Data Mining and Knowledge Discovery (2024) 38:110–127
https://doi.org/10.1007/s10618-023-00959-z

Improving Graph Neural Networks by combining active learning with self-training

Georgios Katsimpras[1] · **Georgios Paliouras**[1]

Received: 11 February 2023 / Accepted: 12 July 2023 / Published online: 11 August 2023

Abstract
In this paper, we propose a novel framework, called STAL, which makes use of unlabeled graph data, through a combination of Active Learning and Self-Training, in order to improve node labeling by Graph Neural Networks (GNNs). GNNs have been shown to perform well on many tasks, when sufficient labeled data are available. Such data, however, is often scarce, leading to the need for methods that leverage unlabeled data that are abundant. Active Learning and Self-training are two common approaches towards this goal and we investigate here their combination, in the context of GNN training. Specifically, we propose a new framework that first uses active learning to select highly uncertain unlabeled nodes to be labeled and be included in the training set. In each iteration of active labeling, the proposed method expands also the label set through self-training. In particular, highly certain pseudo-labels are obtained and added automatically to the training set. This process is repeated, leading to good classifiers, with a limited amount of labeled data. Our experimental results on various datasets confirm the efficiency of the proposed approach.

Keywords Active learning · Self-training · Graph Neural Networks

1 Introduction

With the success of deep learning on various tasks, a new set of methods have emerged, called Graph Neural Networks (GNNs), that achieve remarkable performance on graph-based data (Kipf and Welling 2017; Hamilton et al. 2017).

Responsible editor: Charalampos Tsourakakis.

✉ Georgios Katsimpras
gkatsibras@iit.demokritos.gr

Georgios Paliouras
paliourg@iit.demokritos.gr

[1] NCSR Demokritos, 15341 Athens, Greece

A large number of different GNN approaches have been proposed in the literature, aiming to tackle mostly node classification or link prediction problems (Veličković et al. 2017; Xu et al. 2019; Wu et al. 2019; Zhang and Chen 2018). The predictive ability of these models relies mostly on aggregating information from the direct neighborhood of the node or edge to be labeled. However, when the labeled data is limited, the data chosen to be labeled can affect significantly the efficiency of GNNs (Zhu and Goldberg 2009; Sun et al. 2020).

Towards this direction, recent studies suggest that Active Learning (AL) is an effective technique to improve the models' robustness (Aggarwal et al. 2014; Ren et al. 2021). Specifically, AL iteratively selects data to be labeled and used to train the model. In the AL literature, there are two main data selection strategies (Settles and Craven 2008): (i) uncertainty-based and (ii) distribution-based. The former approach chooses the most uncertain samples based on the entropy of the learned model. The latter approach selects the samples that better represent the underlying distribution of training instances, e.g. using centrality measures such as Pagerank. Both data selection approaches depend heavily on the initial set of labeled samples.

On the other hand, the recent rise of self-training (ST) showed that enlarging the label set with the most confident pseudo-labels generated by the model learned thus far, can improve the representation learning of GNNs (Li et al. 2018; Wang et al. 2021). The core idea of self-training is to automatically augment the label set with unlabeled samples that can be labeled with high confidence by the model. The augmented label set is then used to train the final model. Nevertheless, a known issue of ST methods is that the predicted pseudo-labels may introduce noise, and bias the learning process (Dai et al. 2021). Therefore, great care in selecting the pseudo-labels is needed.

Despite various advances in both AL and ST, there is limited research on combining the two approaches. Some attempts have been made in natural language processing (Kwak et al. 2022; Yu et al. 2022) and computer vision (Chan et al. 2021; Feng et al. 2021), while recently a framework that combines AL with ST has been introduced (Fazakis et al. 2019), using traditional machine learning methods (not graph-based learning). To the best of our knowledge, there is only one approach in the literature (Zhu et al. 2020), that investigates the combination of AL and self-supervision on graph data. These initial results seem encouraging and have been an important motivation for our work.

In particular, the study presented here addresses the problem of node classification with the use of label-efficient techniques. We suggest that combining AL with ST can reduce the labeling effort and improve the training process of GNNs. In order to achieve this, we first design an AL strategy which selects highly uncertain cases from a large unlabeled set of nodes. Additionally, we propose a ST technique which enables GNNs to expand their label set by incorporating pseudo-labels that are predicted with confidence by the model. In this way, the predicted labels of one iteration are used as training data in subsequent iterations. Different from most active learning approaches that rely solely on highly uncertain samples, our method utilizes also highly certain ones, through self-training. Our experiments show that

the iterative application of these two steps can benefit the representation learning of GNNs.

Overall, the main contributions of the paper are the following:

- We propose a new framework that combines AL with ST to reduce labeling cost and boost the performance of GNNs.
- We introduce a simple but effective strategy to obtain reliable pseudo-labels in self-training.
- We confirm through experimentation on several benchmark datasets the effectiveness of the proposed approach.

Notably, our proposed approach can be combined with various GNN backbone architectures, as well as different AL and ST methodologies.

2 Related work

2.1 Graph Neural Networks

One of the most popular GNN approaches is the Graph Convolution Network (GCN) (Kipf and Welling 2017), which performs an iterative propagation through a message passing strategy. In GCN, the node representations are produced by aggregating features (messages) from their neighboring nodes. Other notable GNN architectures include the Graph Attention Network (GAT) (Veličković et al. 2017), that uses a weighted aggregation design and a trainable attention mechanism, GraphSAGE (Hamilton et al. 2017), introducing different aggregation functions, and FiLM (Brockschmidt 2020) which considers both the source and target nodes of each graph edge in the representation learning process. There are many more GNN models in the literature, and a more detailed overview can be found in recent surveys (Zhou et al. 2020, 2022).

2.2 Active learning

Active Learning (AL) (Aggarwal et al. 2014; Settles 2009) is a well-studied research area with applications in various domains, such as text mining (Schröder and Niekler 2020) and computer vision (Beluch et al. 2018). Several new approaches have been introduced in recent years, with the most successful ones utilizing selection criteria based on uncertainty (Yang et al. 2015; Zhu et al. 2008). A widely applied AL strategy is Query-by-Committee (QbC) (Settles 2009), which selects the most informative samples based on the votes of multiple models (i.e. a committee). Although, QbC has been shown to be beneficial, building multiple models can be prohibitive in terms of computational resources. For a more detailed discussion on AL, readers can consult corresponding surveys (Ren et al. 2021; Zhan et al. 2022).

Recently, there has been an increased interest in applying AL on graph-structured data. Early work proposed various selection criteria based on the structure of

the graph (Bilgic et al. 2010; Gu et al. 2013). Different work (Appice et al. 2018) introduced a new AL method for regression problems in graph data. More recently, hybrid approaches (Cai et al. 2017; Gao et al. 2018) proposed the linear combination of different heuristics, including information entropy, embedding representativeness and graph centrality, in order to select the most informative nodes to label. Furthermore, a policy network is proposed in Hu et al. (2020a), in order to sequentially select informative nodes using reinforcement learning. Differently from all these approaches, we enrich the AL process with a self-training strategy to further promote the use of unlabeled data during training.

2.3 Self-training GNNs

When the proportion of labeled nodes in a graph is small, the diffusion of supervision information by GNNs is limited. To address this limitation, several methods enhance GNNs through self-training, which extends the supervision by adding nodes that the current model can classify with high confidence, i.e. pseudo-labels (Dai et al. 2021; Wang et al. 2021; Li et al. 2018). These extra nodes may contain valuable local information which does not appear in the initial training set. As an example, the authors of Yang et al. (2021) proposed the Self-Enhanced GNN (SEG), which expands the labeled node set based on the predictions of the GNN, as trained until that point. Other work (Caron et al. 2018; Wang et al. 2019), employs also clustering to improve the quality of the pseudo-labels. Alternatively, the most confident samples can be selected (Li 2022) by constructing a new graph which consists of homogeneous and heterogeneous edges between labeled and unlabeled data.

A known issue of all of these methods is that the predicted pseudo-labels may introduce label noise, which biases the learning process. In Wang et al. (2021), the authors argue that the noise is due to the under-explored low-confidence samples and propose a weighted self-training strategy. Along the same lines, our approach utilizes low-confidence samples by combining self-training with active learning.

2.4 Hybrid methods

The idea of combining AL with other learning methods has been explored by some of the literature. Integrating AL with semi-supervised learning was proposed by Hao et al. (2020) and Xie et al. (2022) to effectively predict molecular properties and classify graphs, respectively. Different work (Yi et al. 2022) proposed an AL approach that utilizes self-supervised models on pretext tasks to achieve state-of-the-art performance on image classification and semantic segmentation. Focusing specifically on self-training, an AL strategy that incorporates pseudo-labels was proposed in Feng et al. (2021), demonstrating significant performance gains in the task of 3D pose estimation. The combination of AL and ST was also proposed by Chaplot et al. (2021), Kwak et al. (2022) and Yu et al. (2022) to improve image and text classification tasks, respectively.

More recently, in Zhu et al. (2020) AL was combined with contrastive learning (You et al. 2020). Initially, the original graph was augmented twice and the

agreement between the augmented embeddings was maximized. Then, the produced node embeddings were used in AL by selecting the nodes that have the most similar embeddings to their neighbors. To the best of our knowledge, this is the only existing work that combined AL with a form of self-supervision, i.e. contrastive learning, for the task of node classification. In contrast, our work focuses on how to integrate AL with pseudo-labels.

3 Preliminaries

3.1 Notation

We consider an undirected graph $G = (V, E, X)$, where $V = \{v_1, v_2, .., v_N\}$ is the set of nodes and $| V | = N$, E is the set of edges, where each $e_{ij} \in E$ denotes an edge between nodes v_i and v_j, and $X = \{x_1, x_2, .., x_N\}$ indicates the node features with $X \in \mathbb{R}^{N \times R}$, where R is the dimension of node features. Let us define $A = [A_{ij}] \in \mathbb{R}^{N \times N}$ as the adjacency matrix of G, where $A_{ij} = 1$ if e_{ij} exists and $A_{ij} = 0$ otherwise. We denote the degree of a node v as $d_v \in \mathbb{R}^+$ and D as the diagonal degree matrix of G, i.e. $D_{ii} = \sum_i A_{ij}$. Also, each node in V is associated with a true label y_i, and we use $\mathcal{Y} = \{y_1, y_2, ..., y_N\}$ to denote the true label vector. Let $V_L = \{v_1, v_2, .., v_L\}$ be a set of labeled nodes, then $V_U = V - V_L$ is the set of unlabeled nodes.

The learned representation matrix of G, at layer l of a GNN, is represented by $H^l = \{h_1^l, h_2^l, .., h_N^l\}$ and $H^l \in \mathbb{R}^{N \times R^l}$, where h_v^l is the representation vector of node v at layer l, and R^l is the dimension of the representation vector at layer l.

The layer-wise propagation operation of GCN (Kipf and Welling 2017) is modeled as follows:

$$H^{l+1} = \sigma(D^{\frac{-1}{2}} \hat{A} D^{\frac{-1}{2}} H^l W^l) \tag{1}$$

where $\hat{A} = A + I$, in order to add self-loops, W^l is the trainable weight matrix of layer l and σ is an activation function.

3.2 Problem formulation

Given a small number of labeled samples $\mathcal{X}_L = \{(v_i, x_i, y_i)\}_{i=1}^{L}$ and a large number of unlabeled samples $\mathcal{X}_U = \{(v_j, x_j)\}_{j=1}^{U}$, our goal is to train a Graph Neural Network $g(\mathcal{X}, A; \theta) : \mathcal{X} \mapsto \mathcal{Y}$, through an iterative process of k rounds that utilizes active learning and self-training at each round. In particular, we aim to select the T most uncertain, as well as the B most confidently classified samples from $\mathcal{X}_U$ to train a model $g(\hat{\mathcal{X}}, A; \theta)$ with $\hat{\mathcal{X}} = \mathcal{X}_L \cup \{(v_t, x_t, y_t)\}_{t=1}^{T} \cup \{(v_b, x_b, y_b)\}_{b=1}^{B}$.

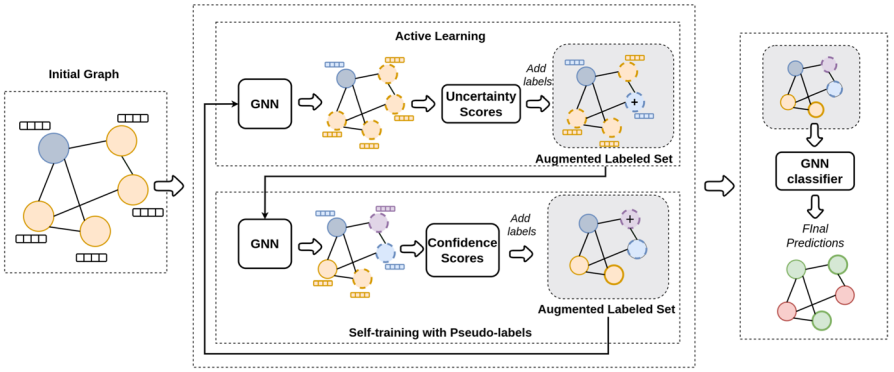

Fig. 1 Overview of the proposed approach for the node classification task. The STAL framework operates in a sequential order, considering the outputs of preceding models

4 Methodology

In Fig. 1, we illustrate the proposed approach for the task of node classification. Given a graph, we first train a GNN model with the initial labeled data. The learned node representations are then used to calculate the uncertainty scores for all unlabeled samples. Then, the labels of the most uncertain samples are requested and added to the initial labeled data. The updated graph is used to train another GNN which generates new class probabilities for all unlabeled samples. These predictions are utilized to identify high-quality pseudo-labels, which will be integrated into the label set. The new graph is fed back to the active learning process, and so on. At the end of the predefined number of iterations, the whole graph is labeled by the trained GNN. Each of two main steps is detailed in the following subsections.

4.1 Active learning

The main objective of AL is to select a subset of unlabeled instances, the labels of which can improve the model's performance. Typically, AL consists of k sampling rounds. At each round, T samples are selected from the unlabeled data, based on strategy ϕ. These samples, together with the other labeled instances, are used to re-train the model. In the context of this study, we investigate the following AL strategies:

Uncertainty A widely used strategy that selects the samples that are the most uncertain according to an information-based measure, such as entropy:

$$\phi_e(v_i) = \sum_c P(y_i^c \mid x_i;g) log P(y_i^c \mid x_i;g) \tag{2}$$

where $P(y_i^c \mid x_i;g)$ is the probability of v_i belonging to class c as predicted by the GNN model g.

Query-by-committee Instead of relying on the uncertainty sampling of a single model, QbC employs a committee of models $C = \{g_1, g_2, ..., g_C\}$. The samples causing the maximal disagreement among committee members are chosen:

$$\phi_{qbc}(v_i) = \sum_{j \neq r} \|\phi_e(v_i;g_j) - \phi_e(v_i;g_r)\| \tag{3}$$

where $\phi_e(v_i;g_j)$ is the entropy score of v_i based on the committee model g_j.

AGE A more recent strategy that incorporates three different query sub-strategies: it combines uncertainty with the density of the node and its centrality. Specifically, it computes the entropy of the predicted label distribution, measures the distance between a node and its cluster center, and calculates the PageRank centrality. These criteria are linearly combined as:

$$\phi_{age}(v_i) = \alpha * \phi_e(v_i) + \beta * \phi_d(v_i) + \gamma * \phi_{PR}(v_i) \tag{4}$$

where $\phi_d(v_i) = 1/(1 + \|h_i^l - CC_i\|)$, $\phi_{PR}(v_i)$ is the PageRank centrality of v_i, $\alpha + \beta + \gamma = 1$ and CC_i is the center of the cluster, in which v_i belongs, as defined in Cai et al. (2017).

Using these strategies, we select the most uncertain samples $\mathcal{X}_T \subset \mathcal{X}_U$ on each AL round as:

$$\mathcal{X}_T = \{(v_i, x_i, \phi_i)\}_{i=1}^{T}; \phi_1 > \phi_2 > ... > \phi_B\} \tag{5}$$

Note that although such uncertainty-based strategies ignore the use of graph-specific properties in selecting nodes for labeling, their performance is often problem-dependent and remain strong baselines, as noted by Cai et al. (2017) and Shui et al. (2020).

4.2 Self-training with confident nodes

In many real-life cases, the number of labeled samples $\mathcal{X}_L$ is relatively small, when compared to the number of unlabeled ones $\mathcal{X}_U$, i.e. $\mathcal{X}_U \gg \mathcal{X}_L$. Self-training addresses scarcity of labeled data by training a model on $\mathcal{X}_L$ and using it to predict high-confidence pseudo-labels for some unlabeled nodes in $\mathcal{X}_U$. These pseudo-labels are then used to augment the initial labeled set. The reliance of self-training on the generated pseudo-labels may introduce noise. Therefore, selecting reliable pseudo-labels is essential.

As shown in Fig. 1, our ST strategy begins with the completion of the AL step, at each iteration. Using the available labels, including the T samples labeled by AL, we train a new GNN model. This model produces a class probability vector for each node in $\mathcal{X}_U$:

$$p(y_i \| x_i, A; \theta) = g(\mathcal{X}_{\mathcal{L}}, A; \theta) \tag{6}$$

A common approach in selecting the pseudo-labels is to keep only the most confident of them, based on the probabilities p. Specifically, a set $\mathcal{X}_B \subset \mathcal{X}_U$ of unlabeled nodes with estimated probabilities higher than a threshold τ_u, i.e. $max(p_i) > \tau_u$.

We argue for the use of classification confidence, instead of the simple probability, in selecting reliable pseudo-labels. In particular, we measure classification confidence, as the difference between the probabilities of the most-probable classes. To this end, we calculate the Euclidean distance between these probabilities, i.e. $r_i = \sqrt{\left(p_i^1 - p_i^2\right)^2}$, where p^1 and p^2 indicate the highest and second highest values of probabilities p respectively, and use r_i as the confidence score for node v_i. Intuitively, the larger the distance, the more separable are the classes, and thus, the more confident we are that the pseudo-labels are accurate. To obtain the most reliable pseudo-labels, we select the top-B such that:

$$\mathcal{X}_B = \{(v_i, x_i, r_i)\}_{i=1}^{B}; r_1 > r_2 > ... > r_B\} \tag{7}$$

Note that the selected pseudo-labeled nodes are not considered labeled in subsequent AL iterations; i.e. pseudo-labeled nodes may be selected for labelling by AL.

4.3 Label prediction

With the help of the AL and ST strategies, we obtain the new augmented labeled set $\hat{\mathcal{X}}$. Using these data, we train a GNN model to produce the new class probabilities. Finally, the model decides on the labels $\mathcal{Y}$ of the nodes as:

$$\mathcal{Y} = argmax(g(\hat{\mathcal{X}}, A; \hat{\theta})) \tag{8}$$

An overview of the STAL framework is given in Algorithm 1.

Algorithm 1 Framework of STAL

Require: $G(V, E)$, A, $\mathcal{X}_L$, k, T, B
Ensure: label predictions $\mathcal{Y}$
  Obtain initial model $g(\mathcal{X}_L, A;\ \theta)$
  for $q = 1, 2, .., k$ **do**
    procedure AL SELECTION
      Select the top-T uncertain nodes $\mathcal{X}_T$ using eq. 5.
      Augment label set $\mathcal{X}_L = \mathcal{X}_L \cup \mathcal{X}_T$.
      Obtain model $g(\mathcal{X}_L, A;\ \theta)$.
    end procedure
    procedure ST SELECTION
      Select the top-B confident nodes $\mathcal{X}_B$ using eq. 7.
      Augment label set $\hat{\mathcal{X}} = \mathcal{X}_L \cup \mathcal{X}_T \cup \mathcal{X}_B$.
      Obtain model $g(\hat{\mathcal{X}}, A;\ \hat{\theta})$.
    end procedure
  end for
  The final predictions: $\mathcal{Y} = argmax(g(\hat{\mathcal{X}}, A;\ \hat{\theta}))$

Table 1 The details of the datasets

Dataset	$\mid V \mid$	$\mid E \mid$	$\mid c \mid$	R
Cora	2708	10,556	7	1433
Citeseer	3327	9104	6	3703
Pubmed	19,717	88,648	3	500
ogbn-arxiv	169,343	1,166,243	40	128

$\mid c \mid$ and R and denote the number of classes and the number of features respectively

Table 2 The search space of hyper-parameters for our experiments

Hyper-parameter	Values
Learning rate	{1e−2, 1e−3, 1e−4}
Dropout	{0, 0.3, 0.5, 0.8}
Layers	{2, 4, 6}
Heads	{2, 4}
Hidden dimension	{16, 32, 64, 128}

5 Experiments

In this section, we evaluate the proposed approach on the node classification task using four benchmark datasets. Furthermore, we conduct ablation experiments to gain insights on the various components of the method.

5.1 Evaluation protocol

We conducted experiments on four benchmark datasets. In particular, we used the public citation networks - Cora, Citeseer and Pubmed (Yang et al. 2016). We also used one larger dataset from the OGB archive (Hu et al. 2020b); namely the ogbn-arxiv. The statistics of the datasets are presented in Table 1.

Regarding the evaluation methodology, we followed common practice in the node classification literature (Cai et al. 2017; Zhu et al. 2020). The nodes of each dataset were partitioned randomly into five folds. For each fold, we further randomly sampled 20 and 30 nodes per class as the training and validation set, respectively. For the assessment of AL, we start with 5 training samples per class and increase the size up to 20. Thus, we perform $k = (20 - 5) \cdot |c|$ rounds, and in each round a single unlabeled sample is selected to be added to the training set. The rest of the nodes are used as the test set. For each fold, we kept the model that performed best on the validation set and evaluated it on the held-out test set. Moreover, we set the number of pseudo-labels introduced by ST at each

Table 3 The results of the STAL variants

Method	Cora	Citeseer	Pubmed	ogbn-arxiv
GCN	77.7±2.2	63.3±3.1	75.0±2.8	61.6±0.6
ST	77.9±2.5	63.5±5.2	74.9±3.1	61.7±1.2
AL_{ϵ}	77.8±3.6	64.0±2.3	74.8±2.9	61.9±1.5
AL_{QbC}	78.1±1.2	64.2±1.6	74.9±2.4	62.1±0.8
AL_{AGE}	79.3±1.5	65.5±2.4	78.1±1.1	61.9±0.3
$STAL_{\epsilon}$	78.9±2.4	67.9±2.8	76.0±2.6	62.5±0.5
$STAL_{QbC}$	78.9±1.3	**68.1±1.6**	76.2±1.8	**62.9±0.02**
$STAL_{AGE}$	**80.8±1.9**	65.6±1	**79.7±1.2**	62.6±0.07
$STAL_{rev}$	78.7±2.1	66.8±3.5	76.0±2.8	62.4±0.7

The scores denote the average accuracy over five random train-test folds. Bold denotes the overall best performance per dataset

iteration to $B = \frac{|\mathcal{V}_{\mathcal{U}}|}{2}$, where $|\mathcal{V}_{\mathcal{U}}|$ is the number of unlabeled nodes, for all datasets. Eventually, we calculated the average accuracy over the five random data folds.

Regarding the hyper-parameters of the methods included in the experiments, we performed grid search in the space presented in Table 2. The Adam optimizer (Kingma and Ba 2015) was used to minimize the cross-entropy losses with weight decay 5e−4 and epsilon value 1e−8 for all models. We used the GNN implementations of PyTorch Geometric (Fey and Lenssen 2019) and trained all models for the same number of epochs,[1] in full-batch mode. Furthermore, we used the implementations of the OGB library[2] for the node classification experimental set-up. We conducted our experiments using one Nvidia RTX A6000 GPU on an AMD Ryzen Threadripper PRO 3955WX CPU. Our code is available at https://github.com/nneinn/STAL.

5.2 Ablation study

In order to assess the importance of different features of the proposed method, we ran a set of experiments, using a GCN as the backbone model.

5.2.1 Variants of STAL

In particular, we evaluated the following baselines and variants of STAL:

- GCN: A vanilla GCN trained with the full set of labels.

[1] We trained our models for 200 epochs for Cora, Citeseer and Pubmed, and 300 epochs for the ogbn-arxiv.

[2] https://github.com/snap-stanford/ogb

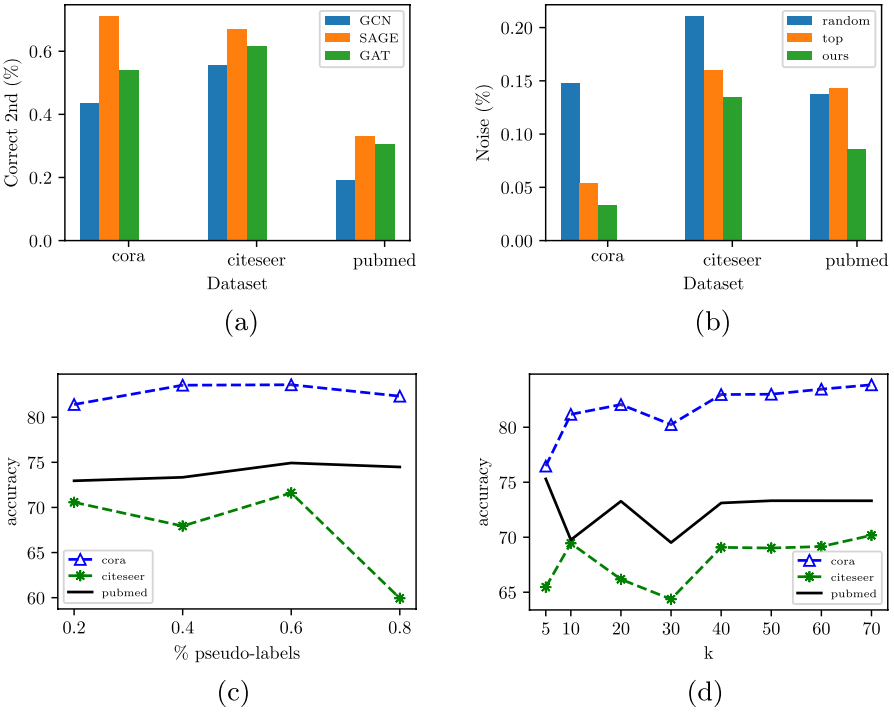

Fig. 2 **a** The percentage of incorrect predictions where the second most probable class was the correct one. **b** The ratio of incorrect pseudo-labels by different selection strategies. Lower values indicate more accurate labelling. **c** The performance of STAL on node classification, when varying the number of pseudo-labels. **d** The performance of STAL with varied number k of AL rounds

- ST: A GCN model that utilizes the self-training strategy only.
- AL_{ϵ}: An AL model that utilizes the uncertainty-based entropy strategy.
- AL_{QbC}: An AL model that utilizes the QbC entropy-based strategy, with a committee of five models.
- AL_{AGE}: An AL model that utilizes the AGE strategy only.
- STAL_{ϵ}: STAL with the entropy strategy.
- STAL_{QbC}: STAL with the QbC entropy-based strategy, using a committee of five models.
- STAL_{AGE}: STAL with the AGE strategy.
- STAL_{rev}: A model similar to STAL_{ϵ} except it applies the self-training strategy before the active learning selection.

The results for all these variants are shown in Table 3. The first observation is that active learning improves the vanilla GCN model. Secondly, the incorporation of self-training in STAL seems to improve the results further, with STAL_{AGE} and STAL_{QbC} achieving the best scores. This result confirms the

value of pseudo-labels, when combined with AL. It is worth mentioning that ST, without the use of AL, did not yield significant improvements. Moreover, our results show that applying ST at the end of each AL round ($STAL_{\epsilon}$) instead of the beginning ($STAL_{rev}$), slightly improves the performance, and therefore, we use the former strategy in the rest of the experiments.

Despite the higher accuracy of QbC in many cases, it can be computationally prohibitive, since it requires retraining multiple models at each iteration of the AL process. Besides, compared to the other strategies, the improvement in accuracy over the simpler AL_{ϵ} approach is not significant. Therefore, we have excluded QbC from the rest of the experiments.

5.2.2 Quality and size of pseudo-labels

As already discussed in Sect. 2.3, the quality of the pseudo-labels can play a significant role in the downstream tasks. To develop a reliable selection strategy for pseudo-labels we investigated the results of GNN models, focusing on their mistakes. Hence, we have observed that GNNs tend to generate incorrect predictions when the top-two class probabilities p_i^1, p_i^2 are close. As shown in Fig. 2a, in the majority of these cases the correct label corresponds to p_i^2. This observation, motivated the design of our ST selection strategy, as presented in Sect. 2.3.

In Fig. 2b, we compare this selection strategy against two baselines: (i) random selection of pseudo-labels, and (ii) selection of the top (most confident) pseudo-labels. Specifically, for each strategy we report the ratio of incorrect pseudo-labels over all pseudo-labels predicted. The proposed pseudo-label selection strategy seems to reduce significantly the errors made by other methods, leading to more robust self-training.

Additionally, we perform a series of experiments to examine how the number of pseudo-labels selected to be added to the labelled set affect the performance of node classification. In particular, we vary the number of pseudo-labels between 20% and 80% of all unlabelled nodes and report the average accuracy over five random runs. The results in Fig. 2c indicate that STAL handles well the incorporation of pseudo-labels, independent of the number of pseudo-labels. Therefore, in the rest of the experiments we opt for a conservative approach, using a small number of pseudo-labels.

5.2.3 Number of active learning rounds *k*

Finally, we study the role of the number of rounds k used in the AL process. It is worth noting that we have set k to be inversely related to the number of selected

Table 4 Node classification results of all models

GNN	Strategy	Cora	Citeseer	PubMed	ogbn-arxiv
GCN	Vanilla	77.7±2.2	63.3±3.1	75.0±2.8	61.6±0.6
	$STAL_{\epsilon}$	78.9±2.4	**67.9 ± 2.8**	76.0±2.6	62.5±0.5
	$STAL_{AGE}$	80.8 ± 1.9	65.6±1	**79.7 ± 1.2**	62.6 ± 0.07
SAGE	Vanilla	72.3±1.8	59.3±1.6	71.3±2.2	59.0±0.4
	$STAL_{\epsilon}$	**81.8 ± 0.8**	66.5±2.2	74.3±1.6	61.6±0.7
	$STAL_{AGE}$	78.2±2.2	67.0 ± 1.9	74.5 ± 2.1	61.8 ± 0.08
GAT	Vanilla	77.4±1.6	60.1±2.6	76.0±2.7	63.1±0.9
	$STAL_{\epsilon}$	81.2 ± 2.5	67.46 ± 1.3	78.4±2.5	63.3±0.3
	$STAL_{AGE}$	80.2±1.3	65.4±1.1	78.8 ± 0.6	**63.7 ± 0.04**

The reported accuracy is the averaged over five random folds. Bold denotes the overall best performance per dataset, whereas the underlined figures the best performance per method

samples T at each AL round, i.e. $T = \frac{(20-5)\cdot|c|}{k}$. Thus, T decreases as k increases and vice versa. When k is maximum ($k = (20-5)\cdot|c|$) T is minimum ($T = 1$). Eventually, the total number of labeled samples remains the same, independent of the number of rounds k. As shown in Fig. 2d, STAL seems to benefit somewhat by large values of k, as was expected, but this comes at a higher manual cost. Therefore, for the rest of the experiments we use $k = (20-5)\cdot|c|$.

5.3 Results for different models

The proposed approach (STAL) can be used with various GNNs and different AL and ST strategies. To demonstrate this flexibility, we conducted experiments with three GNN models: GCN (Kipf and Welling 2017), SAGE (Hamilton et al. 2017) and GAT (Veličković et al. 2017).

The performance of all models, with and without STAL, is shown in Table 4. We observe that STAL improves node classification accuracy by 1–9.5 percentage points on the four datasets. Interestingly, the simple uncertainty-based strategy achieves the best overall score in two cases, although $STAL_{AGE}$ performs slightly better overall.

Figure 3 presents also the performance of the models with varying number of training labels. As shown in the figure, STAL requires considerably fewer labels to reach the performance of the baseline models. In most cases, STAL seems to need just 30–60% of the number of labeled samples required by the baselines, leading to benefits in the labeling effort and enhancing the models' performance.

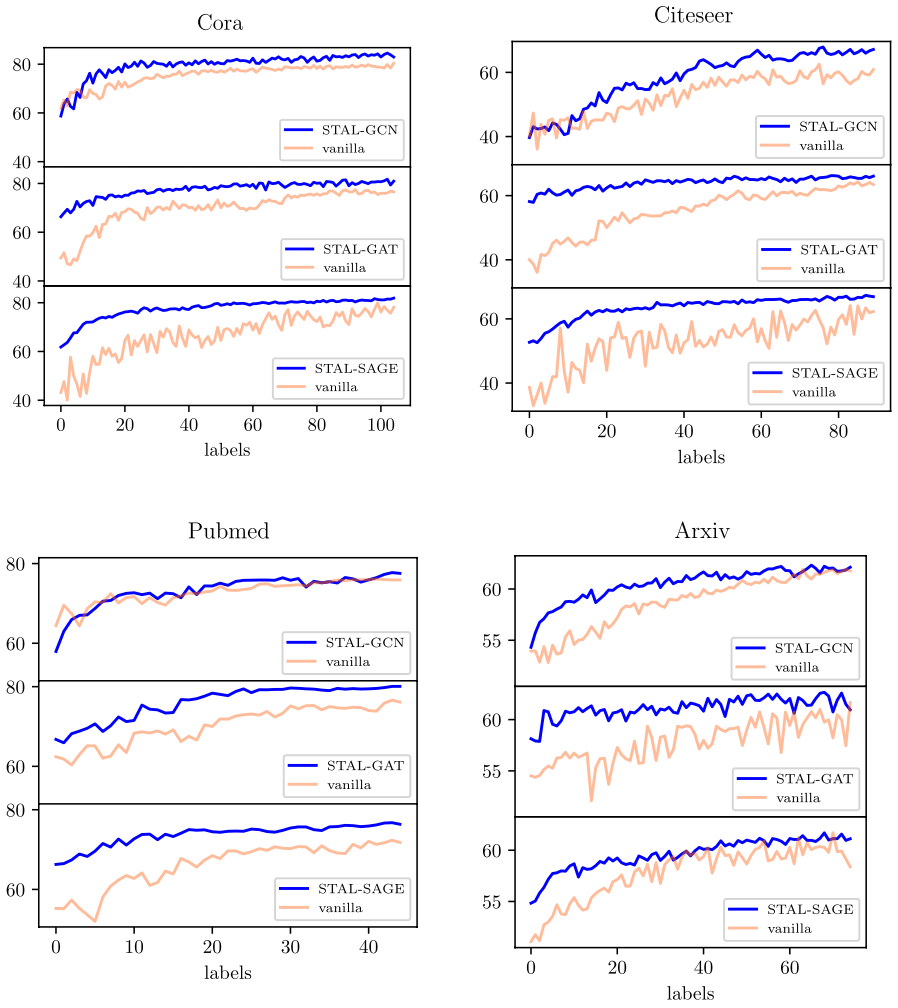

Fig. 3 The performance of 3 different GNNs, with and without using STAL_ϵ, with varying number of labels

6 Conclusion and future work

In this paper, we have proposed STAL, a new approach that combines active learning (AL) with self-training (ST) to improve both label efficiency and performance of GNNs. AL is used to select highly uncertain nodes from a large unlabeled set. In combination with AL, we proposed a simple but efficient ST strategy to identify accurate pseudo-labels. Finally, we incorporated these techniques into a common framework that can be easily used with various AL strategies and GNN backbones. The experimental results verified the effectiveness of our approach in node classification, as well as the contribution of each feature of

the proposed method to the final result. Besides, the experiments, demonstrated the ability of STAL to reduce the labeling cost.

Still, a number of issues remain open to investigate in the future. In this paper we demonstrate the effectiveness of our method on various node classification datasets, acknowledging that there are other tasks, such as link prediction and graph classification, where our approach has not been tested yet. Therefore, further analysis is needed to validate whether the proposed approach can produce similar performance when applied to other downstream tasks. Moreover, in our experiments, we mainly focus on investigating how the combination of AL with ST can produce more accurate models. This improvement in accuracy comes at a higher computational cost due to multiple training rounds that are required by both AL and ST. Accordingly, STAL assumes a sequential architecture where models are trained separately and in a specific order. In the future, we would like to assess a joint methodology that would integrate all components into a single model and reduce its computational cost.

Acknowledgments This works was partially supported by the EU project CREXDATA (Critical Action Planning over Extreme-Scale Data), grant agreement ID: 101092749.

Funding Open access funding provided by HEAL-Link Greece.

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Aggarwal CC, Kong X, Gu Q, Han J, Philip SY (2014) Active learning: a survey. In: Data Classification, pp 599–634

Appice A, Loglisci C, Malerba D (2018) Active learning via collective inference in network regression problems. Inf Sci 460–461:293–317. https://doi.org/10.1016/j.ins.2018.05.028

Beluch WH, Genewein T, Nürnberger A, Köhler JM (2018) The power of ensembles for active learning in image classification. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9368–9377

Bilgic M, Mihalkova L, Getoor L (2010) Active learning for networked data. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 79–86

Brockschmidt M (2020) GNN-film: Graph Neural Networks with feature-wise linear modulation. In: International conference on machine learning, PMLR, pp 1144–1152

Cai H, Zheng VW, Chang KC-C (2017) Active learning for graph embedding. Preprint arXiv:1705.05085
Caron M, Bojanowski P, Joulin A, Douze M (2018) Deep clustering for unsupervised learning of visual features. In: Proceedings of the European conference on computer vision (ECCV), pp 132–149
Chan Y-C, Li M, Oymak S (2021) On the marginal benefit of active learning: Does self-supervision eat its cake? In: ICASSP 2021-2021 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE, pp 3455–3459
Chaplot DS, Dalal M, Gupta S, Malik J, Salakhutdinov RR (2021) Seal: self-supervised embodied active learning using exploration and 3d consistency. Adv Neural Inf Process Syst 34:13086–13098
Dai E, Aggarwal C, Wang S (2021) NRGNN: learning a label noise resistant Graph Neural Network on sparsely and noisily labeled graphs. In: Zhu F, Ooi BC, Miao C (eds) KDD '21: the 27th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, Singapore, pp 227–236. https://doi.org/10.1145/3447548.3467364
Fazakis N, Kanas VG, Aridas CK, Karlos S, Kotsiantis S (2019) Combination of active learning and semi-supervised learning under a self-training scheme. Entropy 21(10):988
Feng Q, He K, Wen H, Keskin C, Ye Y (2021) Active learning with pseudo-labels for multi-view 3d pose estimation. Preprint arXiv:2112.13709
Fey M, Lenssen JE (2019) Fast graph representation learning with PyTorch Geometric. In: ICLR workshop on representation learning on graphs and manifolds
Gao L, Yang H, Zhou C, Wu J, Pan S, Hu Y (2018) Active discriminative network representation learning. In: IJCAI international joint conference on artificial intelligence
Gu Q, Aggarwal C, Liu J, Han J (2013) Selective sampling on graphs for classification. In: Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining, pp 131–139
Hamilton WL, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs. In: Guyon I, von Luxburg U, Bengio S, Wallach HM, Fergus R, Vishwanathan SVN, Garnett R (eds) NIPS, pp 1024–1034. http://dblp.uni-trier.de/db/conf/nips/nips2017.htmlHamiltonYL17
Hao Z, Lu C, Huang Z, Wang H, Hu Z, Liu Q, Chen E, Lee C (2020) Asgn: an active semi-supervised Graph Neural Network for molecular property prediction. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining, pp 731–752
Hu S, Xiong Z, Qu M, Yuan X, Côté M-A, Liu Z, Tang J (2020) Graph policy network for transferable active learning on graphs. Adv Neural Inf Process Syst 33:10174–10185
Hu W, Fey M, Zitnik M, Dong Y, Ren H, Liu B, Catasta M, Leskovec J (2020) Open graph benchmark: datasets for machine learning on graphs. Preprint arXiv:2005.00687
Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: 3rd international conference on learning representations, ICLR 2015—conference track proceedings
Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: 5th international conference on learning representations, ICLR 2017, Toulon, Conference Track Proceedings
Kwak B-w, Kim Y, Kim YJ, Hwang S-w, Yeo J (2022) Trustal: Trustworthy active learning using knowledge distillation. In: Proceedings of the AAAI conference on artificial intelligence, vol 36, pp 7263–7271
Li J (2022) Nang-st: a natural neighborhood graph-based self-training method for semi-supervised classification. Neurocomputing 514:268–284. https://doi.org/10.1016/j.neucom.2022.08.010
Li Q, Han Z, Wu X (2018) Deeper insights into graph convolutional networks for semi-supervised learning. In: McIlraith SA, Weinberger KQ (eds) Proceedings of the thirty-second AAAI conference on artificial intelligence (AAAI-18), the 30th innovative applications of artificial intelligence (IAAI-18), and the 8th AAAI symposium on educational advances in artificial intelligence (EAAI-18), New Orleans, pp 3538–3545. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16098
Ren P, Xiao Y, Chang X, Huang P-Y, Li Z, Gupta BB, Chen X, Wang X (2021) A survey of deep active learning. ACM Comput Surv (CSUR) 54(9):1–40
Schröder C, Niekler A (2020) A survey of active learning for text classification using deep neural networks. Preprint arXiv:2008.07267
Settles B (2009) Active learning literature survey
Settles B, Craven M (2008) An analysis of active learning strategies for sequence labeling tasks. In: Proceedings of the 2008 conference on empirical methods in natural language processing, pp 1070–1079

Shui C, Zhou F, Gagné C, Wang B (2020) Deep active learning: unified and principled method for query and training. In: International conference on artificial intelligence and statistics, PMLR, pp 1308–1318

Sun K, Lin Z, Zhu Z (2020) Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In: The thirty-Fourth AAAI conference on artificial intelligence, AAAI 2020, The thirty-second innovative applications of artificial intelligence conference, IAAI 2020, The tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, pp 5892–5899. https://ojs.aaai.org/index.php/AAAI/article/view/6048

Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2017) Graph attention networks. In: ICLR 2018, Preprint arXiv:1710.10903

Wang C, Pan S, Hu R, Long G, Jiang J, Zhang C (2019) Attributed graph clustering: a deep attentional embedding approach. Preprint arXiv:1906.06532

Wang X, Liu H, Shi C, Yang C (2021) Be confident! Towards trustworthy graph neural networks via confidence calibration. In: Ranzato M, Beygelzimer A, Dauphin YN, Liang P, Vaughan JW (eds) Advances in neural information processing systems 34: annual conference on neural information processing systems 2021, NeurIPS 2021, Virtual, pp 23768–23779

Wu F, Jr, AHS, Zhang T, Fifty C, Yu T, Weinberger KQ (2019) Simplifying graph convolutional networks. In: Chaudhuri K, Salakhutdinov R (eds) Proceedings of the 36th international conference on machine learning, ICML 2019, Long Beach, Proceedings of machine learning research, vol 97, pp 6861–6871. http://proceedings.mlr.press/v97/wu19e.html

Xie Y, Lv S, Qian Y, Wen C, Liang J (2022) Active and semi-supervised graph neural networks for graph classification. IEEE Trans Big Data 8(4):920–932. https://doi.org/10.1109/TBDATA.2021.3140205

Xu K, Hu W, Leskovec J, Jegelka S (2019) How powerful are graph neural networks? In: 7th international conference on learning representations, ICLR 2019, New Orleans. https://openreview.net/forum?id=ryGs6iA5Km

Yang Y, Ma Z, Nie F, Chang X, Hauptmann AG (2015) Multi-class active learning by uncertainty sampling with diversity maximization. Int J Comput Vis 113:113–127

Yang Z, Cohen WW, Salakhutdinov R (2016) Revisiting semi-supervised learning with graph embeddings. In: Balcan M, Weinberger KQ (eds) Proceedings of the 33nd international conference on machine learning, ICML 2016, New York City, JMLR Workshop and conference proceedings, vol 48, pp 40–48. http://proceedings.mlr.press/v48/yanga16.html

Yang H, Yan X, Dai X, Chen Y, Cheng J (2021) Self-enhanced GNN: improving Graph Neural Networks using model outputs. In: International joint conference on neural networks, IJCNN 2021, Shenzhen, IEEE, pp 1–8. https://doi.org/10.1109/IJCNN52387.2021.9533748

Yi JSK, Seo M, Park J, Choi D-G (2022) Pt4al: Using self-supervised pretext tasks for active learning. In: Computer vision–ECCV 2022: 17th European conference, Tel Aviv, Proceedings, Part XXVI, Springer, pp 596–612

You Y, Chen T, Sui Y, Chen T, Wang Z, Shen Y (2020) Graph contrastive learning with augmentations. Adv Neural Inf Process Syst 33:5812–5823

Yu Y, Kong L, Zhang J, Zhang R, Zhang C (2022) Actune: uncertainty-based active self-training for active fine-tuning of pretrained language models. In: Proceedings of the 2022 conference of the North American chapter of the association for computational linguistics: human language technologies, pp 1422–1436

Zhan X, Wang Q, Huang K-h, Xiong H, Dou D, Chan AB (2022) A comparative survey of deep active learning. Preprint arXiv:2203.13450

Zhang M, Chen Y (2018) Link prediction based on Graph Neural Networks. Adv Neural Inf Process Syst 31

Zhou J, Cui G, Hu S, Zhang Z, Yang C, Liu Z, Wang L, Li C, Sun M (2020) Graph Neural Networks: a review of methods and applications. AI Open 1:57–81. https://doi.org/10.1016/j.aiopen.2021.01.001

Zhou Y, Zheng H, Huang X, Hao S, Li D, Zhao J (2022) Graph Neural Networks: taxonomy, advances, and trends. ACM Trans Intell Syst Technol 13(1):15–11554. https://doi.org/10.1145/3495161

Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. In: Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool Publishers. https://doi.org/10.2200/S00196ED1V01Y200906AIM006

Zhu J, Wang H, Yao T, Tsou BK (2008) Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In: Proceedings of the 22nd international conference on computational linguistics (Coling 2008), pp 1137–1144

Zhu Y, Xu W, Liu Q, Wu S (2020) When contrastive learning meets active learning: a novel graph active learning paradigm with self-supervision. Preprint arXiv:2010.16091