We thank all the reviewers for their constructive feedback. Below are the responses to each reviewer.

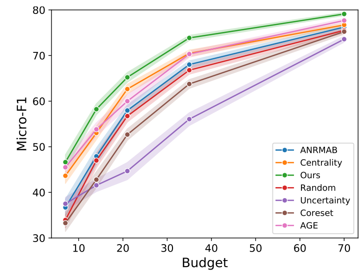

Fig. 1: Performance w.r.t query budget on Cora

Reviewer #1: (1) Number of labeled nodes to train the policy network. We use all the labeled nodes in the training graphs, as one could very easily find some fully labeled graphs to train the policy network. **(2) About ANRMAB.** Yes, theoretically you are right. However, to learn good weights of different heuristics for ANRMAB, at least a moderate number of labeled data are required. In our setting, we focus on very limited query budget, with which it is very difficult to learn good weights. **(3) Performance w.r.t. query budgets.** We agree that it is important to report the classification performance w.r.t. different query budgets in active learning. As an example, we have illustrated the corresponding curves on Reddit 4 in Section 4.4 (Paper). Here, we provide additional results in Fig. 1, where GPA is trained on Reddit $\{1, 2\}$ and evaluated on Cora. We observe similar trends to the results in Section 4.4 (Paper). **(4) Concerns on Table 1&2.** The purpose of Table 1&2 is to compare the performance of different active learning algorithms under the same query budgets of $(5\times\#\text{classes})$. We have compared classification performance w.r.t. different query budgets in Section 4.4 (Paper) and Fig. 1. **(5) Concerns on Section 4.4.** This is a very good point! Following your suggestion, we fix the test budget and change the training budget to see how the performance varies. Fig. 2 shows the results on the Reddit dataset, where graph 1&2 are used for training and graph 4 for testing. The x-axis of the figure corresponds to different query budgets on the training graphs. The results show that a training budget of 30 queries is sufficient to yield good performance, and more budgets will further yield more stable results with lower variance. For the effect of query budgets on classification performance, it has been discussed in the aforementioned answer (3). We will add more results on different graphs in the revised version.

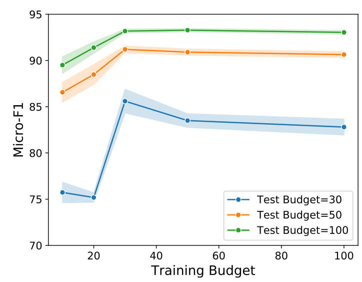

Fig. 2: Performance w.r.t. training budget

Reviewer #2: (1) About "Ignoring Long-term Performance". All the baseline methods except ANRMAB greedily choose the node with the maximal surrogate criterion score to label, which ignore the long-term performance. In contrast, our method uses reinforcement learning to label nodes with maximal long-term performance gain. In experiment, our method outperforms all the greedy methods, which proves our claim. **(2) Definition of the reward and evaluation metric.** Empirically, we use Micro-F_1 score of the classification GNN on test sets as the evaluation metric $\mathcal{M}$ to generate the reward signal R. **(3) Relation to prior work.** Most previous methods use different kinds of greedy strategies to identify informative nodes to label. In this paper we formulate the problem of active learning on graphs as a sequential decision process and propose to train an active learning policy network to maximize the long-term performance score on the end task. We will discuss this in more details in the revised version.

Reviewer #3: (1) Experiments on other tasks. We agree that it would be interesting to evaluate the proposed algorithm GPA on other tasks. Indeed, GPA is very general and can be easily applied to different tasks by changing the reward functions accordingly. Here we take node classification as an example, which is the most fundamental problem on graphs. **(2) Questions about zero-shot node classification.** This is a misunderstanding. Our paper actually focuses on “zero-shot transfer learning” instead of “zero-shot node classification”. “Zero-shot transfer learning” means that the *policy network* learned on labeled training graphs can be directly applied to the unlabeled test graphs without any further fine-tuning. In addition, the active learning setting mainly focuses on identifying informative nodes to label for supervised learning, which is different from unsupervised learning setting. **(3) Interaction between graph policy network and the one for node classification.** The graph policy network selects unlabeled nodes for annotation to train the node classification network. Meanwhile, the performance of the classification network is used as rewards to train the graph policy network. **(4) Transferring to other graphs.** This is a good point. Indeed, our algorithm is not sensitive to the number of classes between source and target graphs, because all the considered state features are not sensitive to the number of classes. Our algorithm parameterizes policy networks with GNNs, which naturally generalize to graphs with different topology. In experiment, we evaluate GPA on graphs with different numbers of classes and different topology, and show compelling results. **(5) Indices of state features.** The fourth state feature is defined as $\mathbf{s}_v^t(4)$ in the equation following L115 on page 3. **(6) Node class prediction probabilities.** Following existing literature, we apply a linear softmax classifier on top of the node representations learned by the classification GNN to get the node class probability. **(7) Evaluation metric for reward.** Empirically, we use Micro-F_1 as the evaluation metric for reward generation. **(8)Action space.** Remember that the action of the policy network is to select an unlabeled node and query for its label in each query step, and thus the action space is defined as the unlabeled nodes in the training set. **(9) Comparison with [5].** [5] considers batch-mode active learning on heterogeneous graphs, which cannot be directly applied to our setting. Also, the idea of [5] is very similar to ANRMAB, where the problem is both formulated with multi-armed bandit, and thus we mainly compare against ANRMAB in the paper.

Reviewer #4: We appreciate your positive feedback, and will revise the paper according to your suggestions.