Budget Allocation Exploiting Label Correlation between Instances

Adithya Kulkarni[1] **Mohna Chakraborty**[2] **Sihong Xie**[3] **Qi Li**[4]

[1,2,4]Department of Computer Science, Iowa State University, Ames, Iowa, USA
[3]AI Thrust, Information Hub, Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, Guangdong, China

Abstract

In this study, we introduce an innovative budget allocation method for graph instance annotation in crowdsourcing environments, where both the labels of instances and their correlations are unknown and need to be estimated simultaneously. We model the budget allocation task as a Markov Decision Process (MDP) and develop an optimization framework that minimizes the uncertainties associated with instance labeling and correlation estimation while adhering to budget constraints. To quantify uncertainty, we employ entropy and derive two strategies: OPTUENT-EXP and OPTUENT-OPT. Our reward function further considers the impact of a worker's label on the entire graph. We conducted extensive experiments using four real-world graph datasets, simulating worker labeling behavior to showcase the effectiveness of our approach. Experimental results demonstrate that our proposed approach can accurately estimate correlations between adjacent nodes while significantly reducing labeling costs. Moreover, across four real-world datasets, our proposed approach consistently outperforms existing baselines in moderate and high budget scenarios, highlighting its robustness and practical scalability.

1 INTRODUCTION

Hiring expert annotators to label a graph can be both time-consuming and costly. A more budget-friendly alternative is to engage non-expert crowd workers. Since these workers lack specialized expertise, it is often recommended to conduct multiple rounds of labeling with different workers to enhance the overall quality. However, compensating crowd workers for each label can quickly escalate costs, especially when repeated labeling is required for every instance.

When operating within a limited data labeling budget, leveraging instance correlations can significantly improve the selection of instances for crowd worker labeling. If two instances are correlated, labeling one can provide valuable insights into the other, allowing the labeling information to propagate across the graph. This means that instead of labeling every instance, it is possible to select a smaller subset to minimize costs strategically. However, a key challenge arises: instance correlations are typically unknown when the graph lacks annotations. Estimating these correlations while simultaneously identifying the optimal subset of instances for labeling is a complex task.

Previous research on budget allocation has largely neglected the intricate challenge of simultaneously estimating instance labels and their correlations within a graph. Most studies have treated instances as independent and identically distributed (i.i.d.), overlooking the potential correlations that exist among them [Frazier et al., 2008, Chen et al., 2013, Li et al., 2016]. While recent work by [Kulkarni et al., 2023] has attempted to address correlations between adjacent nodes, their approach is built on the assumption that these correlations are predetermined—a problematic stance, especially in non-homophily graphs. Moreover, the methodology presented by [Kulkarni et al., 2023] fails to extend naturally to estimating instance correlations, as workers cannot directly annotate edges, and the accuracy estimations used do not apply to edge labels. In contrast, our study introduces a dynamic method for real-time estimation of instance correlations by leveraging labels provided by workers for adjacent nodes. This innovative approach facilitates a more sophisticated allocation of labeling budgets, accounting for both the correlations among instances and the estimation of their labels. By addressing these complexities, we enhance the effectiveness and efficiency of the labeling process.

Our goal is to reduce the uncertainties surrounding both instance labeling and correlation estimation. Since worker-provided labels do not directly annotate instance correlations, traditional accuracy metrics used in previous studies [Kulkarni et al., 2023, Chen et al., 2013] are inadequate

for assessing annotation utility. Instead, we suggest focusing on measuring the uncertainty of labeling results. If a Graph Neural Network (GNN) model can accurately estimate the uncertainty of the graph, it can serve as a robust budget allocator. However, GNN models are known to struggle with the cold-start problem, and they require a sufficient budget to perform effectively [Wu et al., 2020]. Given our goal of leveraging label correlations to significantly reduce labeling costs, it is crucial to address these challenges. To that end, we adopt a Bayesian framework, formulating the budget allocation problem as an entropy optimization challenge. This approach aims to minimize uncertainty in both instance labeling and correlation estimation, ensuring that our strategies are not only effective but also cost-efficient.

To tackle this optimization problem, we decompose the expected uncertainty into a sum of *stage-wise rewards*, inspired by the technique from [Xie and Frazier, 2012]. Our innovative reward function captures the aggregated changes in uncertainty related to the labeling of all instances and the overall correlation estimation across the graph. The reward increases when a worker's label leads to a greater reduction in uncertainty, ensuring that our approach is both effective and efficient.

To effectively propagate labeling information throughout the graph, we first need to estimate the instance correlations for all edges. However, we face a challenge: the absence of worker labels makes it difficult to gauge these correlations. To address this issue, we leverage the intuition that adjacent instances with similar features are likely to exhibit similar correlations. We propose training a random forest regression model (RFR) [Breiman, 2001], using labeled pairs of adjacent instances to infer correlations for unlabeled pairs.

With the estimated instance correlations, we utilize belief propagation (BP) [Pearl, 2022] to disseminate labeling information across the graph. To achieve our objective of minimizing uncertainty, we introduce two strategic policies for selecting instances: OPTUENT-EXP, which prioritizes the instance with the highest expected reward, and OPTUENT-OPT, which focuses on the instance with the highest optimistic reward at each stage. The proposed approaches ensure a targeted and efficient allocation of resources for obtaining worker labels. Although our problem setting superficially resembles active learning, it diverges significantly in assumptions and goals. Unlike active learning, we assume access only to noisy, non-expert crowd workers, require repeated labeling to infer true label distributions, and jointly model uncertainty over both instance labels and their correlations. These distinctions render classical active learning methods unsuitable for our setting.

In summary, this paper makes several key contributions:

1. We are the first to estimate instance correlations between adjacent nodes and leverage these correlations to significantly reduce data labeling costs.
2. We introduce an entropy optimization framework that effectively models the uncertainties involved in both instance labeling and correlation estimation.
3. Our innovative reward function provides a comprehensive assessment of the aggregated uncertainty changes related to label estimation for instances and correlations across the entire graph.
4. We employ a random forest regression model to infer correlations for unlabeled pairs of adjacent nodes and utilize belief propagation to seamlessly disseminate labeling information throughout the graph.
5. Through extensive experiments on four real-world datasets, we empirically demonstrate the effectiveness of our proposed approach[1].

2 RELATED WORKS

The quest to minimize data annotation costs has sparked a wave of research aimed at creating innovative approaches and algorithms for crowdsourcing tasks.

A significant line of work focuses on optimizing instance selection strategies for querying worker labels, often under the assumption of a uniform labeling cost. Among these studies, [Zhou et al., 2014] explores non-sequential instance selection, employing aggregate regret to identify the top K arms with the highest expected rewards in a stochastic n-armed bandit framework. In contrast, several studies [Sheng et al., 2008, Li et al., 2016, Frazier et al., 2008, Chen et al., 2013, Raykar and Agrawal, 2014] focus on sequential instance selection with varying objectives. For instance, [Sheng et al., 2008] and [Li et al., 2016] aim to maximize the number of labeled instances while adhering to quality requirements and budget constraints. While [Sheng et al., 2008] assumes uniform data labeling quality across instances, [Li et al., 2016] posits that easier instances yield higher-quality labels. [Raykar and Agrawal, 2014] seeks to maximize a utility function that accounts for a pull market, where workers may choose to decline jobs from requesters. In a similar vein, [Frazier et al., 2008] and [Chen et al., 2013] aim to enhance labeling accuracy within budget limits. The former proposes a knowledge gradient policy for sequential instance selection, while the latter critiques this policy's consistency, introducing an optimistic variant that proves consistent under infinite budget scenarios.

However, these studies often treat instances as independent and identically distributed (i.i.d.), neglecting potential correlations between them. Related to our work, [Kulkarni et al., 2023] does consider instance correlations and strives to maximize overall labeling accuracy within budget constraints. Nonetheless, they rely on the assumption that these correlations are predetermined, a stance that may be problematic, particularly in non-homogeneous graphs. In this work,

[1] https://github.com/kulkarniadithya/OPTUENT

we make significant advancements by relaxing previous assumptions and dynamically estimating instance correlations as labels are obtained in real time. Given that workers do not provide instance correlations, we introduce a novel entropy-based objective function that minimizes uncertainty in both instance labeling and correlation estimation. Notably, we are the first to estimate instance correlations and leverage this information for budget allocation, effectively reducing data labeling costs. This innovative approach marks a key contribution to the field, enhancing both the accuracy and efficiency of online data labeling processes.

3 PRELIMINARIES

3.1 PROBLEM FORMULATION

Consider an unlabeled graph $G = (V, E)$ comprising N vertices $V = \{v_1, ..., v_N\}$ and M edges $E = \{e_1, ..., e_M\}$. Each vertex $v_i \in V$ represents an instance linked to a true label $l_i \in \{+1, -1\}$. The true label of each instance is characterized by $\theta_{v_i} = P(l_i = +1) \in [0, 1]$, while the correlation between instances is represented by $\omega_{e_k} = P(l_i = l_j) \in [0, 1]$, where $e_k = (v_i, v_j) \in E$. Notably, we do not consider correlations between vertices that lack a connecting edge. Following the framework of [Kulkarni et al., 2023], we assume that all workers are equally reliable, meaning the labels they provide for any vertex $v_i \in V$ at a given timestamp t (denoted by $y_{v_{i_t}}$) are drawn from the underlying label distribution: $y_{v_{i_t}} \sim Bernoulli(\theta_{v_i})$. While we acknowledge that real-world crowdsourcing scenarios can be more intricate than simply drawing worker labels from a Bernoulli distribution, previous studies [Chen et al., 2013, Li et al., 2016] suggest that this assumption is generally valid for real-world datasets.

Given a labeling budget of T where each worker label costs one unit, our goal is to minimize the uncertainty in the estimation of θ_{v_i} for every vertex $v_i \in V$ and ω_{e_k} for every edge $e_k \in E$.

3.2 INSTANCE SELECTION: KG AND OPTKG

The Knowledge Gradient (KG) [Frazier et al., 2008] and Optimistic Knowledge Gradient (OPTKG) [Chen et al., 2013] frameworks treat each instance as independent and identically distributed (i.i.d.), proposing strategies to select instances for label acquisition at each timestamp. Knowledge Gradient (KG) employs a single-step look-ahead approach that greedily identifies the next instance with the highest expected reward defined in Eq. (1)

$$v_t = \underset{v}{\operatorname{argmax}} \left(R(\mathbf{S^t}, v)\right), \text{where} \tag{1}$$
$$R(\mathbf{S^t}, v) \doteq p_1 * R_1(a_v^t, b_v^t) + p_2 * R_2(a_v^t, b_v^t),$$

In contrast, OPTKG selects the next instance based on an optimistic projection of the reward as shown in Eq. (2)

$$v_t = \underset{v}{\operatorname{argmax}} \left(R^+(\mathbf{S^t}, v)\right), \text{where} \tag{2}$$
$$R^+(\mathbf{S^t}, v) \doteq \max(R_1(a_v^t, b_v^t), R_2(a_v^t, b_v^t)).$$

In both Eq. (1) and Eq. (2), a_v^t and b_v^t denote the counts of positive and negative labels for vertex v at timestamp t. The posterior probabilities p_1 and p_2 are calculated as $p_1 = \frac{a_v^t}{a_v^t + b_v^t}$ and $p_2 = \frac{b_v^t}{a_v^t + b_v^t}$, respectively, representing the likelihoods of vertex v being labeled $+1$ or -1. Additionally, the rewards for obtaining labels $+1$ and -1 for vertex v are denoted as $R_1(a_v^t, b_v^t)$ and $R_2(a_v^t, b_v^t)$, respectively.

4 METHODOLOGY

This work addresses the budget allocation problem in instance graphs by leveraging correlations between adjacent nodes to optimize data labeling costs. We adopt a Bayesian framework to systematically reduce uncertainty in both instance labeling and correlation estimation (Section 4.1). Label propagation is formalized in Sections 4.2 and 4.3, while the budget allocation problem is reformulated as an entropy optimization framework to minimize uncertainty across both vertices and edges (Section 4.4). To solve this problem efficiently, we model it as a Markov Decision Process (MDP), enabling the decomposition of expected uncertainty into stage-wise rewards. A novel reward function is introduced to effectively estimate these rewards (Section 4.5), and we propose two efficient approximate policies, for instance selection, ensuring optimal label acquisition at each decision step (Section 4.6).

4.1 BAYESIAN SETUP

The input to our method is an unlabeled graph devoid of any information regarding the true labels of its vertices and edges. Following the approach of [Kulkarni et al., 2023], we initialize θ_{v_i} for each vertex $v_i \in V$ using a Beta prior distribution, specifically Beta(α, β). This initialization can be interpreted as assigning α positive and β negative pseudo-labels to each vertex v_i at the outset.

For each vertex, we define two key probabilities: the marginal probability and the posterior probability. The marginal probability is derived from the Beta initialization and the labels obtained from workers, while the posterior probability incorporates both the marginal probability and the labeling information propagated from neighboring vertices within the graph. This posterior probability is crucial for estimating θ_v for all $v \in V$.

As worker labels are obtained for a vertex v_i, its marginal probability is updated accordingly. In line with the Bayesian framework, we define the state matrix $\mathbf{S^t}$, an $N \times 2$ matrix

that represents the marginal probabilities of the vertices at timestamp t, where 2 corresponds to the two possible labels in our binary classification task. At each subsequent timestamp, the policy determines which vertex to select, and the obtained worker label prompts an update to the marginal probability of that vertex, resulting in a transition to a new state, $\mathbf{S^{t+1}}$.

We note that the new state $\mathbf{S^{t+1}}$ is fully determined by the current state $\mathbf{S^t}$, the selected vertex v_t at timestamp t, and the worker label y_{v_t} obtained for that vertex. This relationship establishes $\mathbf{S^t}$ as a Markovian process. Moreover, the marginal probability for the current vertex v_t is calculated as follows:

$$P_v^t(l = +1|\mathbf{S^t}, v_t) = \frac{\alpha + a_v^t}{\alpha + a_v^t + \beta + b_v^t}, \tag{3}$$

where a_v^t and b_v^t represent the counts of positive and negative worker labels received for vertex v up to timestamp t. Additionally, we have $P_v^t(l = -1|\mathbf{S^t}, v_t) = 1 - P_v^t(l = +1|\mathbf{S^t}, v_t)$. As worker labeling information propagates through the graph, the posterior probabilities for all vertices are updated at each timestamp, reflecting the latest insights gained from the labeling process.

The labeling process described above allows us to establish a filtration $\{\mathcal{F}_t\}_{t=0}^{T-1}$, where $\mathcal{F}_t$ is the σ-algebra generated by the sample path $(v_0, y_{v_0}, ..., v_{t-1}, y_{v_{T-1}})$. In this context, v_t represents any vertex selected from V at timestamp t, and y_{v_t} denotes the corresponding worker label obtained. This filtration implies that the choice of vertex at timestamp t can be fully informed by the historical labeling outcomes up to timestamp $t-1$. Consequently, v_t is $\mathcal{F}_t$-measurable, leading us to define the budget allocation policy as a sequence of vertex selections at each timestamp: $\pi = (v_0, ..., v_{T-1})$.

4.2 INSTANCE CORRELATION ESTIMATION

We utilize Belief Propagation (BP) [Pearl, 2022], a powerful message-passing algorithm, to disseminate labeling information throughout the graph. To implement BP, we first transform the input graph G into a bipartite factor graph FG. This conversion involves adding a factor vertex for each edge $e_k = (v_i, v_j) \in E$, which connects to the vertices v_i and v_j via undirected edges. The resulting factor graph is denoted as $FG = (V \cup F, E')$, where $|E'| = 2|E|$.

Each factor vertex is associated with a function $\phi_{\mathbf{e_k}}$ that specifies the proportion of information to be propagated between vertices v_i and v_j, represented as: $\phi_{\mathbf{e_k}} = \begin{bmatrix} \omega_{e_k}(+1) & \omega_{e_k}(-1) \\ \omega_{e_k}(-1) & \omega_{e_k}(+1) \end{bmatrix}$. To streamline our notation, we will use e_k to refer to the factor vertex associated with edge e_k throughout the remainder of this paper. This simplification enhances clarity while maintaining precision in our discussions. Since the values of ω_{e_k} are initially unknown, we propose to estimate them using the marginal probabilities of the connected vertices. For the edge $e_k = (v_i, v_j) \in E$, we compute the marginal probability of e_k at timestamp t as follows:

$$P_{e_k}^t(+1) = P_{v_i}^t(+1) \times P_{v_j}^t(+1) + P_{v_i}^t(-1) \times P_{v_j}^t(-1), \tag{4}$$

where $P_{v_i}^t$ and $P_{v_j}^t$ represent the marginal probabilities of vertices v_i and v_j at timestamp t, respectively.

An edge is considered labeled if both of its end vertices have received at least one worker label. However, in the early stages of the labeling process, most vertices remain unlabeled, necessitating a method to estimate the marginal probabilities of these unlabeled edges. To achieve this, we employ a Random Forest Regression (RFR) model [Breiman, 2001].

While alternative models, such as neural networks, could also be considered for estimating edge potential, we have found that Random Forest Regression is particularly well-suited to our requirements. It trains quickly, delivers robust performance even with a limited number of labeled instances, and eliminates the need for extensive calibration. Given these advantages, RFR strikes an optimal balance between efficiency and predictive accuracy for our specific application.

Our underlying intuition is that edges connecting nodes with similar attribute vectors are likely to exhibit similar marginal probabilities. To train the model, we concatenate the features of the end vertices v_i and v_j for labeled edges, using this combined feature set as input while treating the marginal probabilities computed according to Eq. (4) as the target for regression. Once trained, the model is then deployed to estimate the marginal probabilities for the unlabeled edges in the graph, enhancing our labeling process significantly.

The calculated marginal probabilities serve as estimates for ω_{e_k}, allowing us to update $\phi_{\mathbf{e_k}}$ as follows: $\phi_{\mathbf{e_k}} = \begin{bmatrix} P_{e_k}^t(+1) & P_{e_k}^t(-1) \\ P_{e_k}^t(-1) & P_{e_k}^t(+1) \end{bmatrix}$. With this update in place, we utilize the constructed factor graph FG to effectively propagate labeling information throughout the entire graph, ensuring that insights gained from labeled edges are shared with their neighbors.

4.3 LABELING INFORMATION PROPAGATION

In the factor graph FG, labeling information is effectively propagated through the exchange of messages between variable vertices and factor vertices. The computation of the message from a variable vertex to a factor vertex is carried out as follows:

$$\underset{v \to f}{\mu}(x_v) = \prod_{f^* \in \mathcal{N}(v) \setminus \{f\}} \underset{f^* \to v}{\mu}(x_v), \tag{5}$$

and the message from the factor vertex to the variable vertex is computed as follows:

$$\mu_{f\to v}(x_v) = \sum_{\substack{x'_f = x_v \\ x'_{v^*} = x_v}} \phi_f(x'_f) \prod_{v^* \in \mathcal{N}(f)\setminus\{v\}} \mu_{v^*\to f}(x'_{v^*}). \tag{6}$$

Here, $x_v \in \{+1, -1\}$ denotes the labeling space for the variable vertex $v \in V$. Furthermore, $\mathcal{N}(v)$ and $\mathcal{N}(f)$ indicate the sets of neighboring vertices for the variable vertex v and factor vertex f, respectively.

At each timestamp, messages are transmitted from the leaf vertices in the graph to a chosen vertex (*forward propagation*), and then from this chosen vertex back to the leaf vertices (*backward propagation*). Each message from a vertex $v \in V$ is initially set to its marginal probability. Following the updates defined in Eq. (5) and Eq. (6), the messages for all internal vertices are also refined. This belief propagation process is iterated multiple times until convergence is achieved, with messages being normalized at each step to prevent underflow. Ultimately, the posterior probability for each variable vertex $v \in V$ at timestamp t is calculated as follows:

$$P_v^t(+1) \propto \frac{\alpha + a_v^t}{\alpha + a_v^t + \beta + b_v^t} \prod_{j \in \mathcal{N}(v)} \underset{j\to v}{\mu}(+1). \tag{7}$$

We can observe that the updates to the posterior probabilities of the vertices are entirely governed by the chosen vertex and the corresponding worker label received.

4.4 OBJECTIVE FUNCTION

Our objective is to minimize the uncertainty associated with instance labeling and instance correlation estimation by the end of the budget at timestamp T. The entropy of the posterior probabilities for vertices and the marginal probabilities for edges serves as a measure of labeling uncertainty. Consequently, we formulate our objective function to minimize the expected entropy of the labeling for both vertices and edges in the graph, conditioned on $\mathcal{F}_t$:

$$\mathcal{H}_T = \text{argmin}\ \mathbb{E}\left(H^T(V) + H^T(E)\right), \tag{8}$$

Here, $H^T(V) = -\sum_v \sum_x P_v^T(x) \log P_v^T(x)$ and $H^T(E) = -\sum_e \sum_y P_e^T(y) \log P_e^T(y)$ are the entropy of vertices and edges in the graph. P_v^T, P_e^T are the posterior probability of vertex $v \in V$ and marginal probability of edge $e \in E$ at the end of budget T, respectively. This formulation effectively captures the uncertainty associated with labeling for both vertices and edges in the graph.

The objective is to identify a policy that minimizes the value function for the objective defined in Eq. (8) by the end of the budget T. Any policy π that successfully minimizes Eq. (9) is considered the optimal policy, denoted as π^*.

$$V(S^T) \doteq \arg\min_{\pi}\ \mathbb{E}^{\pi}\left[\mathbb{E}\left(H^T(V) + H^T(E)\right)\right]. \tag{9}$$

where $V(S^T)$ denotes the value function at the conclusion of budget T, while π represents the policy responsible for selecting instances to obtain worker labels at each timestamp. Additionally, $\mathbb{E}^{\pi}$ signifies the expectation calculated over the sample paths $(v_0, y_{v_0}, ..., v_{t-1}, y_{v_{T-1}})$ generated by the policy π.

4.5 REWARD FUNCTION

We approach the task of identifying the optimal policy π^* for the value function defined in Eq. (9) by framing it as a Markov Decision Process (MDP). The final expected uncertainty is influenced by the selection of instances at each timestamp. To address this, we decompose the final expected uncertainty into a sum of *stage-wise rewards*, utilizing the methodology outlined in [Xie and Frazier, 2012]. While [Xie and Frazier, 2012] primarily addresses an *infinite-horizon* problem that focuses on optimizing stopping times, [Chen et al., 2013] has demonstrated that this technique is also applicable to *finite-horizon* scenarios.

Given that the value function accounts for the total entropy of the vertices and edges within the graph, we define the reward function as the change in this total entropy between two timestamps. A higher reward signifies a greater reduction in uncertainty regarding the labeling of the graph's vertices and edges.

Proposition 1 *The stage-wise expected reward between two timestamps t and $t+1$ is defined as:*

$$\begin{aligned} R(S^t, v_t) =& \mathbb{E}((H^t(V) + H^t(E)) \\ &- (H^{t+1}(V) + H^{t+1}(E))|\mathbf{S^t}, v_t), \end{aligned} \tag{10}$$

then the value function in Eq. (9) becomes:

$$V(S^T) = V(S^0) - \sup_{\pi} \mathbb{E}^{\pi}\left(\sum_{t=0}^{T-1} R(\mathbf{S^t}, v_t)\right). \tag{11}$$

Any policy π that attains the supremum for Eq. (11) is the optimal policy π^*. Here, $V(S^0) = H^0(V) + H^0(E)$. We provide the derivation of Proposition 1 in Appendix B.

Proposition 1 is instrumental in formulating the minimization problem in Eq. (9) as a T-stage Markov Decision Process (MDP) and transforming it into a maximization problem aimed at maximizing the expected reward, as demonstrated in Eq. (11). Since the marginal probability of the edges is derived from the worker labels obtained for the vertices, the T-stage MDP is contingent solely on the state of the vertices at each timestamp. Thus, the T-stage MDP is represented by the tuple $\{T, \{\mathcal{S}^t\}, \mathcal{A}, \mathcal{P}^t, R(\mathbf{S^t}, v_t)\}$. In

this tuple, T signifies the budget, which corresponds to the number of worker labels we can acquire; $\mathcal{S}^t$, the state space at stage t, encompasses all possible states reachable at that stage; $\mathcal{A} = \{1, 2, ..., N\}$ denotes the action space, representing the set of instances eligible for labeling next; $\mathcal{P}^t = \{P_1^t, P_2^t, ..., P_N^t\}$ comprises the posterior probabilities at timestamp t for each vertex $v_i \in V$; and $R(\mathbf{S^t}, v_t)$ is the expected reward defined in Eq. (10). Once a label y_{v_t} is obtained for vertex v at timestamp t, the marginal probability of vertex $v_i \in V$ will be updated accordingly. Therefore, we have

$$\mathcal{S}^t = \left\{ \{p_{1_v}^t, p_{2_v}^t\}_{v=1}^N : p_{1_v}^t, p_{2_v}^t \in [0, 1], p_{1_v}^t + p_{2_v}^t = 1 \right\}. \tag{12}$$

The posterior probabilities of multiple vertices can change as a result of the labeling information propagated from the chosen vertex and the obtained worker label. Additionally, the marginal probabilities of edges may also be affected. Importantly, all these updates are entirely dictated by the selected vertex and the corresponding worker label. Consequently, leveraging the Markovian property of $\{\mathbf{S^t}\}$, it is adequate to consider a Markovian policy [Powell, 2007], where the choice of v_t is made solely based on the current state S^t.

4.6 EFFICIENT APPROXIMATE POLICY

Finding the optimal policy for the value function in Eq. (9) is non-trivial. Therefore, we propose efficient approximate policies designed to select instances that maximize the reward for obtaining worker labels at each timestamp. These approximate policies aim to achieve the supremum of the value function defined in Eq. (11) within the framework of a T-stage Markov Decision Process (MDP). At any state $\mathbf{S^t}$ at timestamp t, when a vertex $v \in V$ is chosen for a worker label, the worker can provide either a label of $+1$ or -1. Consequently, the policies must account for both outcomes when calculating the expected reward. Let $R_1(\mathbf{S^t}, v_t)$, $R_2(\mathbf{S^t}, v_t)$ represent the rewards for obtaining labels $+1$ and -1, respectively. The expected reward can then be expressed as:

$$R(\mathbf{S^t}, v_t) = p_1 R_1(\mathbf{S^t}, v_t) + p_2 R_2(\mathbf{S^t}, v_t), \tag{13}$$

where $p_1 = \frac{\alpha + a_v^t}{\alpha + a_v^t + \beta + b_v^t}$ and $p_2 = \frac{\beta + b_v^t}{\alpha + a_v^t + \beta + b_v^t}$ are marginal probabilities of v at timestamp t. The optimistic reward can be expressed as:

$$R^+(\mathbf{S^t}, v_t) = \max(R_1(\mathbf{S^t}, v_t), R_2(\mathbf{S^t}, v_t)). \tag{14}$$

The first proposed approximate policy, OPTUENT-EXP, selects the instance that offers the highest expected reward at each timestamp, denoted as $\hat{\pi} = (v_0, ..., v_{T-1})$. This strategic choice maximizes the potential benefit of obtaining worker labels, ensuring optimal use of resources throughout the process.

$$v_t = \arg\max_v \left(R(\mathbf{S^t}, v) \doteq p_1 R_1(\mathbf{S^t}, v) + p_2 R_2(\mathbf{S^t}, v) \right). \tag{15}$$

The second proposed approximate policy, OPTUENT-OPT, selects the instance with the highest optimistic reward at each timestamp, represented as $\pi^o = (v_0, ..., v_{T-1})$. This approach strategically prioritizes instances that promise the greatest potential benefits, thereby optimizing the acquisition of worker labels and enhancing the overall effectiveness of the labeling process.

$$v_t = \operatorname*{argmax}_v \left(R^+(\mathbf{S^t}, v) \doteq \max(R_1(\mathbf{S^t}, v), R_2(\mathbf{S^t}, v)) \right). \tag{16}$$

By utilizing Eq. (15) or Eq. (16), we can effectively determine the optimal vertex to target for obtaining the worker label at each timestamp $0 \leq t < T$. This strategic selection process ensures that we maximize the value of our labeling efforts at every stage. The complete procedure is outlined in Algorithm 1.

4.7 PROPOSED POLICIES ARE CONSISTENT

To demonstrate the consistency of the proposed policies, we must show that as the budget T approaches infinity, the sum of entropy for the vertices and edges in the graph converges to a constant value. This constant is defined by the true label of each instance θ_{v_i} for $v_i \in V$ and the instance correlation ω_{e_k} for every edge $e_k \in E$. Thus, as T goes to infinity, each vertex should receive an infinite number of labels, ensuring that the estimated θ_{v_i} aligns with the true label, and the estimated ω_{e_k} converges to its true value.

To establish consistency, we first demonstrate in Appendix D.2 that the random forest regressor achieves over 95% accuracy with a small budget of 40 on the large Cora and Pubmed datasets, indicating rapid convergence. This observation leads us to conclude that as T goes to infinity, changes in edge uncertainty become negligible, allowing us to focus solely on the entropy of vertex labeling. We show that the posterior probability for each vertex $v_i \in V$ is updated based on its marginal probability and that of the leaf vertices in the factor graph FG. The proposed reward function in Eq. 10 is proportional to the change in the marginal probability of the chosen vertex v_t, ensuring that both OPTUENT-EXP and OPTUENT-OPT label each vertex infinitely many times as the budget increases. Given that we assume all workers are equally reliable, this leads to convergence on θ_{v_i} for each $v_i \in V$ and ω_{e_k} for every edge $e_k \in E$. Consequently, the sum of entropy for the vertices and edges converges to a constant value, demonstrating that the proposed policies $\hat{\pi}$ and π^o are consistent. A detailed proof of this consistency is provided in Appendix C.

Input: Graph $G = (V, E)$; Budget T; Beta prior parameters α, β;
Policy $\pi \in \{\text{OPTUENT-EXP}, \text{OPTUENT-OPT}\}$
Output: Posterior label estimates $\{\theta_{v_i}\}_{v_i \in V}$; Edge correlation estimates $\{\omega_{e_k}\}_{e_k \in E}$
Initialize marginal probabilities $\theta_{v_i} \sim \text{Beta}(\alpha, \beta)$ for all $v_i \in V$;
Initialize labeled set $\mathcal{L} \leftarrow \emptyset$;
for $t = 1$ **to** T **do**
  foreach $v_i \in V$ **do**
    Compute marginal edge probabilities using vertex marginals via Eq. 4;
    Estimate edge correlations $\{\omega_{e_k}\}$ for unlabeled edges using Random Forest Regression trained on labeled edge features and Eq. 4;
    Compute posterior probabilities using current marginals and Belief Propagation (Eq. 7);
    Compute rewards $R_1(S^t, v_i), R_2(S^t, v_i)$ for label outcomes $+1$ and -1;
    if $\pi =$ *OPTUENT-EXP* **then**
      Compute expected reward $R(S^t, v_i)$ using Eq. 13;
    else
      Compute optimistic reward $R^+(S^t, v_i)$ using Eq. 14;
    end
  end
  Select $v_t = \arg\max_{v_i \in V} R(S^t, v_i)$ or $R^+(S^t, v_i)$ depending on π;
  Query label $y_{v_t} \sim \text{Bernoulli}(\theta_{v_t})$;
  Update counts (a_v^t, b_v^t) and re-run Belief Propagation to update posterior;
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{y_{v_t}\}$;
end
return Posterior label estimates $\{\theta_{v_i}\}_{v_i \in V}$ and edge correlation estimates $\{\omega_{e_k}\}_{e_k \in E}$

Algorithm 1: Uncertainty-Guided Budget Allocation for Graph Labeling

5 EXPERIMENTS

This section critically assesses our proposed policies, OPTUENT-EXP and OPTUENT-OPT, which strategically select the next vertex to label based on the frameworks established in Eq. (15) and Eq. (16), respectively. This evaluation highlights the effectiveness and robustness of our approaches in optimizing the labeling process.

5.1 DATASET AND EVALUATION METRICS

The performance of our proposed policies is assessed across four distinct graph datasets. Three of these datasets, Cora, Citeseer, and Pubmed [Bojchevski and Günnemann, 2017], are well-established citation networks, while WebKB [Craven et al., 1998] comprises web pages from various computer science departments at universities. We adhere to the guidelines outlined by [Kulkarni et al., 2023] to transform these datasets into binary-class formats. The statistics of the datasets are provided in Table 1. To evaluate the effectiveness of instance labeling, we utilize *accuracy* as our performance metric.

Table 1: Statistics of the Datasets

Dataset	#Vertex	#Pos	#Neg
Cora	2708	1296	1412
Citeseer	3312	1618	1694
Pubmed	19717	7875	11842
WebKB	877	415	462

5.2 EXPERIMENTAL SETTINGS

Our goal is to leverage label correlations to substantially reduce labeling costs. To showcase the effectiveness of our proposed strategies, we specifically concentrate our experiments on low-budget scenarios. This focus highlights the potential of our methods to deliver impactful results even in resource-constrained environments.

We simulate worker labeling behavior across the four datasets: Cora, Citeseer, Pubmed, and WebKB. To begin, we establish the parameter θ_{v_i} for all $v_i \in V$, after which worker labels are generated according to the distribution $y_{v_i} \sim Bernoulli(\theta_{v_i})$. We perform experiments under two distinct settings: one with fixed values of θ_{v_i} set at 0.65, 0.7, 0.75, 0.8, and 0.85 for all $v_i \in V$, and the other with θ_{v_i} sampled from a uniform distribution $\mathcal{U}(0.7, 0.85)$. Due to space constraints, this paper primarily presents results for the fixed setting of $\theta_{v_i} = 0.65$ and the uniform sampling from $\mathcal{U}(0.7, 0.85)$. A detailed discussion of the results for the other fixed values of θ_{v_i} (0.7, 0.75, 0.8, and 0.85) can be found in Appendix F.

At each timestamp, we train a new random forest regression (RFR) model, continuously updating the training data with each newly acquired label from crowd workers. We rigorously evaluate the performance of the RFR model, with detailed results presented in Appendix D.2. According to Eq. (15) and Eq. (16), our goal is to compute the reward for all vertices in the graph and select the one with the highest reward. Given the computational cost of belief propagation, we follow the strategy in [Kulkarni et al., 2023] by uniformly sampling 10 candidate vertices at each timestamp to compute rewards and select the optimal vertex. As shown in Appendix D.1, increasing the sample size beyond 10 offers minimal accuracy gains, confirming that small candidate sets are sufficient for robust performance. We conduct experiments using three random seed values, 11, 42, and 111, and report the mean results for clarity and robustness, alongside

standard deviations presented in Figure 8 in the Appendix. All experiments are performed on a single Nvidia GeForce RTX 3060 GPU, ensuring efficient computation and reliable performance assessments.

5.3 BASELINE METHODS

The reward functions of both OPTKG [Chen et al., 2013] and KG [Frazier et al., 2008] indicate that annotating an unlabeled node is always preferable to annotating a node with one label, which in turn is better than annotating a node with two labels, regardless of the specific labels obtained. Once all nodes receive two labels, OPTKG and KG adopt different selection strategies. As shown in [Chen et al., 2013], these differences emerge when the budget is sufficiently large; for instance, in a simulation with 50 instances, the methods diverge when the budget reaches 3K, which is 60 times the number of instances. However, since the budget for all our experiments is lower than two times the number of instances, both policies behave similarly. Thus, we focus solely on comparing our proposed policies with OPTKG. Overall, our comparisons include the following policies:

1. *Uniform:* This policy randomly samples one vertex from V at each timestamp to obtain worker labels.
2. *OPTKG*: The Optimistic Knowledge Gradient policy [Chen et al., 2013] treats each instance as independent and identically distributed (i.i.d.) and selects the instance with the highest optimistic reward at each timestamp. The reward is defined as the change in the marginal probabilities of vertices between two timestamps, reflecting a proactive approach to label acquisition.
3. *GraphOBA-EXP*: As defined by [Kulkarni et al., 2023], this policy calculates the reward based on the change in the sum of posterior probabilities of vertices in the graph between two timestamps. GraphOBA-EXP selects vertices that maximize expected rewards at each timestamp, relying on belief propagation to effectively disseminate labeling information throughout the graph.
4. *GraphOBA-OPT*: GraphOBA-OPT, also proposed by [Kulkarni et al., 2023], chooses the next vertex based on the optimistic expected reward. Like GraphOBA-EXP, it incorporates belief propagation as a critical component for an effective labeling strategy.

We evaluate our proposed policies against baseline policies across three scenarios: (1) **Without BP and RF:** In this scenario, we compare the Uniform and OPTKG policies directly with our proposed methods, providing a clear baseline without any enhancements (2) **With BP and Without RFR:** In this scenario, we utilize belief propagation (BP) to disseminate labeling information for the Uniform and OPTKG policies, allowing us to assess the impact of BP without the influence of Random Forest Regression (RFR), (3) **With BP and RFR:** In this scenario, we incorporate RFR alongside BP for both the Uniform and OPTKG policies, while applying RFR for the GraphOBA-EXP and GraphOBA-OPT policies. This setup represents a comprehensive evaluation of how our methods perform with the full capabilities of BP and RFR.

Due to space constraints, we present the findings for scenario 3 in the main paper, while the results for scenarios 1 and 2 are detailed in Appendix E. This structure allows us to clearly delineate the effectiveness of our proposed approaches across varying conditions.

5.4 RESULTS AND DISCUSSION

In low-budget settings, all methods exhibit higher variance due to limited initial information, a known challenge in MDP-based sequential decision making. However, our entropy-based selection maintains greater stability in label acquisition compared to greedy baselines, as evidenced by standard deviation plots (Appendix Figure 8). Figure 1 presents a compelling comparison of the proposed policies, OPTUENT-OPT and OPTUENT-EXP, against baseline methods in scenario 3 across the WebKB, Cora, Citeseer, and Pubmed datasets. The analysis includes two settings for θ_v: one fixed at 0.65 and the other sampled from a uniform distribution $\mathcal{U}(0.7, 0.85)$, for $v \in V$. For brevity, we omit the subscript i from the vertices. The results reveal that the Uniform policy, which samples vertices randomly, and the OPTKG policy, which selects vertices in a round-robin manner, perform the weakest among the baseline approaches. In contrast, the GraphOBA-OPT and GraphOBA-EXP policies, which leverage posterior probabilities for vertex selection, demonstrate improved performance. However, our proposed policies, which take into account both the posterior probabilities of vertices and the marginal probabilities of edges, surpass all baseline methods. These findings underscore the importance of accurately estimating instance correlations, which can vary from edge to edge. By effectively capturing these dynamics, our approach significantly reduces data labeling costs, highlighting its practical advantages in real-world applications.

In the setting where θ_v is fixed at 0.65, individual workers can theoretically achieve an accuracy of 0.65 after repeated labeling. In contrast, when θ_v is sampled from a uniform distribution $\mathcal{U}(0.7, 0.85)$, the expected accuracy rises to approximately 0.77. However, due to the limited budget in our experiments, only a small number of vertices can undergo repeated labeling. Despite this constraint, the results demonstrate that our proposed policies consistently outperform individual workers by a substantial margin. Moreover, the baseline methods also show improved performance over individual workers, benefiting from the propagation of labeling information through belief propagation (BP). This effec-

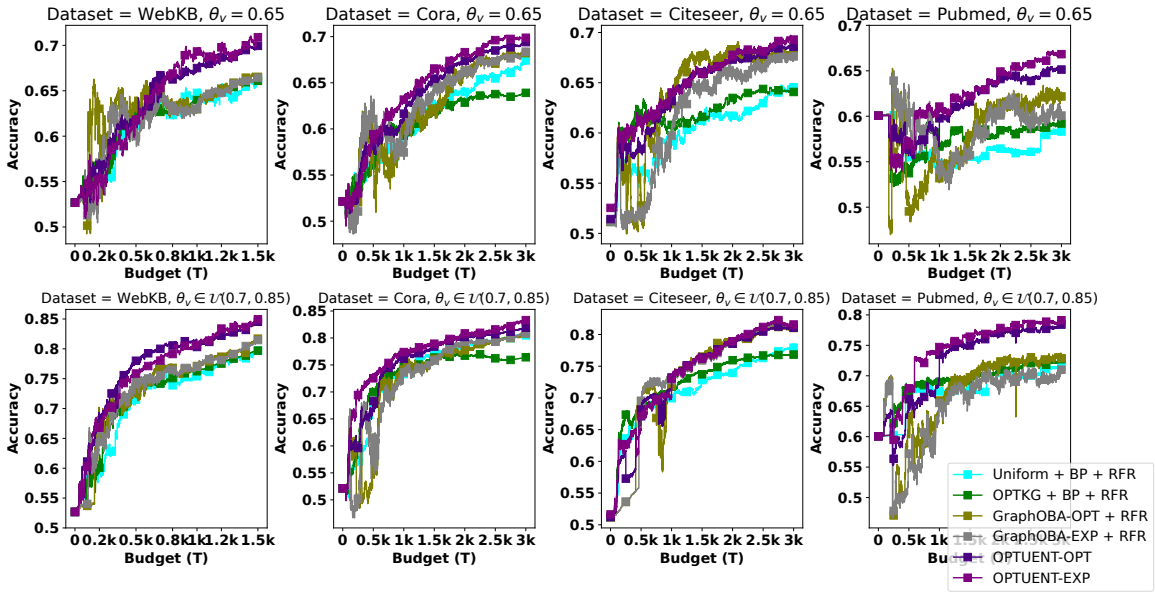

Figure 1: Performance comparison on four graph datasets. The top four plots show the performance comparison of OPTUENT-OPT and OPTUENT-EXP with the baselines following scenario 3 for a fixed $\theta_v = 0.65$, and the bottom four plots show the performance comparison for θ_v sampled from the uniform distribution $\mathcal{U}(0.7, 0.85)$.

tively enhances the labeling process by providing additional context for the labels. This highlights the significant advantages of our approach in leveraging both policy strategies and information propagation to maximize labeling accuracy.

When evaluating performance stability, it's clear that our proposed policies demonstrate greater consistency compared to the baselines. This suggests that the vertices selected by our policies are adept at managing the inherent uncertainty in worker-provided labels. Importantly, our reward function estimation accounts for the potential outcomes of workers delivering labels of $+1$ or -1, with the actual reward ultimately contingent on the label received. By factoring in the influence of worker labels on both the vertices and edges of the graph, our policies achieve a more robust reward computation. This comprehensive approach enhances the resilience of the reward mechanism against uncertainties in worker labeling, further solidifying the effectiveness of our strategies in dynamic labeling environments.

6 CONCLUSION

In this study, we tackle the budget allocation problem as an optimization challenge aimed at minimizing the expected uncertainty surrounding instance labeling and correlation estimation. Leveraging a Markov Decision Process (MDP) framework, we break down the final expected uncertainty into stage-wise rewards that quantify the change in entropy for all vertices and edges across two timestamps. We employ a Random Forest Regression model to estimate the marginal probabilities of edges representing instance correlations, while belief propagation is utilized to disseminate labeling information throughout the graph. We introduce two approximate policies: OPTUENT-EXP, which selects the instance with the highest expected reward, and OPTUENT-OPT, which targets the highest optimistic reward at each timestamp. Our empirical results show that the proposed approaches accurately estimate correlations between adjacent nodes and substantially reduce labeling costs. These findings underscore the value of uncertainty-guided decision-making under tight budget constraints and its potential to generalize to large-scale, real-world graph labeling tasks.

7 ACKNOWLEDGEMENTS

Adithya, Mohna, and Qi were supported in part by the National Science Foundation under NSF grant IIS-2007941. Sihong Xie was supported by the Department of Science and Technology of Guangdong Province (Grant No. 2023CX10X079), the National Key R&D Program of China (Grant No. 2023YFF0725001), the Guangzhou-HKUST(GZ) Joint Funding Program (Grant No. 2023A03J0008), and the Education Bureau of Guangzhou Municipality.

References

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

Xi Chen, Qihang Lin, and Dengyong Zhou. Optimistic knowledge gradient policy for optimal budget allocation in crowdsourcing. In *International conference on machine learning*, pages 64–72. PMLR, 2013.

Mark Craven, Andrew McCallum, Dan PiPasquo, Tom Mitchell, and Dayne Freitag. Learning to extract symbolic knowledge from the world wide web. Technical report, Carnegie-mellon univ pittsburgh pa school of computer Science, 1998.

Dhivya Eswaran, Stephan Günnemann, Christos Faloutsos, Disha Makhija, and Mohit Kumar. Zoobp: Belief propagation for heterogeneous networks. *Proceedings of the VLDB Endowment*, 10(5):625–636, 2017.

Peter I Frazier, Warren B Powell, and Savas Dayanik. A knowledge-gradient policy for sequential information collection. *SIAM Journal on Control and Optimization*, 47 (5):2410–2439, 2008.

Adithya Kulkarni, Mohna Chakraborty, Sihong Xie, and Qi Li. Optimal budget allocation for crowdsourcing labels for graphs. In Robin J. Evans and Ilya Shpitser, editors, *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pages 1154–1163. PMLR, 31 Jul–04 Aug 2023. URL `https://proceedings.mlr.press/v216/kulkarni23a.html`.

Qi Li, Fenglong Ma, Jing Gao, Lu Su, and Christopher J Quinn. Crowdsourcing high quality labels with a tight budget. In *Proceedings of the ninth acm international conference on web search and data mining*, pages 237–246, 2016.

Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 129–138. 2022.

Warren B Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

Vikas Raykar and Priyanka Agrawal. Sequential crowdsourced labeling as an epsilon-greedy exploration in a markov decision process. In *Artificial intelligence and statistics*, pages 832–840. PMLR, 2014.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622, 2008.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Jing Xie and Peter I Frazier. Sequential bayes-optimal policies for multiple comparions with a control. Technical report, Technical report, Cornell University, 2012.

Yuan Zhou, Xi Chen, and Jian Li. Optimal pac multiple arm identification with applications to crowdsourcing. In *International Conference on Machine Learning*, pages 217–225. PMLR, 2014.

A APPENDIX

In the Appendix, we present a detailed derivation for the proof of Proposition 1 and proof that the proposed policies are consistent, ensuring clarity and rigor in our methodology. Additionally, we include a comprehensive discussion of the results from further experiments, offering valuable insights that reinforce our findings.

B PROOF OF PROPOSITION 1

From Eq. (9), we obtain the following value function.

$$V(S^T) \doteq \underset{\pi}{argmin}\, \mathbb{E}^{\pi}\left[\mathbb{E}\left(H^T(V) + H^T(E)\right)\right].$$

We define $V(S^0) = H^0(V) + H^0(E)$. From Eq. (10), the reward function is defined as

$$\begin{aligned} R(S^t, v_t) =& \mathbb{E}((H^t(V) + H^t(E)) \\ &- (H^{t+1}(V) + H^{t+1}(E))|\mathbf{S^t}, v_t). \end{aligned}$$

Substituting the value of $t = 0$, we get

$$R(S^0, v_0) = \mathbb{E}((H^0(V) + H^0(E)) - (H^1(V) + H^1(E))),$$

and substituting the value of $t = 1$, we get

$$R(S^1, v_1) = \mathbb{E}\left[\left(H^1(V) + H^1(E)\right) - \left(H^2(V) + H^2(E)\right)\right]. \tag{17}$$

We can observe that the first term in $R(S^1, v_1)$ and the second term in $R(S^0, v_0)$ get canceled out if we add the rewards for these two timestamps. Therefore, we get

$$\begin{aligned} \sum_{t=0}^{T-1} R(S^t, v_t) =& \mathbb{E}(\left(H^0(V) + H^0(E)\right) \\ &- \left(H^T(V) + H^T(E)\right)) \end{aligned}$$

Substituting $V(S^T)$ and $V(S^0)$ into the equation, we get

$$\sup_{\pi} \mathbb{E}^{\pi}\left(\sum_{t=0}^{T-1} R(S^t, v_t)\right) = V(S^0) - V(S^T).$$

Therefore,

$$V(S^T) = V(S^0) - \sup_{\pi} \mathbb{E}^{\pi}\left(\sum_{t=0}^{T-1} R(S^t, v_t)\right).$$

C PROPOSED POLICIES ARE CONSISTENT

In OPTUENT-OPT, we select the vertex v_t in each iteration as follows:

$$v_t = \operatorname*{argmax}_{v} \left(R^+(\mathbf{S^t}, v) \doteq \max(R_1(\mathbf{S^t}, v), R_2(\mathbf{S^t}, v))\right).$$

The expected reward $R^+(\mathbf{S^t}, v_t)$ depends solely on changes to the marginal probability of vertex v_t due to the obtained label. Since the reward, as specified in Eq. (10), considers the change in entropy of vertices and edges between two timestamps, we have:

$$\begin{aligned} R(S^t, v_t) =& \mathbb{E}((H^t(V) + H^t(E)) \\ &- (H^{t+1}(V) + H^{t+1}(E))|\mathbf{S^t}, v_t), \end{aligned}$$

Since the marginal probability of each vertex is updated based only on its own posterior probability and those of the leaf vertices in the factor graph, as $T \to \infty$, changes in edge entropy become negligible, as evidenced by the empirical experiments in section D.2. Therefore, we focus solely on the entropy of vertex labeling. Therefore, the reward function is updated as:

$$R(S^t, v_t) = \mathbb{E}(H^t(V) - H^{t+1}(V)|\mathbf{S^t}, v_t),$$

where $H^t(V)$ is given by:

$$\begin{aligned} H^t(V) = \sum_{v \in V} &-((1 - h(P_v^t) log(1 - h(P_v^t))) \\ &+ (h(P_v^t)) log(h(P_v^t))), \end{aligned} \tag{18}$$

with $h(x) = max(x, 1 - x)$. The posterior probability $P_v^t(+1)$ can be calculated using Eq. (7), with $P_v^t(-1) = 1 - P_v^t(+1)$.

The reward function $R^+(S^t, v_t)$ remains positive for all t because entropy is a submodular function, and entropy minimization always provides gain, ensuring that each labeling action contributes additional information and prevents entropy from increasing, as long as uncertainty remains. The Beta distribution posterior updates further support this by ensuring that each labeling action increases either a_v^t or b_v^t, thereby reducing node entropy and guaranteeing a nonzero expected reward. Additionally, the greedy selection of the maximum expected reward ensures that the policy always picks the vertex that maximizes entropy reduction, meaning there is always at least one vertex with a positive expected reward. While $R^+(S^t, v_t)$ diminishes over time as nodes become more certain, it never reaches zero at finite t since labeling continues until full certainty is achieved. The condition $\lim_{a_v^t + b_v^t \to \infty} R^+(S^t, v_t) = 0$ ensures eventual convergence but does not imply that rewards vanish during the process. Furthermore, since $R^+(S^t, v_t) > 0$ for all t, from the properties of Beta distributions, we know that as $a_v^t + b_v^t \to \infty$, the variance of the Beta distribution $\text{Var}(P_v^t) = \frac{(\alpha + a_v^t)(\beta + b_v^t)}{(\alpha + a_v^t + \beta + b_v^t)^2 (\alpha + a_v^t + \beta + b_v^t + 1)}$ tends to zero, ensuring convergence of the posterior to a deterministic value.

Consequently, the posterior probability update magnitude decreases:

$$\lim_{a_v^t + b_v^t \to \infty} \left(h(P_v^{t+1}(+1)) - h(P_v^t(+1))\right) = 0. \tag{19}$$

Thus, the reward function satisfies:

$$\begin{aligned} &\lim_{a_v^t + b_v^t \to \infty} R(S^t, v_t) = 0, \quad \text{and hence,} \\ &\lim_{a_v^t + b_v^t \to \infty} R^+(S^t, v_t) = 0. \end{aligned} \tag{20}$$

Implying that OPTUENT-OPT labels each instance infinitely as T increases. Given that we assume workers are reliable, this leads to convergence on θ_{v_i} for each $v_i \in V$ and ω_{e_k} for every edge $e_k \in E$. Thus, the overall entropy for vertices and edges converges to a constant value, confirming that OPTUENT-OPT is a consistent policy.

Consistency of OPTUENT-EXP In the OPTUENT-EXP policy, each iteration selects the vertex v_t as follows:

$$v_t = \operatorname*{argmax}_{v} \left(R(\mathbf{S^t}, v) \doteq p_1 R_1(\mathbf{S^t}, v) + p_2 R_2(\mathbf{S^t}, v)\right).$$

While the initial changes in marginal probabilities for v_t may be similar due to all vertices starting with a Beta prior distribution $Beta(\alpha, \beta)$, their impact on the graph varies based on instance correlations and vertex degrees, leading to different rewards. If the label probability θ_v of vertex $v \in V$ differs from 0.5, then $R_1(\mathbf{S^t}, v)$ may not equal $R_2(\mathbf{S^t}, v)$ if $a_v^t \neq b_v^t$. Even when $\theta_v = 0.5$, rewards can still differ based on worker labels, ensuring $R(\mathbf{S^t}, v_t) \neq 0$ whenever $a_v^t \neq b_v^t$. As the budget increases, changes in instance correlations become negligible, yet the difference between a_v^t and b_v^t ensures $R(\mathbf{S^t}, v_t) \neq 0$.

In OPTUENT-OPT, the policy selects the node with the highest optimistic reward, ensuring that the most uncertain and informative node is labeled at every step. However, in OPTUENT-EXP, the policy selects the node based on the expected reward, which takes into account the probabilities of both possible labeling outcomes. This means that rather than always picking the node with the highest potential

entropy reduction, OPTUENT-EXP chooses nodes that, on average, significantly reduce entropy.

Despite this difference, OPTUENT-EXP still ensures that $R(S^t, v_t) > 0$ for all t because the expected entropy reduction remains positive as long as there are remaining uncertain nodes. While the selection process is more balanced, it does not lead to the premature selection of fully certain nodes. Instead, it systematically reduces uncertainty across the graph, ensuring that every node is labeled sufficiently over time.

Thus, with $R(\mathbf{S^t}, v_t) > 0$ for any positive integers a_v^t and b_v^t, each vertex continues to be labeled indefinitely as $T \to \infty$. As labeling continues, entropy minimization ensures that the posterior probabilities stabilize, leading to accurate label estimation. This guarantees that the estimated labels converge to the true values, proving that OPTUENT-EXP is consistent.

D ABLATION STUDIES

D.1 INFLUENCE OF SAMPLE SIZE

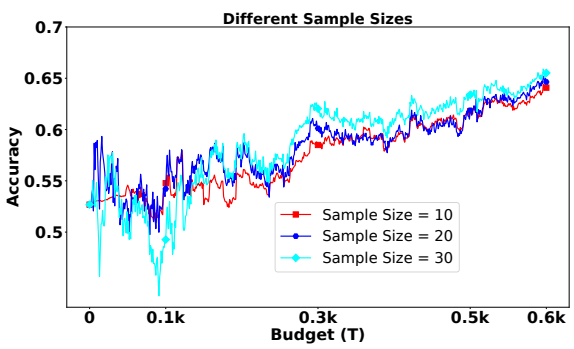

Figure 2: Performance of OPTUENT-EXP with different sample sizes on the WebKB dataset for a fixed $\theta_v = 0.65$.

The experiments presented in Figure 1 and the additional analyses in the Appendix utilize a sample size of 10. While it is intuitive to assume that a larger sample size would yield better performance by providing a greater pool of candidate vertices, our findings suggest otherwise. To test this assumption, we conducted experiments with varying sample sizes of 10, 20, and 30 using the OPTUENT-EXP policy, as illustrated in Figure 2. Remarkably, the results indicate that even with a sample size of just 10, the performance is robust and effective. As the budget increases, the performance gains from larger sample sizes diminish, reinforcing the conclusion that a sample size of 10 is not only sufficient but also optimal for achieving high-quality outcomes in our experiments. This efficiency allows for resource conservation while maintaining competitive performance.

D.2 PERFORMANCE OF RANDOM FOREST REGRESSOR

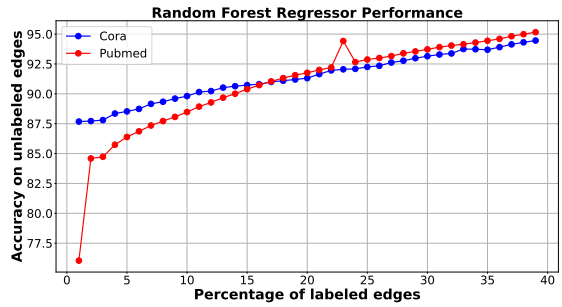

Figure 3: Performance of Random Forest Regressor for Cora and Pubmed datasets.

To evaluate the effectiveness of the Random Forest Regressor, for instance, correlation estimation, we present performance plots for the Cora and PubMed datasets. In these experiments, we assume reliable workers and utilize Equation (4) to compute the marginal probabilities for labeled edges. The regressor is trained exclusively on these labeled edges, and the trained model is subsequently employed to predict the correlations of the remaining unlabeled edges.

As illustrated in Figure 3, the results reveal that the Random Forest Regressor performs remarkably well, even when less than 5% of the edges are labeled. Notably, performance improves significantly with an increase in the proportion of labeled edges, achieving over 90% accuracy with just 15% labeled data for both datasets. These findings demonstrate that the Random Forest Regressor effectively estimates instance correlations, making it a highly suitable model for our proposed task.

E PERFORMANCE COMPARISON FOR SCENARIOS 1 AND 2

In Figures 4 and 5, we present a comparative analysis of the baseline methods under scenarios 1 and 2 using the WebKB, Cora, Citeseer, and Pubmed datasets. In scenario 1, where θ_v is fixed at 0.65, and no belief propagation (BP) or random forest regression (RFR) is employed, we assess the Uniform and OPTKG baselines that treat instances as independent and identically distributed (i.i.d.). In scenario 2, which incorporates BP but excludes RFR, we evaluate the performance of GraphOBA-EXP and GraphOBA-OPT alongside the Uniform and OPTKG baselines, leveraging BP to enhance the propagation of labeling information. The results in Figure 4 clearly demonstrate that our proposed policies, OPTUENT-OPT and OPTUENT-EXP, substantially outperform the Uniform and OPTKG baselines in the absence of BP and RFR, highlighting the importance of effective instance selection even without label propagation.

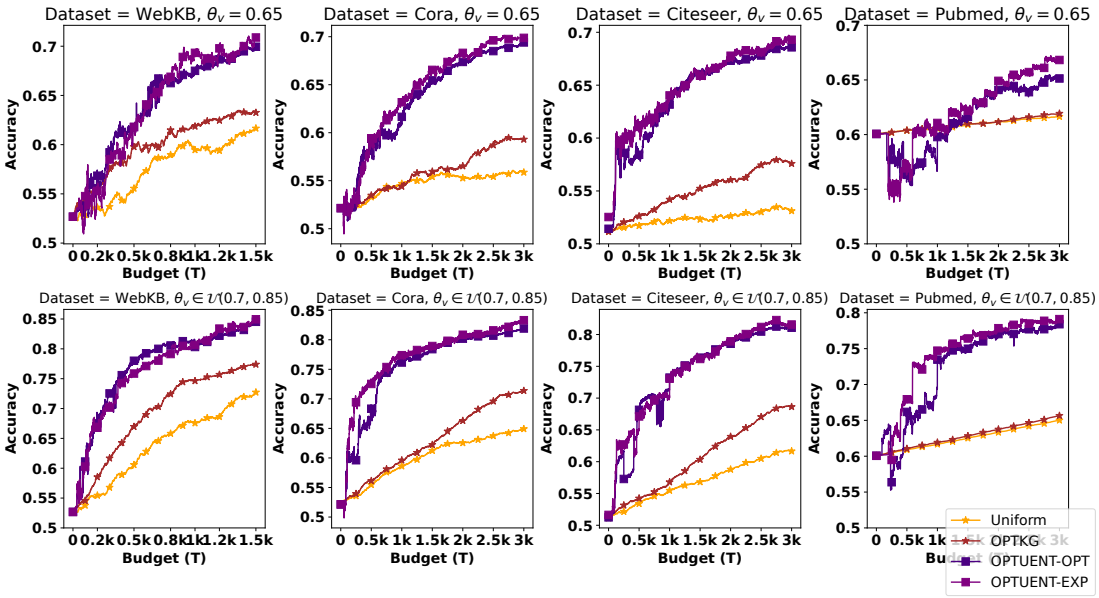

Figure 4: Performance comparison on four graph datasets. The top four plots show the performance comparison of OPTUENT-OPT and OPTUENT-EXP with the baselines following scenario 1 for a fixed $\theta_v = 0.65$, and the bottom four plots show the performance comparison for θ_v sampled from the uniform distribution $\mathcal{U}(0.7, 0.85)$.

In contrast, the performance shown in Figure 5 illustrates a marked improvement when BP is utilized, confirming the findings of Kulkarni et al. [2023] that propagating labeling information significantly enhances performance, even within constrained budgets.

F PERFORMANCE COMPARISON FOR SETTING WITH FIXED θ_v

Figure 6 presents a performance comparison of the OPTUENT-OPT and OPTUENT-ENT policies against baseline methods for the WebKB and Cora datasets. Meanwhile, Figure 7 illustrates similar comparisons for the Citeseer and Pubmed datasets under a fixed θ_v setting, with values set at 0.7, 0.75, 0.8, and 0.85. The results reveal a clear advantage for baselines employing belief propagation, which consistently outperform those treating instances as independent and identically distributed (i.i.d.). Furthermore, the integration of random forest regression significantly enhances the performance of these baseline methods. Notably, when examining the impact of different θ_v values, our proposed policies, OPTUENT-OPT and OPTUENT-ENT, demonstrate remarkable superiority, particularly at lower θ_v values where worker labels tend to be of poorer quality. This underscores the effectiveness of our policies in selecting optimal instances for labeling at each timestamp. Additionally, the ability to estimate instance correlations contributes to improved performance over individual workers across all θ_v values. As θ_v increases and the quality of worker labels improves, we observe a corresponding enhancement in the performance of baselines utilizing both random forest and belief propagation, further emphasizing the critical role that label quality plays in the efficacy of these models.

G ADAPTING PROPOSED APPROACH

The proposed approach is highly adaptable, effectively addressing both binary and multi-class labeling tasks in homogeneous and heterogeneous graphs. For multi-class tasks, we can seamlessly convert them into binary problems using a one-vs-all strategy. While our current framework infers edge labels from node pair labels, transitioning to heterogeneous graphs will require direct edge label annotations, which can be achieved through a Bayesian framework similar to that used for nodes. With these annotations in place, random forests can be employed to estimate edge labels and their associated uncertainties. Additionally, adapting Belief Propagation techniques for heterogeneous networks, such as those proposed by Eswaran et al. [2017], will further enhance the model's robustness.

H LIMITATIONS

Theoretically, the proposed approach can be applied to both binary and multi-class labeling tasks and to both homogeneous and heterogeneous graphs. While this work focuses on homogeneous graphs for binary labeling, the method

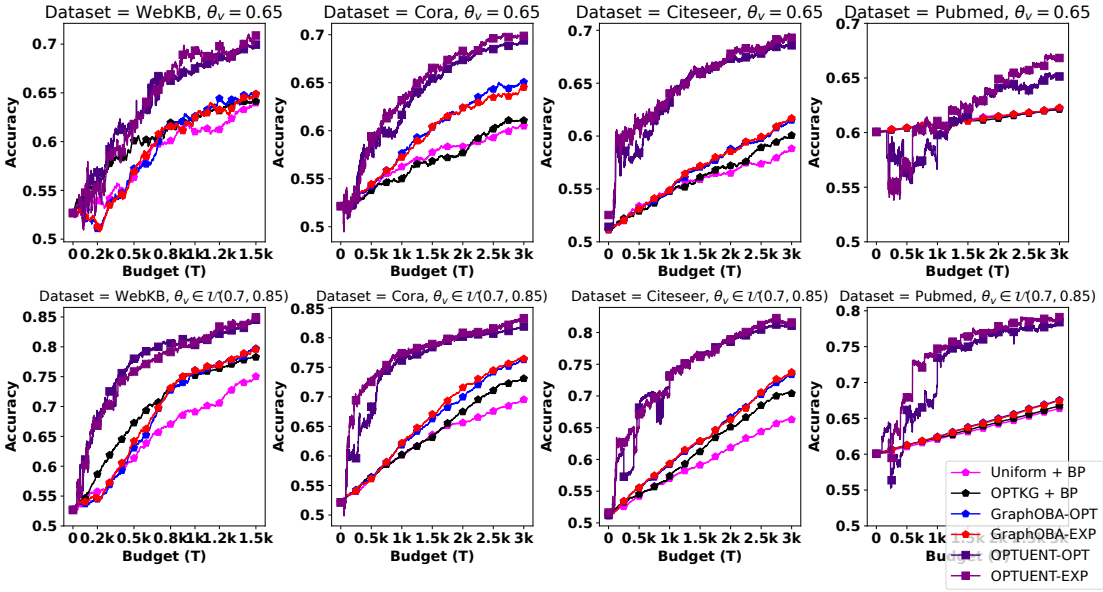

Figure 5: Performance comparison on four graph datasets. The top four plots show the performance comparison of OPTUENT-OPT and OPTUENT-EXP with the baselines following scenario 2 for a fixed $\theta_v = 0.65$, and the bottom four plots show the performance comparison for θ_v sampled from the uniform distribution $\mathcal{U}(0.7, 0.85)$.

is tailored for real-world crowdsourcing scenarios, utilizing simulated worker behavior due to the absence of actual crowd worker labels in our datasets. Details on adapting the method for multi-class labeling and heterogeneous graphs are provided in Appendix G.

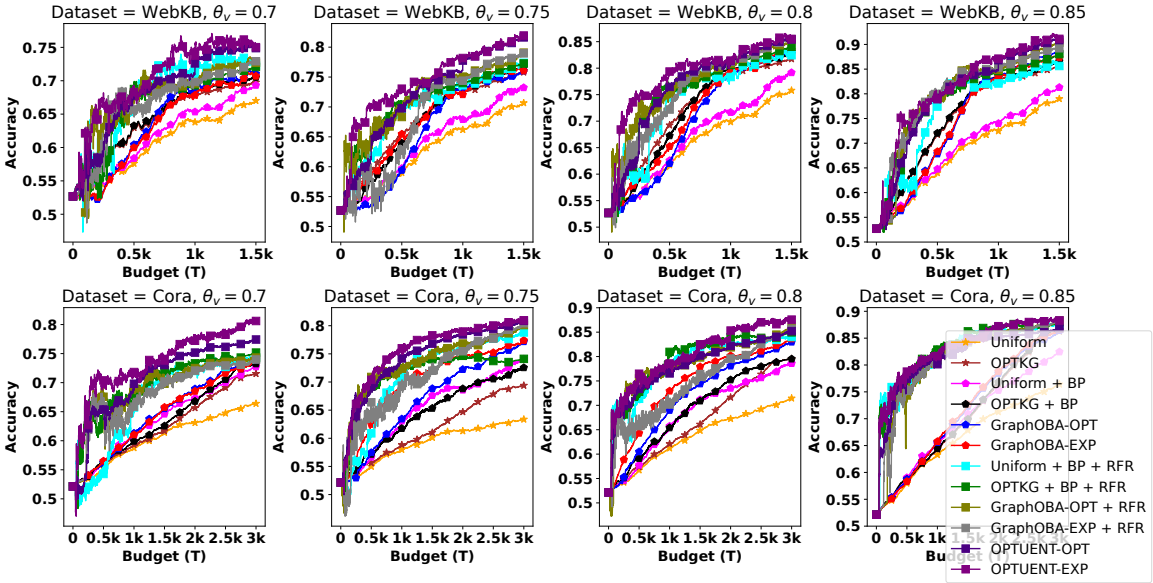

Figure 6: Performance comparison on WebKB and Cora dataset. The top four plots and bottom four show the performance comparison of the proposed OPTUENT with baselines for the WebKB and Cora datasets, respectively, where the value of θ_v is set to 0.7, 0.75, 0.8, and 0.85.

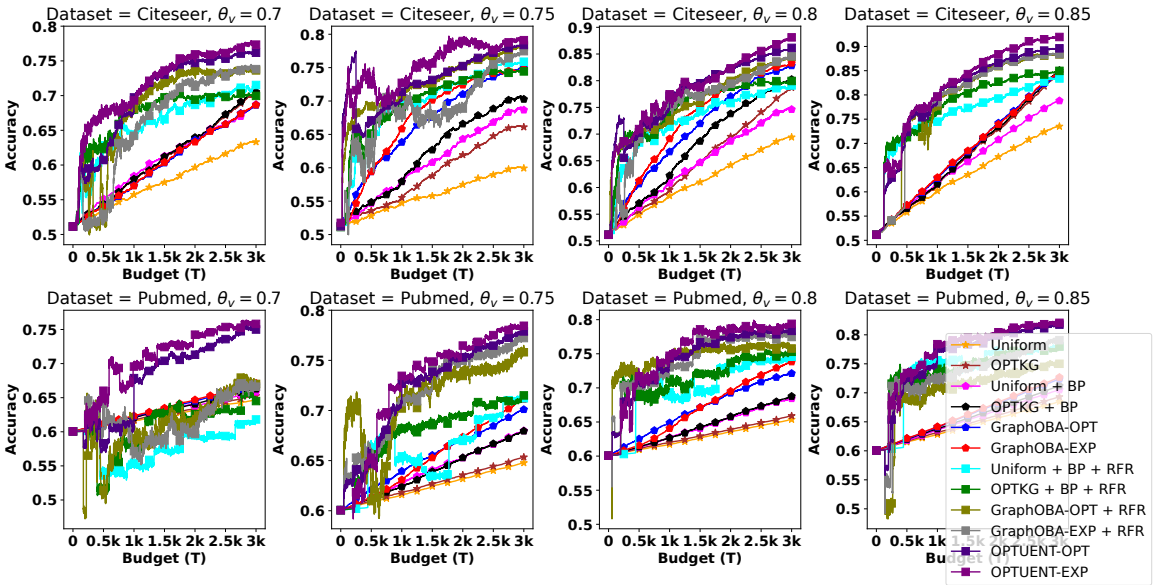

Figure 7: Performance comparison on Citeseer and Pubmed dataset. The top four plots and bottom four show the performance comparison of the proposed OPTUENT with baselines for the Citeseer and Pubmed datasets, respectively, where the value of θ_v is set to 0.7, 0.75, 0.8, and 0.85.

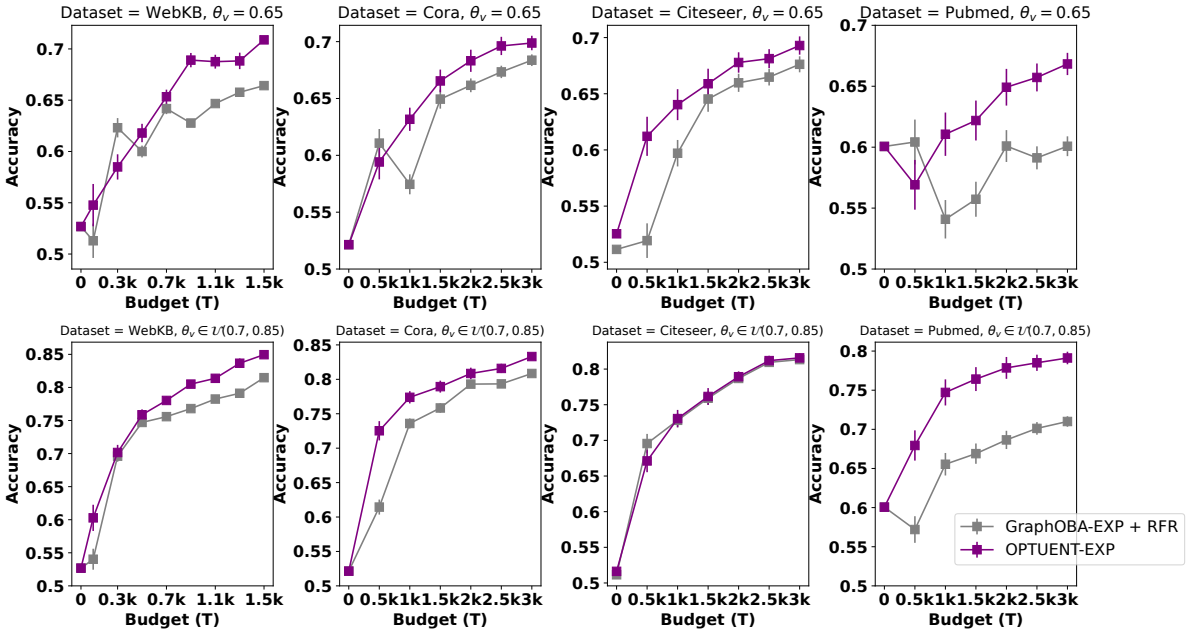

Figure 8: Performance comparison on four graph datasets. The top four plots show the performance comparison between OPTUENT-EXP and GraphOBA-EXP+RFR following scenario 3 for a fixed $\theta_v = 0.65$, and the bottom four plots show the performance comparison for θ_v sampled from the uniform distribution $\mathcal{U}(0.7, 0.85)$. We plot the means and standard deviations for experiments obtained from different seed values of 11, 42, and 111.