Uncertainty for Active Learning on Graphs

Dominik Fuchsgruber*
Technical University of Munich
d.fuchsgruber@tum.de

Tom Wollschläger*
Technical University of Munich
t.wollschlaeger@tum.de

Bertrand Charpentier
Technical University of Munich
b.charpentier@tum.de

Antonio Oroz
Technical University of Munich
a.oroz@tum.de

Stephan Günnemann
Technical University of Munich
s.guennemann@tum.de

Abstract

Uncertainty Sampling is an Active Learning strategy that aims to improve the data efficiency of machine learning models by iteratively acquiring labels of data points with the highest uncertainty. While it has proven effective for independent data its applicability to graphs remains under-explored. We propose the first extensive study of Uncertainty Sampling for node classification: **(1)** We benchmark Uncertainty Sampling beyond predictive uncertainty and highlight a significant performance gap to other Active Learning strategies. **(2)** We develop ground-truth Bayesian uncertainty estimates in terms of the data generating process and prove their effectiveness in guiding Uncertainty Sampling toward optimal queries. We confirm our results on synthetic data and design an approximate approach that consistently outperforms other uncertainty estimators on real datasets. **(3)** Based on this analysis, we relate pitfalls in modeling uncertainty to existing methods. Our analysis enables and informs the development of principled uncertainty estimation on graphs.

1 Introduction

Applications in machine learning are often limited by their data efficiency. Active Learning (AL) allows the learner to query an oracle to label specific data points considered *informative*. A popular strategy for query selection is Uncertainty Sampling (US) [1, 2] that relies on uncertainty estimates to measure the informativeness of each data point. Typically, one disentangles the (irreducible) *aleatoric* uncertainty and the (reducible) *epistemic* uncertainty which the *total* uncertainty about a prediction is composed of [3, 4]. This disentanglement is particularly important for AL: The learner might not benefit much from labeling instances with high irreducible uncertainty while acquiring knowledge about data points for which uncertainty stems from reducible sources can be highly informative [5].

For independent and identically distributed (i.i.d.) data, disentangled US methods for AL have demonstrated high data efficiency in benchmarks [1, 2, 5–8]. Existing efforts in AL for interdependent data like graphs neglect the compositional nature of uncertainty [9–13] and limit themselves to a measure of total uncertainty. This leaves it unclear (i) to which extent uncertainty estimators can effectively inform US for graph data, and (ii) whether disentangling aleatoric and epistemic uncertainty has similar benefits to AL as in i.i.d. settings. The complex nature of graph data makes investigating US methods for AL particularly challenging. Uncertainty estimates should not only capture information about node features independently but also model information about their relationships [14].

In this work, we benchmark state-of-the-art uncertainty estimators (see Relatded Work in Appendix C) against traditional AL strategies and find that US (as well as many other approaches) fall short of surpassing random sampling. We formally approach the question of whether US methods for graphs align well with the AL objective: We derive novel ground-truth uncertainty measures from the

*Equal contribution.

D. Fuchsgruber et al., Uncertainty for Active Learning on Graphs (Extended Abstract). Presented at the Third Learning on Graphs Conference (LoG 2024), Virtual Event, November 26–29, 2024.

underlying data-generating process and disentangle its aleatoric and epistemic components. This generative perspective enables a principled perspective on handling instance interdependence in uncertainty estimation that we leverage for AL: We prove that querying epistemically uncertain nodes is equivalent to maximizing the relative gain in confidence a Bayesian classifier puts on correctly predicting all unlabeled nodes.

The main **contributions** of our work are:

- We provide the first extensive AL benchmark[2] for node classification including both state-of-the-art uncertainty estimators for US and traditional acquisition strategies. Many of the latter as well as most uncertainty estimators do not outperform random acquisition.
- We derive ground-truth aleatoric and epistemic uncertainty for a Bayesian classifier and formally prove the alignment of US with AL.
- We empirically confirm the efficacy of epistemic US both on synthetic and real data by employing an approximation to our proposed ground-truth uncertainty that outperforms other uncertainty estimators off-the-shelf. This enables future uncertainty estimators to achieve competitive AL performance by building on the principles of our work.

2 Ground-Truth Uncertainty from the Data Generating Process

As we show in Appendix D, in contrast to i.i.d. domains, neither traditional nor uncertainty-based AL strategies consistently outperform random queries. This motivates a thorough theoretical examination of whether uncertainty sampling and AL align well for non-i.i.d. problems. However, evaluating the quality of uncertainty estimates is inherently difficult as generally, ground-truth values are unavailable for both the overall predictive uncertainty u^{total} and its constituents $u^{\text{alea}}, u^{\text{epi}}$. Additionally, since epistemic uncertainty pertains to the knowledge of the classifier, it cannot be defined in a model-agnostic manner. Therefore, we analyze uncertainty from the perspective of the underlying (potentially unknown) data-generating process $p(\boldsymbol{X}, \boldsymbol{A}, \mathbf{y})$ with respect to a Bayesian classifier. This lends itself to a definition of ground-truth uncertainty. In the following, we propose confidence measures and relate them to uncertainty as their inverse $\mathrm{conf} := \mathrm{u}^{-1}$.

Definition 2.1. We define the parametrized Bayesian classifier $f^*_\theta(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y}^{\text{gt}}_{\mathcal{O}})$ in terms of the data generating process $p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y})$ as the prediction $\mathbf{c} \in [C]^{|\mathcal{U}|}$ that maximizes:

$$\mathbb{E}_{p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}^{\text{gt}}_{\mathcal{O}})} \left[\mathbb{P} \left[\mathbf{y}_{\mathcal{U}} = \mathbf{c} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}} = \mathbf{y}^{\text{gt}}_{\mathcal{O}}, \theta \right] \right] \tag{1}$$

Here, we denote with $\mathbf{y}^{\text{gt}}_{\mathcal{O}}$ the labels of already observed instances. The predictive distribution $p(\mathbf{y}_{\mathcal{U}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}^{\text{gt}}_{\mathcal{O}})$ encapsulates the total confidence of f^*_θ. The classifier averages its prediction according to a learnable posterior distribution $p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})$ over its parameters, e.g. weights of a GNN. Marginalization yields the total confidence $\mathrm{conf}^{\text{total}}$.

Definition 2.2. We define the total confidence $\mathrm{conf}^{\text{total}}(i, c)$ of f^*_θ in predicting $\mathbf{y}_i = c$ as:

$$\mathbb{E}_{p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}^{\text{gt}}_{\mathcal{O}})} \left[\mathbb{P} \left[\mathbf{y}_i = c \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}} = \mathbf{y}^{\text{gt}}_{\mathcal{O}}, \theta \right] \right] \tag{2}$$

Intuitively, the total confidence captures aleatoric factors through the inherent randomness in the data generating process $p(\mathbf{y}, \boldsymbol{A}, \boldsymbol{X})$. Epistemic uncertainty is incorporated by conditioning on a limited set of observed labels $\mathbf{y}^{\text{gt}}_{\mathcal{O}}$. With a growing labeled set irreducible errors will increasingly dominate total predictive uncertainty. In the extreme case where all labels but one have been observed, i.e. $\mathcal{O} = \mathcal{V} \setminus \{v_i\}$, remaining uncertainty only stems from aleatoric factors.

Definition 2.3. We define the aleatoric confidence $\mathrm{conf}^{\text{alea}}(i, c)$ of f^*_θ in predicting $\mathbf{y}_i = c$ as:

$$\mathbb{E}_{p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}^{\text{gt}}_{-i})} \left[\mathbb{P} \left[\mathbf{y}_i = c \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{-i} = \mathbf{y}^{\text{gt}}_{-i}, \theta \right] \right] \tag{3}$$

Here, we denote with $\mathbf{y}_{-i} = \mathbf{y}^{\text{gt}}_{-i}$ that all nodes excluding the predicted node i are observed as their true values. All remaining lack of confidence is deemed irreducible. Lastly, we define epistemic confidence by comparing aleatoric factors to the overall confidence. As both are defined probabilistically, we consider their ratio:

[2] Find our code at https://figshare.com/s/95911d5944f57d7ab445?file=44372885

Definition 2.4. We define epistemic confidence of f_θ^* in predicting label $\mathbf{y}_i = c$ as:

$$\operatorname{conf}^{\text{epi}}(i, c) := \operatorname{conf}^{\text{alea}}(i, c) / \operatorname{conf}^{\text{total}}(i, c) \tag{4}$$

These definitions directly imply a notion of uncertainty: $\mathrm{u}^{\text{epi}}(i, c) = \mathrm{u}^{\text{total}}(i, c) / \mathrm{u}^{\text{alea}}(i, c)$. Epistemic US thus labels node i when the associated total uncertainty $\mathrm{u}^{\text{total}}(i, \mathbf{y}_i^{\text{gt}})$ is large compared to its aleatoric uncertainty $\mathrm{u}^{\text{alea}}(i, \mathbf{y}_i^{\text{gt}})$. It favors uncertain nodes *when the uncertainty stems from non-aleatoric sources*.

Remark 2.5. In most applications, including US, monotonoic transformations of an uncertainty estimator do not affect its behaviour. Therefore, uncertainty can equivalently be defined as a difference between log-likelihoods instead of likelihood ratios. This definition recovers the well-established additive nature of uncertainty: $\log \mathrm{u}^{\text{total}}(i, \mathbf{y}_i^{\text{gt}}) - \log \mathrm{u}^{\text{alea}}(i, \mathbf{y}_i^{\text{gt}}) = \log \mathrm{u}^{\text{epi}}(i, \mathbf{y}_i^{\text{gt}})$. The same holds for logarithmic confidence definitions. Notably, the core result of our work, which we state next, also holds when defining uncertainty in terms of log-likelihoods.

Theorem 2.6. *Epistemic uncertainty* $\mathrm{u}^{\text{epi}}(i, \mathbf{y}_i^{\text{gt}})$ *of a node* i *is equivalent to the relative gain its acquisition provides to the posterior over the remaining true labels:*

$$\mathrm{u}^{\text{epi}}(i, \mathbf{y}_i^{\text{gt}}) = \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\text{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\text{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\text{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]}$$

Hence, acquiring the most epistemically uncertain node is an optimal AL strategy for f_θ^*.

We provide a proof of Theorem 2.6 in Appendix B. Here, we refer to all unobserved labels excluding $\mathbf{y}_i$ as $\mathbf{y}_{\mathcal{U}-i}$. The ratio that is optimized by epistemic US corresponds to the relative increase in the posterior of the true unobserved labels. That is, it compares the probability the classifier f_θ^* assigns to the remaining unobserved true labels after acquiring the ground-truth label $\mathbf{y}_i^{\text{gt}}$ of node i as opposed to not acquiring its label. High values indicate that the underlying classifier will be significantly more likely to predict the true labels of the remaining nodes after the corresponding query. Thus, a query that maximizes epistemic uncertainty will push the classifier toward predicting the true labels for all remaining unlabeled nodes. This holds for any Bayesian classifier that specifies a posterior $p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})$ over the parameters of the generative process. For example, fitting the parameters of a GNN is an instance of this framework, where all probability mass is put on one estimate of θ. Approaches like Bayesian GNNs explicitly specify this posterior distribution. However, computing exact disentangled uncertainty requires access to unavailable labels $\mathbf{y}_{\mathcal{U}}$ and is therefore impractical. Our analysis motivates the development of tractable approximations to these quantities. Novel US approaches can directly benefit from the theoretical optimality guarantees that this work provides.

3 Uncertainty Sampling with Ground-Truth Uncertainty

To support these theoretical claims, we employ US using the proposed ground-truth epistemic uncertainty as an acquisition function. The uncertainty measures depend on the data-generating process, which is not known in real-world applications. Furthermore, computing aleatoric uncertainty necessitates access to unavailable labels $\mathbf{y}^{\text{gt}}$. We address these issues by proposing a simple approximate disentangled uncertainty estimator that closely follows our analysis. Our results underline the applicability of our theoretical findings to real-world problems. Additionally, we supply an evaluation on synthetic data for which the generating process is known in Appendix H.1. In the following, we outline the concepts behind two different paradigms to approximate ground-truth uncertainty. We provide an extensive formal description of both algorithms in Appendix J.

Multiple Pseudo-Labels (MP). According to Equation (4), one way to obtain epistemic confidence is to approximate total and aleatoric confidence separately and take their ratio. Total confidence is defined in terms of the unknown underlying data generating process $p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y})$ (Definition 2.2). We approximate it directly from the predictive distribution of the classifier f_θ, thereby assuming f_θ

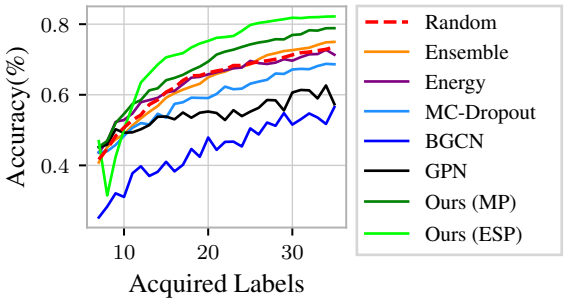


Figure 1: Approximation of disentangled uncertainty against baselines on CoraML. Our framework works well both using multiple pseudo-labels (MP) or taking an expectation over each of them (ESP).

to implicitly model the data distribution: $\mathrm{conf}^{\mathrm{total}}(i, c) \approx f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c}$

Aleatoric confidence (Definition 2.3) also conditions the marginal distribution over $\mathbf{y}_i$ on unavailable labels $\mathbf{y}_{\mathcal{U}-i}$. Therefore, we use the predictions of f_θ as pseudo-labels $\hat{\mathbf{y}}_i = \mathrm{argmax}_c\, f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c}$ and temporarily augment the dataset with *multiple pseudo-labels* by setting $\mathbf{y}_{\mathcal{U}-i} = \hat{\mathbf{y}}_{\mathcal{U}-i}$. The predictive distribution of an auxiliary classifier $f_{\hat{\theta}_i}$ trained on this dataset approximates aleatoric confidence: $\mathrm{conf}^{\mathrm{alea}}(i, c) \approx f_{\hat{\theta}_i}(\boldsymbol{A}, \boldsymbol{X})_{i,c}$.

Expected Single Pseudo-Label (ESP). The right-hand side of Theorem 2.6 is equivalent to epistemic uncertainty and can also be approximated as a proxy for acquisition. For both the numerator and denominator, we estimate the joint probabilities over $\mathbf{y}_{\mathcal{U}-i}$ as a product of marginal probabilities given by a classifier. The probability of the denominator, can be calculated directly from the predictive distribution of f_θ (see Appendix J): $\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \hat{\mathbf{y}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right] \propto f_\theta(\boldsymbol{A}, \boldsymbol{X})^{-1}_{i,\hat{\mathbf{y}}_i}$

Similarly to MP, we compute the probability in the numerator by training a separate classifier after augmenting the training data with a label for $\mathbf{y}_i$. Since for each node i, we condition only on one additional label $\mathbf{y}_i$ at a time, we do not use the pseudo-label $\hat{\mathbf{y}}_i$ but instead take an *expectation* over all realizations $c \in [C]$ with respect to the predictive distribution of f_θ.

Results. We visualize the AL curves of epistemic US using these approaches in Figure 1 and Appendix J. On all datasets, our proposed approximate epistemic uncertainty outperforms other US techniques, often matching the best-performing non-US AL strategy. We observe that ESP approximations perform better in most cases. While both algorithms estimate the same quantity they rely on different approximations. Therefore, it is expected that they behave differently in practice. While MP substitutes the labels of all unobserved nodes $\mathcal{U}$ in the dataset with pseudo-labels simultaneously, the ESP method only considers one pseudo-label at a time and factors in the belief of f_θ by taking an expectation over possible labels. This makes it less prone to erroneous pseudo-labels which, especially for small training sets, are to be expected. Both MP and ESP approximations are also exposed to miscalibration of f_θ and $f_{\hat{\theta}}$, a well-known issue in GNN training [15]. As we discussed in Section 2, message passing GNNs may also not be suited to model the underlying data distribution well. Despite these limitations, approximating and disentangling uncertainty by following our theoretical analysis elevates US from often sub-random to competitive performance levels.

Table 1: Average AUC (↑) of our proposed approximate ground-truth uncertainty versus existing epistemic US. The best estimator is boldfaced and the runner-up is underlined. Our approximate uncertainty consistently outperforms existing epistemic estimators.

	Ensemble	MC-Dropout	Energy	GPN	BGCN	Ours (MP)	Ours (ESP)
CoraML	63.47	59.17	63.97	54.75	44.45	<u>68.16</u>	**71.45**
Citeseer	82.94	78.86	81.59	65.31	58.68	**83.96**	<u>83.43</u>
Pubmed	<u>63.70</u>	58.67	59.64	58.82	55.19	n/f	**64.36**
Photos	<u>84.46</u>	72.42	74.66	54.78	70.83	75.37	**85.52**
Computers	<u>68.38</u>	51.02	59.62	39.21	58.64	n/f	**72.54**

While the performance our approximate approach to ground-truth uncertainty consistently outperforms all existing US methods, our main point is not to propose this as a novel AL strategy. Instead, these results underline the high applicability of our theoretical findings to the development of uncertainty estimators on graphs. We show that epistemic uncertainty is a provably well-suited acquisition proxy for AL on graphs. It identifies highly informative queries in practice when taking into account the theoretical insights of our work. We remark that the paradigm of estimating uncertainty by approximating an unknown generative process may be particularly useful for the domain of graphs as it enables a principled of how interdependence should be accounted for: The non-i.i.d. nature of the problem is implicitly accounted for by estimating a non-i.i.d. data distribution.

4 Limitations

Our work highlights the potential of US for AL on graphs and the role of epistemic uncertainty as an optimal guide. While we propose an effective off-the-shelf approximation of our theoretical analysis our goal is not primarily to provide an efficient AL strategy to be deployed in-practice. We leave the development of uncertainty estimators that align with our theoretical work to future work.

5 Conclusion

Our study sheds light on the potential and challenges of AL on graphs. We introduce ground-truth uncertainty estimates for node classification and prove the alignment of US with AL. An empirical analysis on real data shows that epistemic US is a highly effective AL strategy when uncertainty is properly disentangled compared to existing strategies. We believe this to be a highly relevant result for uncertainty estimation on graphs which, so far, neglected AL as an evaluation criterion. Our work, thus, lays the necessary theoretical groundwork for developing uncertainty estimators on graphs.

Impact Statement

Our paper aims at advancing the field of uncertainty estimation on interdependent data which contributes to making AI systems more reliable. By formally establishing a connection to Active Learning, we enable novel acquisition strategies in the domain of graphs to be more data efficient which in turn can help in reducing their negative environmental impact. Nonetheless, advances in the field of Machine Learning are always exposed to non-benign usage. While we acknowledge the various well-established societal and ethical implications of such research we believe the potential positive impacts of our paper outweigh these risks. We encourage both researchers and practitioners building on our analysis to take special care not only for potential misuse but also to thoroughly validate uncertainty estimators that our work proposes, especially in safety-critical domains.

References

[1] William H. Beluch, Tim Genewein, Andreas Nurnberger, and Jan M. Kohler. The power of ensembles for active learning in image classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018. doi: 10.1109/CVPR.2018.00976. 1, 13, 14

[2] Ajay J. Joshi, Fatih Murat Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379, 2009. 1

[3] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31:105–112, 2009. ISSN 0167-4730. doi: https://doi.org/10.1016/j.strusafe.2008.06.020. 1, 9

[4] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision?, 2017. 1, 9, 11

[5] Vu-Linh Nguyen, Mohammad Shaker, and Eyke Hüllermeier. How to measure uncertainty in uncertainty sampling for active learning. *Machine Learning*, 111, 2022. doi: 10.1007/s10994-021-06003-9. 1, 11, 13

[6] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. *CoRR*, abs/1703.02910, 2017. 13, 14

[7] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning, 2019. 12, 13, 14

[8] Sebastian Schmidt and Stephan Günnemann. Stream-based active learning by exploiting temporal properties in perception with temporal predicted loss, 2023. 1, 11

[9] Xiaojin Zhu, John Lafferty, and Zoubin Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining*, volume 3, 2003. 1, 11

[10] Kwang-Sung Jun and Robert Nowak. Graph-based active learning: A new look at expected error minimization. In *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1325–1329. IEEE, 2016. 11

[11] Florence Regol, Soumyasundar Pal, Yingxue Zhang, and Mark Coates. Active learning on attributed graphs via graph cognizant logistic regression and preemptive query generation. In *International Conference on Machine Learning*, pages 8041–8050. PMLR, 2020. 11, 12, 14, 15

[12] Yuexin Wu, Yichong Xu, Aarti Singh, Yiming Yang, and Artur Dubrawski. Active learning for graph neural networks via node feature propagation, 2021.

[13] Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. Information gain propagation: A new way to graph active learning with soft labels. In *ICLR*, 2022. 1

[14] Maximilian Stadler, Bertrand Charpentier, Simon Geisler, Daniel Zügner, and Stephan Günnemann. Graph posterior network: Bayesian predictive uncertainty for node classification. *Advances in Neural Information Processing Systems*, 34:18033–18048, 2021. 1, 11, 13, 14, 15

[15] Hans Hao-Hsun Hsu, Yuesong Shen, Christian Tomani, and Daniel Cremers. What makes graph neural networks miscalibrated? *Advances in Neural Information Processing Systems*, 35: 13775–13786, 2022. 4, 23

[16] Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning, 2018. 9

[17] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning, 2021. 11

[18] Xueying Zhan, Qingzhong Wang, Kuan-hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan. A Comparative Survey of Deep Active Learning, 2022. 11

[19] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach, 2018. 11, 12

[20] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational Adversarial Active Learning, 2019. 11

[21] Changjian Shui, Fan Zhou, Christian C Gagné, and Boyu Wang. Deep Active Learning: Unified and Principled Method for Query and Training, 2020. 11

[22] C. E. Shannon. A mathematical theory of communication, 1948. 11

[23] Dan Wang and Yi Shang. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014. doi: 10.1109/IJCNN.2014.6889457. 11

[24] Neil Houlsby, Ferenc HuszÃ¡r, Zoubin Ghahramani, and MÃ¡tÃ© Lengyel. Bayesian Active Learning for Classification and Preference Learning, 2011. 11

[25] Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 11

[26] Vu-Linh Nguyen, SÃ©bastien Destercke, and Eyke HÃ¼llermeier. Epistemic Uncertainty Sampling, 2019. 11

[27] Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson. Deep Similarity-Based Batch Mode Active Learning with Exploration-Exploitation. In *2017 IEEE International Conference on Data Mining (ICDM)*, 2017. doi: 10.1109/ICDM.2017.67. 11

[28] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds, 2020. 11

[29] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019. 11

[30] Gautam Dasarathy, Robert Nowak, and Xiaojin Zhu. S2: An efficient graph based active learning algorithm with application to nonparametric classification. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 503–522, Paris, France, 03–06 Jul 2015. PMLR. URL `https://proceedings.mlr.press/v40/Dasarathy15.html`. 11

[31] Jifan Zhang, Julian Katz-Samuels, and Robert Nowak. GALAXY: Graph-based active learning at the extreme. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 26223–26238.

PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhang22k.html. 11

[32] Yifei Ma, Roman Garnett, and Jeff Schneider. σ-optimality for active learning on gaussian random fields. *Advances in Neural Information Processing Systems*, 26, 2013. 11

[33] Ming Ji and Jiawei Han. A variance minimization criterion to active learning on graphs. In *Artificial Intelligence and Statistics*, pages 556–564. PMLR, 2012.

[34] Dimitris Berberidis and Georgios B Giannakis. Data-adaptive active sampling for efficient graph-cognizant classification. *IEEE Transactions on Signal Processing*, 66:5167–5179, 2018. 11

[35] Yayong Li, Jie Yin, and Ling Chen. Seal: Semisupervised adversarial active learning on attributed graphs. *IEEE Transactions on Neural Networks and Learning Systems*, 32:3136–3147, 2020. 11, 13, 15

[36] Shengding Hu, Zheng Xiong, Meng Qu, Xingdi Yuan, Marc-Alexandre Côté, Zhiyuan Liu, and Jian Tang. Graph policy network for transferable active learning on graphs, 2020. 11

[37] Wentao Zhang, Yexin Wang, Zhenbang You, Meng Cao, Ping Huang, Jiulong Shan, Zhi Yang, and Bin Cui. Information gain propagation: a new way to graph active learning with soft labels, 2022. 11

[38] Kaushalya Madhawa and Tsuyoshi Murata. Active learning for node classification: An evaluation. *Entropy*, 22, 2020. 11, 12

[39] Hongyun Cai, Vincent W. Zheng, and Kevin Chen-Chuan Chang. Active learning for graph embedding, 2017. 11, 12

[40] Li Gao, Hong Yang, Chuan Zhou, Jia Wu, Shirui Pan, and Yue Hu. Active discriminative network representation learning. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2142–2148. International Joint Conferences on Artificial Intelligence Organization, 2018. doi: 10.24963/ijcai.2018/296. 12

[41] Xiaoting Li, Yuhang Wu, Vineeth Rakesh, Yusan Lin, Hao Yang, and Fei Wang. Smartquery: An active learning framework for graph neural networks through hybrid uncertainty reduction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4199–4203, 2022. 11

[42] Manali Sharma and Mustafa Bilgic. Evidence-based uncertainty sampling for active learning. *Data Mining and Knowledge Discovery*, 31:164–202, 2016. 11

[43] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2017. 11, 14, 15

[44] Johannes Klicpera, Aleksandar Bojchevski, and Stephan Günnemann. Personalized embedding propagation: Combining neural networks on graphs with personalized pagerank. *CoRR*, abs/1810.05997, 2018. 11, 15

[45] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles, 2017. 11

[46] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning, 2016. 11

[47] Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *International Conference on Learning Representations*, 2020. 11

[48] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks, 2015. 11, 15

[49] Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pages 1–77, 2023. 11

[50] Tom Wollschläger, Nicholas Gao, Bertrand Charpentier, Mohamed Amine Ketata, and Stephan Günnemann. Uncertainty estimation for molecules: Desiderata and methods, 2023. 12

[51] Zhao-Yang Liu, Shao-Yuan Li, Songcan Chen, Yao Hu, and Sheng-Jun Huang. Uncertainty aware graph gaussian process for semi-supervised learning. 34, 2020. 12

[52] Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. Matérn gaussian processes on graphs. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130, pages 2593–2601. PMLR, 2021. 12

[53] Sanghamitra Bandyopadhyay, Ujjwal Maulik, Lawrence B Holder, Diane J Cook, and Lise Getoor. Link-based classification. *Advanced methods for knowledge discovery from complex data*, pages 189–207, 2005. 12, 14

[54] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29:93–93, 2008. 12

[55] C Lee Giles, Kurt D Bollacker, and Steve Lawrence. Citeseer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, pages 89–98, 1998. 12, 14

[56] Galileo Namata, Ben London, Lise Getoor, Bert Huang, and U Edu. Query-driven active surveying for collective classification. In *10th international workshop on mining and learning with graphs*, volume 8, page 1, 2012. 12, 14

[57] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 43–52, 2015. 12

[58] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR, 2019. 13, 24

[59] Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection, 2021. 13

[60] Qitian Wu, Yiting Chen, Chenxiao Yang, and Junchi Yan. Energy-based out-of-distribution detection for graph neural networks, 2023. 13, 14

[61] Andrew Kachites McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3:127–163, 2000. 14

[62] Yash Deshpande, Andrea Montanari, Elchanan Mossel, and Subhabrata Sen. Contextual stochastic block models. In *Neural Information Processing Systems*, 2018. 14

[63] Lukas Gosch, Daniel Sturm, Simon Geisler, and Stephan Günnemann. Revisiting robustness in graph machine learning, 2023. 15

[64] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 15

[65] Mahendra Mariadassou, Stéphane Robin, and Corinne Vacher. Uncovering latent structure in valued graphs: a variational approach. 2010. 17

[66] Tommi S Jaakkola. Tutorial on variational approximation methods. *Advanced mean field methods: theory and practice*, pages 129–159, 2000. 17, 18

[67] John Palowitch, Anton Tsitsulin, Brandon Mayer, and Bryan Perozzi. GraphWorld. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2022. doi: 10.1145/3534678.3539203. 18

[68] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019. 24

A Background

AL for Semi-Supervised Node classification. Let $G = (\mathcal{V}, \mathcal{E})$, be a graph with a set of nodes $\mathcal{V}$ and edges $\mathcal{E} \subseteq \{\{i, j\} : i, j \in \mathcal{V}\}$. With $n = |V|$ number of nodes the adjacency matrix is defined as $\boldsymbol{A} \in \{0, 1\}^{n \times n}$, where $A_{ij} = A_{ji} = 1$ if there exists an edge between node i and node j. The features are represented as matrix $\boldsymbol{X} \in \mathbb{R}^{n \times d}$. In node classification, every node has a label represented by a vector $\mathbf{y} \in [C]^n$. We decompose the set of all nodes into $\mathcal{V} = \mathcal{U} \cup \mathcal{O}$ where $\mathcal{O}$ contains nodes with observed labels $\mathbf{y}_{\mathcal{O}}$ and $\mathcal{U}$ contains remaining unobserved nodes $\mathbf{y}_{\mathcal{U}}$ that we want to infer, where $\mathcal{V} = \mathcal{U} \cup \mathcal{O}$. We consider pool-based AL: In each iteration, the learner queries an oracle for a label $\mathbf{y}_i$ from the set of unobserved labels $\mathbf{y}_{\mathcal{U}}$, adds it to the set of observed labels $\mathbf{y}_{\mathcal{O}}$ and retrains the model.

Uncertainty in Machine Learning. Alongside the predicted label $\mathbf{y}_i$, it is crucial to consider the associated uncertainty [3, 4]. Aleatoric uncertainty $\mathrm{u}^{\mathrm{alea}}$ is the inherent uncertainty that comes from elements like experimental randomness. It can not be reduced by acquiring more data. On the other hand, Epistemic uncertainty $\mathrm{u}^{\mathrm{epi}}$ reflects knowledge gaps, which can be addressed by data acquisition. A commonly employed conceptualization [16] defines the total predictive uncertainty $\mathrm{u}^{\mathrm{total}}$ as encapsulating both reducible and irreducible factors, i.e. $\mathrm{u}^{\mathrm{total}} = \mathrm{u}^{\mathrm{epi}} + \mathrm{u}^{\mathrm{alea}}$.

B Proofs

Theorem 1. Epistemic uncertainty $\mathrm{u}^{\mathrm{epi}}(i, \mathbf{y}_i^{\mathrm{gt}})$ of a node i is equivalent to the relative gain its acquisition provides to the posterior over the remaining true labels:

$$\mathrm{u}^{\mathrm{epi}}(i, \mathbf{y}_i^{\mathrm{gt}}) = \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]}$$

Hence, acquiring the most epistemically uncertain node is an optimal AL strategy for f_θ^*.

Proof.

$$\begin{aligned}
\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}}) &= \int \mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{-i} = \mathbf{y}_{-i}^{\mathrm{gt}}, \theta\right] p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{-i} = \mathbf{y}_{-i}^{\mathrm{gt}}) d\theta \\
&= \mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{-i} = \mathbf{y}_{-i}^{\mathrm{gt}}\right] \\
&= \mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} | \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}}\right] \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]} \mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mid \boldsymbol{X}, \boldsymbol{A}, \mathbf{y}_{\mathcal{O}}\right] \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]} \mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}}) \\
\mathrm{u}^{\mathrm{epi}}(i, \mathbf{y}_i^{\mathrm{gt}}) &= \frac{\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})}{\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})} = \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]}
\end{aligned}$$

First, we insert our definition of aleatoric confidence and marginalize θ. Then, we split $\mathbf{y}_{-i}$ into two parts: observed $\mathbf{y}_{\mathcal{O}}$ and unobserved $\mathbf{y}_{\mathcal{U}-i}$. As we assign the ground-truth labels to both, the exact partition is not relevant. Next, we use Bayes law to get a distribution over the unobserved node labels $\mathbf{y}_{\mathcal{U}}$. Similarly to the first step, we marginalize θ to obtain $\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})$. In the last step, we see that the right term matches our definition of total confidence $\mathrm{conf}^{\mathrm{total}}$. □

Proposition 1. Total uncertainty $\mathrm{u}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})$ of a node i is proportional to the posterior over the unobserved true labels $\mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}}$ after acquiring its label $\mathbf{y}_i^{\mathrm{gt}}$:

$$\mathrm{u}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}}) \propto \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]$$

Proof.

$$
\begin{aligned}
\mathrm{u}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}}) &= \frac{1}{\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})} \\
&= \frac{\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})}{\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})} \frac{1}{\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]} \frac{1}{\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]} \frac{1}{\mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mid \mathbf{y}_{\mathcal{U}_i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}}, \ldots\right]} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]} \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]}{\mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}, \mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots\right]} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots\right]} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}}^{\mathrm{gt}} \mid \ldots\right]} \\
&\propto \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]
\end{aligned}
$$

Here, we abbreviated $\boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}$ with $\ldots$ for clarity. First, we insert the result from Theorem 2.6. Then, we use the definition of conditional probability to replace the inverse of the aleatoric uncertainty. After canceling terms, we observe that $\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}}^{\mathrm{gt}} \mid \ldots\right]$ is a constant with respect to i and arrive at the desired result that the total uncertainty is proportional to the posterior over the true labels $\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]$.As in Proposition H.1, we implicitly marginalized the learnable parameters θ of the Bayesian classifier.

□

Proposition 2. Aleatoric uncertainty $\mathrm{u}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})$ of a node i is proportional to the posterior over the unobserved true labels $\mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}}$ *without* acquiring its label $\mathbf{y}_i^{\mathrm{gt}}$:

$$
\mathrm{u}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}}) \propto \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]
$$

Proof.

$$
\begin{aligned}
\mathrm{u}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}}) &= \frac{1}{\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})} \\
&= \frac{\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})}{\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})} \frac{1}{\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]} \frac{1}{\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \ldots, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]} \frac{1}{\mathbb{P}\left[\mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mid \ldots\right]} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}} \mid \ldots\right]} \\
&= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}}^{\mathrm{gt}} \mid \ldots\right]} \\
&\propto \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} | \ldots\right]
\end{aligned}
$$

Again, we abbreviated $\boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}$ with $\ldots$ for clarity. First, we insert the result from Theorem 2.6 into $\mathrm{conf}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})/\mathrm{conf}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})$. Then, insert the definition of total confidence. We apply the

law of conditional probability, we see that $\mathbb{P}\left[\mathbf{y}_{\mathcal{U}} = \mathbf{y}_{\mathcal{U}}^{\text{gt}} \mid \ldots\right]$ is a constant with respect to i and arrive at the desired result that the aleatoric uncertainty is proportional to the posterior over the true labels $\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\text{gt}} | \ldots\right]$.

□

C Related Work

Active Learning on Independent and Identically Distributed Data. AL has seen substantial exploration in the context of i.i.d. data [17]. Approaches can be divided into three main categories: diversity-based, uncertainty-based, or a combination thereof [18]. Diversity or representation-based methods query data samples that best represent the full dataset, i.e. they opt for a diverse set of data points. Approaches like KMeans or Coreset opt to minimize the difference in model loss between the selected training set and the whole data set [19]. Other approaches use adversarial techniques to estimate the representativeness and diversity of new samples [20, 21]. Uncertainty-based approaches query instances that the classifier is most uncertain about. It is commonly computed from the predictive distribution of a classifier calculating its entropy [22], the margin between the two most likely labels, or as the probability of the least confident label [23]. Houlsby et al. [24] introduce Bayesian Active Learning by Disagreement (BALD), which queries points with high mutual information between the model parameters and the class label. Another line of work uses a loss-prediction module as a proxy for uncertainty [8, 25]. Nguyen et al. [26] leverage disentangled uncertainties [4] for AL and propose that epistemic uncertainty is a better proxy for US than aleatoric estimates. Other approaches linearly combine diversity- and uncertainty-based measures [27] or employ a two-step optimization scheme [18, 28]. BADGE [29] queries diverse and uncertain instances in the gradient space of the model using clustering. Another line of work constructs similarity graphs between instances to enforce diversity among queries [30]. GALAXY [31] proposes further refines this approach by mitigating class imbalance. In contrast, our work concerns settings where the adjacency matrix A is explicitly given .

Active Learning on Interdependent Graph Data. While a plethora of studies exist on AL for i.i.d. data, only a limited amount of work addresses interdependent data like graphs. Previous methods approach AL on graphs from different perspectives, including random fields [9, 32–34], risk minimization [10, 11], adversarial learning [35], knowledge transfer between graphs [36] or querying cheap soft labels [37]. US is typically only considered in terms of the predictive distribution [38] which does not disentangle aleatoric and epistemic components. Also, other uncertainty-reliant approaches do not make that distinction [39–41] even though the literature on i.i.d. data suggests that US benefits from disentangled uncertainty estimators [5, 42]. The exploration of US for AL on graphs beyond total uncertainty remains uncharted territory. Our work targets this gap and showcases the unrealized potential of epistemic uncertainty estimators for AL on graph data.

Uncertainty Estimation on Graphs. US strategies necessitate accurate uncertainty estimates which can be obtained in different ways. In classification, the predictive distribution of deterministic classifiers has been used to obtain aleatoric uncertainty [14]. The Graph Convolutional Network (**GCN**) [43] iteratively updates the representation $H^{(l)}$ of each node using a linear transformation $W^{(l)}$ and subsequent diffusion along the edges and non-linearity σ. Formally, a GCN layer is expressed as $f(H^{(l)}, A) = \sigma(AH^{(l)}W^{(l)})$. The **APPNP** model [44] first transforms the nodes features independently and then diffuses predictions according to approximate Personalized Page Rank (PPR) scores. Bayesian approaches model the posterior distribution over the model parameters. **Ensembles** [45] fall into this category. They approximate the posterior over model parameters through a collection of independently trained models. **Monte Carlo Dropout** (MC-Dropout) [46] instead emulates a distribution over model parameters by applying dropout at inference time. DropEdge [47] proposed to additionally drop edges to reduce over-fitting and over-smoothing. Variational Bayes methods place a prior on the model parameters (**BGCN**) and sample different parameter sets for each forward pass [48]. They allow access to disentangled uncertainty estimates by approximating a distribution over predictions from multiple forward passes. Commonly employed measures of epistemic uncertainty are the mutual information between the model weights and predicted labels [49] or the variance in confidence about the predicted label [14]. Evidential methods like **GPN** [14] disentangle epistemic and aleatoric uncertainty by outputting the parameters of a Dirichlet prior to the categorical predictive distribution. This method has shown strong performance in detecting distribution shifts. Finally, Gaussian processes on graphs, while potentially leading to

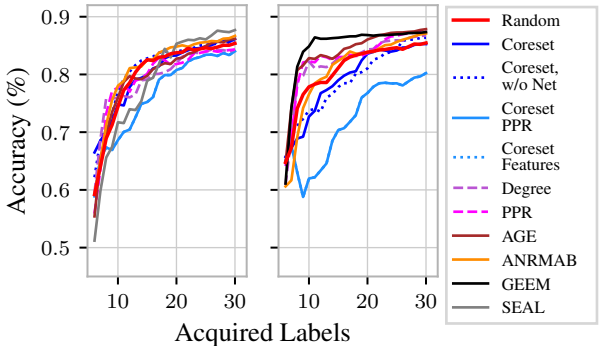


Figure 2: Accuracy of AL strategies on Citeseer using a GCN (left) / SGC (right) classifier. Except for GEEM, which is only tractable for SGCs, traditional AL can not significantly outperform random selection.

strong uncertainty estimates in specific domains [50], they do not disentangle aleatoric and epistemic uncertainty [51, 52].

D Benchmarking Uncertainty Sampling Approaches for Active Learning on Graphs

Previous studies on non-uncertainty-based AL on graphs find that AL strategies struggle to consistently outperform random sampling [38]. We, therefore, ask the question of whether US shows any merit when using state-of-the-art uncertainty estimation. We are the first to design a comprehensive AL benchmark for node classification that not only considers traditional methods but also includes a variety of uncertainty estimators for US.

Experimental Setup. We evaluate AL on five common citation benchmark datasets for node classification: **CoraML** [53], **Citeseer** [54, 55], **PubMed** [56] as well as the co-purchase graphs **Amazon Photos** and **Amazon Computers** [57]. We evaluate the models of Appendix C: **GCN**, **APPNP**, **MC-Dropout**, **BGCN**, **GPN** and **Ensembles** and report average results over multiple AL runs (see Appendix E).

When acquiring multiple labels per iteration, greedily selecting the most promising candidates might overestimate the performance improvement [7]. We therefore only acquire a single label in each iteration, enabling us to analyze the performance of different acquisition strategies without having to consider the potential side effects of batched acquisition. We initially label one node per class and fix the acquisition budget to $4C$.

In addition to a qualitative evaluation of the accuracy curves of different strategies in Figure 2, we report the accuracy after the labeling budget is exhausted in Table 6. Good acquisition functions should achieve higher accuracy at a lower amount of queries, which in turn results in a larger area under (AUC) the visualized curves. After normalization, this metric quantifies the average accuracy which we report as a summary of the AL performance in Table 2.

We proceed as follows: First, we benchmark traditional AL approaches and find that only one consistently outperforms **Random** selection that acquires labels with uniform probability. Then, we show that US fails to even match the performance of random queries in many instances, contrasting its successful application to i.i.d. data. We supply the corresponding AUC and final accuracy scores as well as visualizations on all datasets in Appendix F.

Non-Uncertainty-based Strategies. We first examine strategies that do not exclusively rely on uncertainty. (i) **Coreset** [19] opts to find a core-set cover of the training pool by selecting nodes that maximize the minimal distance in the latent space of a classifier to already acquired instances. (ii) **Coreset-PPR** is similar to Coreset, but we use inverse Personalized Page Rank (PPR) scores as a distance measure to select structurally dissimilar nodes. (iii) **Coreset Features.** Distances between nodes are computed only in terms of input features. (iv) **Degree** and **PPR.** We acquire nodes with the highest corresponding centrality measure. (v) **AGE** [39] and **ANRMAB** [40] combine total predictive uncertainty, informativeness, and representativeness metrics. (vi) **GEEM** [11] uses risk minimization to select the next query. Because of its high computational cost, we follow its authors and employ an

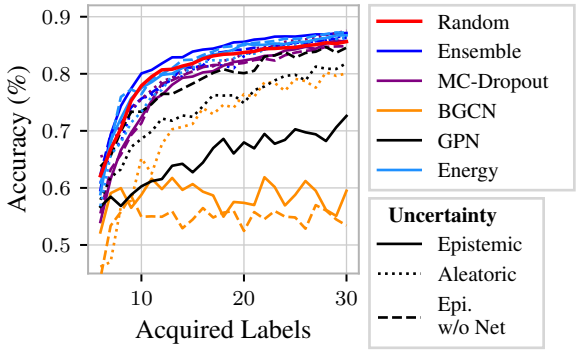


Figure 3: US on Citeseer. No method significantly outperforms random selection.

SGC [58] backbone. (vii) **SEAL** [35] uses adversarial learning to identify nodes dissimilar from the labeled set. If applicable, we also consider setting $\boldsymbol{A} = \boldsymbol{I}$ to exclude structural information similar to [14].

Observations: Figure 2 shows that only GEEM identifies a training set that is significantly more informative than random sampling. The performance of GEEM, however, comes at a high computational cost, as it requires training $\mathcal{O}(nC)$ models in each acquisition which makes it intractable for larger datasets and models beyond SGC. Structural Coreset strategies work well on both co-purchase networks but do not show strong results on other graphs. This highlights a notable gap between many commonly used acquisition functions and a potential optimal strategy.

Uncertainty Sampling. We evaluate US using different uncertainty estimators: (i) **Aleatoric.** Like [14], we compute $\mathrm{u}^{\mathrm{alea}} = -\max_c \mathbf{p}_c$. (ii) **Epistemic.** For models perform multiple predictions (MC-Dropout, Ensembles, BGCNs), we compute epistemic uncertainty as the variance of the confidence $\mathrm{u}^{\mathrm{epi}} = \mathrm{Var}\left[\mathbf{p}_{\hat{c}}\right]$ in the predicted class $\hat{c}$. For GPN, we follow the authors and use evidence as a measure of epistemic confidence. (iii) **Energy.** Energy-based models (EBMs) [59, 60] relate uncertainty to the energy $\mathrm{u} = -\tau \log \sum_c \exp\left(\mathbf{l}_i\right)$ of the predicted logits $\mathbf{l}$. We apply this estimator to the deterministic GCN, APPNP, and SGC models as a surrogate for epistemic uncertainty. Again, we ablate each strategy by withholding structural information ($\boldsymbol{A} = \boldsymbol{I}$).

Table 2: Average AUC (↑) for different acquisition strategies on different models and datasets. We mark the best strategy per model in bold and underline the runner-up. For each dataset, we highlight the overall best strategy with a † symbol.

		Baselines	**Non-Uncertainty**							**Uncertainty**			
	Inputs		*A* & *X*					*A*	*X*	*A* & *X*		*X*	
	Model	Random	Coreset	AGE	ANRMAB	GEEM	SEAL	Coreset PPR	Coreset Inputs	Epi./ (Energy)	Alea.	Epi./ (Energy)	Alea.
CoraML	GCN	62.51	64.35	64.12	64.24	n/a	**66.07**	59.53	61.26	63.97	61.30	65.65	64.33
	APPNP	67.72	67.32	66.12	69.49	n/a	n/a	**71.04**	64.49	64.92	67.68	69.59	66.69
	Ensemble	63.89	60.55	64.80	65.10	n/a	n/a	62.65	65.07	63.47	64.03	64.80	**65.82**
	MC-Dropout	**64.94**	64.37	64.44	64.06	n/a	n/a	62.92	64.35	59.17	63.69	61.82	63.87
	BGCN	45.76	49.37	**51.25**	47.23	n/a	n/a	39.43	44.85	44.45	46.61	42.74	48.11
	GPN	56.50	n/a	n/a	n/a	n/a	n/a	**58.04**	54.02	54.75	57.16	55.89	57.21
	SGC	63.85	65.23	67.56	61.14	**71.39**†	n/a	60.24	59.18	67.51	65.66	65.05	67.13
Pubmed	GCN	61.56	62.61	**69.48**	60.31	n/a	58.62	61.71	56.71	59.64	61.85	59.66	60.34
	APPNP	64.61	63.88	**70.18**†	63.83	n/a	n/a	64.21	56.87	63.09	62.37	62.23	63.95
	Ensemble	59.26	64.25	**68.26**	60.40	n/a	n/a	61.89	56.36	63.70	61.37	59.71	61.15
	MC-Dropout	58.30	62.97	**65.24**	60.50	n/a	n/a	61.43	56.01	58.67	59.07	59.23	62.22
	BGCN	53.59	**59.29**	56.93	52.68	n/a	n/a	53.40	51.40	55.19	52.81	57.09	54.62
	GPN	59.76	n/a	n/a	n/a	n/a	n/a	**62.08**	54.34	58.82	57.24	56.25	59.37
	SGC	56.79	64.48	**69.20**	60.49	64.82	n/a	62.15	52.25	62.04	61.55	61.04	60.74
AmazonPhotos	GCN	79.06	78.58	75.17	79.97	n/a	71.16	70.20	**82.71**	74.66	74.61	79.96	79.63
	APPNP	79.29	81.04	79.02	80.35	n/a	n/a	76.37	**84.24**	79.72	77.45	80.48	77.69
	Ensemble	82.23	80.44	77.45	82.77	n/a	n/a	74.93	84.04	**84.46**	77.85	80.50	81.25
	MC-Dropout	80.32	76.63	74.75	80.21	n/a	n/a	75.32	**82.45**	72.42	73.16	69.68	78.80
	BGCN	71.22	67.15	65.69	70.69	n/a	n/a	59.34	**73.39**	70.83	67.83	72.21	69.19
	GPN	62.80	n/a	n/a	n/a	n/a	n/a	55.59	**65.07**	54.78	60.53	62.90	62.41
	SGC	80.52	82.32	74.01	80.92	**86.43**†	n/a	66.94	84.24	84.01	71.43	80.75	76.38

Observations: In contrast to literature on i.i.d. data [1, 5, 6], we observe none of the US approaches to be effective in Figure 3. While sampling instances with high aleatoric uncertainty matches the performance of random queries, we find epistemic uncertainty to even underperform in many instances. This is surprising, as the efficacy of epistemic US has been demonstrated for i.i.d. data [5, 7]. Only ensemble models match the performance of random sampling and slightly outperform on some datasets. GPN and energy-based approaches can not guide US toward effective queries.

This is an intriguing result as both uncertainty estimators have shown to be highly effective for of out-of-distribution detection [14, 60].

E Experimental Setup

E.1 Active Learning

We discuss AL on graphs, where given an initial set of labeled instances $\mathcal{O} \subset \mathcal{V}$ we aim to acquire a set of the unlabeled nodes $\mathcal{U} \subset \mathcal{V}$ that is optimal in terms of improving the performance of the classifier on the entire graph. That is, we (1) initially label one node randomly drawn from each class. (2) We then re-initialize the model weights and train the classifier until convergence. (3) We employ an acquisition strategy to select one or more unlabeled node(s). (4) We add the acquired label(s) to the training set and repeat the procedure at step (2) until some acquisition budget is exhausted. In contrast to domains where model training is expensive [1, 6, 7], we re-train the classifier after each acquisition iteration. If not stated otherwise, we only acquire one node label in each iteration and fix the acquisition budget to $5C$. The resulting final training pools, therefore, contain fewer instances compared to dataset splits commonly used in other work [14, 43].

E.2 Hyperparameters

As hyperparameter tuning may be unrealistic in AL [11], we do not finetune them on a validation set. One potential strategy to realize hyperparameters is to randomly sample them from the search space over which hyperparameter optimization would be performed. This, however, is expected to lead to notably worse performance for most GNN architectures [11]. Since the goal of our work is to showcase that even under optimal circumstances, contemporary uncertainty estimators can not enable US to be effective, we select one set of hyperparameters reported to be effective by the literature and employ it on all datasets. Specifically, we chose the following values:

Model	Hidden Dimensions	Learning Rate	Max Epochs	Weight Decay	Teleport Probability	Power Iteration Steps	Dropout	Flow Dimension	Number Radial Flow Layers
GCN	[64]	0.001	10,000	0.001	n/a	n/a.	0.0	n/a	n/a.
APPNP	[64]	0.001	10,000	0.001	0.2	10	0.0	n/a	n/a
GPN	[64]	0.001	10,000	0.001	0.2	10	0.5	16	10

Table 3: Hyperparameters of different GNN backbones

E.3 Datasets

Real-World Datasets. We evaluate AL approaches on three common node classification benchmark citation datasets: CoraML [53–55, 61], Citeseer [53–55], PubMed [56]. In all datasets, nodes are papers and edges model citations. We consider undirected edges only and select the largest connected component in the dataset. All datasets use bag-of-words representations and we normalize the input features $\mathbf{x}_i$ node-wise to have a L_2-norm of 1. We report the statistics of each dataset in Table 4.

Dataset	**#Nodes** n	**#Edges** m	**#Input Features** d	**#Classes** c	**Edge Density** $\frac{m}{n^2}$	**Homophily** p	**Intra-Class Edge Density** p	**Inter-Class Edge Density** q	**SNR** $\frac{p}{q}$
CoraML	$2,810$	$15,962$	$2,879$	7	0.20%	78.44%	1.69%	0.06%	28.51
Citeseer	$1,681$	$5,804$	602	6	0.20%	92.76%	4.89%	0.03%	141.36
PubMed	$19,717$	$88,648$	500	3	0.02%	80.24%	0.07%	0.01%	9.48
AmazonComputers	$13,381$	$491,556$	767	10	0.27%	77.72%	3.60%	0.07%	54.74%
AmazonPhotos	$7,484$	$238,086$	745	8	42.47%	82.72%	3.77%	0.10%	36.85%

Table 4: Dataset statistics.

We additionally compute the edge density $\frac{m}{n^2}$ as well as the homophily which is the fraction of edges that link between nodes of the same class. For comparability with CSBMs, we also report the average empirical inter-class edge probabilities p and intra-class edge probabilities q as well as their ratio, the structural signal-to-noise rate (SNR).

Contextual Stochastic Blockmodels (CSBMs).

CSBMs [62] define the following generative process: First, node labels are sampled independently from a prior $p(\mathbf{y})$. For each node, features are then drawn independently from a class-

conditional normal distribution $p(\boldsymbol{X}_i \mid \mathbf{y}_i) \sim \mathcal{N}(\mu_{\mathbf{y}_i}, \sigma_x^2 \boldsymbol{I})$. Each edge in the graph is generated independently according to an affiliation matrix $\boldsymbol{F} \in \mathbb{R}^{c \times c}$, i.e. $p(\boldsymbol{A}_{i,j} \mid \mathbf{y}_i, \mathbf{y}_j) \sim \text{Ber}(\boldsymbol{F}_{\mathbf{y}_i, \mathbf{y}_j})$. This gives rise to an explicit joint distribution over the graph that factorizes as $p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y}) = \prod_{i<j} p(\boldsymbol{A}_{i,j} \mid \mathbf{y}_i, \mathbf{y}_j) \prod_i p(\boldsymbol{X}_i \mid \mathbf{y}_i) \prod_i p(\mathbf{y}_i)$.

We generate CSBM graphs with a fixed number of nodes, classes, and input features according to Appendix A. That is, we first sample class labels $\mathbf{y}_i$ for each node independently from a uniform prior $\mathbb{P}\big[\mathbf{y}_i\big] = 1/C$. We then create links according to the affiliation matrix $\boldsymbol{F}$, where $\mathbb{P}\big[\boldsymbol{A}_{i,j} = 1 | \mathbf{y}_i, \mathbf{y}_j\big] = \boldsymbol{F}_{\mathbf{y}_i, \mathbf{y}_j}$. If not stated otherwise, we use a symmetric and homogeneous affiliation matrix $\boldsymbol{F}$, where we set all diagonal elements to a given intra-class edge probability p and all off-diagonal elements to a given inter-class edge probability q. We enforce a structural signal-to-noise ratio (SNR) $\sigma_A = p/q$ by specifying an expected node degree $\mathbb{E}\left[\text{deg}(v)\right]$ and then solving for:

$$q = \frac{\mathbb{E}\left[\text{deg}(v)\right] c}{n-1} \frac{1}{\sigma_A + c - 1} \tag{5}$$

For each class, we first deterministically draw C vectors of dimension d that all have a pairwise distance of δ_X. We then use a random rotation to obtain class centers μ_c. For each node, we then independently sample its features $\mathbf{x}_i$ from a normal distribution $\mathcal{N}(\boldsymbol{X}_i | \mu_{\mathbf{y}_i}, \sigma_X \boldsymbol{I})$. We refer to the quotient $\frac{\delta_X}{\sigma_X}$ as the feature signal-to-noise ratio. The dimension of the feature space is given by $d = \max(c, \lceil n / \log^2 n \rceil)$, following [63].

E.4 Model Details

If applicable, we use the same GCN [43] backbone for all models. That is, we employ one hidden layer of dimension of 64. The APPNP [44] model uses an MLP with one hidden layer of the same dimension. We diffuse the predictions using 10 steps of power iteration and a teleport probability of 0.2.

If not stated otherwise, ensembles are composed of 10 architecturally identical GCNs. In the case of MCD, we apply dropout with probability 0.1 to each neuron of the GCN backbone independently. The BGCN mimics the GCN but models weights and biases as normal distributions. We follow [48] and regularize these towards a standard normal distribution. Consequentially, we apply a weight of $\lambda = 0.1$ to the regularization loss.

We follow [14] to configure the GPN model: We use 10 radial flow layers to implement the class-conditional density model with a dimension of 16. For diffusion, we implement the same configuration as for APPNP. Other hyperparameters are set as in [14].

For SEAL, we follow the authors and do not re-train the model after each acquisition [35]. Furthermore, we pick the number of training iterations for the discriminator to be $n_d = 5$ and the number of iterations for the generator $n_g = 10$, a combination that we observed to be successful on one dataset and kept fixed for all others as the authors do not provide them directly and hyperparameter optimization is unrealistic in AL [11].

E.5 Training Details

Apart from the GPN, all of the aforementioned models are trained towards the binary-cross-entropy objective using the ADAM [64] optimizer with a learning rate of 10^{-3} and weight decay of 10^{-3}. We also perform early stopping on the validation loss with a patience of 100 iterations. We implement our models in PyTorch and PyTorch Geometric and train on two types of machines: (i) Xeon E5-2630 v4 CPU @ 2.20GHz with a NVIDA GTX 1080TI GPU and 128 GB of RAM. (ii) AMD EPYC 7543 CPU @ 2.80GHz with a NVIDA A100 GPU and 128 GB of RAM .

For each dataset, classifier, and acquisition function, we report results averaged over five different dataset splits and five distinct model initializations. In each dataset split, we a priori fix 20% of all nodes as a test set that is reused in every subsequent dataset split and can never be acquired by any strategy.

F Additional Metrics and Plots

We supply an evaluation of contemporary non-uncertainty-based acquisition strategies on different models as well as various uncertainty estimators for US over all datasets listed in Appendix E.3. We briefly summarize the key insights:

(i) **CoraML.** Figures 4a to 4d show that apart from GEEM no AL strategy is significantly more effective than random sampling. For an SGC classifier, GEEM can identify high-quality training sets. In terms of US approaches, only ensemble methods perform somewhat better than random acquisition. (ii) **Citeseer.** Figures 3, 4e, 4g and 4h show that no AL strategy can match the performance of the GEEM. Again, risk minimization is the strongest performing approach. Only ensembles and energy-based models can compete with random acquisition when concerning epistemic uncertainty estimates. We verify the intriguing trend observed in Appendix D that epistemic uncertainty proxies seem to significantly underperform random queries when being employed for US. This supports our conjecture that contemporary estimators may not properly disentangle or only partially model uncertainty. (iii) **PubMed.** In Figures 4i to 4l, GEEM is outperformed by the AGE baseline. For this dataset, acquiring central nodes (PPR, AGE) appears to be a successful strategy. Interestingly, ANRMAB fails to exploit centrality despite it, in theory, being capable of doing so. US appears to be effective only when employing ensemble methods. (iv) **AmazonPhotos.** On this dataset, GEEM outperforms a centrality-based (PPR) Coreset approach (see Figures 4m to 4p. Concerning US, we again find that only ensemble methods outperform random sampling while other estimators lead to significantly worse performance. (v) **Amazon Computers.** Similar to the other co-purchase network, we observe Coreset PPR to be a strong proxy for AL on the AmazonComputers dataset (Figures 4q to 4t). For this dataset, all US approaches–including ensembles– perform significantly worse than a random strategy.

This affirms the statements of Appendix D: Among non-uncertainty strategies, only GEEM can consistently outperform random queries. In turn, most US approaches consistently underperform random queries, a trend that may be indicative of improperly disentangled uncertainty. Only ensembles show small merit in most cases and, at least, do not fall short of random queries.

We supplement our findings by reporting the accuracy of each classifier after the acquisition budget is exhausted in Table 6. The corresponding rankings align well with the respective AUC scores Table 5. Both metrics are averaged over all 5 dataset splits and model initializations and we also report the standard deviations.

G Computing Aleatoric and Total Confidence on CSBMs

Computing the aleatoric confidence is straightforward directly employing Definition 2.3 using Bayes rule and the generative process up to a tractable normalization constant.

$$\mathrm{conf}^{\mathrm{alea}}(i, c) \propto p(\boldsymbol{A}, \boldsymbol{X} \mid \mathbf{y}_{-i}^{\mathrm{gt}}, \mathbf{y}_i = c) p(\mathbf{y}_i = c) \tag{6}$$

Computing the epistemic confidence, however, turns out to have exponential complexity in the number of unlabeled nodes $|\mathcal{U} - i|$:

$$\begin{aligned}
\mathrm{conf}^{\mathrm{epi}}(i, c) &= p(\mathbf{y}_i = c \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}) && (7)\\
&= \sum_{\mathbf{y}_{\mathcal{U}-i}} p(\mathbf{y}_i = c = \mathbf{y}_{\mathcal{U}-1} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}) && (8)\\
&\propto \sum_{\mathbf{y}_{\mathcal{U}-i}} p(\boldsymbol{A}, \boldsymbol{X} \mid \mathbf{y}_i = c, \mathbf{y}_{\mathcal{U}-i}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}) p(\mathbf{y}_i = c, \mathbf{y}_{\mathcal{U}-i}) && (9)
\end{aligned}$$

Approximating Epistemic Confidence. For larger graphs, this quickly becomes intractable and we therefore rely on a variational mean-field approximation of the joint distribution $p(\mathbf{y}_{\mathcal{U}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})$ to obtain marginals similar to [65, 66].

$$p(\mathbf{y}_{\mathcal{U}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}) \approx q(\mathbf{y}_{\mathcal{U}}) := \prod_{i \in \mathcal{U}} q_i(\mathbf{y}_i) \tag{10}$$

Since the variational distributions q_i are discrete, we can fully describe them with parameters $\gamma_{i,c} := q_i(\mathbf{y}_i = c)$. The ELBO of this variational problem is given by:

$$\begin{aligned}
J(\gamma) &= \log p(\boldsymbol{X}, \boldsymbol{A}, \mathbf{y}_{\mathcal{O}}) + \mathbb{KL}\left[q(\mathbf{y}_{\mathcal{U}}) \,\middle\|\, p(\mathbf{y}_{\mathcal{U}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}) \right] && (11)\\
&= \mathbb{E}_{\mathbf{y}_{\mathcal{U}} \sim q} \left[\log p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{U}} \mid \mathbf{y}_{\mathcal{O}}) \right] + \mathbb{H}\left[q(\mathbf{y}_{\mathcal{U}}) \right] + \text{const} && (12)\\
&= \sum_{i,c} \gamma_{i,c} \log p(\mathbf{y}_i = c) + \sum_{i<j} \sum_{c,c'} \gamma_{i,c} \gamma_{j,c'} \log p(\boldsymbol{A}_{i,j} \mid \mathbf{y}_i = c, \mathbf{y}_j = c') && (13)\\
&+ \sum_{i,c} \gamma_{i,c} \log p(\boldsymbol{X}_i \mid \mathbf{y}_i = c) + \text{const} && (14)
\end{aligned}$$

Here, we introduced $\gamma_{j,c} = \mathbb{1}(c = \mathbf{y}_j^{\mathrm{gt}})$ for all $j \in \mathcal{O}$ for convenience. This gives rise to a constrained optimization problem where $\sum_c \gamma_{i,c} = 1$ for all $i \in \mathcal{U}$. Solving this problem analytically gives rise to the equations:

$$\gamma_{i,c} \propto \exp\left(\log p(\mathbf{y}_i = c) + \sum_{j \neq i} \sum_{c'} \gamma_{j,c'} \log p(\boldsymbol{A}_{i,j} \mid \mathbf{y}_i = c, \mathbf{y}_j = c') + \log p(\boldsymbol{X}_i \mid \mathbf{y}_i = c) \right) \tag{15}$$

The marginal probabilities $\gamma_{i,c}$ can then directly be optimized by the resulting fixed point iteration scheme, where after each iteration the probabilities are normalized:

$$\log \gamma_{i,c}^{(t+1)} = \log p(\mathbf{y}_i = c) + \sum_{j \neq i} \sum_{c'} \gamma_{j,c'}^{(t)} \log p(\boldsymbol{A}_{i,j} \mid \mathbf{y}_i = c, \mathbf{y}_j = c') + \log p(\boldsymbol{X}_i \mid \mathbf{y}_i = c) \tag{16}$$

The quality of the acquisition function given by the epistemic confidence relies on the quality of the approximated marginals $p(\mathbf{y}_i \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})$, i.e. the total confidence $\mathrm{conf}^{\mathrm{total}}(i, \cdot)$. To verify

that the proposed variational scheme indeed provides reasonable approximations, we report the absolute approximation errors $|q(\mathbf{y}_i = c) - p(\mathbf{y}_i = c \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})|$ for CSBM graphs such that exact computation of the true marginals is tractable, i.e. up to 12 nodes.

Figure 6 shows the distribution of approximation errors averaged over five different samples from a CSBM generative distribution with a structural SNR of $\sigma_A = 2.0$ and a feature SNR $\sigma_X = 1.0$ as well as an expected node degree $\mathbb{E}[\deg(v)] = 4.0$. In general, we expect the approximation to be of higher quality the more decoupled the marginals $p(\mathbf{y}_i \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})$ are [66]. While the median error does not exceed 5%, we observe some outliers for larger graphs. In such cases, the employed approximation is inaccurate. Nonetheless, in Appendix H we observe that even in the face of sometimes poor approximations, the proposed uncertainty framework achieves strong results. We suspect a stronger approximation to perform even better.

H Uncertainty Sampling with Ground-Truth Uncertainty

Since for real-world datasets, the data generating process is not known, we perform our analysis on CSBMs defined in Appendix A and discuss a practical approximation later in Section 3. This allows us to compute the uncertainty estimates of Definitions 2.2 to 2.4 directly by evaluating the explicit joint likelihood of the generative process $p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y})$ (see Appendix G). While the optimality of epistemic US holds for any data-generating process, we focus on CSBMs as they have been extensively studied as proxies for real data in node classification [67]. To isolate the effect of correctly disentangling uncertainty, we also assume the parameters of the underlying CSBM to be known to the Bayesian classifier f^*_θ. Therefore, any discrepancies in US performance are purely linked to the disentanglement into aleatoric and epistemic factors.

We compare the performance of US using the proposed uncertainty measures to contemporary uncertainty estimators over 5 graphs with 100 nodes and 7 classes sampled from a CSBM distribution $p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y})$ in Appendix H.

Observations: In agreement with Theorem 2.6, epistemic uncertainty significantly outperforms random queries as well as aleatoric and total uncertainty which we explain formally in the following Propositions H.1 and H.2. We continue to analyze which aspects of uncertainty modelling are crucial for its successful in AL.

Disentangling Uncertainty. We first discuss why acquiring nodes with high total uncertainty $\mathrm{u}^{\mathrm{total}}$ performs worse than isolating epistemic factors: Total uncertainty favors not only informative queries but also tends to acquire labels of nodes that are associated with a high aleatoric uncertainty $\mathrm{u}^{\mathrm{alea}}$.

Proposition H.1. *Total uncertainty* $\mathrm{u}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}})$ *of a node* i *is proportional to the posterior over the unobserved true labels* $\mathbf{y}^{\mathrm{gt}}_{\mathcal{U}-i}$ *after acquiring its label* $\mathbf{y}_i^{\mathrm{gt}}$*:*

$$\mathrm{u}^{\mathrm{total}}(i, \mathbf{y}_i^{\mathrm{gt}}) \propto \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}^{\mathrm{gt}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]$$

We provide a proof of Proposition H.1 in Appendix B. Acquiring nodes with maximal total uncertainty maximizes the posterior of the remaining unlabeled set $\mathbf{y}_{\mathcal{U}-i}$. This is problematic as one way to increase this posterior probability is to remove an aleatorically uncertain node i from the unlabeled set. Such a query will not push the posterior of the remaining nodes in $\mathbf{y}_{\mathcal{U}}$ towards their true labels and instead improve the posterior by excluding nodes that are inherently difficult to predict. In contrast, the epistemic acquisition evaluates the joint posterior in relation to the effect of removing node i from the unlabeled set (see Theorem 2.6). In fact, acquiring aleatorically uncertain nodes directly removes inherently ambiguous nodes from the unlabeled set.

Proposition H.2. *Aleatoric uncertainty* $\mathrm{u}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}})$ *of a node* i *is proportional to the posterior over the unobserved true labels* $\mathbf{y}^{\mathrm{gt}}_{\mathcal{U}-i}$ without *acquiring its label* $\mathbf{y}_i^{\mathrm{gt}}$*:*

$$\mathrm{u}^{\mathrm{alea}}(i, \mathbf{y}_i^{\mathrm{gt}}) \propto \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}^{\mathrm{gt}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]$$

We provide a proof for Proposition H.2 in Appendix B. Proposition H.2 explains why we observe aleatoric US to be ineffective in Appendix H. It optimizes the posterior of the remaining labels $\mathbf{y}^{\mathrm{gt}}_{\mathcal{U}-i}$ without considering the acquisition of $\mathbf{y}_i^{\mathrm{gt}}$. Such queries do not align with AL as they neglect the additional information obtained in each iteration. To optimize predictions on all remaining nodes it is crucial to properly disentangle uncertainty into aleatoric and epistemic components and acquire epistemically uncertain labels.

Modelling the Data Generating Process. We also investigate the importance of the uncertainty estimator to faithfully model the true data-generating process $p(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y})$. To that end, we ablate our proposed uncertainty measures but only consider a Bayesian classifier that erroneously exclusively models present edges $(i, j) \in \mathcal{E}$ while neglecting $(i, j) \notin \mathcal{E}$: $\hat{p}(\boldsymbol{A}, \boldsymbol{X}, \mathbf{y}) := \prod_{i<j,(i,j)\in\mathcal{E}} p(\boldsymbol{A}_{i,j} \mid \mathbf{y}_i, \mathbf{y}_j) \prod_i p(\boldsymbol{X}_i) \prod p(\mathbf{y}_i)$.

We specifically pick this inaccurate model to ignore non-existing edges because of its strong resemblance to contemporary GNN architectures used at the backbone of uncertainty estimators discussed in Appendix D. They rely on variations of the message-passing framework which propagates information exclusively along existing edges $\mathcal{E}$. In Appendix H, we observe that employing disentangled ground-truth uncertainty based on an inaccurate generative process neglecting non-existing edges harms US even with proper uncertainty disentanglement. Thus, our analysis reveals another potential shortcoming of contemporary uncertainty estimators for graphs: They may fail to accurately learn the underlying data-generating process and thus be incapable of assessing uncertainty faithfully.

Pitfalls in US for Graphs. Overall, we identify two aspects required for effective US on graphs that may explain the lackluster performance of existing estimators: Most importantly, uncertainty needs to be *disentangled into aleatoric and epistemic components*. Only epistemic US theoretically and empirically aligns well with AL. Furthermore, we observe an additional performance gain when the Bayesian classifier faithfully *models the data generating process*. Hence, we propose that uncertainty estimators should be designed with special care for isolating epistemic uncertainty while at the same time being expressive enough to capture the unknown underlying data-generating process.

Also looking at larger graph sizes ($n = 1000$), we find epistemic US to be the strongest approach in terms of the Bayesian classifier while total uncertainty can not match its performance but also outperforms random sampling. We also observe a decrease in performance when the generative process is modelled incorrectly. Also contemporary estimators fail to outperform random queries, confirming the results reported in Section 2 for larger graph sizes.

H.1 Ablation of the Proposed Acquisition Strategy on Different CSBM Configurations

We also ablate the proposed acquisition strategy on different configurations of the CSBM. That is, we sample graphs from a CSBM with homogeneous and symmetric affiliation matrices and vary both the structural SNR $\frac{p}{q}$ as well as the feature SNR $\frac{\delta_X}{\sigma_X}$. The underlying CSBM has 100 nodes and 4 classes and labels are sampled from a uniform prior (see Appendix E.3).

We perform AL on five graphs sampled from each configuration independently and measure the absolute improvement Uncertainty Sampling achieves in terms of AUC and accuracy after a budget of $5C = 20$ is exhausted. In Figure 8a, we show the average improvement in AUC for each SNR configuration. We also supply a similar visualization for the accuracy after the budget is exhausted in Figure 8b.

We find that our approach achieves the strongest improvement in both metrics when the structural SNR σ_A is neither too high nor too small: Large values make the classification problem too easy and hence AL strategies do not have to carefully query informative nodes as the overall performance is strong even in very label scare regimes. Interestingly, our ground-truth uncertainty estimator shows strong merit when the node features are noisy, indicating that it is crucial to pick structurally informative nodes in the graph in these regimes.

At the same time, when the structural SNR is too low (in particular drops below 1.0), our method fails to outperform random acquisition: We attribute this to the mean-field approximation to the total confidence $\mathrm{conf}^{\mathrm{total}}$ described in Appendix G: We observed that for graphs with low structural SNR, the fixed point iteration of Equation (16) does not converge. Hence, the approximated marginal probabilities are poor and both the prediction as well as the uncertainty estimation based on it deteriorate.

I Visualization of Ground-Truth Uncertainty on a Toy Example

In the following, we illustrate the behavior of this acquisition function on a small CSBM graph. Figure 9a shows a sample graph with three classes and nine nodes. The greyed-out histograms represent the aleatoric confidence. The distributions correspond to the total confidence, and the size

of the nodes indicates the inverse of the ratio of both, i.e., the epistemic uncertainty, which guides our acquisition. We note that nodes 7 and 8 are the most promising candidates, as their aleatoric prediction is confident, and in the initial step since all nodes are unlabeled, the total confidence is uniform across all nodes due to the homogenous structure of the affiliation matrix.

(a) Initial state **(b)** State after first acquisition

Figure 9: Example of CSBM with 3 classes and 9 nodes. **Left**, we can see the initial state without any labeled node. On the **right**, we can see the state after we acquired node 7.

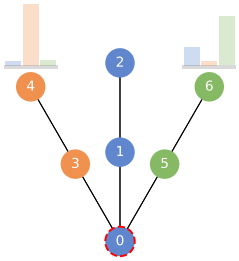
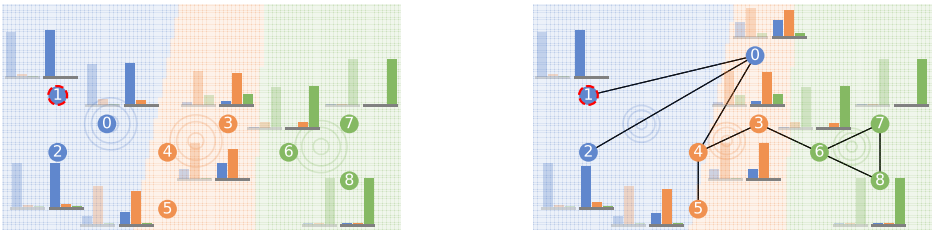


Figure 10: Aleatoric uncertainty is different for symmetric nodes when the affiliation matrix is not symmetric itself.

Figure 9b depicts the subsequent iteration following the acquisition of node 7. Due to the additional information introduced, node 8 exhibits reduced epistemic uncertainty. Consequently, the most promising nodes to consider next are nodes 0 and 4, as their total confidence remains relatively low, and they display higher aleatoric confidence than their neighbors. This is because both nodes connect to two nodes from the same class, bolstering their confidence levels.



(a) Decision boundary given only the features **(b)** Decision boundary given the features **and** structure

Figure 11: Example of CSBM with one labeled node (1). The shaded area represents the decision of the classifier. **Left**, only using the feature information. **Right** feature and structural information.

Figure 11a presents predictions made solely on feature information, with shaded regions and concentric circles demonstrating decision boundaries and class feature distributions, respectively. It is evident that incorporating feature information significantly increases confidence. Then, in Figure 11b, we adjust the position of node 0 away from the blue center and introduce the structural information. The shaded regions in this figure represent the prediction for node 0 and a noticeable shift to the right

compared to Figure 11a can be observed. This shift demonstrates the impact of structural information on the classifier's decision boundary.

Finally, Figure 10 underscores the significance of modeling non-existing edges in uncertainty estimation. Despite their symmetric neighborhoods, the aleatoric uncertainty (without feature information) differs for nodes 4 and 6 as in the underlying CSBM model the green class is less likely to not associate with the blue class. This reveals that sensible uncertainty estimators need to also utilize the *absence of edges*.

J Approximating Disentangled Uncertainty on Real-World Data

J.1 Multiple Pseudo-Labels (MP)

Equation (4) computes epistemic confidence as a ratio of total and aleatoric confidences. For a given classifier $f_\theta(\boldsymbol{A}, \boldsymbol{X})$ that should be improved with AL, we interpret its predictive distribution as total confidence which is defined in Definition 2.2.

$$\mathrm{conf}^{\mathrm{total}}(i, c) = \mathbb{E}_{p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}})} \left[\mathbb{P}\left[\mathbf{y}_i = c \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}} = \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}, \theta \right] \right] \approx f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c} \tag{17}$$

Note that total confidence is defined as the marginal predictive distribution associated with a node i after conditioning on the training set $\mathbf{y}_{\mathcal{O}} = \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}$. This conditioning is represented by the posterior over the model parameters $p(\theta \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}})$ that is obtained through training the classifier. In the case of deterministic GNNs, this posterior collapses to a point estimate.

Estimating the aleatoric confidence defined in Definition 2.3 is more challenging as the predictive distribution of a node i needs to be conditioned on unavailable ground-truth labels $\mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}}$. We approximate those by using the predictions of f_θ as pseudo-labels:

$$\hat{\mathbf{y}}_j = \mathrm{argmax}_{c \in [C]} \, f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j,c} \tag{18}$$

Similar to the approximation for total confidence, we employ the predictive distribution of a classifier to estimate aleatoric confidence. Since the set of labels that this classifier must be conditioned on the pseudo-labels $\hat{\mathbf{y}}_{\mathcal{U}-i}$ as well, a separate classifier must be trained. For each node i that the aleatoric confidence should be predicted, we get a different set of labels the classifier needs to be trained on: $\mathbf{y}_{\mathcal{O}}^{\mathrm{gt}} \cup \hat{\mathbf{y}}_{\mathcal{U}-i}$. This implies that $\mathcal{O}(n)$ auxiliary classifiers need to be trained in each iteration of AL using this approximate method. For each classifier $f_{\hat{\theta}_i}$, we use its predictive distribution as an estimate of aleatoric confidence:

$$\mathrm{conf}^{\mathrm{alea}}(i, c) = \mathbb{E}_{p(\hat{\theta}_i \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}, \hat{\mathbf{y}}_{\mathcal{U}-i})} \left[\mathbb{P}\left[\mathbf{y}_i = c \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}} = \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}, \mathbf{y}_{\mathcal{U}-i} = \hat{\mathbf{y}}_{\mathcal{U}-i}, \hat{\theta} \right] \right] \approx f_{\hat{\theta}_i}(\boldsymbol{A}, \boldsymbol{X})_{i,c} \tag{19}$$

Again, the posterior $p(\hat{\theta}_i \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}^{\mathrm{gt}}, \hat{\mathbf{y}}_{\mathcal{U}-i})$ over the weights of the auxiliary classifier are obtained through training on both available ground-truth labels as well as pseudo labels. This implies that each auxiliary classifier $f_{\hat{\theta}_i}$ is trained on $n-1$ labeled nodes. To get a scalar estimate of epistemic uncertainty that serves as an acquisition proxy, we take the ratio of both approximations according to Equation (4) and evaluate it at $c = \hat{\mathbf{y}}_i$ as the true class label of node i is not available:

$$\mathrm{conf}^{\mathrm{epi}}(i, \hat{\mathbf{y}}_i) = \frac{\mathrm{conf}^{\mathrm{alea}}(i, \hat{\mathbf{y}}_i)}{\mathrm{conf}^{\mathrm{total}}(i, \hat{\mathbf{y}}_i)} \tag{20}$$

When employing the MP approximation as a strategy, in each iteration, we query the label of node $i^* = \mathrm{argmax}_{i \in \mathcal{U}} \, \mathrm{conf}^{\mathrm{epi}}(i, \hat{\mathbf{y}}_i)$ that maximizes (approximate) epistemic uncertainty.

J.2 Expected Single Pseudo-Label (ESP)

An alternative approach is to not approximate epistemic uncertainty as a ratio of total and aleatoric confidences but directly estimate the right-hand side of Theorem 2.6 which we prove to be equivalent.

Both in the numerator and denominator, we need to compute joint probabilities over $\mathbf{y}_{\mathcal{U}-i}$ which we approximate as a product of marginal probabilities predicted by the classifier. To that end, the joint probability in the denominator is conditioned on $\mathbf{y}_{\mathcal{O}}$ only, so we again can make use of the predictive distribution of the classifier f_θ to be trained. As the denominator of Theorem 2.6 requires evaluating the joint probability at the unavailable ground-truth labels $\mathbf{y}^{\text{gt}}_{\mathcal{U}-i}$, we again have to rely on the pseudo-labels $\hat{\mathbf{y}}$ predicted by f_θ.

$$
\begin{aligned}
\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}^{\text{gt}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right] &\approx \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \hat{\mathbf{y}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right] \\
&\approx \prod_{j \in \mathcal{U}-i} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j, \hat{\mathbf{y}}_j} \\
&= \prod_{j \in \mathcal{U}-i} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j, \hat{\mathbf{y}}_j} \frac{f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i}}{f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i}} \\
&= \frac{\prod_{j \in \mathcal{U}} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j, \hat{\mathbf{y}}_j}}{f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i}} \\
&\propto \frac{1}{f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i}}
\end{aligned}
\tag{21}
$$

In the last line of Equation (21), we recognized that the term $\prod_{j \in \mathcal{U}} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j, \hat{\mathbf{y}}_j}$ is independent of node i which epistemic uncertainty is to be approximated for. Since for AL, we query a node that maximizes epistemic uncertainty, we can discard constant multiplicative factors that are the same for the computation of each node.

For any node i, approximating the numerator of Theorem 2.6 requires conditioning on its unavailable true label $\mathbf{y}^{\text{gt}}_i$. One possible remedy would be to again use the pseudo-label $\hat{\mathbf{y}}_i$ instead. However, since we need to only condition on one additional approximate label, it is also feasible to factor in the belief of f_θ about this label more accurately by taking an expectation over all possible realizations $c \in [C]$ with respect to the predictive distribution $f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,:}$. In contrast, when computing MP approximation using Equation (19), we condition on all $\mathbf{y}_{\mathcal{U}-i}$ simultaneously: Taking an expectation with respect to the belief of the classifier f_θ would thus require $\mathcal{O}(C^{|\mathcal{U}|})$ evaluations, each of which involves training a surrogate model from scratch, which is intractable.

Similar to computing the MP approximation, we emulate conditioning on additional observations by training a separate classifier on augmented data. That is, for conditioning on the observation $\mathbf{y}_i = c$, we train a model $f_{\hat{\theta}_{i,c}}$ and interpret its predictive distribution as the marginals of the joint distribution we try to approximate. Plugging everything together, we can estimate the numerator of Theorem 2.6 for a node i as follows:

$$
\begin{aligned}
\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}^{\text{gt}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}^{\text{gt}}_i\right] &\approx \mathbb{E}_{c \sim f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,:}} \left[\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}^{\text{gt}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = c\right]\right] \\
&= \sum_c \mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}^{\text{gt}}_{\mathcal{U}-i} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = c\right] f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c} \\
&\approx \sum_c \left(\prod_{j \in \mathcal{U}-i} f_{\hat{\theta}_{i,c}}(\boldsymbol{A}, \boldsymbol{X})_{j, \tilde{\mathbf{y}}^{(i,c)}_j} \right) f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c}
\end{aligned}
\tag{22}
$$

Note that here we approximate $\mathbf{y}^{\text{gt}}_{\mathcal{U}-i}$ not with the pseudo-labels of f_θ, but instead use the pseudo-labels of the surrogate classifier $f_{\hat{\theta}_{i,c}}$ which are defined as:

$$
\tilde{\mathbf{y}}^{(i,c)}_j := \operatorname{argmax}_{c \in [C]} f_{\hat{\theta}_{i,c}}(\boldsymbol{A}, \boldsymbol{X})_{j,c} \tag{23}
$$

We can now combine both Equations (21) and (22) and compute an estimate of the epistemic uncertainty associated with a node i:

$$
\begin{aligned}
\mathrm{u}^{\mathrm{epi}}(i, \mathbf{y}_i^{\mathrm{gt}}) &= \frac{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}, \mathbf{y}_i = \mathbf{y}_i^{\mathrm{gt}}\right]}{\mathbb{P}\left[\mathbf{y}_{\mathcal{U}-i} = \mathbf{y}_{\mathcal{U}-i}^{\mathrm{gt}} \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}}\right]} \\
&\approx \frac{\sum_c \left(\prod_{j \in \mathcal{U}-i} f_{\hat{\theta}_{i,c}}(\boldsymbol{A}, \boldsymbol{X})_{j, \tilde{\mathbf{y}}_j^{(i,c)}}\right) f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c}}{\frac{\prod_{j \in \mathcal{U}} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j, \hat{\mathbf{y}}_j}}{f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i}}} \\
&= \frac{\sum_c \left(\prod_{j \in \mathcal{U}-i} f_{\hat{\theta}_{i,c}}(\boldsymbol{A}, \boldsymbol{X})_{j, \tilde{\mathbf{y}}_j^{(i,c)}}\right) f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c}}{\prod_{j \in \mathcal{U}} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{j, \hat{\mathbf{y}}_j}} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i} \\
&\propto \sum_c \left(\prod_{j \in \mathcal{U}-i} f_{\hat{\theta}_{i,c}}(\boldsymbol{A}, \boldsymbol{X})_{j, \tilde{\mathbf{y}}_j^{(i,c)}}\right) f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i,c} f_\theta(\boldsymbol{A}, \boldsymbol{X})_{i, \hat{\mathbf{y}}_i}
\end{aligned}
\tag{24}
$$

J.3 Implementation

Efficiency. Both MP and ESP algorithms require training auxiliary classifiers on augmented datasets in order to estimate probabilities conditioned on unavailable observations. In each iteration of AL, the MP approximation requires training $|\mathcal{U}| \in \mathcal{O}(n)$ classifiers. The ESP approximation additionally takes an expectation over the unobserved class of each node, resulting in the training of $\mathcal{O}(nc)$ additional models per query. Notably, the auxiliary models used in the MP approximation are trained on $n-1$ labels while the classifiers $f_{\hat{\theta}_{i,c}}$ the RP algorithm relies are only trained on $|\mathcal{O}|+1$ labels. Therefore, even though the complexity of the MP is better in theory, in practice the runtime of the ESP approximation is significantly shorter than the MP paradigm. In fact, the MP approximation did not finish an AL run within 72 hours for two datasets (see Table 1).

Backbone Architecture and Training. Because of the aforementioned efficiency limitations, we use an SGC model as the backbone classifier for our proposed framework. Effectively, SGC is a logistic regression model fit to diffused node features $\boldsymbol{X}$. We use the SAGA solver which is efficient for larger datasets to approximate the aleatoric uncertainty of the ESP as it uses pseudo-labels of all nodes, and we rely on liblinear in all other cases. Furthermore, we account for class imbalances and use a regularization weight of $\lambda = 1.0$. To mimic the GNNs used by other baselines, we diffuse the node features $\boldsymbol{X}$ for 2 iterations. While acquisition requires training and evaluating auxiliary models $f_{\hat{\theta}}$ on pseudo-labels $\hat{\mathbf{y}}$ or $\tilde{\mathbf{y}}$ for both approximation frameworks, we only train the underlying classifier f_θ that we report numbers on using ground-truth labels iteratively revealed by the oracle.

Discussion. We also point out reasons for the proposed method to not be at its full efficacy yet due to the various assumptions and approximations we make. Specifically, different sources of error in estimating epistemic uncertainty can stem from (i) The classifiers f_θ and $f_{\hat{\theta}_{i,c}}$ may not faithfully model the true generative process, as described in Section 2, which results in suboptimal performance. (ii) The pseudo-labels $\hat{\mathbf{y}}$ may not match the true labels $\mathbf{y}$, which in turn leads to errors in approximating aleatoric confidence and querying epistemic confidence at the correct label. (iii) The classifiers may be poorly calibrated, a tendency exhibited by some GNN architectures [15]. In fact, we observed similar experiments using a GCN instead of an SGC as a backbone to be unsuccessful. While both MP and ESP approximation paradigms aim to estimate the same quantity, they rely on different assumptions and approximations. Therefore, it is expected that they behave differently in practice. Nonetheless, they both are indicative of the potential properly disentangled uncertainty brings to AL. Indeed, as Table 1 verifies even under various approximations disentangling uncertainty greatly improves US. We suspect the ESP to perform significantly better because it relies less on pseudo-labels: It factors in the belief of the classifier f_θ more accurately and only trains surrogate models on one unobserved class label.

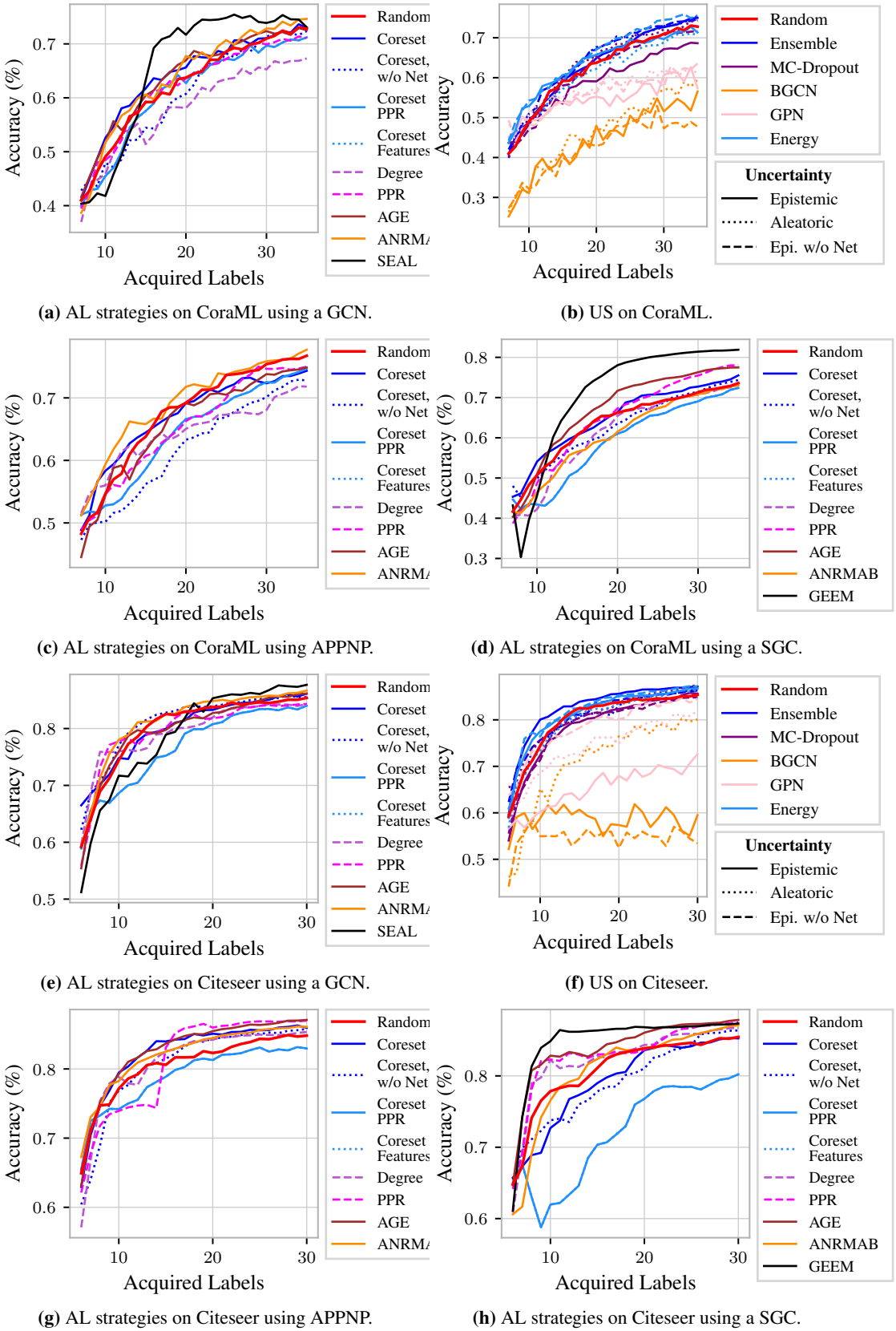
Figures 1 and 12a to 12d showcase the practical applicability of this framework. We compare both MP and ESP approximations to the best-performing US and non-US AL strategies. MP outperforms other uncertainty estimators on Citeseer, while showing worse-than-random performance on Amazon Photos. The ESP approximation consistently yields strong results that outperform other epistemic uncertainty estimators. This underlines that disentangled epistemic uncertainty, in many instances, has the potential to be an effective guide for US. Future work can build upon the results of our analysis

by, for example, finding more sensible and efficient approximations to uncertainty disentangling and equipping estimators with the capabilities to describe a broader family of data-generating processes.

J.4 Approximating Ground-Truth Uncertainty without Graph Inductive Biases

We verify the impact of considering network effects when approximating ground-truth uncertainty to apply the proposed algorithm. Again, we use an SGC backbone and compute uncertainty using the aforementioned ESP approach, see Appendix J.2. We ablate estimating the marginal probabilities from the classifier that is aware of the network (by diffusing the input features) against predictions from the same model that ignores the graph. Notably, we only omit network effects when approximating uncertainty. Both training and evaluation is done with consideration for instance interdependence to enable a fair comparison between the two US strategies.

Figure 13 shows that in all settings, performance deteriorates when ignoring network effects. This is in line with the discussion in **??**: The graph allows a faithful approximation of the underlying generative process and enables the ESP algorithm to identify informative queries. As has been shown by [58], modelling instance interdependence sufficiently mitigates the need for complex feature transformations. While a recent line of works regarding Energy-based models (EBMs) [68] explores how to interpret classifiers on i.i.d. data as surrogates for the data generating process, for such problems typically significantly more complex feature transformations are required. Since we can not rely on strong inductive biases implied by the graph structure, faithfully approximating the generating process and thus also ground-truth uncertainty becomes more challenging.



(a) AL strategies on CoraML using a GCN.

(b) US on CoraML.

(c) AL strategies on CoraML using APPNP.

(d) AL strategies on CoraML using a SGC.

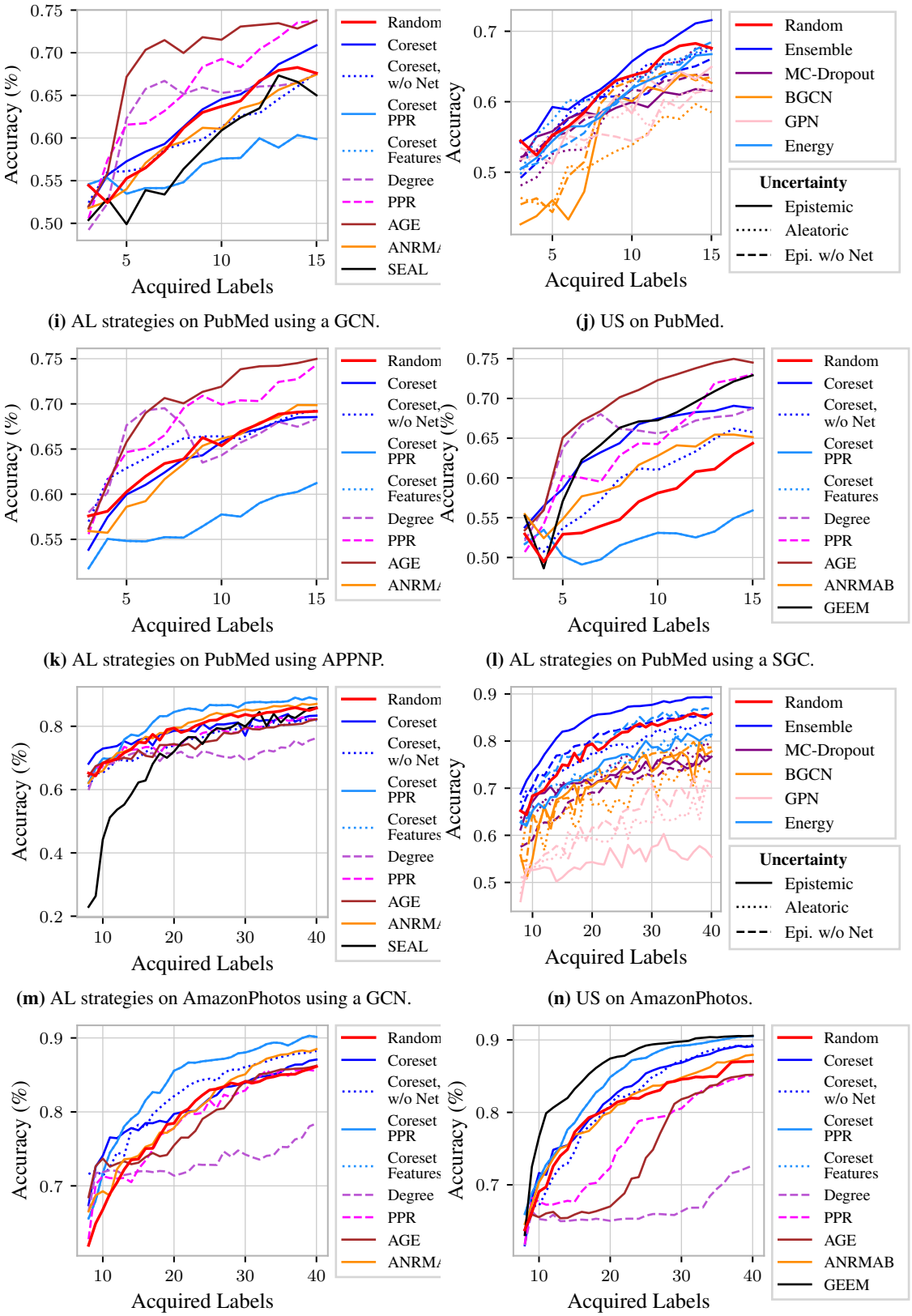
(e) AL strategies on Citeseer using a GCN.

(f) US on Citeseer.

(g) AL strategies on Citeseer using APPNP.

(h) AL strategies on Citeseer using a SGC.

Figure 4: Accuracy curves of AL strategies, both non-uncertainty-based as well as US on different datasets for different models.



(i) AL strategies on PubMed using a GCN.

(j) US on PubMed.

(k) AL strategies on PubMed using APPNP.

(l) AL strategies on PubMed using a SGC.

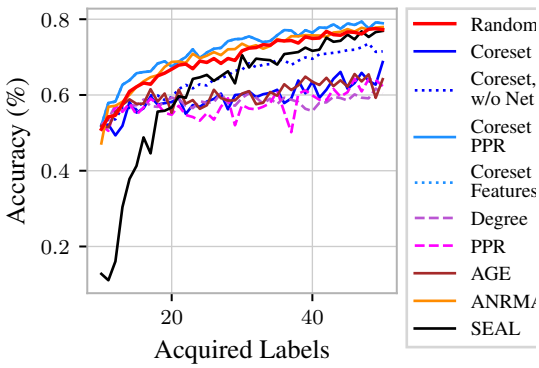
(m) AL strategies on AmazonPhotos using a GCN.

(n) US on AmazonPhotos.

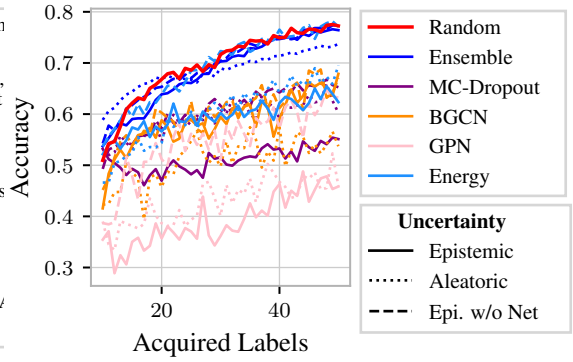
(o) AL strategies on AmazonPhotos using APPNP.

(p) AL strategies on AmazonPhotos using a SGC.

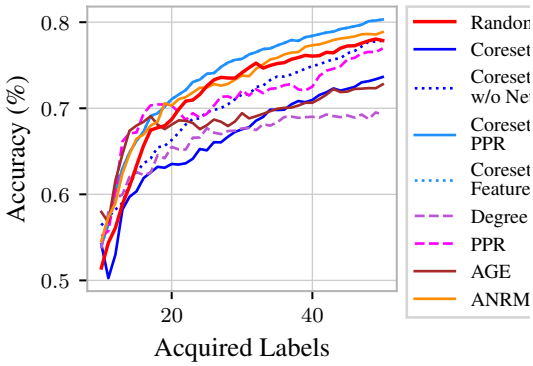
Figure 4: Accuracy curves of AL strategies, both non-uncertainty-based as well as US on different datasets for different models (cont.).



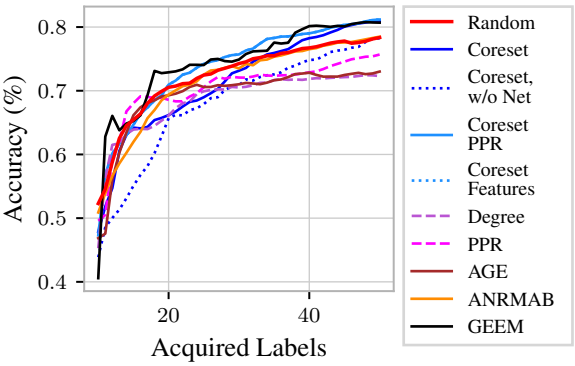
(q) AL strategies on AmazonComputers using a GCN.



(r) US on AmazonComputers.



(s) AL strategies on AmazonComputers using APPNP.



(t) AL strategies on AmazonComputers using a SGC.

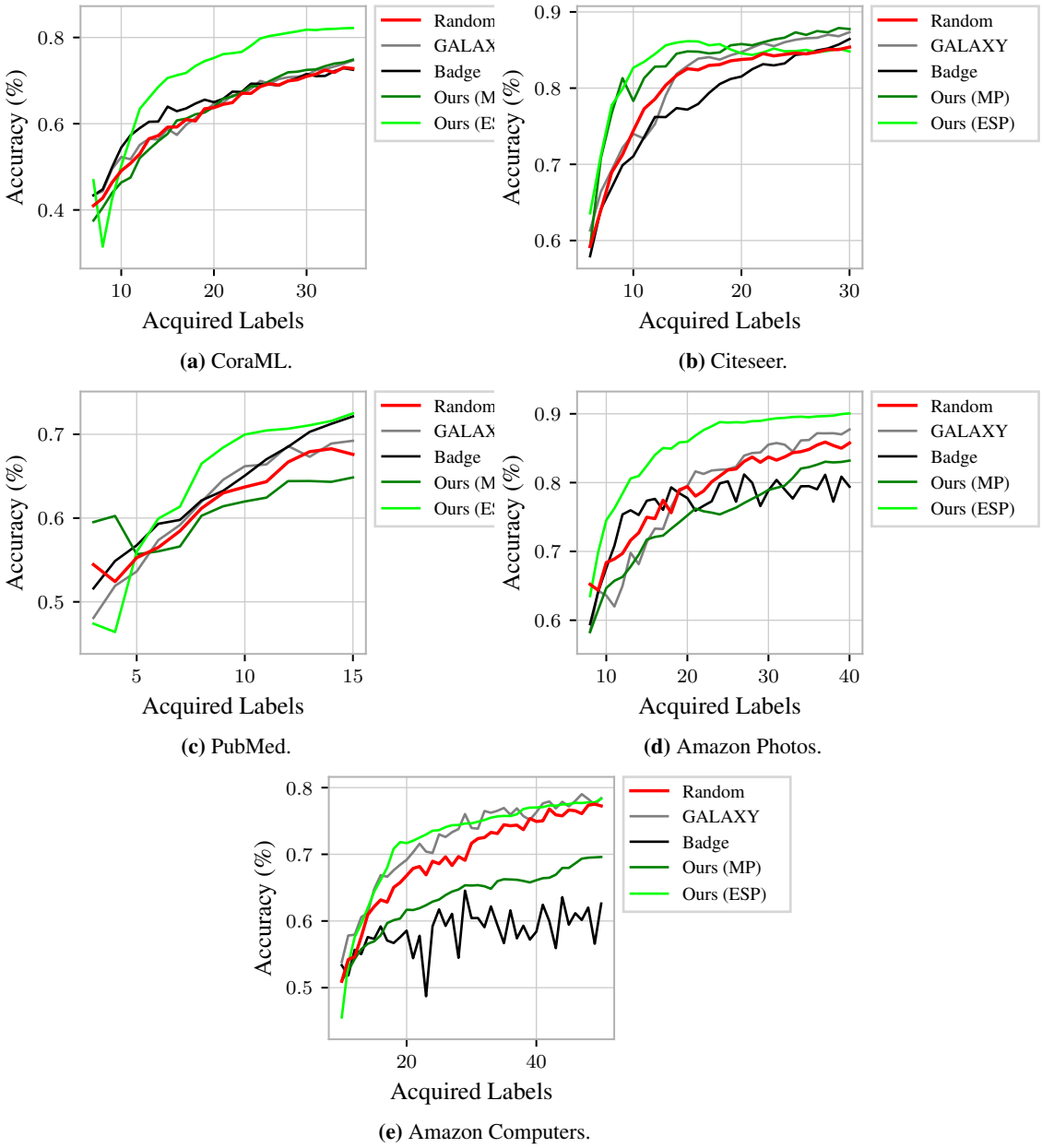
Figure 4: Accuracy curves of AL strategies, both non-uncertainty-based as well as US on different datasets for different models (cont.).

Table 5: Average AUC (↑) for different acquisition strategies on different models and datasets and the corresponding standard deviation over 5 different dataset splits and 5 model initializations each. We mark the best strategy per model in bold and underline the second best. For each dataset, we additionally mark the overall best model and strategy with the † symbol.

		Baselines	**Non-Uncertainty**												**Uncertainty**			
	Inputs		*A & X*							*A*			*X*		*A & X*		*X*	
	Model	Random	Coreset	AGE	ANRMAB	GEEM	SEAL	GALAXY	Badge	Coreset PPR	PPR	Degree	Coreset w/o Net	Coreset Inputs	Epi./ (Energy)	Alea.	Epi./ (Energy)	Alea.
CoraML	GCN	62.51±4.31	64.35±4.27	64.12±3.67	64.24±4.27	n/a	**66.07±3.94**	63.24±5.49	64.51±4.14	59.53±7.57	62.16±4.53	58.39±5.01	60.73±4.33	61.26±4.35	63.97±6.29	61.30±5.63	65.65±4.28	64.33±4.17
	APPNP	67.72±5.70	67.32±4.20	66.12±3.36	69.49±5.52	n/a	n/a	n/a	n/a	**71.04±6.28**	65.52±3.48	64.57±4.79	62.01±5.26	64.49±4.38	64.92±7.02	67.68±5.09	69.59±4.28	66.69±5.03
	Ensemble	63.89±6.15	60.55±5.52	64.80±4.33	65.10±7.29	n/a	n/a	n/a	n/a	62.65±7.08	65.47±4.26	62.41±3.04	60.52±7.37	65.07±5.11	63.47±6.20	64.03±6.68	64.80±5.14	**65.82±3.01**
	MC-Dropout	**64.94±4.93**	64.37±4.58	64.44±4.37	64.06±4.94	n/a	n/a	n/a	n/a	62.92±6.38	63.91±5.68	61.65±3.90	59.58±4.58	64.35±4.29	59.17±5.57	63.69±4.39	61.82±5.19	63.87±4.87
	BGCN	45.76±3.26	49.37±3.47	51.25±4.48	47.23±3.04	n/a	n/a	n/a	n/a	39.43±3.87	**54.64±2.54**	48.68±3.68	43.65±3.73	44.85±3.15	44.45±2.28	46.61±3.07	42.74±3.57	48.11±3.16
	GPN	56.50±4.73	n/a	n/a	n/a	n/a	n/a	n/a	n/a	58.04±4.56	**59.88±3.05**	58.74±3.53	n/a	54.02±3.83	54.75±3.21	57.16±5.25	55.89±4.29	57.21±5.12
	SGC	63.85±6.05	65.23±5.14	67.56±3.20	61.14±5.50	**71.39†±3.37**	n/a	n/a	n/a	60.24±7.98	65.05±2.85	61.60±4.34	62.94±5.41	59.18±5.42	67.51±3.64	65.66±5.70	65.05±4.71	67.13±4.11
Citeseer	GCN	80.34±3.20	80.36±2.81	80.14±3.00	81.59±3.11	n/a	79.19±4.86	81.06±3.59	78.41±4.78	81.51±2.97	79.93±2.16	79.72±2.47	81.41±2.28	76.87±5.65	81.59±3.36	80.15±5.10	**82.26±2.82**	76.79±4.07
	APPNP	80.77±5.34	82.62±2.53	**83.00±2.59**	82.31±3.10	n/a	n/a	n/a	n/a	80.70±4.49	81.37±4.85	81.36±3.32	80.89±5.26	79.20±7.16	80.37±6.14	77.77±8.04	81.94±4.95	79.15±4.96
	Ensemble	81.14±3.99	77.61±4.28	82.37±2.44	81.43±3.62	n/a	n/a	n/a	n/a	81.59±3.71	81.42±2.61	80.31±2.45	80.18±4.60	79.13±4.71	**82.94±2.72**	80.26±6.82	81.06±4.56	77.76±5.03
	MC-Dropout	**81.90±3.19**	78.97±3.47	79.86±3.31	80.46±2.99	n/a	n/a	n/a	n/a	81.09±3.38	79.86±3.58	80.82±2.85	79.40±4.15	78.62±4.98	78.86±5.47	80.97±5.28	78.85±4.73	76.02±4.46
	BGCN	70.17±3.64	73.86±2.14	**76.30±1.42**	70.70±3.28	n/a	n/a	n/a	n/a	69.32±3.41	73.77±2.73	75.55±1.47	68.59±3.91	67.51±3.64	58.68±7.96	71.23±3.80	54.85±6.96	70.33±4.14
	GPN	77.07±5.06	n/a	n/a	n/a	n/a	n/a	n/a	n/a	72.83±6.88	79.52±2.53	**80.65±1.69**	n/a	71.33±7.55	65.31±10.21	74.28±7.63	78.73±5.24	77.28±4.85
	SGC	81.04±5.06	79.38±4.44	84.21±1.95	81.03±3.28	**85.25†±1.80**	n/a	n/a	n/a	81.59±4.15	83.25±2.74	82.60±1.49	79.21±7.73	72.04±5.66	75.13±9.12	78.85±6.36	72.94±8.35	82.90±1.77
Pubmed	GCN	61.56±4.52	62.61±5.20	**69.48±4.12**	60.31±6.68	n/a	58.62±5.27	62.05±5.73	63.34±6.01	61.71±7.24	66.07±4.01	63.84±5.40	60.33±6.04	56.71±6.44	59.64±6.02	61.85±5.43	59.66±5.23	60.34±5.98
	APPNP	64.61±6.05	63.88±6.47	**70.18†±4.50**	63.83±5.85	n/a	n/a	n/a	n/a	64.21±6.62	68.24±4.44	66.09±4.25	65.51±4.18	56.87±8.10	63.09±7.08	62.37±5.82	62.23±7.74	63.95±6.34
	Ensemble	59.26±6.36	64.25±6.60	**68.26±4.28**	60.40±6.98	n/a	n/a	n/a	n/a	61.89±5.92	63.48±7.18	62.78±7.21	59.19±6.73	56.36±5.41	63.70±6.26	61.37±4.97	59.71±8.14	61.15±4.90
	MC-Dropout	58.30±6.21	62.97±6.29	65.24±6.30	60.50±4.68	n/a	n/a	n/a	n/a	61.43±5.41	**65.72±3.92**	65.29±5.23	57.71±5.70	56.01±6.09	58.67±7.52	59.07±7.03	59.23±6.99	62.22±4.68
	BGCN	53.59±5.46	**59.29±4.21**	56.93±3.14	52.68±2.92	n/a	n/a	n/a	n/a	53.40±4.46	59.12±2.49	55.93±5.47	53.63±4.50	51.40±4.15	55.19±3.29	52.81±4.77	57.09±2.93	54.62±5.29
	GPN	59.76±6.46	n/a	n/a	n/a	n/a	n/a	n/a	n/a	62.08±6.79	**64.64±3.83**	58.13±4.94	n/a	54.34±7.91	58.82±8.90	57.24±8.31	56.25±6.23	59.37±6.96
	SGC	56.79±6.28	64.48±5.35	**69.20±4.83**	60.49±6.30	64.82±3.68	n/a	n/a	n/a	62.15±5.47	63.88±4.65	65.14±4.69	59.57±4.45	52.25±6.89	62.04±5.24	61.55±6.65	61.04±5.84	60.74±6.28
AmazonPhotos	GCN	79.06±3.86	78.58±2.44	75.17±5.50	79.97±4.20	n/a	71.16±3.02	78.87±4.84	76.97±5.12	70.20±8.55	76.30±3.13	71.00±5.92	75.40±4.41	**82.71±3.03**	74.66±7.12	74.61±5.99	79.96±4.84	79.63±4.59
	APPNP	79.29±6.17	81.04±2.86	79.02±5.26	80.35±4.96	n/a	n/a	n/a	n/a	76.37±7.76	79.03±3.83	73.25±8.33	82.31±3.29	**84.24±4.37**	79.72±6.19	77.45±6.13	80.48±4.97	77.69±5.83
	Ensemble	82.23±2.91	80.44±4.48	77.45±3.41	82.77±4.62	n/a	n/a	n/a	n/a	74.93±8.07	77.32±4.53	75.17±6.07	76.95±5.38	84.04±3.15	**84.46±3.69**	77.85±6.69	80.50±4.83	81.25±4.27
	MC-Dropout	80.32±3.75	76.63±4.47	74.75±3.45	80.21±3.25	n/a	n/a	n/a	n/a	75.32±6.70	74.33±3.46	69.03±6.60	75.71±4.62	**82.45±2.84**	72.42±6.79	73.16±6.19	69.68±8.19	78.80±4.17
	BGCN	71.22±3.86	67.15±4.70	65.69±5.06	70.69±3.38	n/a	n/a	n/a	n/a	59.34±7.49	64.51±4.28	61.23±5.84	69.78±4.24	**73.39±4.16**	70.83±3.36	67.83±5.83	72.21±2.84	69.19±5.80
	GPN	62.80±4.22	n/a	n/a	n/a	n/a	n/a	n/a	n/a	55.59±4.09	62.17±4.08	56.77±4.94	n/a	**65.07±2.74**	54.78±3.57	60.53±4.17	62.90±2.61	62.41±2.75
	SGC	80.52±4.89	82.32±2.58	74.01±5.96	80.92±4.04	**86.43†±4.13**	n/a	n/a	n/a	66.94±7.86	76.16±4.06	66.63±8.12	81.47±5.08	84.24±3.59	84.01±4.72	71.43±8.37	80.75±4.23	76.38±7.40
AmazonComputers	GCN	69.80±3.43	59.36±4.83	60.07±6.85	70.70±2.66	n/a	61.51±4.11	72.32±2.47	58.48±4.40	61.22±5.54	57.45±4.48	58.14±5.29	65.13±4.31	**72.34±2.75**	59.62±7.26	60.17±6.98	69.34±4.75	69.87±3.77
	APPNP	71.69±3.34	66.62±3.55	68.89±3.32	72.69±3.27	n/a	n/a	n/a	n/a	65.75±6.76	70.91±3.52	66.42±4.73	70.22±4.44	**73.83±3.56**	62.03±8.44	62.26±10.26	68.79±5.46	71.72±3.79
	Ensemble	72.56±3.06	64.19±4.49	60.69±5.43	72.73±4.16	n/a	n/a	n/a	n/a	64.13±6.76	60.59±7.02	63.16±5.33	67.20±6.40	**75.39†±2.68**	68.38±5.04	68.47±7.74	69.49±8.35	73.67±3.30
	MC-Dropout	68.06±4.65	56.58±4.30	57.04±4.61	69.01±2.92	n/a	n/a	n/a	n/a	64.43±6.67	55.88±4.81	56.72±6.12	61.86±3.16	**71.05±2.88**	51.02±9.05	59.31±7.96	60.71±7.40	70.66±3.97
	BGCN	60.52±3.72	45.65±3.02	46.72±3.74	60.32±4.31	n/a	n/a	n/a	n/a	39.86±8.46	43.61±4.17	45.52±2.93	58.85±3.41	**60.79±3.89**	58.64±5.96	51.11±6.23	60.19±3.82	59.20±3.65
	GPN	52.26±5.90	n/a	n/a	n/a	n/a	n/a	n/a	n/a	34.71±4.90	49.30±5.00	45.21±5.32	n/a	**56.32±3.79**	39.21±3.92	42.95±4.53	54.83±4.84	53.71±3.91
	SGC	72.39±2.74	71.53±3.86	69.31±4.37	71.62±3.79	**74.49±3.43**	n/a	n/a	n/a	58.52±0.44	70.35±3.84	68.70±3.24	68.14±5.46	73.91±3.30	61.02±8.74	59.34±10.50	66.53±5.88	69.24±5.52

Table 6: Average final classification accuracy (↑) for different acquisition strategies on different models and datasets and the corresponding standard deviation over 5 different dataset splits and 5 model initializations each. We mark the best strategy per model in bold and underline the second best. For each dataset, we additionally mark the overall best model and strategy with the † symbol.

		Baselines	**Non-Uncertainty**												**Uncertainty**			
	Inputs		*A & X*							*A*			*X*		*A & X*		*X*	
	Model	Random	Coreset	AGE	ANRMAB	GEEM	SEAL	GALAXY	Badge	Coreset PPR	PPR	Degree	Coreset w/o Net	Coreset Inputs	Epi./ (Energy)	Alea.	Epi./ (Energy)	Alea.
CoraML	GCN	72.80±3.74	72.95±4.17	72.54±3.61	74.64±3.46	n/a	73.08±6.13	74.67±4.82	72.51±4.05	69.33±6.04	71.10±4.26	67.31±5.93	72.36±4.27	71.21±4.18	71.34±5.39	70.69±7.56	**75.59±2.82**	74.08±3.79
	APPNP	76.74±4.45	74.33±4.02	74.86±2.76	77.71±4.59	n/a	n/a	n/a	n/a	76.11±4.08	74.99±2.24	71.81±3.78	72.78±3.84	74.69±3.38	72.29±5.83	75.61±3.88	**77.91±3.03**	75.29±3.71
	Ensemble	74.25±4.81	68.32±3.58	73.82±2.05	75.40±4.56	n/a	n/a	n/a	n/a	72.76±4.57	72.96±2.64	69.36±3.30	69.77±5.69	74.12±3.77	74.98±3.58	75.51±3.48	74.63±3.64	**75.62±2.29**
	MC-Dropout	73.86±5.46	72.44±5.59	72.40±4.09	73.44±5.36	n/a	n/a	n/a	n/a	70.91±5.15	72.06±5.10	69.58±3.78	71.42±5.19	74.06±1.94	68.64±4.57	74.42±4.15	71.67±4.63	**75.86±4.55**
	BGCN	59.09±5.86	61.08±6.05	61.20±8.35	57.80±6.47	n/a	n/a	n/a	n/a	46.46±7.81	**64.79±4.96**	59.40±5.58	56.22±4.89	55.73±7.12	56.53±7.54	59.48±7.56	47.72±7.13	57.87±7.44
	GPN	64.24±5.57	n/a	n/a	n/a	n/a	n/a	n/a	n/a	59.66±7.23	**66.85±3.48**	65.10±5.94	n/a	60.52±6.49	57.34±8.78	61.08±8.07	63.55±7.16	65.97±7.69
	SGC	73.59±4.99	75.50±3.70	77.49±1.44	72.99±4.52	**81.89†±1.58**	n/a	n/a	n/a	66.40±6.16	78.02±1.26	73.14±3.15	74.35±4.61	72.39±4.66	77.90±2.49	75.23±3.96	76.59±2.79	77.03±3.59
Citeseer	GCN	85.38±2.24	86.13±1.80	86.15±1.99	86.68±1.89	n/a	**87.69±2.00**	87.35±1.64	86.45±1.39	86.03±1.72	84.31±1.58	84.31±1.59	85.51±1.90	84.01±4.56	86.74±2.45	86.39±2.89	87.21±1.98	87.39±1.95
	APPNP	84.82±3.78	85.96±2.64	87.09±1.55	86.09±2.08	n/a	n/a	n/a	n/a	85.31±3.32	86.99±1.48	85.33±1.68	85.78±2.98	82.97±5.03	86.55±3.81	84.49±5.34	85.83±3.38	**87.27±1.90**
	Ensemble	85.63±3.26	85.00±2.65	86.07±2.35	86.77±2.16	n/a	n/a	n/a	n/a	86.19±3.25	85.07±1.50	84.90±1.59	86.05±1.40	83.98±3.30	**87.15±1.54**	86.41±4.64	86.27±2.37	**87.15±2.28**
	MC-Dropout	86.64±1.90	86.52±1.88	85.50±1.83	86.07±2.77	n/a	n/a	n/a	n/a	86.01±2.24	84.56±1.95	85.23±1.13	85.12±2.47	84.74±3.07	85.63±2.50	87.59±2.09	84.84±3.57	**88.06†±2.00**
	BGCN	78.89±7.20	80.45±3.60	79.59±4.58	79.37±4.66	n/a	n/a	n/a	n/a	75.14±6.50	80.44±4.52	**81.05±3.44**	76.46±5.00	76.41±3.51	59.50±13.96	80.08±5.25	53.42±12.61	78.44±5.55
	GPN	80.96±4.76	n/a	n/a	n/a	n/a	n/a	n/a	n/a	77.06±7.62	83.35±2.24	82.46±2.34	n/a	79.14±4.86	72.61±7.77	81.87±5.27	**84.58±3.86**	81.60±3.65
	SGC	85.45±3.72	85.26±2.90	**87.84±1.10**	87.09±1.86	87.33±1.95	n/a	n/a	n/a	87.17±2.14	87.49±1.05	86.73±1.73	86.37±2.99	80.23±4.86	75.63±10.15	86.19±3.54	80.16±10.35	87.78±1.98
Pubmed	GCN	67.60±5.10	70.87±5.94	**73.79±3.92**	67.44±8.55	n/a	65.01±7.30	69.23±5.44	72.14±5.16	66.73±5.80	73.72±3.72	67.57±6.48	67.66±5.04	59.87±7.29	66.75±8.22	67.84±6.27	68.46±7.13	65.62±7.53
	APPNP	69.18±5.57	68.54±7.37	**74.98†±4.48**	69.86±5.37	n/a	n/a	n/a	n/a	69.19±5.81	74.33±3.52	68.33±4.29	69.24±5.44	61.23±9.30	70.86±5.61	68.16±6.96	68.09±8.57	68.93±6.57
	Ensemble	66.10±6.96	70.89±6.16	**72.65±3.98**	68.74±6.24	n/a	n/a	n/a	n/a	68.24±5.75	69.44±6.61	66.11±6.92	65.03±5.24	60.25±6.79	71.57±6.75	67.35±5.07	66.10±6.61	68.07±5.96
	MC-Dropout	66.45±4.99	71.19±7.79	71.15±6.03	65.50±4.56	n/a	n/a	n/a	n/a	67.88±6.79	**72.20±3.46**	70.11±7.06	62.32±7.93	60.82±6.69	61.55±8.42	67.70±7.60	63.82±7.51	69.75±4.93
	BGCN	62.57±8.73	66.70±4.98	63.77±5.06	60.72±6.22	n/a	n/a	n/a	n/a	59.02±9.08	**68.97±3.86**	58.06±5.41	60.36±6.70	54.49±6.33	62.67±5.46	58.51±7.89	63.49±6.15	61.25±7.83
	GPN	63.75±8.43	n/a	n/a	n/a	n/a	n/a	n/a	n/a	66.96±7.94	**67.79±5.93**	61.50±4.74	n/a	54.32±9.25	64.94±7.58	63.31±8.19	61.74±6.70	65.38±9.24
	SGC	64.36±5.81	68.78±3.81	**74.51±4.79**	65.14±7.04	72.91±5.03	n/a	n/a	n/a	67.71±5.66	73.05±4.78	68.76±4.42	65.76±4.13	55.91±8.35	70.29±6.11	67.61±5.85	67.19±7.03	68.18±6.85
AmazonPhotos	GCN	85.76±3.16	83.37±2.57	82.20±2.98	87.07±3.40	n/a	85.92±2.50	87.72±4.07	79.40±7.22	75.73±8.56	82.28±3.15	76.27±5.99	81.76±3.61	**88.58±2.45**	81.39±7.57	81.12±5.41	86.82±4.51	84.73±4.33
	APPNP	86.14±4.30	87.07±1.60	86.24±2.13	88.49±2.49	n/a	n/a	n/a	n/a	81.93±7.54	85.52±2.33	78.44±6.60	88.24±1.60	**90.14±1.73**	86.07±4.41	84.31±4.21	86.65±3.87	85.28±4.15
	Ensemble	87.97±2.14	84.82±3.48	83.32±4.04	88.10±3.30	n/a	n/a	n/a	n/a	79.82±7.86	83.36±4.01	78.60±5.09	83.51±3.15	**89.96±1.61**	89.28±2.36	83.92±6.09	85.53±4.31	87.17±3.75
	MC-Dropout	86.04±3.39	83.37±3.20	80.08±7.52	87.53±3.21	n/a	n/a	n/a	n/a	79.07±6.75	80.45±9.57	72.42±6.13	82.11±6.62	**88.85±2.26**	76.69±8.01	78.69±5.88	76.44±8.43	87.09±2.69
	BGCN	79.22±7.13	69.90±9.01	73.08±9.53	**82.32±4.67**	n/a	n/a	n/a	n/a	59.80±14.79	69.34±10.50	62.35±12.45	76.64±6.02	77.46±10.16	77.81±8.13	73.15±9.69	78.66±8.01	77.17±7.59
	GPN	64.59±8.87	n/a	n/a	n/a	n/a	n/a	n/a	n/a	56.95±9.79	71.68±9.41	59.87±9.57	n/a	**72.46±10.43**	55.44±7.09	71.29±8.57	71.33±9.67	70.29±11.14
	SGC	87.04±2.72	89.13±1.79	85.20±2.79	87.95±2.43	**90.57†±2.83**	n/a	n/a	n/a	72.18±8.94	85.08±2.70	72.66±6.71	89.34±2.13	90.52±1.51	90.22±2.48	78.75±7.06	85.71±3.10	84.25±5.51
AmazonComputers	GCN	77.25±3.74	68.69±7.28	64.17±9.53	77.93±2.95	n/a	76.92±2.00	78.42±2.18	62.61±11.82	67.04±6.67	62.32±9.62	60.49±8.10	71.51±4.11	**78.94±2.69**	62.35±11.46	69.54±8.69	77.08±4.57	77.84±3.28
	APPNP	77.87±2.37	73.64±4.12	72.78±4.66	78.86±3.23	n/a	n/a	n/a	n/a	70.73±6.69	76.96±3.02	69.24±5.44	77.85±2.75	**80.31±2.32**	67.25±9.27	71.19±10.07	78.44±4.37	79.87±3.42
	Ensemble	77.88±2.90	69.20±5.50	66.15±4.00	78.79±4.64	n/a	n/a	n/a	n/a	71.33±3.58	65.49±5.88	66.32±5.77	71.42±5.38	**80.50±2.38**	76.41±3.94	73.71±7.31	77.38±5.33	80.38±4.37
	MC-Dropout	75.86±3.93	63.65±11.06	62.67±8.51	75.07±3.73	n/a	n/a	n/a	n/a	70.53±7.71	59.78±11.42	58.54±7.70	72.24±4.90	**78.17±3.71**	55.12±11.72	60.21±11.38	65.25±10.33	77.00±3.59
	BGCN	63.23±9.50	39.84±15.30	45.25±13.93	65.87±9.88	n/a	n/a	n/a	n/a	35.80±15.73	45.34±12.66	44.23±12.92	63.64±7.52	62.32±13.64	**67.87±10.76**	53.71±13.66	67.06±8.51	64.32±11.97
	GPN	59.89±13.72	n/a	n/a	n/a	n/a	n/a	n/a	n/a	32.00±8.75	60.63±10.59	46.42±16.56	n/a	**64.79±10.60**	45.85±13.64	45.95±14.32	63.25±3.79	64.24±6.71
	SGC	78.35±2.44	80.98±2.86	73.01±4.68	78.49±2.34	80.69±2.65	n/a	n/a	n/a	64.92±0.75	75.69±3.56	72.31±3.54	78.24±3.16	**81.19†±2.25**	68.70±8.90	67.44±10.26	73.46±6.06	78.38±4.05



(a) CoraML.

(b) Citeseer.

(c) PubMed.

(d) Amazon Photos.

(e) Amazon Computers.

Figure 5: US using the i.i.d. GALAXY and BADGE baselines versus our approach (ESP) and random sampling.

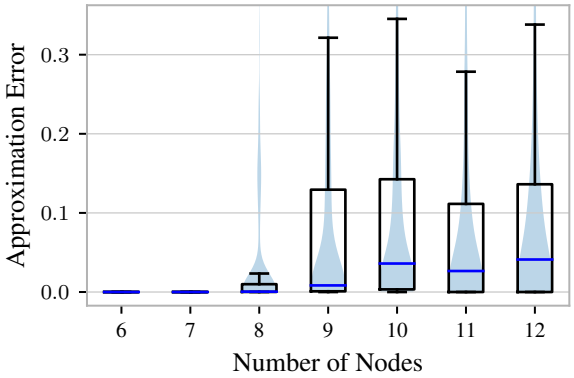
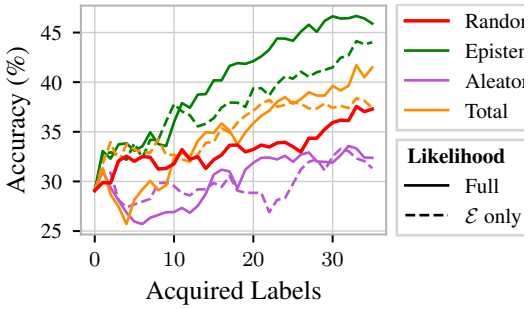
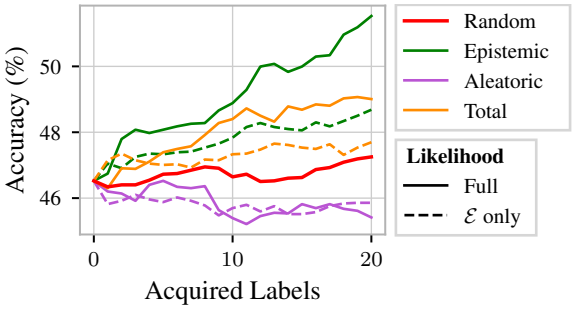


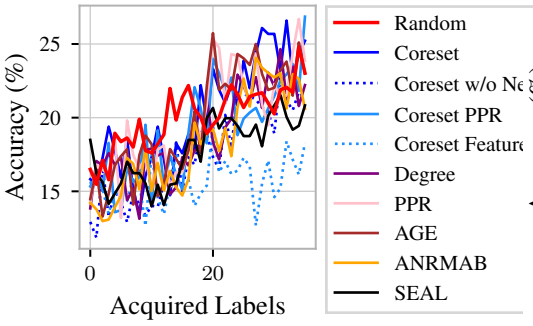
Figure 6: Absolute error distribution between approximate total confidence $q(\mathbf{y}_i)$ and true total confidence $p(\mathbf{y}_i \mid \boldsymbol{A}, \boldsymbol{X}, \mathbf{y}_{\mathcal{O}})$ for graphs of different sizes.



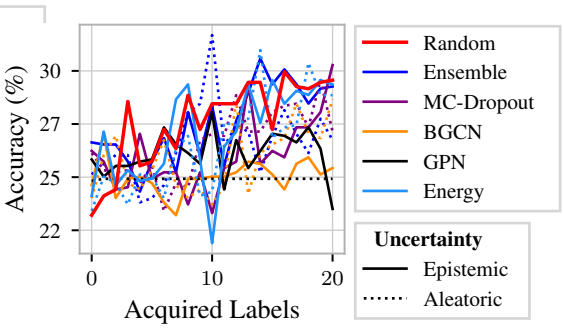
(a) AL using ground-truth uncertainty on an CSBM with 100 nodes and 7 classes.



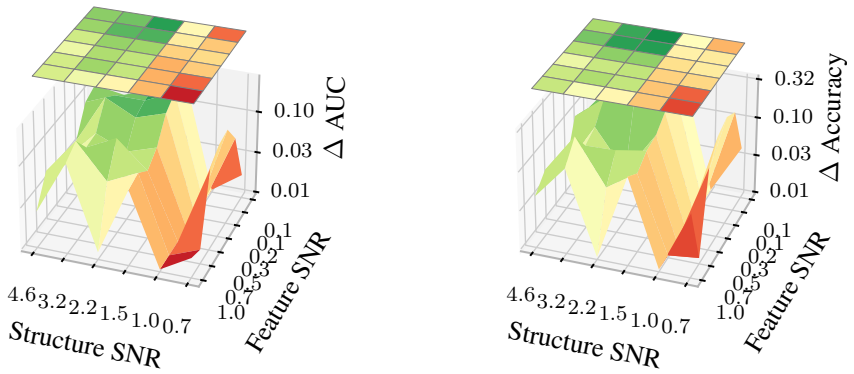
(b) AL using ground-truth uncertainty on an CSBM with 1000 nodes and 4 classes.



(c) AL using contemporary US strategies on a CSBM with 100 nodes and 7 classes.



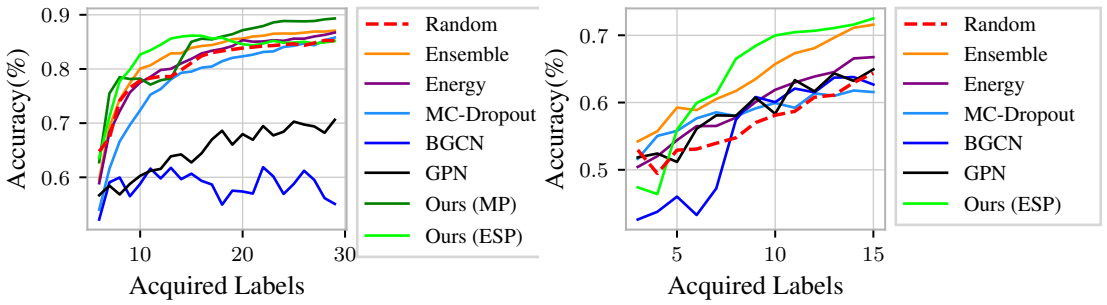
(d) AL using contemporary US strategies on a CSBM with 1000 nodes and 4 classes.



(a) Improvement in AUC

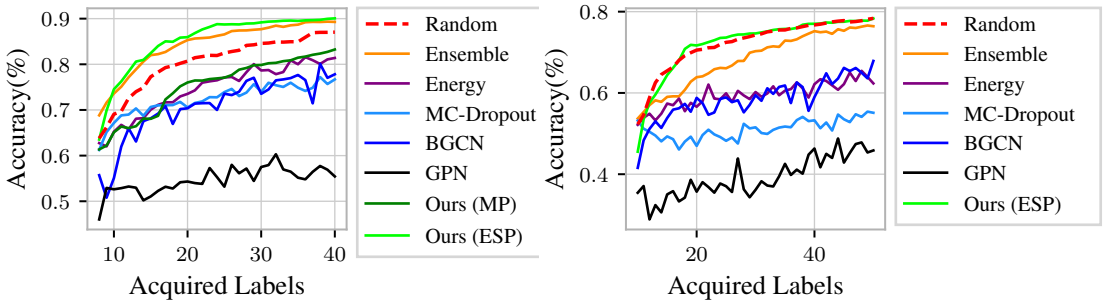
(b) Improvement in test accuracy

Figure 8: Evaluating the absolute improvement of Uncertainty Sampling using epistemic uncertainty over random acquisition for different structure and feature SNRs.



(a) Our proposed framework versus other US strategies on Citeseer.

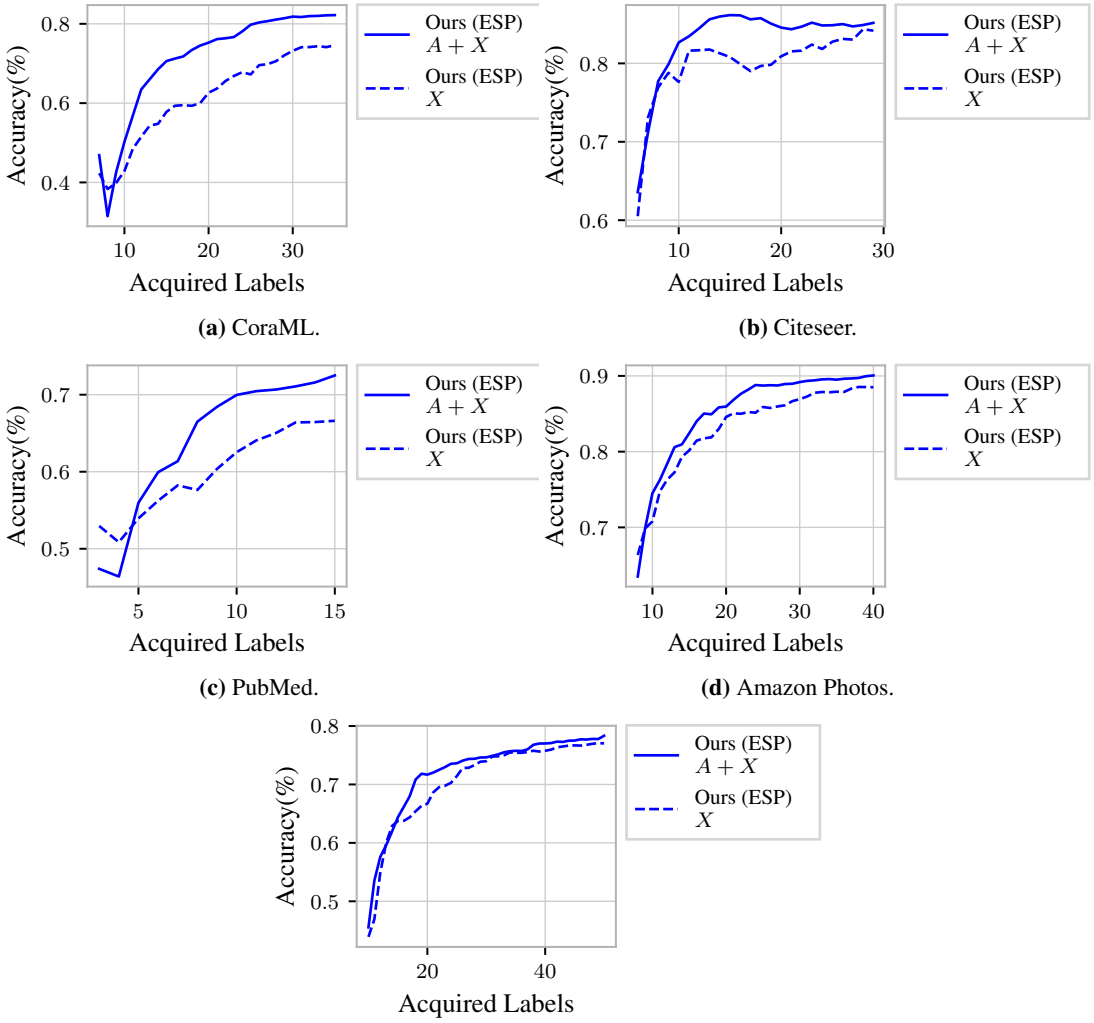
(b) Our proposed framework versus other US strategies on PubMed.



(c) Our proposed framework versus other US strategies on Amazon Photos.

(d) Our proposed framework versus other US strategies on Amazon Computers.

Figure 12: Our proposed uncertainty disentanglement framework applied to an SGC classifier using the MP or ESP approximation.



(a) CoraML.

(b) Citeseer.

(c) PubMed.

(d) Amazon Photos.

(e) Amazon Computers.

Figure 13: US using our proposed approximation (ESP) utilizing the features ($\boldsymbol{A} + \boldsymbol{X}$) and network effects versus only using features ($\boldsymbol{X}$).