CSE 584, Machine Learning
The Pennsylvania State University
Fall 2024

Homework #1

Jeongwon Bae (945397461)

[Paper 1] Cost-Effective Active Learning for Deep Image Classification
(IEEE Transactions on Circuits and Systems for Video Technology, 2017)

The main problem this paper addresses is how to train deep convolutional neural networks (CNNs) for image classification efficiently when there is limited labeled data. This issue arises due to several related factors, such as the shortage of labeled data, the high expense of manual labeling, the limitations of traditional active learning when applied to deep learning, inconsistencies between active learning assumptions and CNN training processes, and the lack of effective use of unlabeled data in existing methods. Especially, traditional active learning methods face challenges when used with deep learning models because they generally select only a small number of informative samples, which is not enough for fine-tuning CNNs or building strong feature representations. Additionally, theres a significant disconnect between the fixed feature representation assumed by many active learning methods and CNN training, which simultaneously optimizes feature learning and classifier training.

The Cost-Effective Active Learning (CEAL) framework [1] addresses these issues with a novel approach that merges active learning concepts with deep learning techniques. The core of CEAL lies in its dual sample selection strategy, which balances selecting informative uncertain samples with leveraging high-confidence samples for more effective learning. First, the CEAL framework initializes the CNN with a small set of labeled samples. The framework identifies a small number of the most uncertain samples from the unlabeled dataset using common active learning criteria such as least confidence, margin sampling, or entropy. The selection criteria are based on $p(y_i = j|x_i; \mathcal{W})$ which denotes the probability of x_i belonging to the jth class, where $\mathcal{W}$ are CNN parameters.

(1) *Least confidence:* Rank the unlabeled samples in ascending order based on their least confidence score, denoted as lc_i. This is calculated by:

$$lc_i = \max_j p(y_i = j|x_i; \mathcal{W}). \tag{1}$$

A low probability for the most likely class indicates that the classifier has high uncertainty about the sample.

(2) *Margin sampling:* Rank the unlabeled samples in ascending order based on their margin score, ms_i. This is defined as:

$$ms_i = p(y_i = j_1|x_i; \mathcal{W}) - p(y_i = j_2|x_i; \mathcal{W}), \tag{2}$$

where j_1 and j_2 are the labels of the two most probable classes predicted by the classifier. A smaller margin indicates greater uncertainty, suggesting the classifier is less certain about the samples correct classification.

(3) *Entropy:* Rank the unlabeled samples in descending order according to their entropy value, en_i, which is computed as:

$$en_i = -\sum_{j=1}^{m} p(y_i = j|x_i; \mathcal{W}) \log p(y_i = j|x_i; \mathcal{W}). \tag{3}$$

This method considers the probabilities of all class labels and uses entropy to assess uncertainty, with higher entropy values reflecting greater uncertainty about the sample.

These samples are then manually labeled and added to the training set. Although few in number, these samples have a significant impact on improving the decision boundary of the classifier. Simultaneously, CEAL identifies a large number of samples that the current model can classify with high-confidence. These samples are automatically assigned pseudo-labels without human intervention. The pseudo-label y_i is defined as:

$$j^* = \arg\max_j p(y_i = j|x_i; \mathcal{W}), \quad y_i = \begin{cases} j^*, & \text{if } en_i < \delta, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

where $y_i = 1$ denotes that x_i is regarded as a high-confidence sample and δ is a threshold that has to be set to a large value to guarantee the high reliability of assigning a pseudo-label. This process serves as a form of data augmentation, providing additional training data to improve feature learning. Next, the CNN parameters $\mathcal{W}$ are fine-tuned using standard back propagation with two types of samples generated by the active user. Afterward, high-confidence samples have their labels removed and are reassigned to the pool of unlabeled samples. To ensure the reliability of high-confidence sample selection, the threshold is updated at the end of each iteration t as follows:

$$\delta = \begin{cases} \delta_0, & t = 0, \\ \delta - dr * t, & t > 0, \end{cases} \tag{5}$$

where δ_0 is the initial threshold, dr controls the threshold decay rate. In conclusion, the CEAL framework can be summarized by Algorithm 1.

Algorithm 1 CEAL Learning Algorithm

Input: Unlabeled sample set D^U, initial labeled sample set D^L, selection size for uncertain samples K, confidence threshold δ, decay rate for the threshold dr, maximum number of iterations T, and fine-tuning interval t

Output: CNN parameters $\mathcal{W}$

```
Initialize the CNN parameters W using the labeled data D^L
while the maximum iteration count T has not been reached do
    Select and add K uncertain samples to D^L using Eq. (1), (2), or (3)
    Identify high-confidence samples D^H using Eq. (4)
    if iteration t is reached then
        Fine-tune the CNN parameters W with both D^H and D^L
        Update the confidence threshold δ using the decay rate dr using Eq. (5)
    end if
end while
return W
```

CEAL is one of the first approaches to successfully integrate active learning principles with deep convolutional neural networks for image classification tasks, addressing the longstanding challenge of applying active learning to deep learning models. The framework employs a novel dual approach to sample selection, combining uncertain and high-confidence samples to improve decision boundaries and enhance feature learning. By automatically pseudo-labeling high-confidence samples, CEAL reduces manual annotation needs, making deep learning more accessible for data-scarce applications. It uses an adaptive thresholding technique to gradually include more samples in the pseudo-labeling process as performance improves. CEAL introduces a progressive learning process that continuously improves model performance with minimal human intervention, suited for scenarios with evolving unlabeled data. The paper provides extensive experimental results on challenging datasets (CACD [2] and Caltech-256 [3]), demonstrating the effectiveness of CEAL in both improving classification accuracy and reducing the need for manual annotation. These advancements offer a practical solution to data scarcity and labeling costs, potentially expanding deep learning applications to previously prohibitive domains.

The CEAL framework, while innovative, has potential limitations. Its effectiveness depends heavily on the initial model quality, risking error propagation through incorrect pseudo-labels if the initial model is poor. The approach may reinforce model biases by relying on high-confidence predictions, potentially overlooking less common but important patterns in the data. Despite aiming for efficiency, the iterative nature of CEAL, involving repeated model fine-tuning, could still incur significant computational costs, especially for large datasets or complex CNN architectures. Performance may be sensitive to hyperparameter choices, such as the initial threshold for high-confidence sample selection and its adjustment rate, which could be challenging to optimize across different datasets. The paper lacks strong theoretical guarantees on convergence or optimality, relying primarily on empirical results. Scalability to very large datasets with millions of images or thousands of classes remains unclear, as the paper demonstrates performance on moderately sized datasets. The framework might struggle with out-of-distribution samples or classes significantly different from the initial labeled set. By focusing on high-confidence samples for pseudo-labeling, CEAL may not fully explore the feature space, potentially missing important decision boundaries in low confidence regions. Its applicability to domains beyond image classification, such as natural language processing or speech recognition, is uncertain.

Despite these potential limitations, the CEAL framework represents a significant step forward in addressing the challenges of deep learning with limited labeled data. It offers a novel and promising approach to cost-effective deep image classification, effectively addressing the challenge of limited labeled data. By integrating active learning principles with deep learning techniques and introducing innovative strategies like complementary sample selection and adaptive pseudo-labeling, CEAL pushes the boundaries of what's possible in scenarios with scarce labeled data. While there are certainly areas for improvement and further research, the framework's demonstrated performance and potential for real-world impact make it a valuable contribution to the field of machine learning and computer vision.

Paper 2) Cold-start Active Learning through Self-supervised Language Modeling
(Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing)

This paper addresses a critical challenge in natural language processing (NLP) by developing efficient active learning methods for deep learning models, particularly in low-resource scenarios. While modern transformer-based models have revolutionized NLP, they require large amounts of labeled data to perform well, which becomes a bottleneck in domains where data annotation is expensive, time-consuming, or restricted by privacy concerns. Traditional active learning methods aim to select the most informative examples for labeling but often struggle with deep learning models due to poorly calibrated confidence scores and the high computational cost of retraining after each query. The authors also highlight limitations in the existing strategy, BADGE [4], which combines uncertainty and diversity sampling. Despite improvements over older methods, BADGE suffers from computational inefficiency and relies on model warm-starting, with its high-dimensional gradient embeddings making distance comparisons both less meaningful and expensive. Additionally, BADGE and other warm-start approaches depend on unreliable confidence scores and labels when trained on limited data. These drawbacks motivate the authors to propose a new method that addresses the cold-start problem while more efficiently balancing uncertainty and diversity.

To tackle these challenges, the authors propose a novel approach [5], which is Active Learning by Processing Surprisal (ALPS). ALPS solves the problem by leveraging the pre-trained knowledge encoded in large language models like BERT [6] to guide the selection of examples for labeling. The paper sets the stage by introducing key concepts and notations used throughout the study. It defines the pre-trained encoder, typically BERT, which maps input sequences to hidden representations. The fine-tuned model, built upon this encoder, is then described for the downstream classification task. The authors also outline the active learning setup for sentence classification, detailing how the algorithm iteratively selects batches of sentences from an unlabeled pool for labeling.

ALPS operates by computing "surprisal embeddings" for each sentence in the unlabeled data pool. This process involves passing the unmasked sentence through BERT's Masked Language Model (MLM) head and calculating the cross-entropy loss for a randomly selected 15% of tokens against their true labels. The authors provide a theoretical foundation for this approach by defining the language model probability p_m as $p_m(w_i|x) = m(x;\phi)_{i,j}$, where x is the entire input context. They argue that this definition can be extended to unmasked inputs, based on BERT's pre-training procedure.

The information-theoretic surprisal is defined as $I(w) = -\log_p(w|c)$, which is the negative log-likelihood of word w given context c. Each non-zero entry in the surprisal embedding is shown to estimate the surprisal of its corresponding token within the sentence context, capturing the model's surprise at encountering certain words and providing a measure of how informative or unusual a sentence is from the pre-trained model's perspective. These surprisal embeddings undergo L2-normalization to enhance clustering performance.

ALPS then applies k-means clustering to these normalized embeddings and selects sentences closest to each cluster center for labeling. This method strikes a balance between choosing surprising (potentially informative) examples and maintaining a diverse sample set. Interestingly, while BADGE employs k-means++ for clustering, experiments reveal that standard k-means performs better with surprisal embeddings in the ALPS framework.

ALPS introduces a truly cold-start active learning method that does not require any task-specific labeled data to begin the selection process. The experimental results demonstrate its effectiveness across multiple datasets. For instance, on the AG News dataset, ALPS achieves an F1 score of

about 0.8 with only 400 labeled examples, while random sampling requires about 800 examples to reach the same performance level.

Also, the authors demonstrate that ALPS outperforms both traditional uncertainty-based methods and more recent approaches across various text classification tasks. Figure 2 shows that ALPS consistently achieves the highest test accuracy across all four datasets (AG News [7], IMDB [8], PubMed [9], and SST-2 [10]) as detailed in Table 1, especially in the earlier iterations.

Dataset	Domain	Train	Dev	Test	Number of Classes
AG NEWS	News articles	110,000	10,000	7,600	4
IMDB	Sentiment reviews	17,500	7,500	25,000	2
PUBMED 20k RCT	Medical abstracts	180,040	30,212	30,135	5
SST-2	Sentiment reviews	60,615	6,736	873	2

Table 1: Sentence classification datasets used in experiments.

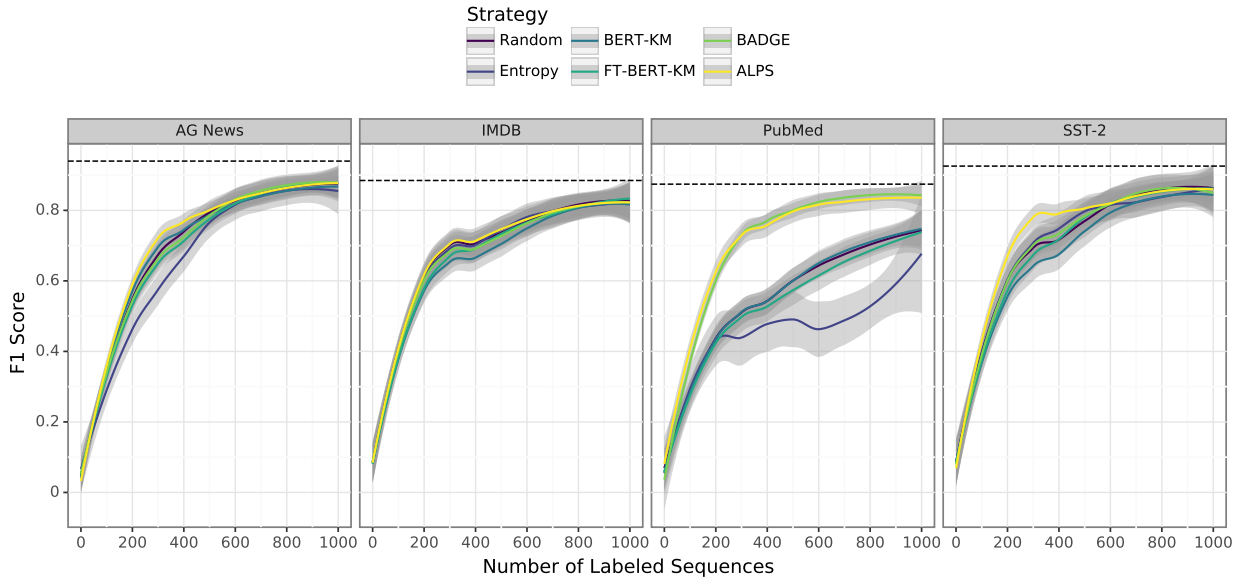

Figure 1: The test accuracy of simulated AL over ten iterations, with 100 sentences selected in each iteration, is shown. The dashed line indicates the test accuracy when the model is fine-tuned using the full dataset. In general, models trained on data selected by ALPS achieve the highest test accuracy, particularly in the early iterations.

BERT-KM: Clusters pre-trained, l_2-normalized BERT embeddings using k-means, and selects the nearest samples to the k cluster centers. The algorithm is similar to ALPS, but uses BERT embeddings.

FT-BERT-KM: This method is similar to BERT-KM, but it uses BERT embeddings, $h(x; W_{t-1})$, from the previously fine-tuned model instead of the pre-trained ones.

By leveraging the MLM loss from pre-trained language models as a proxy for uncertainty, ALPS outperforms traditional uncertainty-based methods. The results show that ALPS consistently achieves higher accuracy than the Entropy method across all datasets, particularly in the early iterations. For example, on the IMDB dataset, ALPS reaches an F1 score of 0.8 with about 600 labeled examples, while Entropy requires over 800. The method is computationally efficient, able to select a batch of examples in a single pass. The runtime comparisons show that ALPS is significantly faster than competing methods like BADGE and FT-BERT-KM. For instance, on the PubMed dataset, ALPS takes 24 minutes per iteration compared to 70 minutes for BADGE and 79 minutes for FT-BERT-KM.

The performance of ALPS may be contingent on the pre-trained language model's quality and relevance to the target domain. In highly specialized fields or tasks that diverge significantly from the pre-training data, the surprisal embeddings might not provide sufficiently informative signals for effective example selection. This could potentially restrict the method's applicability in niche areas or unconventional tasks. While ALPS has demonstrated impressive results in text classification, its efficacy in other NLP domains such as sequence labeling or text generation remains unexplored. The current research primarily addresses sentence-level tasks, and adapting the approach to token-level or document-level applications might necessitate substantial modifications.

ALPS does not directly tackle the problem of class imbalance within datasets. Although the clustering component may help to some extent by fostering diversity, there's still potential for enhancement, particularly when dealing with severely skewed label distributions, which are prevalent in real-world scenarios. Finally, the evaluation of ALPS has been conducted mainly in simulated active learning environments, which is a common practice in the field. However, practical implementation could present additional hurdles, such as dealing with inconsistent or noisy human annotations, which aren't fully addressed in the current study. Further investigation is needed to assess ALPS's effectiveness in more realistic conditions and to devise strategies for managing these practical challenges.

Despite areas that could be improved, the ALPS method marks a major step forward in active learning for NLP. By effectively using the strengths of pre-trained language models, it provides a practical and efficient solution to the cold-start issue in active learning. Its ability to swiftly identify informative samples without needing training specific to a task makes it particularly useful for quickly deploying models in new fields or for tasks with limited labeled data. As NLP continues to face challenges around data efficiency and model adaptation, methods like ALPS are likely to become increasingly important in narrowing the gap between the capabilities of large pre-trained models and the real-world limitations of practical applications.

Paper 3) SEAL: Semisupervised Adversarial Active Learning on Attributed Graphs
(IEEE Transactions on Neural Networks and Learning Systems, 2021)

The paper addresses a key challenge in using machine learning on graph-based data, focusing on the inefficiencies and less-than-ideal performance of current active learning methods for node classification. The motivation for this research comes from the rapid increase in graph-structured data across fields like social networks, citation networks, and financial systems. However, algorithms using graph-structured data require a large number of labeled nodes to work effectively, which is often expensive and time-consuming to acquire.

Current active learning approaches for graph-based node classification are limited by their use of simple combinations of strategies, like uncertainty sampling and graph centrality, which operate in separate scoring spaces. These methods fail to capture the intricate relationships between graph structure and node attributes. Moreover, most active learning techniques treat the query strategy and classifier as separate processes, missing out on potential mutual enhancement.

While graph neural networks (GNNs) [11]have shown promise in learning rich representations from graph data, existing AL methods don't fully leverage GNNs for more effective instance selection. There's a need for unified, adaptive AL strategies that consider both graph topology and node attributes when selecting informative nodes for labeling. As real-world graphs grow in size and complexity, scalability becomes crucial, requiring AL methods to handle large graphs efficiently without compromising selection quality.

The Semisupervised Adversarial Active Learning (SEAL) framework [12] tackles the problem of minimizing labeling efforts while maintaining high accuracy in node classification tasks. This approach aims to make graph-based machine learning more accessible, particularly in scenarios where resources are limited. By harnessing the expressive capabilities of graph neural networks and incorporating an innovative active learning strategy based on adversarial techniques, SEAL offers a comprehensive solution. The framework is composed of several interconnected elements that work together to enhance the overall learning process.

At the core of SEAL is the graph embedding network, which is based on a modified Graph Convolutional Network (GCN). This network takes the entire graph $G = \{V, E, X\}$, where V is the set of nodes, E is the set of edges, and X is the node feature matrix. The network produces low-dimensional embeddings H_L and H_U for labeled and unlabeled nodes, respectively. The goal is to encode both the graph structure and node attributes, ensuring that the labeled and unlabeled nodes become indistinguishable to the discriminator. The objective is captured by the following loss function:

$$J_G = ||\mathbb{E}_{\mathbf{x}\sim L^+}[D^{(n)}(G^{(m)}(x))] - \mathbb{E}_{\mathbf{x}\sim U^-}[D^{(n)}(G^{(m)}(x))]||^2 + J_{GCN}$$

where the first term represents the feature matching loss, and J_{GCN} is the standard GCN cross-entropy loss used for node classification.

SEAL also features a Pool Tuning (PT) mechanism, which dynamically redistributes nodes between labeled and unlabeled pools. The network calculates prediction probabilities for all nodes, and highly confident unlabeled nodes are moved to a pseudo-labeled (p-labeled) pool, while less confident nodes remain in the unlabeled (p-unlabeled) pool. This process is governed by the following expressions:

$$L^+ = L \cup \{x_i \in U | P(\hat{y}|x_i) > \delta\}$$

$$U^- = \{x_i \in U | P(\hat{y}|x_i) \leq \delta\}$$

where δ is a tunable threshold, L^+ is the pseudo-labeled pool, and U^- is the refined unlabeled

pool. This mechanism helps the model focus on the most uncertain predictions and narrows down the search space for selecting informative instances.

The Semisupervised Discriminator Network $D(\cdot)$ is a three-layer fully connected neural network with units (128, 128, K), where K represents the number of classes.. It takes node embeddings from $G(\cdot)$ for both pseudo-labeled (L^+) and pseudo-unlabeled (U^-) pools as input and outputs $K+1$ probabilities K class probabilities and 1 probability of being unlabeled. The loss function is defined as:

$$J_D = \alpha \cdot J_{sup} + J_{unsup}$$

where J_{sup} is the supervised cross-entropy loss on labeled data, and J_{unsup} is the unsupervised adversarial loss. The detailed formulation is:

$$J_{sup} = -\mathbb{E}_{\mathbf{x}\sim L}[\log P(\mathbf{y}|\mathbf{x}, \mathbf{y} < K+1)]$$

$$J_{unsup} = -\left(\mathbb{E}_{\mathbf{x}\sim L^+}[\log D(G^{(m)}(\mathbf{x}))] + \mathbb{E}_{\mathbf{x}\sim U^-}[\log(1 - D(G^{(m)}(\mathbf{x})))]\right)$$

$$D(G^{(m)}(\mathbf{x})) = 1 - P(\mathbf{y} = K+1|\mathbf{x})$$

The supervised loss J_{sup} is calculated using only the originally labeled data L via cross-entropy, whereas the unsupervised loss J_{unsup} is computed through adversarial training, using both the pseudo-labeled nodes L^+ and the pseudo-unlabeled nodes U^-. Here, $D(G^{(m)}(x))$ represents the probability that node x is p-labeled.

The optimal solution that minimizes both J_{sup} (the supervised loss) and J_{unsup} (the unsupervised adversarial loss) is defined by the following equations:

$$e^{l_j(\mathbf{x})} = c(\mathbf{x})p(\mathbf{x} = j, \mathbf{y}), \forall j < K+1,$$

$$e^{l_{K+1}(\mathbf{x})} = c(\mathbf{x})p(\mathbf{y} = K+1, \mathbf{x}),$$

where $l_j(\mathbf{x})$ represents the logit for class j, and $c(\mathbf{x})$ is an undetermined scaling function. In other words, the solution that perfectly minimizes J_{unsup} also perfectly minimizes J_{sup}. This consistency ensures that optimizing J_{unsup} aids in improving supervised performance as well. Therefore, by jointly minimizing the two loss functions, the SEAL framework expects to better approximate the optimal solution.

The Adversarial Learning Mechanism [13] adopts a GAN-like adversarial process between $G(\cdot)$ and $D(\cdot)$. $G(\cdot)$ tries to fool $D(\cdot)$ by making labeled and unlabeled node distributions indistinguishable, while $D(\cdot)$ attempts to distinguish between labeled and unlabeled nodes and predict node classes. The two networks are optimized alternately to strike a balance where $D(\cdot)$ becomes a strong measure of node informativeness.

Active scoring and node selection are done through a scoring function $\text{div}(x_{U^-}, L^+) = 1 - D(x_{U^-})$, and the selection criterion is defined as:

$$x^* = \arg\max_{x_i \in U^-} \text{div}(x_i, L^+)$$

This process selects the unlabeled node with the highest divergence score for labeling.

The training algorithm begins by initializing the parameters of $G(\cdot)$ and $D(\cdot)$, followed by pretraining for n_p epochs. In the main training loop, $G(\cdot)$ is updated by minimizing J_G, pool tuning is performed to generate L^+ and U^-, and $D(\cdot)$ is updated by minimizing J_D. If the budget allows, the most informative node is selected for labeling, and the labeled and unlabeled pools are updated. This loop continues until convergence or budget exhaustion.

The SEAL framework introduces several novel contributions to the field of active learning on attributed graphs, supported by comprehensive experimental results. One of the key innovations of SEAL is its unified adversarial framework, which seamlessly integrates graph neural networks with the active learning query engine. This unified approach has demonstrated superior performance across multiple datasets, including Citeseer, Cora [14], and DBLP [15]. For example, SEAL achieved a Micro-F1 score of 73.2% on the Citeseer dataset, outperforming other state-of-the-art methods such as ANRMAB (71.9%) [16] and AGE [17] (71.4%).

Another notable feature of SEAL is its introduction of a novel semisupervised adversarial learning (SAL) component, which uses a unique discriminator structure with multiple outputs specifically designed for active learning on graphs. Ablation studies show that the SAL component significantly enhances SEAL's performance. On the Cora dataset, for instance, SEAL without SAL achieved a Micro-F1 score of 82.7%, while the full SEAL model reached 84.5%, highlighting the effectiveness of the semisupervised approach.

SEAL also improves upon existing methods by generating a unified informativeness score in a common latent space, as opposed to the naive combination of different active learning strategies. This unified scoring mechanism leads to more effective instance selection. On the DBLP dataset, SEAL achieved a Macro-F1 score of 89.6%, compared to ANRMAB's 87.2% [16] and AGE's 86.5% [17], demonstrating the advantage of this approach.

A key component of SEAL's framework is its pool tuning mechanism, which focuses the discriminator on the most uncertain predictions. This step enhances both stability and efficiency. For example, on Citeseer, SEAL without pool tuning achieved a Micro-F1 score of 71.8%, while the full SEAL model improved to 73.2%, underscoring the importance of pool tuning in the learning process.

SEAL also incorporates a feature matching loss in its graph embedding network to mitigate overfitting and mode collapse, improving training stability. Experiments comparing SEAL with and without feature matching show that this feature leads to more consistent results. For instance, on the Cora dataset, the standard deviation of Micro-F1 scores across multiple runs dropped from 0.8% without feature matching to 0.5% with it, demonstrating more stable performance.

Extensive experiments were conducted on three real-world networks: Citeseer, Cora, and DBLP, each with varying sizes and characteristics. SEAL consistently outperformed baseline methods across all datasets, with the performance gap most noticeable on larger graphs. For example, on DBLP, SEAL achieved a 2.4% higher Micro-F1 score than ANRMAB, highlighting its scalability and robustness.

From a scalability perspective, SEALs computational complexity is proven to be linear with respect to the number of edges and nodes, which is supported by empirical evidence. On the Pubmed dataset, which consists of 19,717 nodes, SEAL's training time scaled linearly as the graph size increased, while competitors like ANRMAB [16] exhibited quadratic growth. At 19,000 nodes, SEAL completed training in approximately 200 seconds, compared to over 800 seconds for ANRMAB [16], highlighting SEAL's superior scalability.

In terms of labeling efficiency, SEAL achieves higher accuracy with fewer labeled nodes. On Citeseer, SEAL reached 72% Micro-F1 with just 66 labeled nodes, while ANRMAB [16] required 120 labeled nodes to achieve similar performance. This trend of improved labeling efficiency was consistent across all datasets, with SEAL maintaining a performance advantage throughout the active learning process.

While the SEAL framework offers notable advancements in active learning on attributed graphs, it also presents several potential downsides and limitations that warrant consideration. One key con-

cern is the added complexity and computational overhead introduced by the adversarial framework and the multiple components, such as the graph embedding network, pool tuning, and semisupervised discriminator. This complexity can lead to increased computational demands, particularly during the training phase, which may pose challenges for very large graphs or environments with limited computational resources.

Another limitation is SEAL's sensitivity to hyperparameter choices, particularly the pool tuning threshold (δ) and the balance factor (α) in the discriminator loss. Determining the optimal values for these parameters often requires extensive tuning, which can be both time-consuming and dataset-dependent, limiting the method's practical usability. Additionally, while feature matching helps mitigate some instability issues, adversarial learning approaches are still prone to training instabilities, and the paper does not provide a detailed analysis of the stability of the framework across different initializations or training runs.

The paper also lacks strong theoretical guarantees on the convergence or optimality of the proposed method, focusing more on empirical results. This lack of theoretical grounding makes it difficult to predict or bound SEAL's performance in new, unseen scenarios. Additionally, the current implementation of SEAL relies heavily on a specific GCN architecture for the graph embedding network, and it's unclear how well the method would perform with other types of GNNs or graph embedding techniques, raising questions about its generalizability across different architectures.

In terms of scalability, while the authors demonstrate improved scalability compared to some baselines, the evaluation is limited to graphs with tens of thousands of nodes. SEAL's performance and efficiency on much larger graphs, with millions of nodes or edges, remain untested, leaving its scalability in extreme cases uncertain. Similarly, the paper does not offer a thorough analysis of how different types or qualities of node features influence SEALs performance, which could be important for understanding its effectiveness across diverse datasets.

Despite these potential drawbacks, SEAL represents a significant step forward in the field of machine learning on graph-structured data, offering innovative solutions to many challenges in active learning. However, understanding and addressing these limitations will be crucial for further advancing the framework and expanding its real-world applicability.

References

[1] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-effective active learning for deep image classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 12, pp. 2591–2600, 2017.

[2] B.-C. Chen, C.-S. Chen, and W. H. Hsu, "Cross-age reference coding for age-invariant face recognition and retrieval," in *Computer Vision – ECCV 2014*, Cham: Springer International Publishing, 2014, pp. 768–783.

[3] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*. Mar. 2007.

[4] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, and A. Agarwal, "Deep batch active learning by diverse, uncertain gradient lower bounds," in *International Conference on Learning Representations*, 2020.

[5] M. Yuan, H.-T. Lin, and J. Boyd-Graber, "Cold-start active learning through self-supervised language modeling," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Nov. 2020, pp. 7935–7948.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186.

[7] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'15, Montreal, Canada: MIT Press, 2015, pp. 649–657.

[8] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 142–150.

[9] F. Dernoncourt and J. Y. Lee, "PubMed 200k RCT: A dataset for sequential sentence classification in medical abstracts," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 308–313.

[10] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 1631–1642.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[12] Y. Li, J. Yin, and L. Chen, "SEAL: Semisupervised adversarial active learning on attributed graphs," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3136–3147, 2021.

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc., 2014.

[14] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[15] D. Zhang, J. Yin, X. Zhu, and C. Zhang, "Attributed network embedding via subspace discovery," *Data Min. Knowl. Discov.*, vol. 33, no. 6, pp. 1953–1980, Nov. 2019, ISSN: 1384-5810.

[16] L. Gao, H. Yang, C. Zhou, J. Wu, S. Pan, and Y. Hu, "Active discriminative network representation learning," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, Jul. 2018, pp. 2142–2148.

[17] H. Cai, V. W. Zheng, and K. C.-C. Chang, "Active learning for graph embedding," *ArXiv*, 2017.