangeability relaxes the i.i.d. assumption. In § 3 we discuss in detail the settings (e.g. transductive vs. inductive) under which this assumption is satisfied for semi-supervised node classification. More generally, we prove that semi-supervised learning with (subset-) permutation-equivariant models preserves exchangeability. Interestingly, even in cases where exchangeability may be violated, it is still possible to provide strong guarantees while incurring a coverage penalty that is proportional to the degree of distribution shift (Barber et al., 2022).

Although full conformal prediction has a significant computational cost, *split conformal prediction* is fast, easy to implement, and model and data-distribution independent (Vovk et al., 2005; Shafer & Vovk, 2008). Since it uses the model as a black box, there is no need to retrain or modify it. Along with a provable coverage guarantee, the sets are interpretable and can be used to communicate with non-expert stakeholders, making CP readily applicable to different domains like medicine (Vazquez & Facelli, 2022), electricity market forecasting (Kath & Ziel, 2021), and robotics (Luo et al., 2023). Contrary to full conformal, split conformal prediction sacrifices statistical efficiency for computational efficiency, while there are extensions that sit in the middle of this tradeoff, e.g. cross-conformal prediction (Vovk, 2015), and CV+/Jackknife+ (Barber, 2020; Barber et al., 2021). We focus on the split conformal setting, but our diffusion-based approach can be extended to CV+/Jackknife+.

[1]CISPA Helmholtz Center for Information Security [2]University of Cologne. Correspondence to: Soroush H. Zargarbashi <sayed.haj-zargarbashi@cispa.de>, Simone Antonelli <simone.antonelli@cispa.de>, Aleksandar Bojchevski <a.bojchevski@uni-koeln.de>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

An important ingredient of CP is the conformity score $s(\boldsymbol{x}, y)$ which quantifies the agreement between an observation $\boldsymbol{x}$ and a candidate label y. While the coverage guarantee holds for any scoring function s, the output sets are more efficient (i.e. smaller on average) the closer s is able to track the true conditional label distribution (see § 4 for a detailed discussion). Our key insight is that the network structure for homophilous graphs contains valuable information which we leverage to refine the node-wise conformity scores. Specifically, our main contributions are:

- A method called Diffusion Adaptive Prediction Sets (DAPS) to smooth node-wise conformity scores resulting in prediction sets with comparable or better efficiency and significantly improved singleton hit ratio.
- Theoretical insights into when smoothing is beneficial, and a rigorous discussion of graphs and exchangeability.
- The first thorough empirical evaluation of conformal prediction for transductive node classification.

In contrast to existing baselines, our method produces meaningful sets even without access to the class distribution. This considerably expands its applicability, e.g. to cloud-based models that only provide a prediction. DAPS is effective and simple – which we argue is its biggest strength.

2. Background

First, we review the concept of standard conformal prediction (as applied to e.g. image classifiers) without considering any additional structure like network homophily. We also cover the current state-of-the-art conformity scores.

Let $\pi(\boldsymbol{x}) \in \triangle^K$ be the distribution over $K = |\mathcal{Y}|$ class labels predicted by some classifier f (pre-)trained on $\mathcal{D}_{\text{train}}$. For example, $\pi(\boldsymbol{x}) = \sigma(f(\boldsymbol{x}))$ where σ is the softmax applied on the last layer of a neural network f, but any classifier that outputs a distribution is applicable. Given access to calibration data $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_n, y_n)\}$ we can construct a prediction set $\mathcal{C}(\boldsymbol{x}_{n+1}) \subseteq \mathcal{Y}$ for an unseen test example $\boldsymbol{x}_{n+1}$ with the following coverage guarantee $\mathbb{P}\left[y_{n+1} \in \mathcal{C}(\boldsymbol{x}_{n+1})\right] \geq 1 - \alpha$, where α is the user-specified significance level. The only assumption is that $\mathcal{D}_{\text{cal}} \cup (\boldsymbol{x}_{n+1}, y_{n+1})$ is exchangeable.

Theorem 1 (Vovk et al. (2005)). *Let $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n+1}$ be exchangeable. For any score function $s : \mathcal{X} \times \mathcal{Y} \mapsto \mathbb{R}$ and any significance level $\alpha \in (0, 1)$, define quantile $\hat{q} := \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(x_i, y_i)\}_{i=1}^{n}\right)$ and prediction sets as $\mathcal{C}_\alpha(\boldsymbol{x}_{n+1}) = \{y : \ s(\boldsymbol{x}_{n+1}, y) \geq \hat{q}\}$. We have*[1]

$$1 - \alpha + \frac{1}{(n+1)} \geq \mathbb{P}\left[y_{n+1} \in \mathcal{C}(\boldsymbol{x}_{n+1})\right] \geq 1 - \alpha \quad (1)$$

[1]The upper bound holds when there are no ties between the scores, but in practice, ties are broken by adding random noise.

The conformity score function $s(\boldsymbol{x}, y)$ quantifies the agreement between an observation $\boldsymbol{x}$ and a candidate label y[2].

Theorem 1 provides a *marginal* coverage guarantee that holds true on average for all $\boldsymbol{x}$. It has been shown that without strong unrealistic assumptions, coverage guarantees conditional on a given $\boldsymbol{x}$ are impossible (Vovk, 2012; Barber et al., 2019). Additionally, with $l = \lfloor (n+1)\alpha \rfloor$, Vovk (2012) shows that the coverage follows a Beta distribution

$$\mathbb{P}\left[y_{n+1} \in \mathcal{C}_\alpha(\boldsymbol{x}_{n+1}) \mid \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}\right] \sim \mathbf{Beta}(n+1-l, l)$$

This means that if we resample the calibration set, the empirical coverage on the test set will be centered on $1 - \alpha$. Two conclusions are directly implied: (i) the number of calibration samples has an effect on the concentration (and the variance) of the coverage probability (and other metrics), and (ii) the coverage is also upper-bounded, which in trivial cases where α is smaller than the model's accuracy, may lead to systematic miscoverage (for details see § 6.1).

Conformity scores. An obvious idea for the conformity score is $s(\boldsymbol{x}, y) := \pi(\boldsymbol{x})_y$ where $\pi(\boldsymbol{x})_y$ is the predicted probability for class y, which is known as threshold prediction sets (TPS) (Sadinle et al., 2018). However, this scoring method has the tendency to undercover hard examples and overcover trivial ones (Angelopoulos & Bates, 2021). Hence, a popular alternative is the *adaptive* prediction sets (APS) (Romano et al., 2020) method. Assuming we have access to an oracle, let $p(y \mid \boldsymbol{x})$ be the ground-truth conditional label distribution. We can form $\mathcal{C}_\alpha$ by including classes one by one, from the most likely class to the least likely, until the cumulative probability becomes $> 1-\alpha$. This is the motivation behind APS. In place of the oracle, APS uses the estimated $\pi(\boldsymbol{x})$ defining $s(\boldsymbol{x}, y) := -\left(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y\right)$ where $\rho(\boldsymbol{x}, y) := \sum_{c=1}^{K} \pi(\boldsymbol{x})_c \mathbb{1}\left[\pi(\boldsymbol{x})_c > \pi(\boldsymbol{x})_y\right]$ is the sum of all classes predicted as more likely than y, and $u \in [0, 1]$ is a uniform random value that breaks potential ties between different scores (Stutz et al., 2022)[3].

One drawback of APS is that it results in large sets. To overcome this, Angelopoulos et al. (2021) propose a regularization approach, called *regularized* adaptive prediction sets (RAPS), penalizing labels that are less likely, and thus encouraging smaller sets. Formally, let $o(\boldsymbol{x}, y) := |\{c \in \mathcal{Y} : \pi(\boldsymbol{x})_y \geq \pi(\boldsymbol{x})_c\}|$ be the rank of y, the proposed score is $s(\boldsymbol{x}, y) = -\left(\rho(\boldsymbol{x}, y) + u \cdot \pi(\boldsymbol{x})_y + \nu \max\left(o(\boldsymbol{x}, y) - k, 0\right)\right)$, where ν and k are hyperparameters. Intuitively, the regularization term penalizes classes that are at the bottom of the rank (after k) proportionally to ν, so to be selected for the predictive set, a lower quantile is needed.

[2]Conformity scores can equivalently be defined as measuring the nonconformity (disagreement).

[3]This randomization helps achieve an exact $1 - \alpha$ coverage.

3. Graphs and Exchangeability

Let $G = (\boldsymbol{X}, \boldsymbol{A})$ be a graph where $\boldsymbol{X}$ is the matrix of node features and $\boldsymbol{A}$ the adjacency matrix. Let $\mathcal{V}_l$ and $\mathcal{V}_u$ be disjoint sets of labeled and unlabeled nodes and $\mathcal{V} = \mathcal{V}_l \cup \mathcal{V}_u$. In all settings we split $\mathcal{V}_l = \mathcal{V}_d \cup \mathcal{V}_c$ into a disjoint development (training + validation) $\mathcal{V}_d$ and a calibration $\mathcal{V}_c$ set. We discuss three settings, focusing mostly on the first.

Transductive case.[4] Here the model has access to the entire graph during training, calibration, and testing. We assume an arbitrary fixed graph and that the union of calibration and unlabeled nodes $\mathcal{V}_c \cup \mathcal{V}_u$ is exchangeable. The set of development nodes $\mathcal{V}_d$ may have arbitrary dependencies. In our experiments, we sample all of the labeled nodes $\mathcal{V}_l$ uniformly at random so the exchangeability of $\mathcal{V}_c \cup \mathcal{V}_u$ is satisfied by construction since any node has an equal chance to land in $\mathcal{V}_c$ or $\mathcal{V}_u$. It is plausible that in a real-world application, the labeling budget is randomly allocated, but any other exchangeable sampling strategy is permitted. Importantly, while the classifier has access to the node features and the neighborhood structure for all nodes in the calibration set $\mathcal{V}_c$, their labels are *not* revealed during training. In other words, the classifier itself (and thus the score) cannot distinguish between calibration $\mathcal{V}_c$ and unlabeled $\mathcal{V}_u$ nodes.

Simultaneous inductive case. This setting is identical to the transductive case except the classifier is trained only on the subgraph induced by $\mathcal{V}_d$. The rest of the graph, including the calibration and the unlabeled test nodes, are simultaneously revealed after training and before calibration.

Inductive case. The classifier is again trained only on the subgraph induced by $\mathcal{V}_d$. We calibrate on the extended subgraph induced by $\mathcal{V}_d \cup \mathcal{V}_c$. The rest of the unlabeled test nodes may arrive either one at a time or in batches.

Exchangeability. We show that the conformity scores from a transductive (and simultaneous inductive) semi-supervised GNN are exchangeable. With $s(v, y \mid \boldsymbol{X}, \boldsymbol{A}) = s(v, y)$ denote the score for node v and a candidate label y, which may depend on all other nodes. We omit $\boldsymbol{X}$ and $\boldsymbol{A}$ for brevity.

Proposition 1. *Assume that $\mathcal{V}_c \cup \mathcal{V}_u$ is exchangeable. Let $\pi(G) = \boldsymbol{\Pi} \in \triangle^{|\mathcal{V}| \times K}$ be a matrix where row v is the label distribution for node v predicted by any permutation equivariant GNN classifier $\pi(\cdot)$ trained on the entire graph G and only using labels for nodes in $\mathcal{V}_d$. Then the scores $s(v, y) = \boldsymbol{\Pi}_{vy}$ where $\boldsymbol{\Pi}_{vy}$ is the predicted probability for node v and class y, are exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

All omitted proofs are provided in § C. The gist here is that exchangeability is preserved since the classifier is permutation equivariant and does not distinguish between calibration $\mathcal{V}_c$ and unlabeled $\mathcal{V}_u$ nodes. Proposition 1 implies that the APS and RAPS scores are also exchangeable in this setting. More importantly, it implies that conformal prediction is applicable to node classification and that the coverage guarantee must hold. Our experimental evaluation confirms this since we can always obtain the desired coverage. Proposition 1, can also be trivially extended to the general transductive semi-supervised setting (e.g. on images) as long as permutation equivariance is satisfied.

Beyond exchangeability. In the inductive case exchangeability is violated whenever the conformity scores for calibration nodes are affected by a change in the graph. Specifically, as soon as a test node becomes part of the receptive field of any calibration node (e.g. its 2-hop neighborhood for a 2-layer GNN). Nonetheless, even in this case conformal prediction can still provide coverage guarantees, incurring only a penalty on the coverage that is proportional to the magnitude of the distribution shift as measured by the total variation distance (see Barber et al. (2022) for more details). We consider this setting in § E.10. Clarkson (2022) uses the same approach to adapt conformal prediction for inductive node classification assuming exchangeability to be violated in both inductive and transductive settings. Although this assumption is correct for the inductive scenario, our Proposition 1 shows that exchangeability is not violated in the transductive case.

Sparsity. We are often interested in the sparsely-labeled setting where we have access to a relatively small set of labeled nodes for training, validation, and now calibration. Using an unrealistically large validation set, e.g. larger than the training set, is one pitfall that can skew the evaluation results of GNNs (Shchur et al., 2018). Moreover, if we happen to have access to a large labeled set it is probably more effective to use it for training than validation. These concerns equally apply to the calibration set. Thus, under label scarcity one reasonable strategy is to split the labeled nodes into training, validation and calibration sets of the *same size*. Since most graphs are sparse themselves, sparsity leads to a serious issue for the NAPS approach proposed by Clarkson (2022). Adapting the weighted variant of CP from Barber et al. (2022), NAPS assigns a weight of one to adjacent nodes, and a weight of zero otherwise. Now, due to the two sources of sparsity, many of the unlabeled test nodes have no calibration nodes in their immediate neighborhood. This means we cannot form any valid prediction sets for them (since all weights are zero), which happens for up to 79% of nodes as we show in § 6.1. This problem persist regardless of whether we are in the inductive or transductive setting. Our approach does not suffer from the same issue and we can always obtain valid predictions for all nodes.

[4] In the conformal prediction literature, full conformal prediction is sometimes referred to as transductive, while split conformal is referred to as inductive. This is orthogonal to the use of the terms transductive and inductive in semi-supervised node classification where they indicate which part of the graph is seen during training. In this paper, we mostly focus on split conformal prediction and GNNs that are trained in a transductive manner.

Graph generative models and exchangeability. There is a rich body of work on exchangeability and graph generative models such as the Stochastic Block Model (SBM) (Holland et al., 1983). There we can make a distinction between node-exchangeable models like the SBM that generate either dense or empty graphs with probability one (Lloyd et al., 2012), and edge-exchangeable models (Cai et al., 2016) that can exhibit sparsity. Since most real world graphs are sparse one can conclude that edge-exchangeability is a more reasonable assumption. However, this literature is orthogonal to our work since it concerns *generative* models, while our focus is on the transductive node classification setting where we assume an arbitrary given graph. Here, our exchangeability assumption is w.r.t. the node labels, regardless of how the features vectors and the graph structure is generated. Since we sample the set of calibration nodes uniformly at random, exchangeability is satisfied by construction.

4. Properties of Conformal Scores

Efficiency. On the surface, conformal prediction seems to sidestep the need for direct uncertainty quantification where the goal is to provide calibrated probability estimates, making it a convenient alternative. However, CP is highly dependent on the choice of the scoring function, which in turn depends on how well the unknown conditional label distribution is approximated by the model. Even if we can obtain a good approximation, all scoring functions are not created equal. As shown by Romano et al. (2020), assuming an oracle model that returns the true $p(y \mid \boldsymbol{x})$, the scores produced by APS provide the *smallest possible* sets that satisfy the conditional coverage guarantee. We conjecture an interesting implication of this result that, to the best of our knowledge, has not been discussed before: we can use efficiency to compare models. Let S_α^f be the average set size at significance α using APS and probabilities estimated by some model f. Similarly, let S_α^g be the average set size for model g. If it holds $S_\alpha^f < S_\alpha^g$ for all α, i.e. CP on top of the model f always produces more efficient sets, then we can conclude that it is likely that f better approximates the oracle model. In § B we discuss how to estimate the efficiency without having to explicitly compute it on the test set, and we use this insight to select the calibration hyperparameters.

Score distribution. To understand the effect of different scoring functions, we examine the distribution of conformity scores. In Fig. 1, we show the distribution for APS, RAPS and DAPS (introduced in § 5), where True denotes the scores for ground-truth labels in the calibration set and False denotes all other scores. We see that the penalty term in RAPS causes the distribution of low-ranked scores to concentrate on $K - k$ locations. This makes RAPS unstable since a small shift in the quantile threshold (e.g. due to sampling) can have a large effect on the outcome, causing

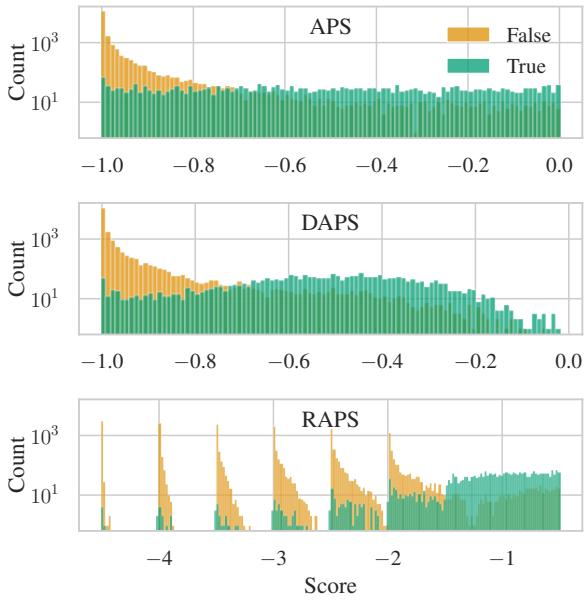

Figure 1. Histogram of conformity scores for ground-truth labels (True) and all other labels (False) on `CoraML` for `GCN`.

RAPS to have a large variance, which we experimentally verify (see § 6). Moreover, Einbinder et al. (2022) show that for the oracle, the (conditional) distribution of true adaptive conformity scores is uniform. RAPS scores strongly deviate from this ideal case implying that it would be difficult to conclude how well the predicted sets capture the true $p(y \mid \boldsymbol{x})$. Diffused scores deviate from distribution mildly.

Evaluation metrics and the singleton hit ratio. In the conformal prediction literature, the most commonly used metric to compare different methods is efficiency (assuming valid coverage). We argue that another important metric is the singleton hit ratio defined as the fraction of examples with a *correct* prediction set of size one, i.e. singletons that contain the true label. For example, in a real-world application it is reasonable to automatically process all singleton predictions – simply predict the single class. However, a predicted set of size ≥ 1 might trigger inspection by a human expert who would have to decide how to handle the uncertain observation. Therefore, maximizing the singleton hit ratio would minimize the inspection effort in this example. As we will show in § 6, our approach significantly improves the singleton hit ratio even though it is not specifically designed to do so. It is also important to note that blindly optimizing for efficiency may not always be a good idea. Observations that are truly aleatorically uncertain should indeed have larger sets.

5. Diffused Adaptive Prediction Sets

Our proposed DAPS exploits the graph structure by updating the node-wise conformal scores $s(v, y)$ based on

neighborhood diffusion. We define the diffused score as

$$\hat{s}(v, y) = (1 - \lambda)s(v, y) + \frac{\lambda}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} s(u, y) \quad (2)$$

where $\mathcal{N}_v$ is set of v's neighbors, and λ is a diffusion parameter. Practically, given the matrix of node-wise scores $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times K}$ the neighborhood diffused scores are $\hat{\boldsymbol{H}} = (1 - \lambda)\boldsymbol{H} + \lambda \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{H}$ where $\boldsymbol{D}$ is the degree matrix. We show that diffusion preserves exchangeability.

Proposition 2. *Let $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times K}$ be any matrix where row v is the conformal scores for all classes y for node v, and let $\boldsymbol{H}$ be exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. Then the diffused scores $\hat{\boldsymbol{H}}$ are also exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

5.1. Homophily and Theoretical Benefits of Diffusion

The predicted label distribution approximates the ground truth. To understand when diffusion is beneficial, we compare the approximation error before and after diffusion.

Theorem 2. *Let $\boldsymbol{\pi}_i$ be the model's approximation of the ground-truth conditional probability vector $\boldsymbol{p}_i$, and let the diffused distribution be $\hat{\boldsymbol{\pi}}_i = (1 - \lambda)\boldsymbol{\pi}_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j$. Assume that the $G = (\boldsymbol{X}, \boldsymbol{A})$ is constructed such that $\boldsymbol{A}_{ij} = 1$ iff $||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq \Delta$ where $||\cdot||$ is the total variation norm. Diffusion improves the approximation error $\epsilon_i = ||\boldsymbol{\pi}_i - \boldsymbol{p}_i||$, i.e. $||\hat{\boldsymbol{\pi}}_i - \boldsymbol{p}_i|| < \epsilon_i$ if $\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \Delta < \epsilon_i$.*

Assuming that the graph is constructed in a homophilous manner – here edges are formed only between nodes with ground-truth distributions closer than some Δ – Theorem 2 shows that diffusion helps whenever the average approximation error in the neighborhood plus a Δ penalty is smaller than the node's own error. Diffusion is beneficial since a better approximation of $\boldsymbol{p}_i$ leads to more efficient sets.[5]

Efficiency is affected by more than just the model accuracy. We can amplify or decrease the absolute probabilities without affecting the accuracy by e.g. rank-preserving transformations such as temperature scaling or uniform perturbations (see § D for a discussion). Moreover, in § 6 we show that diffusion has a negligible (positive) impact on accuracy while significantly improving efficiency and singleton hit ratio. This shows that diffusion does not change the most likely label, but rather refines the distribution of labels.

Simulation. To illustrate the effect of diffusion we create synthetic data where the true label distribution is known. Then we simulate approximation errors by perturbing the nodes – each node is randomly perturbed with noise which either has a large magnitude with probability $p_s = 0.20$ or small magnitude with probability $1 - p_s$ (see § D for more details). The graph is constructed for $\Delta = 0.4$. In Fig. 2 we see that the diffused probability vectors can correct the introduced perturbations. Intuitively, if the neighborhood of a node is mostly unperturbed, its probability vector can be reconstructed from its neighbors as long as their probability vectors are similar enough (i.e. we have homophily).

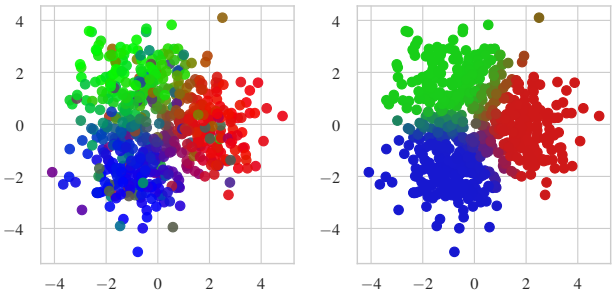

Figure 2. Diffusion (right) corrects the perturbed synthetic data (left). The RGB color shows the probability of each class.

5.2. Generalizations of Neighborhood Diffusion

We generalize diffusion beyond the 1-hop neighborhood. For instance, we can incorporate the k-hop neighbors as $\hat{\mathbf{H}} = \lambda_0 \mathbf{H} + \sum_{i=1}^{k} \lambda_i (\boldsymbol{D}^{-1}\boldsymbol{A})^i \times \mathbf{H}$. We investigate the 2-hop variant with parameters λ_0 and λ_1. Inspired by label propagation (LP), we define another variant which we call *score propagation (SP)*, where each node propagates its scores to its neighbors. We define the iterative score update as $\hat{\boldsymbol{H}}^{(t)} = (1 - \lambda)\hat{\boldsymbol{H}}^{(0)} + \lambda \hat{\boldsymbol{H}}^{(t-1)} \boldsymbol{D}^{-1}\boldsymbol{A}$, where $\hat{\boldsymbol{H}}^{(0)} = \boldsymbol{H}$ is the initial node-wise score, $\lambda \in (0, 1)$ can be interpreted as the teleport probability in the corresponding random walk, and $\boldsymbol{D}^{-1}\boldsymbol{A}$ is the degree-normalized adjacency matrix. Similar to LP, the close-form solution is $\hat{\boldsymbol{H}} = (1 - \lambda)\left(\boldsymbol{I} - \lambda \boldsymbol{D}^{-1}\boldsymbol{A}\right)^{-1} \boldsymbol{H}$. In practice, we do not perform the matrix inverse and run $t = 10$ iterations which is enough for convergence. Note, since both of these transformations are permutation equivariant it is easy to see that they also preserve exchangeability like the 1-hop variant (see proof of Proposition 2). In § 6 we show that the 2-hop and SP variants improve over the 1-hop variant, however, for simplicity, in most experiments we focus on the latter. For code and computational complexity analysis see § A.

5.3. Hard Predictions

In some cases, the model outputs only the most likely label and there is no information on the predicted distribution of labels. For example, cloud-based models may provide such hard predictions on purpose for privacy protection since this makes membership inference attacks more difficult. In this case, $\pi(\boldsymbol{x})_y = 1$ for the predicted label, and 0 for

[5] The result in Theorem 2 is about diffusion in probability space, while Eq. 2 performs diffusion on the scores. We investigate both variants in § E.8. Moreover, it's easy (but notationally more cumbersome) to derive a similar result for the score diffusion.

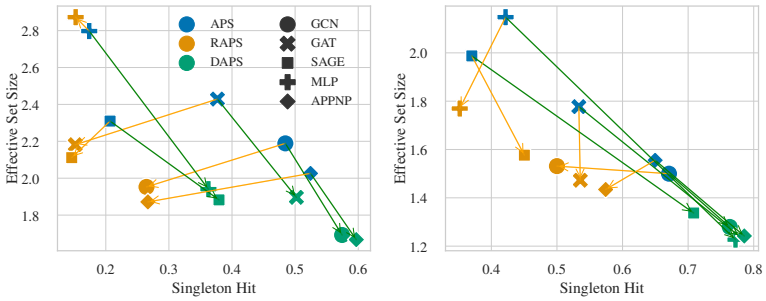

Figure 3. Comparing efficiency and singleton hit ratio for all models on `CoraML` (left) and `CoauthorPhysics` (right) datasets with adaptive coverage.

Dataset	RAPS Eff ↓	RAPS SH ↑	DAPS Eff ↓	DAPS SH ↑
CoraML	-0.24	-0.22	-0.50	0.09
PubMed	0.07	-0.13	0.02	-0.01
CiteS	0.08	-0.24	-0.40	0.10
Co-CS	-0.93	-0.09	-1.03	0.13
Co-Phy	0.03	-0.17	-0.22	0.09
Amz-C	-0.35	-0.16	-0.77	0.19
Amz-P	-0.02	-0.23	-0.48	0.16

Table 1. Performance relative to APS across all small datasets for `GCN` model. DAPS is best overall.

all other labels. Therefore, the (empirical) distribution of scores computed by APS will degenerate with all scores concentrated on one of two locations, introducing many ties between the scores. While the coverage guarantee will still hold due to the built-in randomization with uniform noise, the resulting prediction sets will be less informative. Moreover, RAPS is not applicable at all since it penalizes scores based on rank, but here there is no rank information (all values are either 0 or 1). In contrast, DAPS will diffuse the scores based on the neighborhood and recover some of the missing information about the distribution of labels.

We provide some intuition on why diffusion works in this case. Bahri & Jiang (2021) have shown that even if we only have access to hard labels sampled from the true $p(y \mid \boldsymbol{x})$, the estimate $p_k(y \mid \boldsymbol{x})$ computed as the average label among the k nearest neighbors, approaches the true distribution at a minimax optimal rate for an appropriately chosen value of k. Assuming that the graph is constructed based on the k nearest neighbors in feature space, $p_k(y \mid \boldsymbol{x})$ coincides with the diffused $\hat{p}(y \mid \boldsymbol{x})$ for $\lambda = 1$. Thus, if all predictions are correct, $\hat{\pi}(y \mid \boldsymbol{x})$ also approaches the true distribution at the same optimal rate. Since the model is never perfectly accurate, there will be some estimation error, however, in practice we observe that diffusion indeed provides a significant performance boost (see § 6). We leave it for future work to theoretically characterize this setting in more detail.

6. Experimental Evaluation

We study the: (i) the impact of diffusion on efficiency and singleton hit ratio for semi-supervised node classification, (ii) the stability of all methods to random sampling and their sensitivity to hyperparameters, (iii) and the performance when we only have hard predictions. We compare our approach with the two strongest baselines APS and RAPS, which do not explicitly take the graph structure into account (although implicitly the graph is used to produce the probability vectors $\boldsymbol{\pi}_i$). For experimental evaluation, we put our main focus on the transductive case. We also provide some experiments for inductive and simultaneous inductive settings in § E.10. Moreover, in § E.2 we study the combination of the regularization from RAPS plus diffusion (although it's not recommended due to the instability of RAPS). In § E.4 we show that the margin score (Wijegunawardana et al., 2020) also benefits from diffusion.

Models and datasets. We evaluate conformal prediction considering five different models: `GCN` (Kipf & Welling, 2017), `GAT` (Velickovic et al., 2018), `GraphSAGE` (Hamilton et al., 2017), and `APPNP`(Klicpera et al., 2019) as structure-aware models and `MLP` as a structure-independent model. We evaluate our approach on 10 datasets. The common citation graphs `CoraML` (McCallum et al., 2004), `CiteSeer` (Sen et al., 2008), `PubMed` (Namata et al., 2012), `CoraFull`(Bojchevski & Günnemann, 2018), `Coauthor Physics` and `Coauthor CS` (Shchur et al., 2018). The co-purchase graphs `Amazon Photos` and `Amazon Computers` (McAuley et al., 2015; Shchur et al., 2018). And two large graphs, `OGBN Arxiv` (Wang et al., 2020) and `OGBN Products` (Bhatia et al., 2016). `CoraML`* and `CoraFull`* are variants considering the largest connected component. Datasets statistics are in § G.

Evaluation procedure. We randomly split the nodes into train/validation/calibration/test sets. Since GNNs are sensitive to splits, especially in the sparsely labeled setting (Shchur et al., 2018), we train 10 different models with different train/validation splits and report the average. We randomly select 20 nodes per class for training/validation. As described in § B, we split the calibration set into two sets, one for tuning parameters like λ, and one for actual calibration. To reflect a realistic scenario, the size of the calibration (and tuning) set is the same as as the training set size. Since APS is parameter-free, for fairness we increase its calibration set such that it uses the same total number of labels. We report average results over 100 calibration/test splits. The calibration/test labels are never used for model training. See § G for details on the labeling budget.

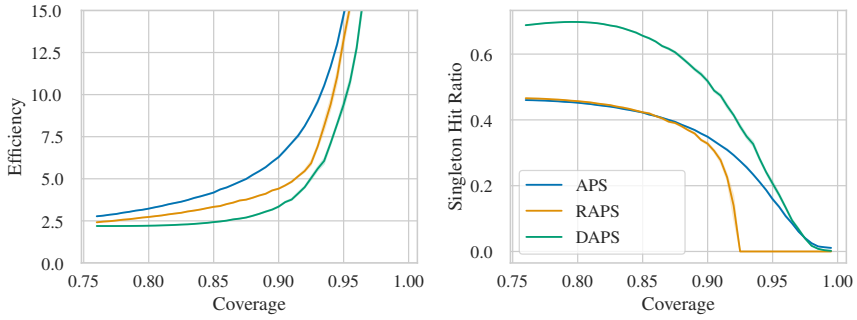

Figure 4. DAPS scales to large datasets such as `OGBN Products` and provides a strong improvement in both efficiency (left) and singleton hit ratio (right) for any coverage.

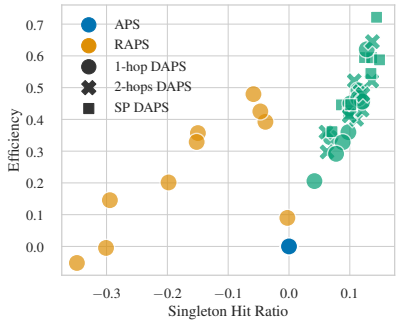

Figure 5. RAPS vs. DAPS variants relative to the APS baseline for `CoraML`/`GCN`.

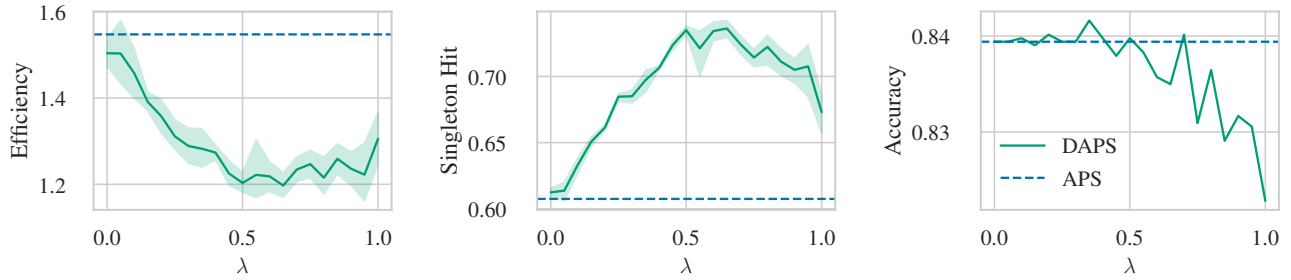

Figure 6. DAPS for different values of λ for a `GCN` model trained on `CiteSeer`. From left to right: the effect of diffusion on (i) efficiency (ii) singleton hit ratio (iii) accuracy. The change in accuracy is negligible while significantly improving the other two metrics.

6.1. Comparing Conformal Prediction Sets for GNNs

Efficiency and singleton hit ratio. As discussed in § 4, efficiency (average set size) and the singleton hit ratio are two important metrics. We consider two settings: a fixed coverage with $\alpha = 0.08$ and an adaptive coverage, related to the actual accuracy of the model, which we discuss in § E.3. Fig. 3 shows that DAPS slightly improves efficiency while significantly increasing the singleton hit ratio compared to APS and RAPS. We also study the performance across different coverage guarantees. Fig. 4 shows the result for the large `OGBN Products` dataset, again seeing that DAPS performs best overall. We provide a comprehensive report on all datasets and models in § E.9 with similar conclusions. In § E.6 we also compare empty, singleton, and multi-sets.

On Fig. 4 (right) we observe a mild increase in the singleton hit ratio for values of $1 - \alpha$ close to the model's accuracy. Since the coverage is distributed as a Beta distribution, there is both an upper and a lower bound on the coverage meaning that in these conditions CP is forced to discard potential singleton sets to satisfy the upper bound. As the coverage gets closer to the accuracy, there is a potential for improvement in the singleton hit ratio. The inflection point around 78.4% (model's accuracy with diffusion) is followed by a trade-off between coverage and the singleton hit ratio.

Generalizations of diffusion. So far we focused on 1-hop diffusion since it is simple and computationally inexpensive, making it practical. We also evaluate the 2-hop and score propagation (SP) variants introduced in § 5.2. In Fig. 5, for a `GCN` model on `CoraML`, we see that both variants provide further improvements. We leave it as a future work to study what is the optimal form of diffusion.

Efficiency and accuracy. Model accuracy plays a significant role in conformal prediction. One might wonder whether the improvements from DAPS stem primarily from improved accuracy. Fig. 6 (right) shows this is not the case. In most cases DAPS does not increase the accuracy (when predicting $\arg\max_y \hat{s}(v, y)$), while significantly increasing efficiency and singleton hit ratio. We further investigate this phenomenon in § E.5, and find a large subset of λ values that lead to improvement in CP metrics, with the optimal value of around $\lambda = 0.5$ for most datasets and models.

Stability. GNNs are known to be sensitive to data splits, so we study the stability of all methods across varying conditions. We explore two settings: (i) we tune the parameters on a single given split and we evaluate the same parameters on all other splits, (ii) we tune the parameters for a single α and we evaluate on all other values of α. In Fig. 7 (left) we see that the variance of RAPS across both metrics is significantly larger compared to APS and DAPS (as visually

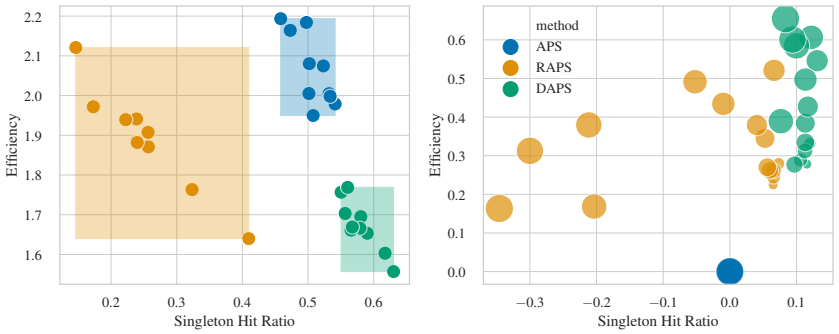

Figure 7. Stability for a GCN model on CoraML across different initial splits (left, absolute difference) and different coverage guarantees (right, enhancement relative to APS).

$\lvert\mathcal{V}_c\rvert$	NAPS		DAPS
	Eff ↓	N/A ↓	Eff ↓
5.2%	2.24	79%	1.81
10%	2.34	67%	1.82
20%	2.44	47.9%	1.81
50%	2.58	19.5%	1.80

Table 2. Comparison between DAPS and NAPS ($k = 1$) for different calibration set sizes in a transductive setting. N/A means "not applicable". Note that DAPS is always applicable to all test nodes.

shown by the rectangles). In Fig. 7 (right) we see the relative enhancement over APS. Each circle corresponds to a different α and its size shows the magnitude. Unlike RAPS, DAPS provides a consistent enhancement for all values.

Comparison with NAPS. As discussed in § 3 due to sparsity many test nodes are not adjacent to any calibration nodes. As a result, NAPS fails to return a prediction set for them. Table 2 shows that on CoraML with GCN under the default evaluation setting with $|\mathcal{V}_c| = 5.2\%$, NAPS is non-applicable and unable to make predictions for 79% of test nodes (see § E.11 for details). Even if we make the size of the calibration set unrealistically large, NAPS still fails for many test nodes. DAPS is always applicable regardless of the size of $\mathcal{V}_c$, and returns more efficient sets.

Hard predictions. We compare only with APS since, as we discussed in § 4, RAPS is not applicable when only hard predictions are given. If we naively use APS the coverage guarantee cannot be satisfied since we will still get ties despite the built-in randomization (see § E.1 for details). To make APS applicable we add a small constant ϵ to all classes and renormalize (increasing 0 to ϵ and decreasing 1 to $1 - |C|\epsilon$). For comparability, we do the same for DAPS even though it does not need it. In Fig. 8 we see that APS catastrophically fails to provide useful sets. In contrast, DAPS still works reasonably well, and its performance is close to the soft baseline that has access to the full distribution. Moreover, we see that for all methods the empirical coverage matches the guaranteed coverage, which is a good sanity check against theoretical and implementation bugs. The variance is also on the expected order of magnitude (recall that the coverage is distributed as $\mathbf{Beta}(n + 1 - l, l)$).

Limitations. Diffusion relies on homophily. In § E.9 we study graphs with lower homophily such as OGBN Arxiv, where as expected diffusion does not provide a significant boost, and has different trade-offs than RAPS. Moreover, during tuning we can always select $\lambda = 0$ to disable diffusion. More importantly, a general limitation of conformal prediction is that the guarantees only hold marginally (over all test nodes). Recall, that conditional coverage is impossible without additional strong assumption (Barber et al., 2019). Thus, we have to be careful when interpreting the results. Nonetheless, in practice we observe good empirical coverage for different subsets of nodes. In Fig. 9 we investigate coverage conditional on the class label, the set size and the node degree and see that diffusion again provides a strong benefit. In § E.7 we investigate empirical conditional coverage and find that APS and DAPS are comparable. Finally, coverage is also upper-bounded (see Theorem 1).

7. Related Work

Conformal prediction. First introduced by Vovk et al. (2005), CP provides distribution-free guarantees assuming only exchangeability (Lei & Wasserman, 2014; Shafer & Vovk, 2008). Most works provide guarantees on the marginal coverage of the true label, however, CP can be generalized to any user-defined risk function (Angelopoulos et al., 2022). Improving efficiency is often the goal. APS, RAPS and our DAPS do not change the model, while Stutz et al. (2022) improve efficiency by simulating calibration during training. Einbinder et al. (2022) encourage uniformity via a loss to improve conditional coverage. Fisch et al. (2022) add a constraint on the false positive rate. There is also significant effort in generalizing CP beyond exchangeability. Gendler et al. (2022) address the adversarial setting. Tibshirani et al. (2019) define a weighted CP to handle covariate shift. Finally, Barber et al. (2022) propose a general framework where their guarantee has a penalty term proportional to the degree of distribution shift.

Uncertainty quantification on graphs. There are few studies on uncertainty quantification for graph-based models such as GNNs (Abdar et al., 2020). One challenge is the interdependence between the nodes which prevents us from e.g. directly applying methods designed for i.i.d. data. Stadler et al. (2021) explicitly model epistemic and aleatoric

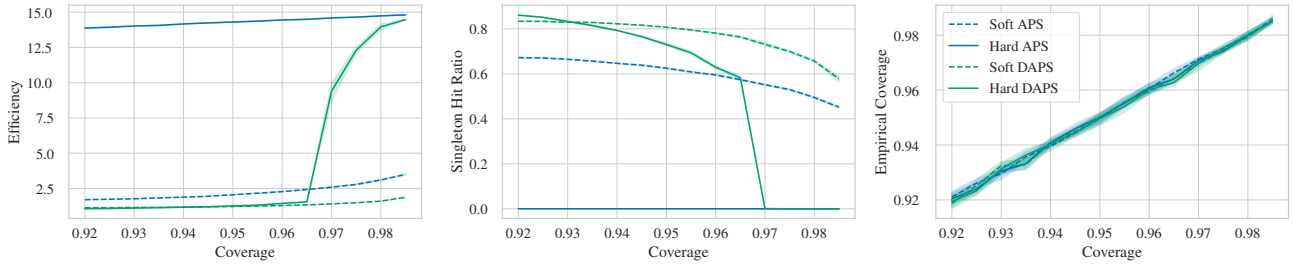

Figure 8. Comparison of APS and DAPS when using hard predictions on the `Coauthor-CS` dataset for a `GCN` model. We evaluate three metrics: efficiency (left), singleton hit ratio (middle), and empirical coverage (right). Dashed lines are recalls from CP approaches with access to the predicted softmax probabilities while solid line correspond to the same approach applied over the hard (one-hot) predictions.

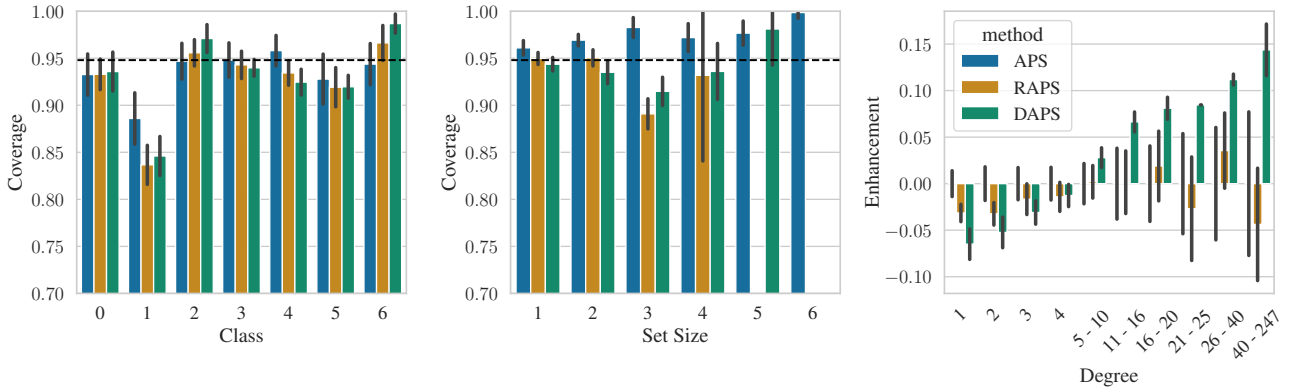

Figure 9. Empirical coverage conditional on the class label (left), prediction set size (middle), and node degree (right). On the middle plot the support for each set size is different per method. On the right plot the lines without any bar show the s.d. of APS from its mean.

uncertainty by propagating node-wise estimates along the graph. In § F we show that their approach is orthogonal and can be combined with our CP guarantees. They define three axioms for uncertainty quantification with structural dependency. DAPS aligns with the third, indicating that a node's aleatoric uncertainty should increase when connected to conflicting nodes or nodes with higher aleatoric uncertainty. Hsu et al. (2022b) study calibration and temperature scaling, and Hsu et al. (2022a) study edge-wise calibration. A few works study out-of-distribution detection (Liu et al., 2023; Huang et al., 2022; Bazhenov et al., 2022).

Conformal prediction on graphs. Wijegunawardana et al. (2020) is the first to apply CP on graphs. They propose a margin-based score, which unlike DAPS, does not explicitly account for the graph structure. In § E.9 we show that their score also benefits from diffusion. Nonetheless, we argue that using APS as the base score is more suitable, since similar to TPS, the margin score may undercover hard examples. Recently, Clarkson (2022) introduces NAPS for the inductive case using the beyond-exchangeability technique from Barber et al. (2022), dismissing the transductive case as unsuitable. In § 3, we highlighted the major limitations of NAPS (see also § E.11 for a longer discussion). Alongside our main focus on the transductive scenario, we provide additional experiments on the inductive setting in § E.10. Finally, Kang et al. (2022) derives a variant of Jackknife+ for GCN. Different from existing works, in our score we explicitly leverage homophily while still providing a valid coverage guarantee.

8. Conclusion

We propose conformal prediction sets that explicitly account for the graph structure. The key insight is that diffusing the conformity scores along the graph leads to improved uncertainty quantification in presence of homophily. We discuss exchangeability for graphs and GNNs, and the theoretical conditions under which diffusion is beneficial. Our method, DAPS, performs on par or better than the baselines in efficiency and significantly better w.r.t. singleton hit ratio.

Acknowledgements

This work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition. We thank Vijay Lingam, and Mohammad Sadegh Akhondzadeh for their feedback on the first draft.

References

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P. W., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *ArXiv preprint*, 2020.

Abdar, M., Pourpanah, F., Hussain, S., Rezazadegan, D., Liu, L., Ghavamzadeh, M., Fieguth, P. W., Cao, X., Khosravi, A., Acharya, U. R., Makarenkov, V., and Nahavandi, S. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Inf. Fusion*, 2021.

Angelopoulos, A. N. and Bates, S. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *ArXiv*, 2021.

Angelopoulos, A. N., Bates, S., Jordan, M. I., and Malik, J. Uncertainty sets for image classifiers using conformal prediction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, 2021.

Angelopoulos, A. N., Bates, S., Fisch, A., Lei, L., and Schuster, T. Conformal risk control. *ArXiv preprint*, 2022.

Bahri, D. and Jiang, H. Locally adaptive label smoothing improves predictive churn. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, 2021.

Barber, R. F. Is distribution-free inference possible for binary regression? *Electronic Journal of Statistics*, (2), 2020. doi: 10.1214/20-EJS1749.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. The limits of distribution-free conditional predictive inference. *Information and Inference: A Journal of the IMA*, 2019.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. Predictive inference with the jackknife+. *The Annals of Statistics*, (1), 2021. doi: 10.1214/20-AOS1965.

Barber, R. F., Candès, E. J., Ramdas, A., and Tibshirani, R. J. Conformal prediction beyond exchangeability. 2022.

Bazhenov, G., Ivanov, S., Panov, M., Zaytsev, A., and Burnaev, E. Towards ood detection in graph classification from uncertainty estimation perspective. *ArXiv*, 2022.

Berti, P. and Rigo, P. A glivenko-cantelli theorem for exchangeable random variables. *Statistics & probability letters*, (4), 1997.

Bhatia, K., Dahiya, K., Jain, H., Kar, P., Mittal, A., Prabhu, Y., and Varma, M. The extreme classification repository: Multi-label datasets and code, 2016.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Cai, D., Campbell, T., and Broderick, T. Edge-exchangeable graphs and sparsity. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016.

Clarkson, J. Distribution free prediction sets for node classification. In *Learning on Graphs Conference*, 2022.

Dvoretzky, A., Kiefer, J., and Wolfowitz, J. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, 1956.

Einbinder, B.-S., Romano, Y., Sesia, M., and Zhou, Y. Training uncertainty-aware classifiers with conformalized deep learning. In *Advances in Neural Information Processing Systems*, 2022.

Fey, M. and Lenssen, J. E. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

Fisch, A., Schuster, T., Jaakkola, T. S., and Barzilay, R. Conformal prediction sets with limited false positives. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, Proceedings of Machine Learning Research, 2022.

Gendler, A., Weng, T., Daniel, L., and Romano, Y. Adversarially robust conformal prediction. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, Proceedings of Machine Learning Research, 2017.

Hamilton, W. L., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017.

Hein, M., Andriushchenko, M., and Bitterwolf, J. Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, 2019. doi: 10.1109/CVPR.2019.00013.

Holland, P. W., Laskey, K. B., and Leinhardt, S. Stochastic blockmodels: First steps. *Social networks*, (2), 1983.

Hsu, H. H.-H., Shen, Y., and Cremers, D. A graph is more than its nodes: Towards structured uncertainty-aware learning on graphs. In *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022a.

Hsu, H. H.-H., Shen, Y., Tomani, C., and Cremers, D. What makes graph neural networks miscalibrated? In *Advances in Neural Information Processing Systems*, 2022b.

Huang, T., Wang, D., Fang, Y., and Chen, Z. End-to-end open-set semi-supervised node classification with out-of-distribution detection. In *International Joint Conference on Artificial Intelligence*, 2022.

Hüllermeier, E. and Waegeman, W. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.*, 2021.

Kang, J., Zhou, Q., and Tong, H. Jurygcn: Quantifying jackknife uncertainty on graph convolutional networks. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Kath, C. and Ziel, F. Conformal prediction interval estimation and applications to day-ahead and intraday power markets. *International Journal of Forecasting*, (2), 2021.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Klicpera, J., Bojchevski, A., and Günnemann, S. Predict then propagate: Graph neural networks meet personalized pagerank. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019.

Kuchibhotla, A. K. Exchangeability, conformal prediction, and rank tests. *arXiv: Methodology*, 2020.

Lei, J. and Wasserman, L. A. Distribution-free prediction bands for non-parametric regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2014.

Liu, Y., Ding, K., Liu, H., and Pan, S. Good-d: On unsupervised graph out-of-distribution detection. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, 2023. ISBN 9781450394079. doi: 10.1145/3539597.3570446.

Lloyd, J. R., Orbanz, P., Ghahramani, Z., and Roy, D. M. Random function priors for exchangeable arrays with applications to graphs and relational data. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, 2012.

Luo, R., Zhao, S., Kuck, J., Ivanovic, B., Savarese, S., Schmerling, E., and Pavone, M. Sample-efficient safety assurances using conformal prediction. In *Algorithmic Foundations of Robotics XV*, 2023. ISBN 978-3-031-21090-7.

McAuley, J. J., Targett, C., Shi, Q., and van den Hengel, A. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, Santiago, Chile, August 9-13, 2015*, 2015. doi: 10.1145/2766462.2767755.

McCallum, A., Nigam, K., Rennie, J. D. M., and Seymore, K. Automating the construction of internet portals with machine learning. *Information Retrieval*, 2004.

Namata, G., London, B., Getoor, L., and Huang, B. Query-driven active surveying for collective classification. 2012.

Romano, Y., Sesia, M., and Candès, E. J. Classification with valid and adaptive coverage. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Sadinle, M., Lei, J., and Wasserman, L. A. Least ambiguous set-valued classifiers with bounded error levels. *Journal of the American Statistical Association*, 2018.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B., and Eliassi-Rad, T. Collective classi!cation in network data. 2008.

Shafer, G. and Vovk, V. A tutorial on conformal prediction. *Journal of Machine Learning Research*, (12), 2008.

Shchur, O., Mumme, M., Bojchevski, A., and Günnemann, S. Pitfalls of graph neural network evaluation. *Relational Representation Learning Workshop, NeurIPS 2018*, 2018.

Stadler, M., Charpentier, B., Geisler, S., Zügner, D., and Günnemann, S. Graph posterior network: Bayesian predictive uncertainty for node classification. In *Advances in*

Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, 2021.

Stutz, D., Dvijotham, K., Cemgil, A. T., and Doucet, A. Learning optimal conformal classifiers. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

Tibshirani, R. J., Barber, R. F., Candès, E. J., and Ramdas, A. Conformal prediction under covariate shift. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

Vazquez, J. and Facelli, J. C. Conformal prediction in clinical medical sciences. *Journal of Healthcare Informatics Research*, 2022.

Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

Vovk, V. Conditional validity of inductive conformal predictors. *Machine Learning*, 2012.

Vovk, V. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, (1), 2015.

Vovk, V., Gammerman, A., and Shafer, G. Algorithmic learning in a random world. 2005.

Wang, K., Shen, Z., Huang, C., Wu, C.-H., Dong, Y., and Kanakia, A. Microsoft Academic Graph: When experts are not enough. *Quantitative Science Studies*, (1), 2020. ISSN 2641-3337. doi: 10.1162/qss_a_00021.

Wijegunawardana, P., Gera, R., and Soundarajan, S. Node classification with bounded error rates. 2020.

A. Conformal Prediction Algorithm

Assuming that a black box model f has been chosen, as well as a score function $s(\cdot, \cdot)$, the construction of a conformal prediction set is straightforward. On a hold-out calibration set, we compute conformal scores for each pair of input and the corresponding true class. Then we sort them and save the α quantile as a calibration quantile variable. During the evaluation procedure, for each datapoint, we evaluate the conformal score for each of the classes, and we take those with scores higher than the quantile as elements of the prediction set. See Fig. 10 for a better intuition about the position of the quantile with respect to true classes and false classes. Algorithm 1 outlines the steps to obtain a prediction set with a coverage guarantee equal to a user-selected $1 - \alpha$.

Algorithm 1 Conformal prediction pseudo-code

Input: Model f
Score function s
Held-out[a] labeled calibration data $\mathcal{D}_{\text{cal}} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$
Unseen new input $(\boldsymbol{x}_{n+1}, ?)$
Coverage guarantee $1 - \alpha$
$\forall (\boldsymbol{x}_i, y_i) \in \mathcal{D}_{\text{cal}}$ compute $s(\boldsymbol{x}_i, y_i) = s_i$
Sort all scores $\mathcal{S} = \{s_i\}_{i=1}^{n}$
Set $\hat{q} := \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \mathcal{S}\right)$
Compute $s(\boldsymbol{x}_{n+1}, y_j)$, for all $y_j \in \mathcal{Y}$
Return: $\mathcal{C}_\alpha(\boldsymbol{x}_{n+1}) = \{y_j : s(\boldsymbol{x}_{n+1}, y_j) \geq \hat{q}\}$

[a] Recall that in the transductive setting, the feature and graph structure of the calibration (and the test nodes) are available to the model during training, but their labels are not. Thus, held-out here refers to the labels.

In addition to the mentioned algorithm, the Python implementation including the code to reproduce reported results is accessible at `https://github.com/soroushzargar/DAPS`.

A.1. Computational Complexity of DAPS

Alongside the time complexity of conformal prediction, to apply DAPS or its generalizations we have to consider additional computation. Simple diffusion takes $\mathcal{O}(E)$, and its k-hop generalization takes $\mathcal{O}(k \cdot E)$ additional runtime. This complexity is added to the whole procedure and we need to run it only once (for transductive setting). The complexity of the SP generalization is dominated by the complexity to compute the propagated scores. In practice, we do not compute the inverse matrix but rather use only a few (e.g. 10) steps of power iteration which is enough to get a good approximation. There is a rich literature on scalable approximations to personalized PageRank that is also applicable here. We highlight that we applied DAPS to `OGBN Products` a graph with more than 2.4 million nodes (with a wall-clock time of 631 ms).

Experimental setting. We based our implementation on PyTorch Geometric (Fey & Lenssen, 2019). Given the computation efficiency of DAPS, we run all our experiments both on CPU (Intel(R) Xeon(R) Platinum 8368 CPU @ 2.40GHz) and, even if not necessary, on GPU (NVIDIA A100-SXM4-40GB).

B. Tuning Calibration Parameters

Split conformal prediction relies on held-out labeled data for calibration. It is not always realistic to assume that a large proportion of data is accessible for this purpose. This restriction becomes more critical in sparsely-labeled semi-supervised node classification tasks since the goal is to predict all node labels in the graph based on a small proportion of training nodes. Methods like DAPS and RAPS require additional labeled data for tuning calibration parameters. Naively, one might require two other labeled sets, analogous to the calibration/evaluation sets used in the final algorithm, in order to estimate the effect of different hyperparameters (e.g. different values of λ). We show that tuning can be conducted using only one set, which we call the tuning set. Our tuning procedure is the same as the procedure in RAPS, which uses a tuning set to select λ and k. Here, we only provide a principled justification of why this is a good idea. Specifically, we show that the expected set size on the tuning set is almost surely the same as the expected set size on the test set.

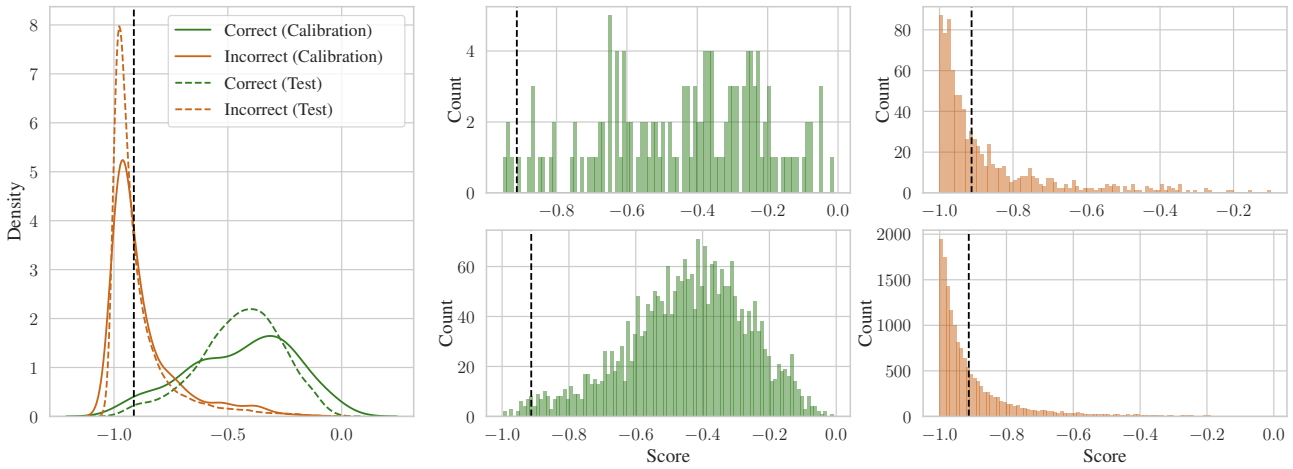

Figure 10. Scores with respect to the selected quantile. A density plot of the scores (left) where solid lines show the calibration set and dashed lines show the test set. The 4×4 boxes (middle and right) show a histogram of calibration scores (upper row) and test scores (lower row) for both true classes (green) and false classes (orange). On all plots, the dashed black line shows the place of the α quantile.

The expected set size is determined by: (1) performing a calibration over tuning scores, (2) defining prediction sets for elements in the same tuning set, and finally (3) computing the effective set size for nodes in the tuning set. The calculated number is an approximation for the effective set size over the rest of the unlabeled nodes. Formally, the expected set size for the conformal prediction with $1-\alpha$ coverage and the holdout tuning set $\mathcal{I}_\tau$ is derived as

$$\mathbb{E}[\text{ESS}_\alpha(\mathcal{D})] = \frac{\sum_{\boldsymbol{x}_i \in \mathcal{I}_\tau} \sum_{j \in \{1,\dots,C\}} \mathbb{I}\left(s(\boldsymbol{x}_i, y^{(j)}) > \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right)\right)}{|S_\tau|} \tag{3}$$

where $s(\boldsymbol{x}_i, y^{(j)})$ is the score value for class j and node i in the tuning set. This could be rewritten as $\mathbb{E}[\text{ESS}_\alpha(\mathcal{D})] = \frac{1}{|\mathcal{D}|}\sum_{\boldsymbol{x} \in \mathcal{D}} \mathbb{E}[\text{ESS}_\alpha(\boldsymbol{x})]$.

For simplicity, we consider binary classification, but the extension to multiple classes is trivial. Assume $S = \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n \sim \mathcal{D}$ is sampled exchangeably from the test dataset. Note that in this notation y_i is the true class for $\boldsymbol{x}_i$; we also call the false class as y_i'. For $1-\alpha$ coverage, we take $\hat{q} := \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right)$. There exists a α^r for the derived value $\hat{q}$ for which we have $\hat{q} = \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha^r) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i')\}_{i=1}^n\right)$. For each individual node $\boldsymbol{x}_i$ we assume that the probability of class $y^{(1)}$ is η_i, hence the probability of $y^{(2)}$ is equal to $1-\eta_i$. The expected set size for the node x_i is based on two independent random events; $y^{(1)} \in \mathcal{C}(x_i)$, and $y^{(2)} \in \mathcal{C}(\boldsymbol{x}_i)$. If each of these classes is selected in the prediction set, the expected set size increases by one unit; hence the expected prediction set size for the set $C(\boldsymbol{x}_i)$ is

$$\mathbb{E}[|\mathcal{C}(\boldsymbol{x}_i)|] = 1 \times \mathbb{P}[y^{(1)} \in \mathcal{C}(\boldsymbol{x}_i)] + 1 \times \mathbb{P}[y^{(2)} \in \mathcal{C}(\boldsymbol{x}_i)] \tag{4}$$

We can expand $\mathbb{P}[y^{(j)} \in \mathcal{C}(\boldsymbol{x}_i)]$ to two conditions based on whether $y^{(j)}$ is true, and a supplementary term that determines whether the set contains $y^{(j)}$ given that it is true or false:

$$\mathbb{E}[|\mathcal{C}(\boldsymbol{x}_i)|] = [\eta_i(1-\alpha) + (1-\eta_i)(1-\alpha^r)] + [(1-\eta_i)(1-\alpha) + (\eta_i)(1-\alpha^r)] = (1-\alpha) + (1-\alpha^r) \tag{5}$$

Based on definition of α^r we have

$$\mathbb{E}[|\mathcal{C}(\boldsymbol{x}_i)|] = 1 - \alpha + \sum_{j: y^{(j)} \neq y_i} \mathbb{I}\left(s(\boldsymbol{x}_i, y^{(j)}) > \text{Quantile}\left(\frac{\lfloor (n-1)(\alpha) \rfloor}{n}; \{s(\boldsymbol{x}_i, y_i)\}_{i=1}^n\right)\right) \tag{6}$$

This yields an expected prediction set size for any node inside the tuning set. Similarly we have $\mathbb{E}[\text{ESS}_\alpha(\mathcal{I})] = \frac{1}{|\mathcal{I}|}\sum_{\boldsymbol{x}_i \in \mathcal{I}} \mathbb{E}[\text{ESS}_\alpha(\boldsymbol{x}_i)]$. As the tuning set $\mathcal{I}$ is exchangeably sampled from $\mathcal{D}$, the plug-in estimator is unbiased (see

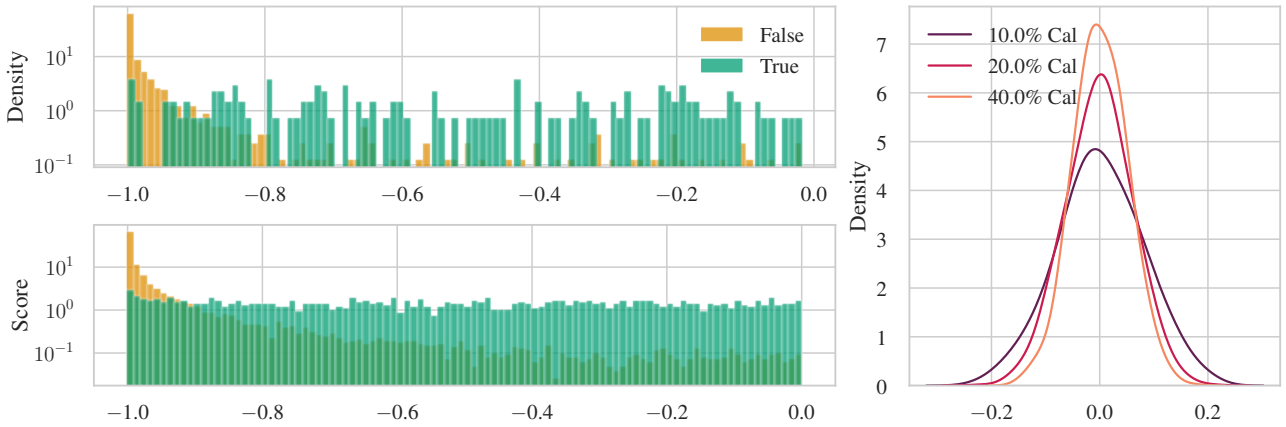

Figure 11. The density histogram of the scores in the calibration set (upper left) and the evaluation set (lower left) alongside the marginal difference between the estimated set size and the actual effective set size (right). As shown in the figure, the estimation error for the average set size is centered around zero and the concentration of this error is correlated with the size of the calibration set. Here we use the APS score, but similar results hold for all other scores.

(Berti & Rigo, 1997)), hence $\mathbb{E}[\text{ESS}_\alpha(\mathcal{D})] = \mathbb{E}[\text{ESS}_\alpha(\mathcal{I})]$. Note, for the special case of i.i.d., one can also use the Dvoretzky–Kiefer–Wolfowitz inequality (Dvoretzky et al., 1956) to characterize the approximation error, however, we omit this discussion here since we focus on the more general exchangeable setting.

In our experiments, we use a tuning set with the same size as the calibration set (and the training/validation sets) of just 20 nodes per class on average. Thus, the total sum of labeled nodes used is still relatively small. This is in contrast to e.g. applications of CP in computer vision, where a large number of labels are available. Both RAPS and DAPS share the same random indices for the splits. We exclude the tuning set from any further evaluation or calibration steps. In other words, to make sure that the coverage guarantee is not violated we do not reuse the tuning set during the final calibration which uses "fresh" data. As an empirical verification of the above statement, we empirically compare our estimate of the expected average set size (i.e. the efficiency) with the average set size on the test set. In Fig. 11 we see that the distribution of errors has a mean around zero and the variance scales with the number of calibration samples.

C. Proofs

In this section, we provided the proofs that were omitted from the main paper.

Proposition 1. *Assume that $\mathcal{V}_c \cup \mathcal{V}_u$ is exchangeable. Let $\pi(G) = \mathbf{\Pi} \in \triangle^{|\mathcal{V}| \times K}$ be a matrix where row v is the label distribution for node v predicted by any permutation equivariant GNN classifier $\pi(\cdot)$ trained on the entire graph G and only using labels for nodes in $\mathcal{V}_d$. Then the scores $s(v, y) = \mathbf{\Pi}_{vy}$ where $\mathbf{\Pi}_{vy}$ is the predicted probability for node v and class y, are exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

Proof of Proposition 1. Let $g(\boldsymbol{X}, \boldsymbol{A})$ be the function that takes the entire graph, trains the model $\pi(\cdot)$ using only the labels for the nodes in $\mathcal{V}_d$, and returns the prediction only for the calibration and test nodes $\mathcal{V}_c, \mathcal{V}_u$. Since, we assume that $\pi(\cdot)$ is permutation equivariant, this implies that g is also permutation equivariant w.r.t. a *subset* of nodes. To see this, we construct the matrix $\mathbf{\Pi}$ as the prediction matrix $\mathbf{\Pi} = \pi(\boldsymbol{X}, \boldsymbol{A})$. This matrix consists of a block of labeled nodes, corresponding to the nodes in $\mathcal{V}_d$, and two blocks of nodes corresponding to $\mathcal{V}_c$ and $\mathcal{V}_u$ respectively. Without loss of generality, let the first $|\mathcal{V}_d|$ rows correspond to $\mathcal{V}_d$, the next $|\mathcal{V}_c|$ rows correspond to $\mathcal{V}_c$, and the final $|\mathcal{V}_u|$ rows correspond to $\mathcal{V}_u$, i.e. $\mathbf{\Pi} = [\mathbf{\Pi}^d, \mathbf{\Pi}^c, \mathbf{\Pi}^u]^T$ where $\mathbf{\Pi}^d, \mathbf{\Pi}^c, \mathbf{\Pi}^u$ correspond to the three blocks. Then, $g(\boldsymbol{X}, \boldsymbol{A}) = [\mathbf{\Pi}^c, \mathbf{\Pi}^u]^T$. Since π is permutation equivariant, we have that for any permutation ω, it holds $g(\omega\boldsymbol{X}, \omega\boldsymbol{A}) = \omega[\mathbf{\Pi}^c, \mathbf{\Pi}^u]^T$. Now, the result directly follows given the assumption that the nodes in $\mathcal{V}_c \cup \mathcal{V}_u$ are exchangeable and the fact that permutation equivariant functions preserve exchangeability (Kuchibhotla, 2020). □

Recall, that the while we do have and do use the labels for $\mathcal{V}_c$ for calibration, these labels are not used during training.

Proposition 2. *Let $\boldsymbol{H} \in \mathbb{R}^{|\mathcal{V}| \times K}$ be any matrix where row v is the conformal scores for all classes y for node v, and let $\boldsymbol{H}$ be exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. Then the diffused scores $\hat{\boldsymbol{H}}$ are also exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$.*

Proof of Proposition 2. Similar to the proof of Proposition 1, the diffusion of the scores defined in Eq. 2 that results in $\hat{\boldsymbol{H}}$ is permutation equivariant for all nodes, and hence also for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. To see this notice that $\hat{\boldsymbol{H}} = (1-\lambda)\boldsymbol{H} + \lambda \boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{H}$ is a special case of a message passing GNN layer. It follows that the diffused scores are also exchangeable for all $v \in (\mathcal{V}_c \cup \mathcal{V}_u)$. □

Theorem 2. *Let $\boldsymbol{\pi}_i$ be the model's approximation of the ground-truth conditional probability vector $\boldsymbol{p}_i$, and let the diffused distribution be $\hat{\boldsymbol{\pi}}_i = (1-\lambda)\boldsymbol{\pi}_i + \frac{\lambda}{|\mathcal{N}_i|}\sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j$. Assume that the $G = (\boldsymbol{X}, \boldsymbol{A})$ is constructed such that $\boldsymbol{A}_{ij} = 1$ iff $||\boldsymbol{p}_i - \boldsymbol{p}_j|| \leq \Delta$ where $||\cdot||$ is the total variation norm. Diffusion improves the approximation error $\epsilon_i = ||\boldsymbol{\pi}_i - \boldsymbol{p}_i||$, i.e. $||\hat{\boldsymbol{\pi}}_i - \boldsymbol{p}_i|| < \epsilon_i$ if $\frac{1}{|\mathcal{N}_i|}\sum_{j \in \mathcal{N}_i} \epsilon_j + \Delta < \epsilon_i$.*

Proof of Theorem 2. To show $\|\boldsymbol{p}_i - \hat{\boldsymbol{\pi}}_i\| < \|\boldsymbol{p}_i - \boldsymbol{\pi}_i\| = \epsilon_i$ we use the definition of $\hat{\boldsymbol{\pi}}_i$

$$\|\boldsymbol{p}_i - \hat{\boldsymbol{\pi}}_i\| = \|\boldsymbol{p}_i - (1-\lambda)\boldsymbol{\pi}_i - \frac{\lambda}{|\mathcal{N}_i|} \cdot \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j\| \tag{7}$$

Leveraging the fact that $\boldsymbol{p}_i = (1 - \lambda \cdot \boldsymbol{p}_i) + \lambda \cdot \boldsymbol{p}_i$, we have

$$\begin{aligned}
\left\| \boldsymbol{p}_i - (1-\lambda)\boldsymbol{\pi}_i - \frac{\lambda}{|\mathcal{N}_i|} \cdot \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j \right\| &= \left\| (1-\lambda)(\boldsymbol{p}_i - \boldsymbol{\pi}_i) + \lambda \left(\boldsymbol{p}_i - \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \boldsymbol{\pi}_j \right) \right\| && (8) \\
&\leq (1-\lambda)\|\boldsymbol{p}_i - \boldsymbol{\pi}_i\| + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{p}_i - \boldsymbol{\pi}_j\| && (9) \\
&\leq (1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\boldsymbol{p}_i - \boldsymbol{\pi}_j\| && (10) \\
&\leq (1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\|\boldsymbol{p}_i - \boldsymbol{p}_j\| + \|\boldsymbol{p}_j - \boldsymbol{\pi}_j\|) && (11) \\
&\leq (1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\Delta + \epsilon_j) && (12)
\end{aligned}$$

Where we repeatedly use the triangle inequality.

Conclusively, we want to prove that

$$\begin{aligned}
(1-\lambda)\epsilon_i + \frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \lambda\Delta < \epsilon_i && (13) \\
\frac{\lambda}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \lambda\Delta < \lambda\epsilon_i && (14)
\end{aligned}$$

By dividing everything by λ

$$\frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \epsilon_j + \Delta < \epsilon_i \tag{15}$$

Which is the basic assumption of the proposition. □

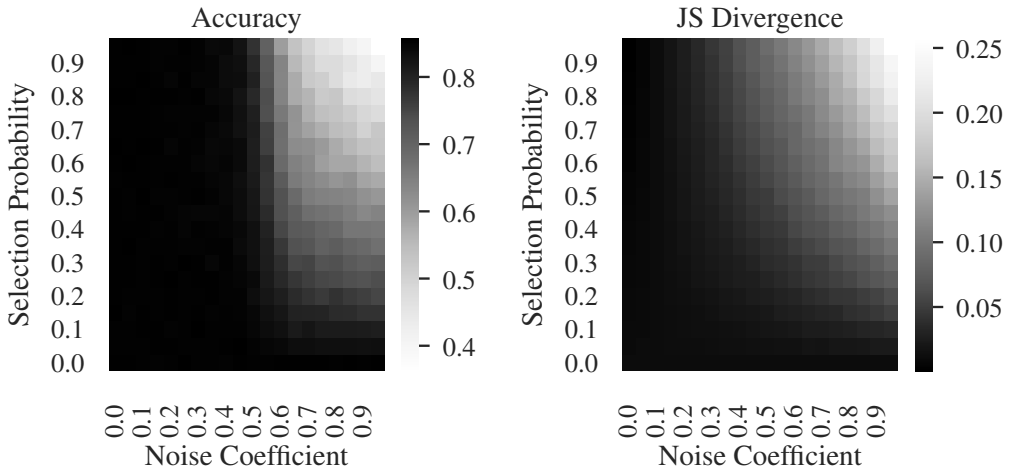

Figure 12. Accuracy and Jensen Shannon divergence for various degrees of perturbation over the ground-truth label distribution. The x-axis shows the intensity of noise applied to highly perturbed nodes (ϵ_h) and the y-axis shows the probability of the node being highly perturbed (p_s). A highly accurate model does not imply a good approximation of the true label distribution.

D. Synthetic Experiment with Access to the Ground-Truth Distribution

Additional to the provided theoretical insights on the effect of neighborhood diffusion, we also carried out a supplementary experiment utilizing synthetic data. In § 5.1 we discussed a perturbation of the ground-truth data. We use the following perturbation scheme:

$$\pi(x_i) = \pi_i = \begin{cases} p_i + u \cdot \epsilon_h & \text{if Bernoulli}(p_s) = 1 \\ p_i + u \cdot \epsilon_l, & \text{otherwise} \end{cases} \tag{16}$$

where ϵ_h is a high-magnitude perturbation coefficient, ϵ_l is a low-magnitude perturbation coefficient, u is a uniform random variable, and p_s defines the probability of a node being highly perturbed. We make sure that the resulting probability vector is normalized. The result is a perturbed distribution that aligns with ground truth for many nodes (with a very small shift) while highly perturbing the small proportion of selected nodes.

For the experiment in Fig. 2 we set $\epsilon_h = 0.7$, $\epsilon_l = 0.1$, and $p_s = 0.2$. First, we generate a dataset based on two different multivariate Gaussian distributions. Each point in the resulting dataset would be a conditional probability vector indicating how much the point is likely to belong to each class. Labels are also sampled from the same distribution. Here the graph is the k-NN graph with $k = 15$. In Fig. 12 we study the effects of changing the high-magnitude perturbation coefficient ϵ_h from 0.0 to 0.9. We measure the effect on the accuracy and on the Jensen–Shannon divergence to the ground-truth probability distribution. We see that a perturbation up to a certain degree can cause a large JS divergence from the ground-truth while preserving the rank of the class with the highest probability (and thus the accuracy). This shows that an accurate model does not guarantee a good approximation of the label distribution.

Potential issue with smoothed probabilities. One potential pitfall of the diffusion approach which is also observable in the synthetic data experiment (see Fig. 2) is that it results in less confident probability vectors. This is a general phenomenon since there is an aggregation involved in the transformation. One can notice the resulting underconfidence in Fig. 1 where for DAPS we see a decrease at the end (highly-confident) part of the histogram. However, the denoising effect of diffusion makes such a valuable enhancement, that the effect of smoothing is negligible. This issue can be mitigated with temperature scaling, however we leave this for future work.

E. Additional Experiments and Experimental Details

E.1. Technical Details for the Hard Prediction Case

As mentioned in § 4 and further evaluated in § 6, one additional application of neighborhood diffusion is in cases where only the hard predictions of nodes are provided. Technically, in such cases methods like APS can not perform as expected since the one-hot probability vector causes many ties in the score space. Even the built-in uniform randomization in APS can not overcome this problem. Inspecting the definition of APS, we see that the classes with probability of 0 are directly mapped to -1 which again results in ties despite the randomization. As a technical solution to this problem, we increase each non-top class by a small constant $\epsilon = 0.001$ and decrease the top class by $|C| \cdot \epsilon$. After this step, the resulting vectors can be used with APS and we can obtain valid coverage. However, APS results in large prediction sets for almost all inputs. This makes sense, since there is no information about the ranking of the (non-top) classes. As we discussed before, this is also the reason why RAPS is not applicable, even with the ϵ transformation. For a fair comparison, we apply the same transformation to DAPS even though it can work without it.

E.2. Combination of Scoring Functions

In addition to individually comparing DAPS and RAPS (see § E.9), another idea is to apply each of the approaches as an add-on to the other. This means that we can perform the regularization defined in RAPS on top of the diffusion in DAPS or vice versa. We evaluate the enhancement of such combined scores (relative to APS) in Fig. 13. We see that diffusion provides additional benefits to RAPS. However, we do not recommend using these combinations since they inherit the instabilities of RAPS and its sensitivity to different initial splits and different α values.

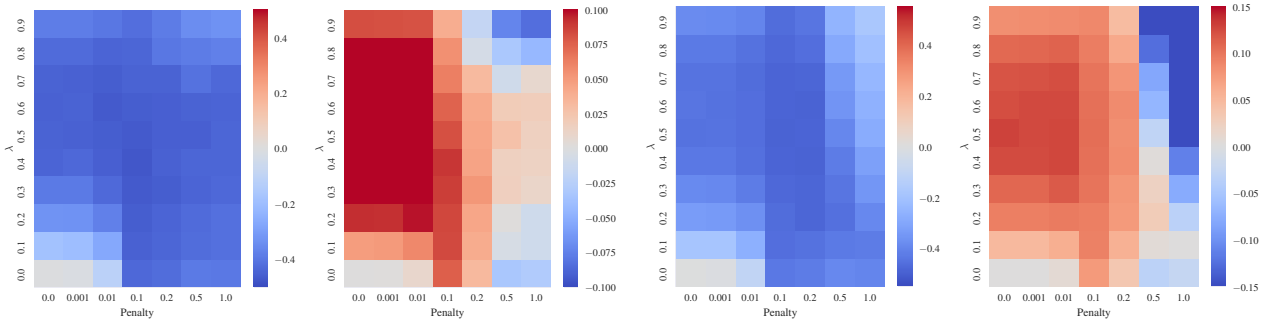

Figure 13. Applying RAPS regularization over DAPS (left two plots), and diffusion over RAPS (right two plots). For each pair of plots the left subplot shows the enhancement in efficiency and the right subplot shows the enhancement in singleton hit ratio (the results are relative to APS). All plots of this figure have the $k = 0$ for RAPS.

E.3. Adaptive Coverage Guarantee

Many recent studies report their results on a fixed coverage guarantee. We argue that an adaptive coverage guarantee (which we define next) is more meaningful. In any case, we report results using both fixed and adaptive coverage. To make the coverage adaptive, for each dataset-model pair, we set $1 - \alpha$ relative to the accuracy of the selected model for the selected dataset. In our examples, this value is set to a weighted average between the model accuracy and 100% with weights $(\frac{1}{3}, \frac{2}{3})$. This results in a value of $1 - \alpha$ that is always larger than the accuracy. For example, if the model has accuracy of 97% it is less informative to use a fixed $1 - \alpha = 0.9$ which is often the default. Here the adaptive coverage $1 - \alpha = 0.99$ is more realistic. For all results in the main paper, except Fig. 7 (right), we use adaptive coverage, the concrete selected values are given in Table 5. The value of α is important since the performance of CP is sensitive to the distance between $1 - \alpha$ and the model's accuracy (see Fig. 4 as one of many examples).

In other words, assuming everything else is the same, a model with higher accuracy is expected to exhibit superior performance under a fixed coverage guarantee. Hence, it is essential to examine different methods for both fixed and adaptive coverage. Fig. 14 illustrates a significant difference between the two settings (fixed and adaptive coverage). Nonetheless, DAPS performs well in both settings. Since the accuracy of `Coauthor Physics` and `Coauthor CS` is relatively high (see Table 5), all models including APS seem to perform well for fixed coverage, which is not true for the adaptive coverage.

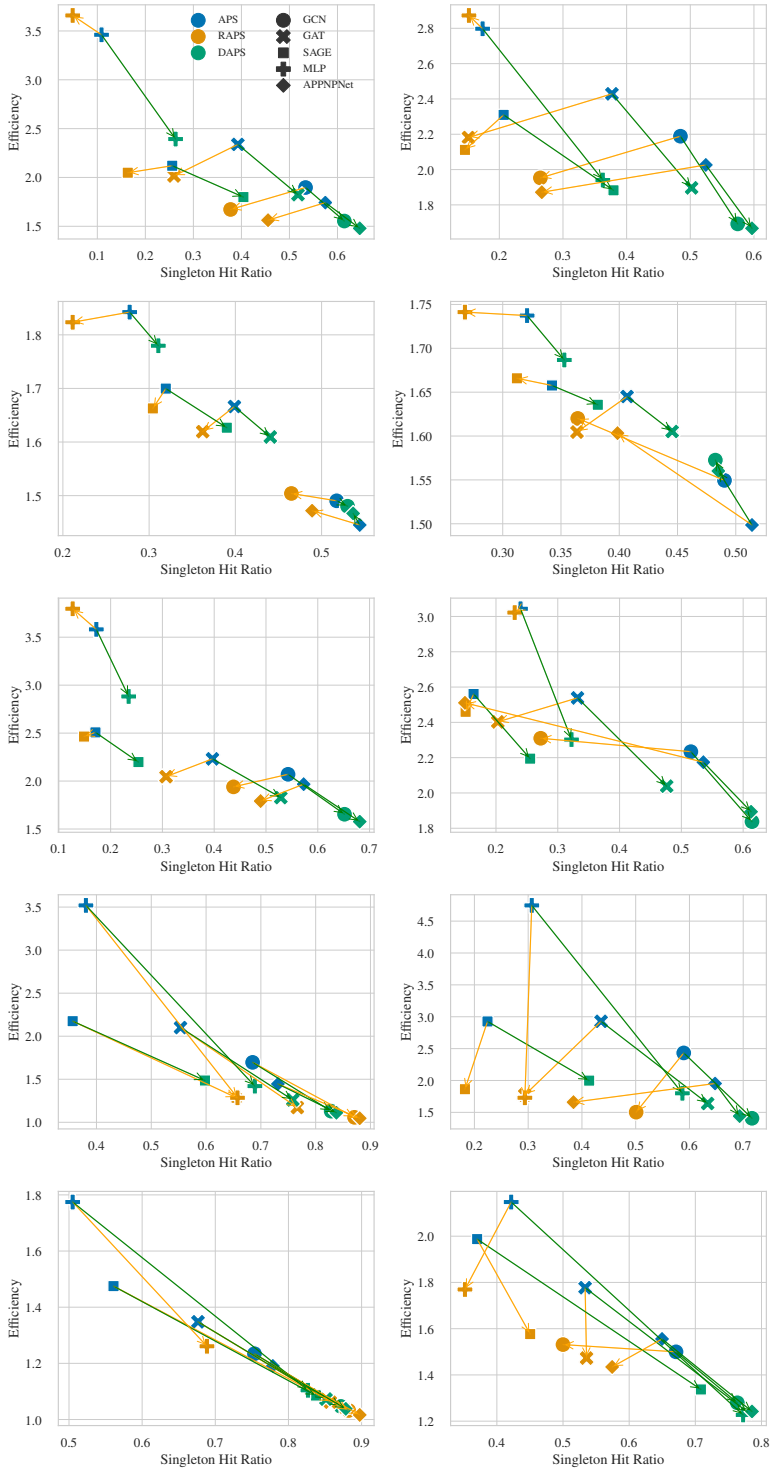

Figure 14. The Pareto plot of different CP approaches for different datasets. From top to bottom, each row illustrates the evaluation of the approaches, namely APS, RAPS and DAPS, on `CoraML`, `PubMed`, `CiteSeer`, `Coauthor CS`, and `Coauthor Physics` respectively. The left plot in each row is regarding an experiment on 92% fixed coverage, and the right plot illustrates the result for adaptive coverage. DAPS performs best on average for both fixed and adaptive coverage.

E.4. Margin Scoring Function

In this study, we focused on applying diffusion on top of the baseline APS score function. However, this approach is equally applicable to any other score function as well. This is also true for the regularization idea behind RAPS presented in Angelopoulos et al. (2021). In § 4 we argued that APS is the most suitable choice. However, since Wijegunawardana et al. (2020) proposes a different scoring function called "margin scoring", we also evaluate our diffusion method on top of that as well. The score function is defined as $s(\boldsymbol{x}, y) = \pi(\boldsymbol{x})_y - \max_{y' \neq y} \pi(\boldsymbol{x})_{y'}$. Fig. 15 presents the evaluation of the diffusion and regularization variants of the margin scoring function. Again, we see that both provide similar improvements on top of the margin baseline. Since the margin score has issues with undercovering hard examples and overcovering easy examples, similar to TPS, we do not advocate for its use even though it appears to have good efficiency.

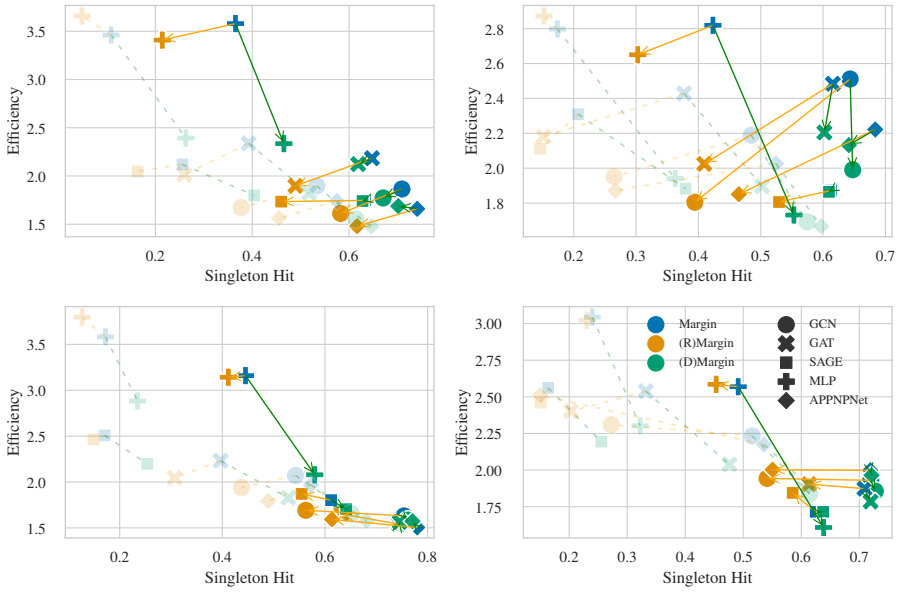

Figure 15. Conformal prediction with the margin score for the `CoraML` (top), and `CiteSeer` (bottom) datasets. The evaluation is based on fixed 92% coverage (left) and adaptive coverage (right). The transparent plot recalls the result of using APS as a reference score.

E.5. Sensitivity to λ, Efficiency and Accuracy

As a supplementary discussion to § 6.1 we present the result of the same experiment as Fig. 6 over some other dataset/model pairs. Fig. 16 shows that in almost every case the impact of the proposed diffusion on the accuracy is insignificant while it enhances the conformal set efficiency and singleton hit ratio. This gives a more intuitive sense that the diffusion framework results in more efficient sets by enhancing the approximation of the probabilities instead of increasing the model's accuracy.

E.6. Empty, Singleton and Multi-class Prediction Sets

For a given input, CP either returns a single class, a set of classes, or an empty set. One might prefer to increase the proportion of singleton sets as they can be applied without any further postprocessing. We compare the proportion of zero, singleton and multi-class prediction sets in Fig. 17 for different coverages spanning from a threshold below the model's accuracy (which is a trivial area for CP) up to near 100% coverage. This experiment shows that DAPS results in fewer empty sets and more singleton sets, which aligns with the higher singleton hit ratio covered in § 6. As shown in the figure, all CP methods tend to increase the number of empty-set predictions as the coverage guarantee decreases (and becomes lower than the model's accuracy). It is likely that with lower values of coverage guarantees, CP tends to result in a smaller false positive ratio. Note, the result varies across different runs of the experiment which is a direct result of label scarcity and the small calibration set, but the order is the same in the majority of observations.

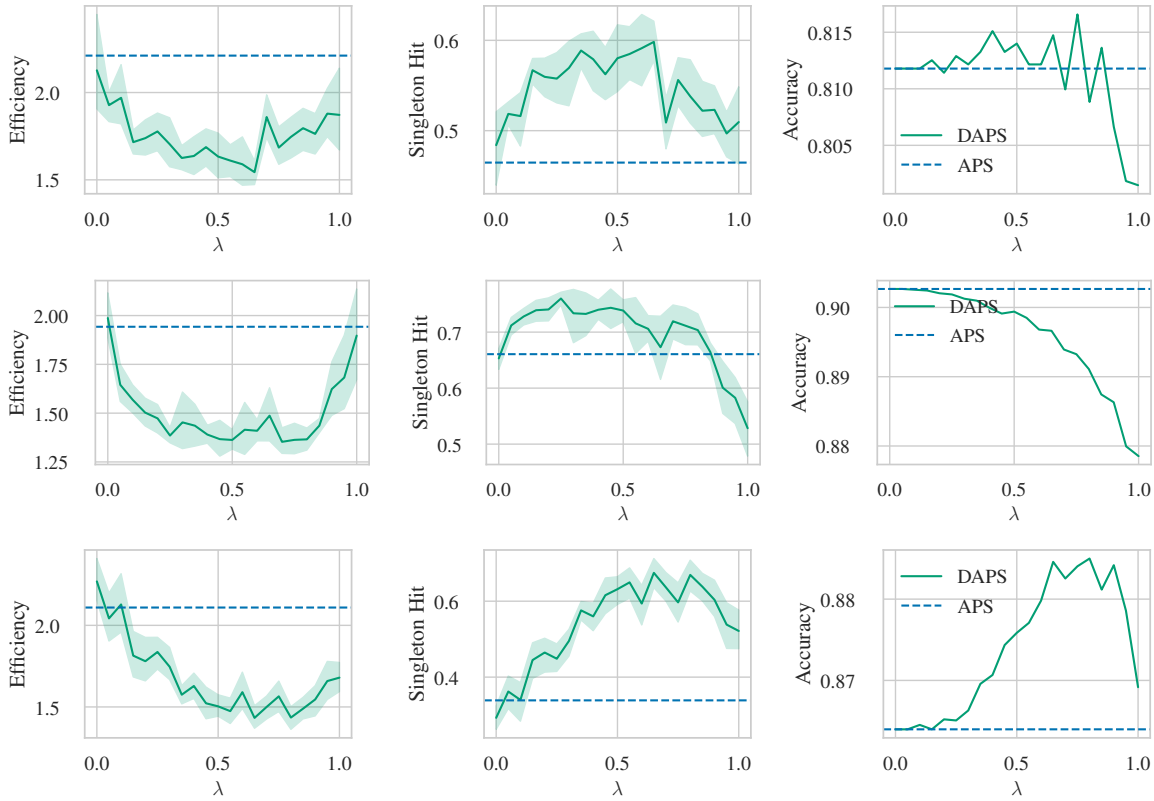

Figure 16. The effect of different λ values on (left) efficiency and (middle) singleton hit ratio of conformal prediction alongside its impact on (right) the accuracy of the model. Rows refer to experiments conducted on (top) `CoraML/GCN`, (middle) `CoauthorCS/APPNP`, and (bottom) `Amazon-Photo/GraphSAGE` respectively.

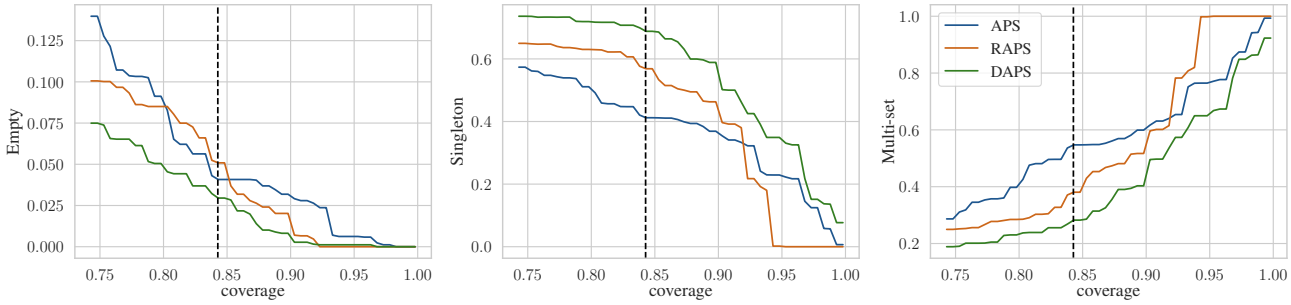

Figure 17. Comparison of APS, RAPS, and DAPS over different coverage guarantees for `CoraML/GCN` by the proportion of empty (left), singleton (middle) and multi-set (right) prediction sets. The dashed line in all plots show the model's accuracy over the test set.

E.7. Empirical Evaluation of Conditional Coverage

Evaluating CP's deviation from conditional coverage requires access to ground truth $p(\boldsymbol{x}, y)$. However, Romano et al. (2020) propose an approximation that is adaptable to limited data. The procedure involves searching for a slab ($S_{\boldsymbol{v},a,b} = \{\boldsymbol{x} \in \mathbb{R}^p : a < \boldsymbol{v}^T\boldsymbol{x} < b\}$) in which the empirical coverage is at its lowest. With a finite number of datapoints, in order to avoid finite-sample negative bias, a valid slab must contain an acceptable proportion of data points (e.g. 10%). The slab's identifier vector $\boldsymbol{v}$ is chosen uniformly at random in the feature space and is normalized. We chose the optimal parameters a^*, b^*, and $\boldsymbol{v}^*$ on 25% of the test data, using the rest to evaluate the coverage. See Romano et al. (2020) for more details. Fig. 18 shows that DAPS performs better than or on par with APS. This comparison is shown for different values of $1 - \alpha$.

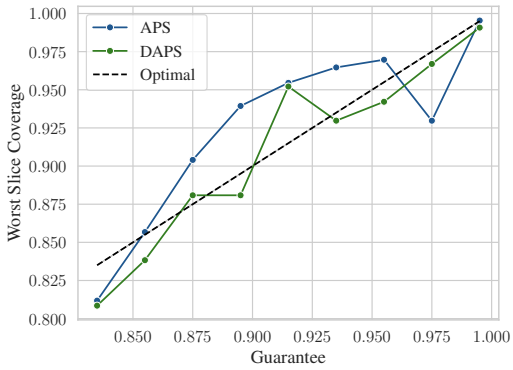

Figure 18. Comparison of worst-slice coverage for APS and DAPS across different coverage guarantees $1 - \alpha$. Results are shown for `CoraML/GCN`. Note approaches that are closer to the optimal dashed line are better.

E.8. Transformations for the Probability Space

Although we usually apply regularization (in RAPS), and diffusion as an enhancement on top of APS scores, we also examine the scenario where we apply those transformations over probability vectors (softmax outputs) as well. In both cases, we apply APS on top of the result and compare it with the conventional APS. Since APS accepts probability vectors as input, we need to represent the output of those transformations in a probability space. While, DAPS with $\lambda \in [0, 1]$ does not require any normalization to return a probability vector (since it is a convex combination), RAPS needs to be normalized (since the penalty changes the output range). To represent the regularization result in form of a probability vector, we apply a min-max normalization over elements, such that the minimum is equal to zero. Then we divide them by their summation. Fig. 19 shows the comparison. With RAPS, we observed a significant decrease in efficiency and singleton hits while DAPS is similar to conventional APS. It is better to apply both RAPS and DAPS in the score rather than the probability space.

E.9. Transductive Semi-Supervised Node Classification

With a brief review of experimental results provided in § 6, this section presents a comprehensive report of all results obtained. We compare DAPS alongside the baseline APS, and RAPS in the form of Pareto plots where two different metrics (effective set size and singleton hit) are evaluated at the same time. Blue points in the plot show baseline values (APS results) for different dataset/model pairs. Corresponding orange and green points are respectively showing RAPS, and 1-hop DAPS. Each baseline is connected to the two other results by an arrow of the same color. Fig. 14 shows the result on co-author networks and Fig. 20 shows a similar evaluation on the co-purchase networks. We also conducted the same experiment on `CoraFull`* and `CoraML`*, for which the result is reported in Fig. 21. Although we have shown the adaptability of DAPS to large networks like `OGBN Products` in Fig. 4, we also evaluated our method on another large `OGBN Arxiv` dataset.

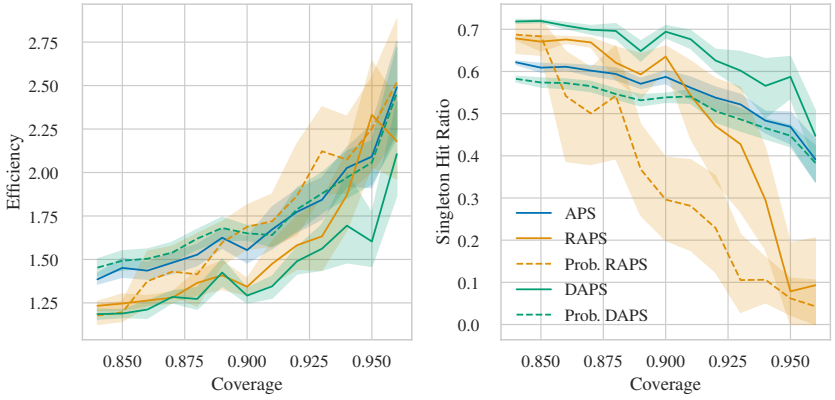

Figure 19. Comparison of DAPS and RAPS in probability space (dashed lines) and in APS (solid lines) score space.

As shown in Fig. 22, for OGBN Arxiv DAPS achieves a marginal enhancement and RAPS returns the most efficient sets among other methods. In spite of losing efficiency, DAPS outperforms RAPS in terms of singleton hit ratio for many coverage values. It is noteworthy that we did not expect a significant improvement for DAPS in this experiment since OGBN Arxiv does not have a high homophily score which is an essential requirement for this approach. For the same reason, we do not expect a significant enhancement in CoraFull* as well.

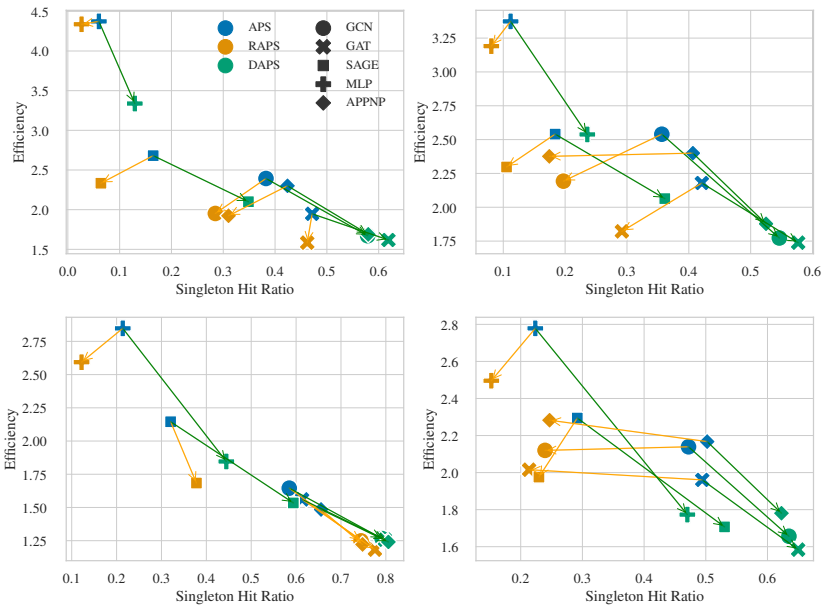

Figure 20. Pareto plot of different CP approaches for co-purchase datasets; APS, RAPS, and DAPS. The first column shows the result for the fixed 92% coverage and the right column shows the result for adaptive coverage. Rows from above to below refer to (1) Amazon Computers and (2) Amazon Photo datasets.

E.10. Other Variants of Semi-Supervised Node Classification

While our focus is on the transductive setting, we conduct additional experiments for other settings as well. Here we provide our experimental results on inductive and simultaneous inductive settings. For both cases, we trained our model on the inductive subgraph restricted to only training and validation nodes. Definitions and theoretical analysis of exchangeability for these settings are provided in § 3. For the inductive setting, we add calibration nodes to the training graph (creating the induced subgraph of training/validation/calibration nodes) during CP's calibration step. The rest of the evaluation nodes (with their connections) are added one at a time. Upon each modification, we update the model predictions. The prediction set for each node is computed immediately upon its arrival. These updates in the graph structure lead to distribution shift and conclusive violation of exchangeability as shown in Fig. 23. As a result, the $1-\alpha$ coverage is not guaranteed anymore.

For the simultaneous inductive setting, we utilized the same model. However, this time all nodes in the rest of the network (consisting of calibration nodes and unlabeled nodes) were connected to the training graph simultaneously. After this we update the model predictions using the final graph. As shown in Fig. 24 the coverage guarantee is still valid since exchangeability is not violated. The only difference between this setting and the transductive setting is the performance of the underlying model which is reflected in the conformal prediction metrics.

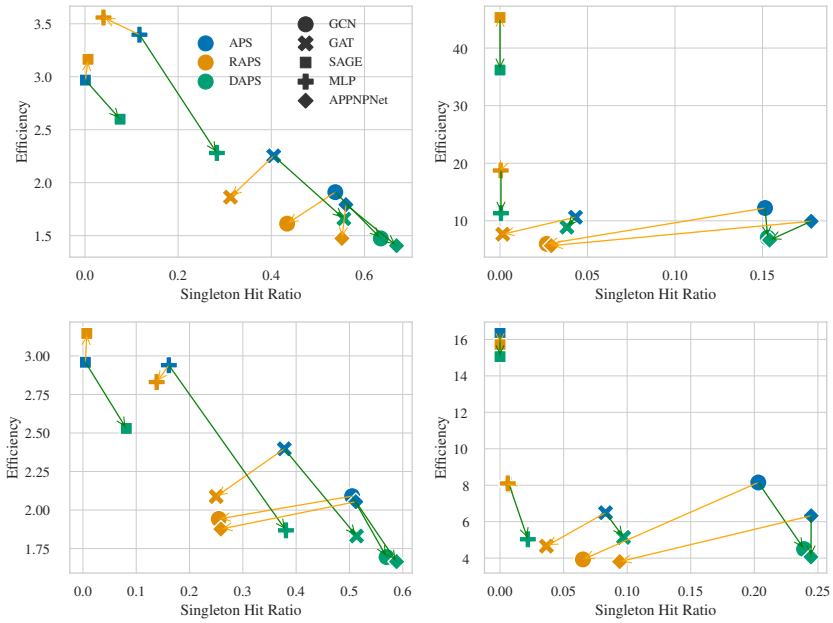

Figure 21. Pareto plot on the effective set size and singleton hit over the datasets `CoraML`* (first column) and `CoraFull`* (second column). (First row) shows the result for a fixed coverage (92%) and (second row) over the adaptive coverage.

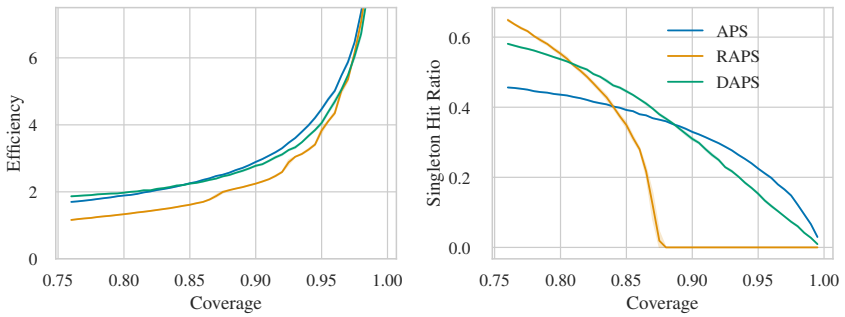

Figure 22. Efficiency (left) and singleton hit ratio (right) for the `OGBN Arxiv` dataset. Since we have less homophily DAPS sacrifices efficiency to improve singleton hits.

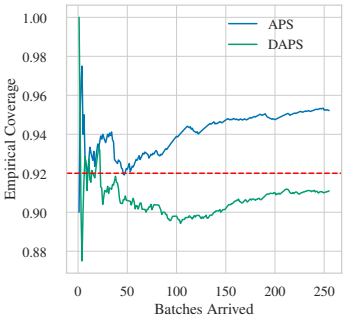

Figure 23. Inductive evaluation setting for `Cora-Ml/GCN`. Accuracy is 82%.

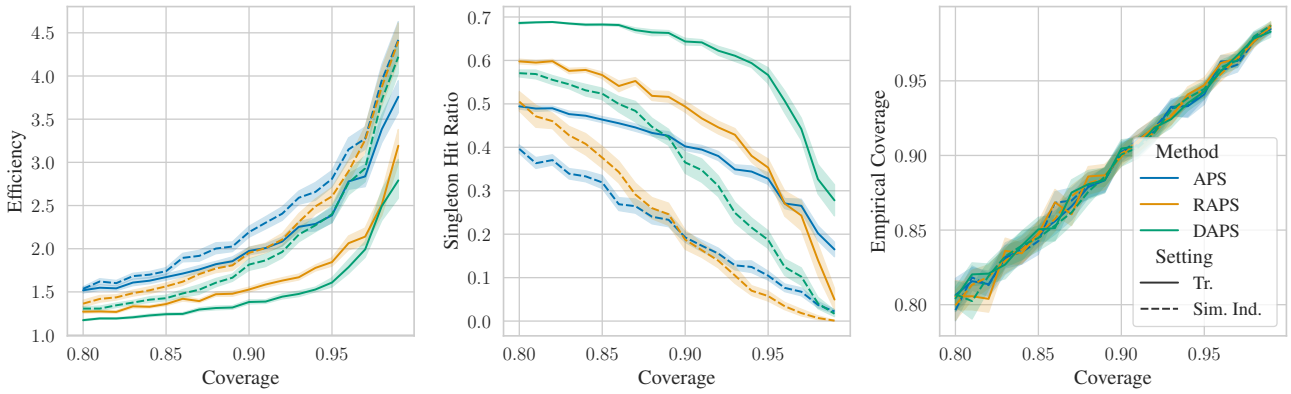

Figure 24. Simultaneous inductive setting on `CoraML` with a `GCN`. Solid lines are recalling the same result for transductive setting while dashed lines show the results of the simultaneous inductive setting. DAPS always leads to an improvement.

E.11. Discussion of Neighborhood Adaptive Prediction Sets (NAPS)

For the inductive semi-supervised node classification, Clarkson (2022) proposes NAPS (Neighborhood APS) which is built upon Barber et al. (2022) to adapt conformal prediction "beyond exchangeability". Applying conformal prediction without exchangeability leads to a gap between the empirical (real) coverage and the specified $1-\alpha$. This gap is bounded by

$$\text{Coverage Gap} \leq \frac{\sum_{i=1}^{n} w_i \cdot d_{TV}(Z, Z^i)}{1+\sum_{i=1}^{n} w_i} \tag{17}$$

where w_i corresponds to a weight over i-th datapoint ($w_i \in [0,1]$), and Z^i is the result of swapping i-th datapoint in the calibration set Z with the test point. Conclusively, a better weight assignment can result in a smaller coverage gap. In NAPS, a weight of $w_i = 1$ is assigned in case the test node is within k-hop distance of the i-th calibration node. The study suggests using NAPS only on large homophilous networks with $k = 1$ or 2. Note, the coverage gap is a theoretical property and it is not straightforward to compute or estimate it in practice.

As we discussed in § 3 NAPS is not applicable when the graph is sparse and the number of labeled nodes is limited. When the size of the calibration set is realistic, many test nodes will have no nodes with non-zero weights from the calibration set, making CP inapplicable to them. This happens regardless of the inductive or transductive setting. Even if somehow sparsity is not an issue, since NAPS assigns weights from $\{0,1\}$, a substantial proportion of the calibration set will be effectively discarded for each node. Consequently, the smaller calibration set leads to a less concentrated coverage distribution (i.e. less concentrated Beta) and less statistical power. It is worth noting that the claim by Clarkson (2022) that NAPS (or any other CP score) is not to applicable for transductive settings does not hold as shown by Proposition 1.

In Table 2 we compared DAPS with NAPS in the transductive setting for different calibration set sizes. We see that as we approach a realistic labeling budget, a considerable amount of nodes are excluded from the CP procedure (due to all zero weights). This observation holds true even when the algorithm is applied in the inductive setting using the same dataset since the source of this limitation remains the same.

E.12. Comparison Over Training Checkpoints

As CP is built on top of a model, to evaluate CP, it becomes important how well the model is trained. The question is how the enhancement made by DAPS (in comparison to conventional APS) changes during training. Since the categorical cross-entropy loss encourages the model to predict a concentrated "one-hot" label distribution, it is expected that the predicted probabilities become more over-confident when training the model for too many epochs. This may be an issue for APS. DAPS uses structural information to propagate the scores and overcomes this issue. To support this discussion we compared DAPS and APS for a GNN model during different checkpoints of the model's training. The results in Table 3 show the enhancements made by DAPS has an increasing trend of improvement while the model becomes more accurate.

F. Complementarity with Other Methods for Uncertainty Quantification

Another interesting insight is that if we are provided with a good uncertainty estimation model, applying conformal prediction on top should return even better results. To show this, we evaluate CP with the Graph Posterior Network (`GPN`) (Stadler et al., 2021) on `CoraML`. In particular, we compute the scores for the conformal prediction based on the class probabilities

Table 3. Comparison between DAPS and APS over different checkpoints during the model training.

Checkpoint	Acc	APS		DAPS		Difference	
		Eff Set Size	Singleton Hit	Eff Set Size	Singleton Hit	Eff Set Size	Singleton Hit
1	0.21	5.53	0.00	5.38	0.00	0.15	0
2	0.60	3.69	0.00	3.51	0.00	0.18	0
3	0.72	2.38	0.08	2.17	0.17	0.22	0.09
4	0.76	2.26	0.14	1.99	0.26	0.27	0.12
5	0.81	2.23	0.22	1.78	0.40	0.45	0.23
6	0.84	2.25	0.34	1.58	0.56	0.67	0.22

as a measure of the aleatoric uncertainty which is also the kind of uncertainty captured by conformal prediction. The results are shown in Fig. 25. Comparing GPN and GCN and we can see that even though their performance is close, we can note an improvement for the APS and RAPS methods when using an uncertainty-aware model like GPN. DAPS is on par for both underlying models, which suggests that our method captures the aleatoric uncertainty regardless of the model.

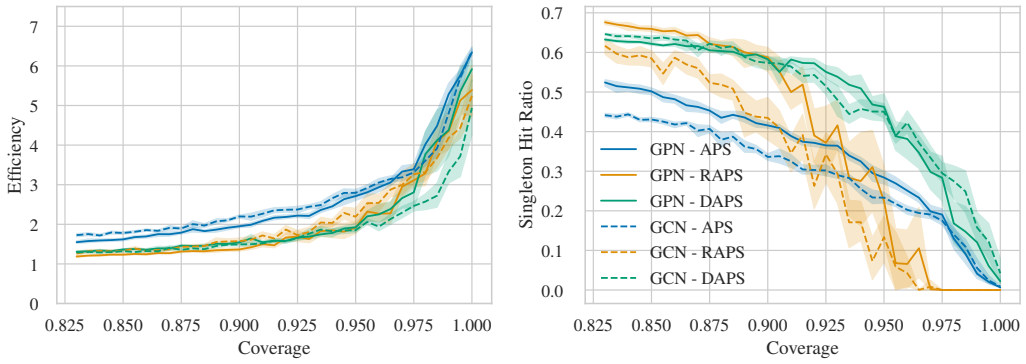

Figure 25. Comparison of the approaches on GPN and GCN model over CoraML dataset. While GPN helps APS and RAPS, especially for singleton hit, our approach DAPS is already able to provide an uncertainty quantification with a vanilla GCN.

G. More Details on Datasets and Models

Table 4 displays the statistics of the datasets used for the evaluation. The datasets marked by $\star$ refer to the largest component. Moreover, for CoraFull* we remove the classes (and the respective nodes) that have a number of samples less than 50 in order to have the same number of nodes per class in each train/validation split. Table 5 summarizes the model's accuracies on every dataset, and the selected adaptive coverage as explained in § E.3.

Table 4. Statistics of the datasets. The labeled node column includes all nodes that are assumed to be labeled in each experiment which is a summation of training, validation, tuning, and calibration nodes.

Dataset Name	Vertices	Attributes	Edges	Classes	Homophily	Labeled Nodes
CoraML	2995	2879	16316	7	78.85%	18.7%
CoraML*	2810	2879	15962	7	78.44%	14.95%
CoraFull*	18712	8710	124848	67	56.69%	20.41%
PubMed	19717	500	88648	3	80.23%	1.2%
CiteSeer	4230	602	10674	6	94.94%	10.8%
Coauthor CS	18333	6805	163788	15	80.80%	6.5%
Coauthor Physics	34493	8415	495924	5	93.14%	1.2%
Amazon Computers	13752	767	491722	10	77.72%	5.8%
Amazon Photo	7650	745	238162	8	82.72%	8.4%
OGBN Products	2449029	100	123718280	47	80.75%	11.24%
OGBN Arxiv	169343	128	1166243	40	65.51%	88.9%

Table 5. Accuracy report for datasets and models involved in the analysis.

Dataset	Model	Accuracy	Best Accuracy	Adaptive Coverage
CoraML	GCN	82.3 ± 0.9	83.9	94.0
	GAT	79.8 ± 3.1	84.4	93.1
	SAGE	79.9 ± 1.7	82.2	93.2
	MLP	63.4 ± 1.9	65.6	87.6
	APPNPNet	83.5 ± 0.8	84.9	94.4
PubMed	GCN	79.5 ± 2.0	82.0	93. 0
	GAT	77.8 ± 3.2	82.3	92.5
	SAGE	75.7 ± 2.2	79.8	91.7
	MLP	69.9 ± 0.9	71.6	89.8
	APPNPNet	79.4 ± 2.3	82.2	93.0
CiteSeer	GCN	83.7 ± 1.4	85.9	94. 5
	GAT	83.2 ± 0.9	84.2	94.3
	SAGE	78.2 ± 2.3	80.8	92.6
	MLP	62.6 ± 1.5	65.2	87.3
	APPNPNet	84.9 ± 1.2	87.0	94.9
Coauthor CS	GCN	90.9 ± 0.8	91.8	96.9
	GAT	88.6 ± 1.1	90.3	96.1
	SAGE	88.6 ± 1.2	90.2	96.1
	MLP	88.0 ± 0.6	88.9	95.9
	APPNPNet	91.1 ± 0.5	91.7	97.0
Coauthor Physics	GCN	92.2 ± 1.2	93.4	97.3
	GAT	91.1 ± 1.1	92.9	97.0
	SAGE	92.0 ± 0.7	92.9	97.3
	MLP	87.1 ± 1.3	89.8	95.6
	APPNPNet	93.1 ± 0.9	94.5	97.7
Amazon Computers	GCN	80.2 ± 2.9	83.0	93.3
	GAT	81.8 ± 1.9	84.6	93.8
	SAGE	74.6 ± 3.1	78.1	91.4
	MLP	60.0 ± 5.3	67.3	86.4
	APPNPNet	80.5 ± 2.2	84.3	93.4
Amazon Photo	GCN	89.0 ± 2.3	91.0	96.3
	GAT	89.3 ± 1.2	91.3	96.4
	SAGE	82.4 ± 3.5	86.4	94.0
	MLP	74.2 ± 2.2	76.4	91.2
	APPNPNet	89.4 ± 1.6	91.7	96.4