# Active Heterogeneous Graph Neural Networks with Per-step Meta-Q-Learning

Yuheng Zhang*, Yinglong Xia†, Yan Zhu†, Yuejie Chi‡, Lei Ying§, Hanghang Tong*
*University of Illinois at Urbana-Champaign, †Meta, ‡Carnegie Mellon University, §University of Michigan
*{yuhengz2, htong}@illinois.edu, †{yxia, yzhu}@fb.com, ‡yuejiechi@cmu.edu, §leiying@umich.edu

***Abstract*—Recent years have witnessed the superior performance of heterogeneous graph neural networks (HGNNs) in dealing with heterogeneous information networks (HINs). Nonetheless, the success of HGNNs often depends on the availability of sufficient labeled training data, which can be very expensive to obtain in real scenarios. Active learning provides an effective solution to tackle the data scarcity challenge. For the vast majority of the existing work regarding active learning on graphs, they mainly focus on homogeneous graphs, and thus fall in short or even become inapplicable on HINs. In this paper, we study the active learning problem with HGNNs and propose a novel meta-reinforced active learning framework MetRA. Previous reinforced active learning algorithms train the policy network on labeled source graphs and directly transfer the policy to the target graph without any adaptation. To better exploit the information from the target graph in the adaptation phase, we propose a novel policy transfer algorithm based on meta-Q-learning termed per-step MQL. Empirical evaluations on HINs demonstrate the effectiveness of our proposed framework. The improvement over the best baseline is up to 7% in Micro-F1.**

***Index Terms*—Active learning, Meta-Reinforcement learning, Heterogeneous graph neural networks**

## I. Introduction

Heterogeneous information networks (HINs) also known as heterogeneous graphs (HGs) consist of multiple types of nodes and edges. Hence, HINs have a great capacity for encoding rich semantic information and are ubiquitous in real world applications such as recommendation systems [11], biomedicine [30] and security systems [8]. Recently, various heterogeneous graph neural networks (HGNNs) [27] [7] have been proposed to process HINs. They rely on a semi-supervised learning paradigm which requires sufficient labeled training data. Nevertheless, the annotation cost could be very expensive, especially in high-stake decision-making scenarios, e.g., medical diagnosis [5] and molecular biology [10]. Active learning [20] [18] [1] is an effective solution to tackle the data scarcity challenge. Effective active learning methods are capable of identifying the samples which provide maximal information for model training. Thus, a high-performance HGNN could be obtained with limited labeling cost which broadens the HGNN applications. However, the sparse literature on active learning on graphs [2] [9] [29] focus on homogeneous graphs which do not bear the complex relations and structure in HINs. Extending the existing algorithms to HINs is highly non-trivial. We need to carefully consider the characteristics of HINs and model the composite relations between different kinds of nodes. This requires us to measure the informativeness of nodes from different perspectives of semantic information.

In this paper, we study the active learning problem with HGNNs and propose a novel **Met**a-**R**einforced **A**ctive learning framework MetRA. We observe that the scenario of active learning could be viewed as the interaction between a learning agent and the oracle, and the goal of the agent is to maximize the performance gain of the machine learning model. Hence, we formulate the active learning problem as a Markov Decision Process (MDP) and employ deep Q-learning techniques [15] [26] to learn the labeling policy. Compared with heuristic strategy based methods [2], deep reinforcement learning (RL) based active learning methods have two main advantages: (1) It allows the labeling policy to take multiple state metrics into account and learn a non-linear combination of different active learning strategies. (2) Through parameterizing our Q-network as an HGNN, we leverage a heterogeneous message passing mechanism to aggregate information from neighbors with different kinds of relations. This design enables us to fully exploit the graph structural information and address the heterogeneity challenge.

In particular, we address a fundamental problem in the reinforced active learning framework: *How to train our Q-network*? Since we could only run the active learning process once and deep RL is hungry for data, it is implausible to directly train the Q-network on the target graph. Previous works [9] [29] deal with this problem by training the policy network on source graphs with full label information and directly transferring it to the target graph without any adaptation. This strategy relies on the assumption that the optimal labeling policies for the source and target graphs are very similar. To better exploit the information from the target graph, we propose to incorporate meta-reinforcement learning techniques [4] into our active learning framework. Specifically, our algorithm consists of two phases: the meta-training phase and the adaptation phase. During the meta-training phase, we interact with the fully labeled source graph and collect transitions to train our Q-network. In the adaptation phase, we recycle the transitions from the meta-training replay buffer to adapt the previously learned policy. Considering that the optimal labeling policy needs to use different strategies according to the status of the HGNN, we creatively perform the adaptation in a *per-step manner*, which is different from the commonly used meta-reinforcement learning [4]. In each time step, we measure the similarity between the transitions from

the meta-training replay buffer and the state of the target graph. The transitions with higher similarity are more beneficial for the policy adaptation and we leverage them to adapt the current policy using off-policy updates. Compared with the direct transfer strategy, we take the state of the target graph into account and pick the important transitions from the meta-training replay buffer which boosts the transfer performance.

The main contributions of this paper can be summarized as follows.

- **Problem Formulation.** To our best knowledge, we are the first to formulate active learning problem with HGNNs as markov decision process (MDP) and employ deep RL techniques to learn the labeling policy.
- **Algorithmic Design.** We propose a novel meta-reinforced active learning framework MetRA. A per-step meta-Q-learning algorithm is developed to boost the model performance in the adaptation phase.
- **Empirical Evaluations.** Extensive experiments are conducted on HIN tasks to demonstrate the effectiveness of our proposed framework.

## II. Problem statement

In this section, we first introduce the definitions and preliminaries related to HINs, and then we formally define the active learning problem with heterogeneous graph neural networks (HGNNs).

**Definition 1. Heterogeneous Information Network [21].** Heterogeneous information network (HIN) is defined as a network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}, \mathcal{R}, f_{\mathcal{A}}, f_{\mathcal{R}})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. $\mathcal{A}$ and $\mathcal{R}$ denote the set of node types and the set of edge types respectively, where $|\mathcal{A}| + |\mathcal{R}| > 2$. Each node is associated with a node type mapping function $f_{\mathcal{A}} : \mathcal{V} \to \mathcal{A}$. Similarly, each edge is associated with a edge type mapping function $f_{\mathcal{R}} : \mathcal{E} \to \mathcal{R}$.

**Definition 2. Meta-path [22].** A Meta-path $\mathcal{P}$ is a path defined as $A_1 \xrightarrow{R_1} A_2 \xrightarrow{R_2} \dots \xrightarrow{R_l} A_{l+1}$, where $A_1, \cdots, A_{l+1} \in \mathcal{A}$ and $R_1, \cdots, R_l \in \mathcal{R}$. It describes the composite relation $R = R_1 \circ R_2 \circ \cdots \circ R_l$ between nodes with type $A_1$ and type $A_{l+1}$ and contains high-order structure.

Here, our HGNN is built on HAN [27] which aggregates information from meta-path based neighbors. A semantic-level attention is employed to learn the importance of different meta-paths. The details of HAN could be found in [27].

The process of active learning could be viewed as the interaction between a learning agent and an oracle (e.g. a human annotator). The agent acquires the label information from the oracle and constructs the training set. The goal of the agent is to maximize the performance of the machine learning model while minimizing the annotation cost. In this paper, we focus on the active learning problem with HGNNs and formulate it as follows.

**Problem 1. Active learning with HGNNs.** In each iteration,

**Given:** (1) a heterogeneous information network $\mathcal{G}$, (2) a HGNN model $f$, (3) an oracle providing label information, (4) Unlabeled pool $\mathcal{U}$, (5) query batch size $b$ and total query budget $B$.

**Find:** a set of $b$ nodes from $\mathcal{U}$ which are most informative to improve the training of $f$. The interaction process is continued until the query budget $B$ is used up.

## III. Methodology

In this section, we present the details of the meta-training phase and the adaptation phase respectively. The overall framework is shown in Figure 1.

### A. Meta-training Phase

In the active learning task, we need to interactively select a batch of nodes in each iteration and maximize the performance gain of HGNN. Therefore, this problem could be naturally formulated as a fully cooperative multi-agent task. Each agent needs to select an unlabeled node forming the query batch, they need to cooperate with each other to achieve the maximal team reward which is equivalent to the HGNN performance on downstream tasks. We formally describe the task as a tuple $\mathcal{M} = \langle \mathcal{N}, \mathcal{S}, \mathcal{U}, P, r, \gamma \rangle$, where $\mathcal{N} := \{1, \dots, n\}$ denotes the set of agents and $\mathcal{S}$ denotes the set of states. The state $s \in \mathcal{S}$ depicts the status of the HGNN model and the characteristics of HIN $\mathcal{G}$. It is shared across different agents. At each time step, every agent $i \in \mathcal{N}$ observes state $s$ and chooses an unlabeled node as its action $u_i$, which forms the joint action $\mathbf{u} \in \mathcal{U}$. It results in a joint reward $r(s, \mathbf{u})$ and a transition to the next state $s' \sim P(\cdot|s, \mathbf{u})$. $\gamma \in [0, 1)$ is the discount factor. Next, we introduce the details of state $s$, action $\mathbf{u}$, and reward $r$ respectively.

*1) State:* Following [29], we use the state representation from three perspectives to measure the informativeness of each unlabeled node.

(1) **Uncertainty.** Machine learning models tend to make mistakes on uncertain samples. Acquiring the labels of uncertain nodes is very helpful for improving the performance of HGNNs. Since we focus on the node classification task, we use the entropy of the predicted probability distribution as the uncertainty metric.

(2) **Centrality.** Nodes with high centrality tend to play important roles in the graph and their labels provide more information for model training. Since we are dealing with HINs consisting of multiple types of nodes and edges, we have used two methods to calculate the centrality. The first method is to split the graph into multiple homogeneous graphs according to the meta-paths and calculate the node centrality separately. The other method is to ignore the heterogeneity and view the entire graph as a homogeneous graph. From the empirical evaluation, we observe that the latter method performs better and PageRank centrality [16] [25] are employed as the centrality metric.

(3) **Node Embedding Distance.** Suppose our training set all consists of nodes with high centrality and high entropy, the distribution of the training set will be quite different from the test set. Selecting the nodes with diverse node

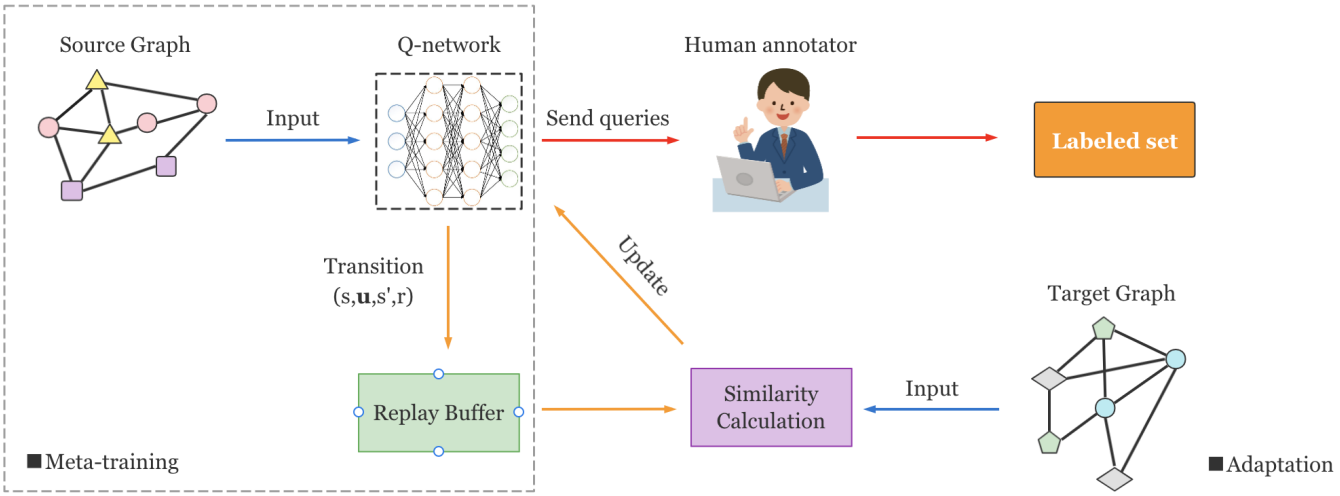



Fig. 1. Illustration of the proposed MetRA framework. During the meta-training phase, our Q-network is interacting with the labeled source graph to collect transitions for training. When Q-network is transferred to the target graph, we measure the similarity between source transitions and the target graph state in each iteration. Transitions with high similarity are selected to adapt Q-network using off-policy updates. After the adaptation, Q-network selects nodes from the unlabeled pool and queries for their labels. The labeled nodes are added to the labeled set for training the HGNN model.

embeddings could increase the diversity of the training set and thus circumvent the distribution shift problem.

The overall state $s$ could be viewed as an attributed HIN, the node attribute is a combination of uncertainty, centrality, and node embedding distance. The details of state calculation can be found in the appendix.

*2) Action:* Observing state $s$, each agent $i$ selects an unlabeled node $u_i$ and forms the joint action $\mathbf{u} = \{u_1, u_2, \cdots, u_n\}$, i.e. the selected batch of nodes. The oracle will provide the corresponding label information which updates the labeled training set.

*3) Reward:* Since our goal is to improve the performance of HGNN model $f$, it is straightforward to define the reward as the performance gain on the validation set. We use the classification accuracy to evaluate the model and the reward is the improvement after training $f$ on the updated training set.

*4) HGNN Q-network:* To fully leverage the characteristics of HIN, we implement our Q-network as an HGNN and predict the individual Q-value for each candidate unlabeled node. We employ a heterogeneous message passing mechanism to learn the node embedding which considers the high-order meta-path structure. For node $i$ and a meta-path $\mathcal{P}_k$, we aggregate the information from the meta-path based neighbors $N_i^{\mathcal{P}_k}$ with a GCN layer [13]:

$$z_i^{\mathcal{P}_k} = \frac{1}{d_i+1}\mathbf{W}_{\mathcal{P}_k} z_i + \sum_{j \in N_i^{\mathcal{P}_k}} \frac{1}{\sqrt{(d_i+1)(d_j+1)}} \mathbf{W}_{\mathcal{P}_k} z_j, \tag{1}$$

where $d_i$ and $d_j$ are degrees of node $i$ and $j$ in the view of meta-path $\mathcal{P}_k$, and $\mathbf{W}_{\mathcal{P}_k}$ is the learnable parameters for meta-path $\mathcal{P}_k$. Considering the meta-paths of source graphs and target graphs may contain different semantic information, $\mathbf{W}_{\mathcal{P}_k}$ are shared across different meta-paths. Suppose we have $K$ meta-paths $\{\mathcal{P}_1, \mathcal{P}_2, \cdots, \mathcal{P}_K\}$ starting from node $i$, we can then obtain $K$ node embeddings $\{z_i^{\mathcal{P}_1}, z_i^{\mathcal{P}_1}, \cdots, z_i^{\mathcal{P}_K}\}$. The final embedding $z_i$ is the element-wise mean value of the $K$ embeddings:

$$z_i = \frac{1}{K}\sum_{k=1}^{K} z_i^{\mathcal{P}_k}. \tag{2}$$

Once we have the embedding $z_i$ of node $i$, a linear layer is used to predict the individual Q-value $Q_i(s, u_i)$.

*5) Q-network Training.:* Due to the combinatorial explosion of the joint action space, it is intractable to directly model the joint Q-value function. Following the value decomposition idea in [23] [29], we make the assumption that joint Q-value $Q(s, \mathbf{u})$ can be additively decomposed into individual Q-value $Q_i(s, u_i)$,

$$Q(s, \mathbf{u}) \approx \frac{1}{n}\sum_{i=1}^{n} Q_i(s, u_i). \tag{3}$$

Based on our assumption, the Q-value of the selected batch is approximated with the average of individual Q-values.

We employ deep Q-learning techniques [15] to train our Q-network. Q-learning aims to optimize the temporal difference (TD) error [24], which is defined as the expectation over transitions $\tau = (s_t, \mathbf{u_t}, r_{t+1}, s_{t+1})$ as follows,

$$\mathbb{E}_{\tau \sim \mathcal{B}}[(Y_t - Q(s_t, \mathbf{u}_t))^2], \tag{4}$$

where $\mathcal{B}$ denotes the replay buffer and $Y_t$ denotes the TD target. To improve the training stability, we use double DQN [26] to calculate the TD target $Y_t$. Double DQN decouples the action selection and action evaluation. Actions are selected by

the online network and are evaluated by the target network. With this design, the TD target $Y_t$ is calculated as follows,

$$Y_t \equiv r(s_t, \mathbf{u}_t) + \gamma Q(s_{t+1}, \arg\max_{\mathbf{u}_{t+1}} Q(s_{t+1}, \mathbf{u}_{t+1}; \theta); \theta'), \quad (5)$$

where $\theta$ denotes the the parameters of the online network and $\theta'$ denotes the parameters of the target network.

### B. Adaptation Phase

After the meta-training phase, we have obtained the active learning policy trained on the labeled source graph. Considering that the source graph and the target graph may come from different domains and have different data distributions, directly transferring the policy to the target graph [9] is not the optimal solution. Inspired by meta-reinforcement learning (meta-RL) [6] [17], we seek to incorporate Meta-Q-Learning (MQL) [4] techniques into the adaptation phase. MQL is an off-policy meta-RL algorithm proposed for Q-learning. The key idea is to update the Q-network with transitions from meta-training replay buffer in the adaptation phase. However, the original MQL algorithm cannot be directly applied to our active learning scenario due to two reasons: (1) MQL relies on the assumption that the agent can do some exploration on the target task. To pick transitions that are similar to the target domain, a binary classifier is trained to discriminate between transitions from source tasks and transitions from target tasks. The prediction from the classifier is employed as the similarity metric. Nevertheless, for the active learning task, we only have one chance to interact with the environment. After the queries are sent to the oracle, the performance of the HGNN model is finalized and cannot be changed. Hence, we cannot collect transitions from the target graph to train the binary classifier. (2) MQL is designed for robotic manipulation tasks and the state could be described as a vector. In contrast, we need to deal with graph-structured data which has complex relations between nodes.

To address the problems mentioned above, we propose a novel policy adaptation algorithm termed per-step MQL. Since the optimal policy needs to take different strategies in different time steps, we conduct the policy adaptation in a per-step manner. In each time step, we measure the similarity between the transitions from the meta-training replay buffer and the current state of the target graph. Transitions with high similarity are recycled to update our Q-network with a weighted loss function. Since the only information from the target graph is the current state, it is implausible to train a binary classifier and use the prediction from the classifier as the similarity metric. Instead, we calculate the similarity in an unsupervised manner. For a transition $(s, \mathbf{u}, s', r)$, only the Q-values of the selected nodes $u_i$ are considered in the TD error. Therefore, we focus on the selected nodes instead of the entire graph when measuring the similarity. Considering our Q-network is implemented as an HGNN, it takes both the node state and the graph structural information into account. Hence, we leverage the Euclidean distance between the node embeddings from the Q-network to calculate the similarity. We formulate the problem as a maximum weight bipartite graph matching problem [28]. Specifically, the selected nodes from the source graph $v_s$ and the candidate nodes with the highest Q-values from the target graph $v_t$ are considered as the two types of nodes in the bipartite graph. The weight of the edge $e_{ij}$ is defined as the negative Euclidean distance between the embedding of $v_s^i$ and the embedding of $v_t^j$. A matching in the bipartite graph is a subset of the edges satisfying that no two edges share the same node. Our goal is to find a matching in which each node is covered and the weights of the chosen edges are maximal. The optimal matching is obtained with Hungarian algorithm [14].

With the similarity measurement, we optimize the following objective to update our Q-network in each time step $t$,

$$\mathbb{E}_{\tau \sim \mathcal{B}_t}[\beta(\tau; S_{\mathcal{G}_t}, \mathcal{B}_t) \mathrm{TD}^2(\theta)], \quad (6)$$

where $S_{\mathcal{G}_t}$ denotes the state of the target graph $\mathcal{G}_t$, $\mathcal{B}_t$ denotes the meta-training replay buffer containing transitions in time step $t$, $\beta(\tau; S_{\mathcal{G}_t}, \mathcal{B}_t)$ denotes the similarity between the transition $\tau$ and state $S_{\mathcal{G}_t}$, and $\mathrm{TD}^2(\cdot)$ denotes the squared TD error described in Eq. (4). Our proposed per-step MQL reweighs the transitions according to their similarities to the state of the target graph. Compared with the direct transfer strategy [9] [29], our method exploits the information from the target graph better, which is in turn beneficial for improving the active learning performance during the adaptation phase.

## IV. Experiments

TABLE I
THE STATISTICS OF THE DATASETS.

| Datasets | Nodes | Edges | Meta-paths |
|---|---|---|---|
| DBLP | author (A):4057<br>paper (P):14328<br>term (T):7723<br>venue(V):20 | A-P:19645<br>T-P:85810<br>V-P:14328 | APA<br>APTPA<br>APVPA |
| ACM | paper (P):4019<br>author (A): 7167<br>subject (S):60 | A-P:13407<br>S-P:4019 | PAP<br>PSP |
| AMiner | paper (P):6564<br>author (A):13329<br>reference (R):35890 | A-P:18007<br>R-P:58831 | PAP<br>PRP |

### A. Experimental Setup

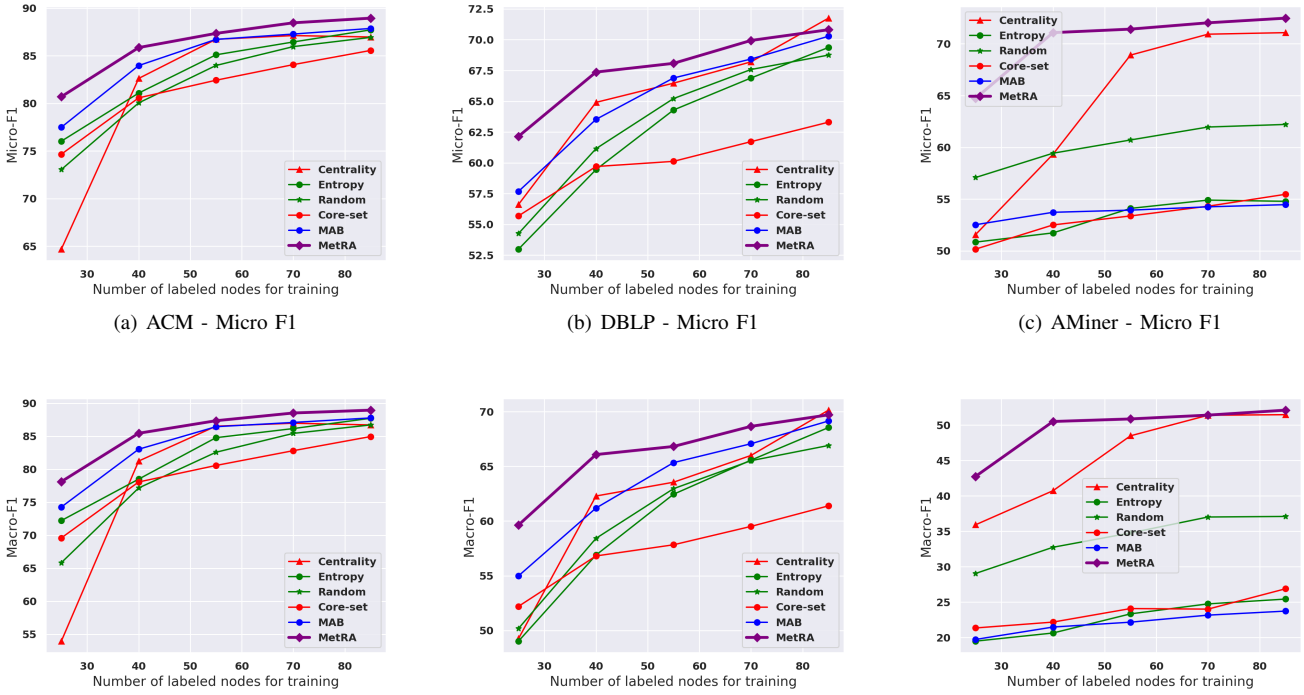
*1) Datasets:* Our proposed MetRA is evaluated on the following three HIN node classification datasets, whose statistics is summarized in Table I.

*2) Baselines:* We compare our method with both heuristic strategy based methods and reinforcement learning based method.

(1) Random: In each iteration, we select unlabeled nodes from a uniform distribution.

(2) Centrality: In each iteration, we calculate the PageRank centrality of unlabeled nodes and select nodes with the highest centrality.

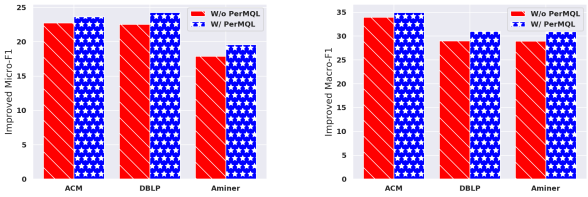
(3) Entropy: In each iteration, we calculate the entropy of the prediction from model $f$ and select nodes with the highest entropy.




(a) ACM - Micro F1 (b) DBLP - Micro F1 (c) AMiner - Micro F1

(d) ACM - Macro F1 (e) DBLP - Macro F1 (f) AMiner - Macro F1

Fig. 2. Active node classification performance on HINs with different number of labeled nodes for training. Our method MetRA is marked as the purple line.




(a) Micro F1 (b) Macro F1

Fig. 3. Ablation study of per-step MQL on HINs. The query budget is set to be 40.

(4) Coreset [19]: For each unlabeled node, we calculate the minimum distance between the node embedding and the labeled nodes' embeddings. We select nodes with the maximum distance.

(5) MAB: We adopt ActiveHNE [3] for active learning with HGNNs. A multi-armed bandit (MAB) framework is employed to learn the combination weights of these three heuristic metrics. The performance gain from previous time steps is used as the reward for MAB to dynamically adjust the weights during the active learning process.

*3) Evaluation Metrics:* We use Macro-F1 and Micro-F1 as the metrics to evaluate our proposed method and baselines. We use 1,000 labeled nodes as the test set and randomly sample 500 labeled nodes from the remaining nodes for validation. We run 20 independent experiments with different initializations for the HGNN classification model. In each active learning time step, the HGNN model is tested 5 times and the average performance on the test set is reported.

*4) Implementation Details:* We use Adam optimizer [12] to train both the classification model and Q-network. The initial number of samples is set to be 10, and the query batch size is set to be 15. For HGNN classification model, a meta-path based GCN layer [13] of 64 embedding dimension is employed for each type of meta-path. The learning rate is set to be 0.01. For HGNN Q-network, the embedding dimension of meta-path based GCN layer is set to be 8. During the meta-training phase, Q-network is trained with learning rate 0.001 and batch size 16. In the adaptation phase, the learning rate is set to be 0.0005. The similarity threshold $\lambda$ is set to be 0.5. For ACM dataset, the source graph is DBLP dataset. For DBLP and AMiner datasets, the source graph is ACM dataset.

*5) Overall Active Learning Performance on HINs:* We present the active node classification performance on HINs in Figure 2. We can see that our proposed MetRA consistently outperforms other baselines under different query budgets. In particular, when the number of labeled nodes is 25, MetRA outperforms the best baseline by at least 3%, 4%, 7% Micro-F1 on ACM, DBLP, and AMiner dataset respectively. The superior performance achieved by MetRA can be attributed to two folds: (1) We employ deep reinforcement learning (RL) techniques to learn the optimal active learning policy. Deep RL enables us to learn a non-linear combination of different heuristic metrics. By implementing the Q-network as an HGNN, we could fully leverage the composite relations in HINs to estimate the Q-value for each candidate unlabeled

node. (2) With our proposed per-step MQL, we could take the information from the target graph into account during the policy transfer process. Although the 'MAB' baseline also employs RL techniques to learn the combination weights of different state metrics, it does not consider the composite relation information which plays a vital role in active learning for HINs.

*6) Ablation Study of per-step MQL:* We compare the performance between methods with and without per-step MQL algorithm and results are shown in Figure 3. The query budget is set to be 40. The y-axis represents the improved classification performance after adding the selected nodes to the training set. The results demonstrate that our proposed per-step MQL consistently improves the policy transfer performance on all three datasets. In particular, the improvement brought by method 'w/ PerMQL' is more than 2.5%, 6.5%, 6.5% Macro-F1 of method 'w/o PerMQL' on ACM, DBLP, and AMiner respectively.

## V. Conclusion

In this paper, we study the active learning problem with HGNNs and propose a novel meta-reinforced active learning framework MetRA. To address the heterogeneity challenge, we implement the Q-network as an HGNN to fully leverage the rich composite relation information. Different from the existing work on active GNNs, we propose a novel per-step MQL algorithm to facilitate policy transfer. It enables us to take the information from the target graph into account during the policy adaptation phase. In the future, we plan to extend our active learning framework to other graph machine learning tasks such as link prediction and recommendation.

## References


[1] Y. Ban, Y. Zhang, H. Tong, A. Banerjee, and J. He, "Improved algorithms for neural active learning," *arXiv preprint arXiv:2210.00423*, 2022.

[2] H. Cai, V. W. Zheng, and K. C.-C. Chang, "Active learning for graph embedding," *arXiv preprint arXiv:1705.05085*, 2017.

[3] X. Chen, G. Yu, J. Wang, C. Domeniconi, Z. Li, and X. Zhang, "Activehne: Active heterogeneous network embedding," *arXiv preprint arXiv:1905.05659*, 2019.

[4] R. Fakoor, P. Chaudhari, S. Soatto, and A. J. Smola, "Meta-q-learning," *arXiv preprint arXiv:1910.00125*, 2019.

[5] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The World Wide Web Conference*, 2019, pp. 417–426.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1126–1135.

[7] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of The Web Conference 2020*, 2020, pp. 2331–2341.

[8] B. Hu, Z. Zhang, C. Shi, J. Zhou, X. Li, and Y. Qi, "Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 946–953.

[9] S. Hu, Z. Xiong, M. Qu, X. Yuan, M.-A. Côté, Z. Liu, and J. Tang, "Graph policy network for transferable active learning on graphs," *arXiv preprint arXiv:2006.13463*, 2020.

[10] W. Hu, B. Liu, J. Gomes, M. Zitnik, P. Liang, V. Pande, and J. Leskovec, "Strategies for pre-training graph neural networks," *arXiv preprint arXiv:1905.12265*, 2019.

[11] J. Jin, J. Qin, Y. Fang, K. Du, W. Zhang, Y. Yu, Z. Zhang, and A. J. Smola, "An efficient neighborhood-based interaction model for recommendation on heterogeneous graph," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 75–84.

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[13] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[14] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[16] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[17] K. Rakelly, A. Zhou, C. Finn, S. Levine, and D. Quillen, "Efficient off-policy meta-reinforcement learning via probabilistic context variables," in *International conference on machine learning*. PMLR, 2019, pp. 5331–5340.

[18] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," *ACM computing surveys (CSUR)*, vol. 54, no. 9, pp. 1–40, 2021.

[19] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," *arXiv preprint arXiv:1708.00489*, 2017.

[20] B. Settles, "Active learning literature survey," 2009.

[21] Y. Sun and J. Han, "Mining heterogeneous information networks: a structural analysis approach," *Acm Sigkdd Explorations Newsletter*, vol. 14, no. 2, pp. 20–28, 2013.

[22] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[23] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[24] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine learning*, vol. 3, no. 1, pp. 9–44, 1988.

[25] H. Tong, C. Faloutsos, and J.-Y. Pan, "Fast random walk with restart and its applications," in *Sixth international conference on data mining (ICDM'06)*. IEEE, 2006, pp. 613–622.

[26] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[27] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. S. Yu, "Heterogeneous graph attention network," in *The World Wide Web Conference*, 2019, pp. 2022–2032.

[28] D. B. West *et al.*, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.

[29] Y. Zhang, H. Tong, Y. Xia, Y. Zhu, Y. Chi, and L. Ying, "Batch active learning with graph neural networks via multi-agent deep reinforcement learning," in *AAAI*, 2022.

[30] M. Zitnik, F. Nguyen, B. Wang, J. Leskovec, A. Goldenberg, and M. M. Hoffman, "Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities," *Information Fusion*, vol. 50, pp. 71–91, 2019.


## Acknowledgment


This work is supported by NSF (IIS-1947135 and DMS-2134079), the NSF Program on Fairness in AI in collaboration with Amazon (1939725), DARPA (HR001121C0165), NIFA (2020-67021-32799), and ARO (W911NF2110088). The content of the information in this document does not necessarily reflect the position or the policy of the Government or Amazon, and no official endorsement should be inferred. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.