Cross-domain-aware Worker Selection with Training for Crowdsourced Annotation

Yushi Sun*, Jiachuan Wang*, Peng Cheng†, Libin Zheng‡, Lei Chen§*, Jian Yin‡
*Hong Kong University of Science and Technology, Hong Kong, China
†East China Normal University, Shanghai, China
‡Sun Yat-sen University, Guangzhou, China
§Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China
ysunbp@cse.ust.hk, jwangey@connect.ust.hk, pcheng@sei.ecnu.edu.cn,
zhenglb6@mail.sysu.edu.cn, leichen@hkust-gz.edu.cn, issjyin@mail.sysu.edu.cn

***Abstract*—Annotation through crowdsourcing draws incremental attention, which relies on an effective selection scheme given a pool of workers. Existing methods propose to select workers based on their performance on tasks with ground truth, while two important points are missed. 1) The historical performances of workers in other tasks. In real-world scenarios, workers need to solve a new task whose correlation with previous tasks is not well-known before the training, which is called cross-domain. 2) The dynamic worker performance as workers will learn from the ground truth. In this paper, we consider both factors in designing an allocation scheme named cross-domain-aware worker selection with training approach. Our approach proposes two estimation modules to both statistically analyze the cross-domain correlation and simulate the learning gain of workers dynamically. A framework with a theoretical analysis of the worker elimination process is given. To validate the effectiveness of our methods, we collect two novel real-world datasets and generate synthetic datasets. The experiment results show that our method outperforms the baselines on both real-world and synthetic datasets.**

***Index Terms*—crowdsourcing, worker selection, cross-domain**

I. Introduction

The quality of the labeled data is of great importance for the performance of machine learning, especially for supervised learning models [25]. To get high-quality annotations for large-scale datasets, recruiting domain experts is too expensive and thus unacceptable. With a limited budget, annotation through selecting crowdsourcing workers is preferable and has drawn attention in recent years [9], [41]. Worker selection is one of the most important issues in the quality control consideration of crowdsourcing [51], which focuses on identifying workers with high performance from the worker pool. How to design an allocation scheme to effectively and efficiently select high-performance crowd workers remains a challenging problem.

In order to select workers through worker quality estimation, existing methods [29], [47], [48], [51] consider different factors in the crowdsourcing process: 1) workers' responses to the golden questions [29]; 2) additional social network interactions for worker trustworthiness estimation [47], [51]; 3) assumption of worker skills, which are hidden states between worker performance and tasks [48].

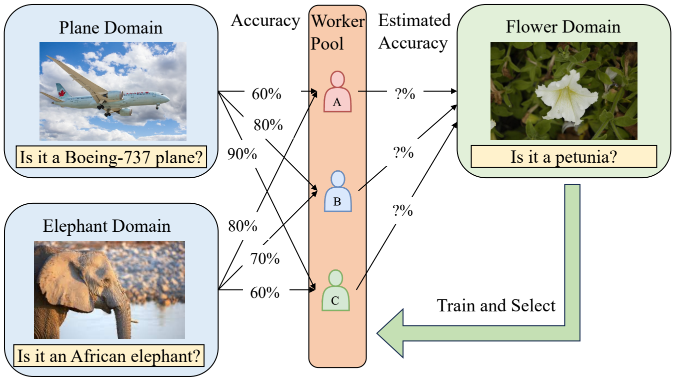


Fig. 1. Cross-domain worker selection. The left shows the two prior domains: plane and elephant. The right shows the target domain: flower. We record workers' historical accuracy on the two prior domains and estimate the accuracy on the target domain, to effectively train and select desired workers.

In order to obtain large-scale manually labeled data for business, many companies such as JD.com, Inc. [3], Alibaba [1], and Baidu [2] have their own commercial crowdsourcing platforms with worker pools. The answering history of workers stored in commercial crowdsourcing platforms can be helpful in selecting high-quality workers to complete tasks in a new domain [6], which is not well-explored in the existing worker selection methods [29], [47], [48], [51]. We refer to the tasks of new topics requiring workers to annotate as *target domain tasks*, while the tasks of historical topics are *prior domain tasks*. In the beginning, the correlations between the target domain and these prior domains are not well-known, which is called *cross-domain*. The performance of workers in the prior domain can help predict their performance in the target domain. As shown in Figure 1, given the classification performance on the elephants and planes of workers A, B, and C, we can obtain a rough idea of their domain knowledge of distinguishing living creatures and distinguishing machines, which is helpful for us to select the proper workers to work on tasks on other domains, such as flowers. Intuitively, workers with good performance in distinguishing elephants are likely to be sensitive to color and shape differences (since different kinds of elephants are similar in size but different in color and shape). In contrast, workers who perform well in distinguishing planes are likely to be good at identifying size differences (since different kinds of planes are similar in color and shape yet different in size). Given this prior domain knowledge, workers who are potentially good at distinguishing flowers (relying on color and shape differences) can be identified. We can train these identified workers by demonstrating the ground truth answers to them so that they can actively learn the characteristics of petunia and achieve better performance on the annotations. After that, we can select the best workers as the desired worker candidates for the target domain. In this way, the *golden questions* from the target domain are fully utilized: not only used for estimating the cross-domain worker quality to select the best candidates but also used to boost the annotation performance of workers on the target domain through worker training.

However, transferring and incorporating workers' performance profiles across different domains is challenging. Explicitly defining the mappings between the domains and the skill sets requires a comprehensive understanding of the domain tasks, which needs expert effort and thus fails to scale well in reality. Therefore, we propose automatically and inherently capturing the relationship between each domain and the required skills to ensure feasibility and scalability in real-world applications. Workers' performances are modeled based on reasonable assumptions for inner- and inter-domain. To be more specific, we apply normal distributions to model workers' performances on each domain following the modeling done by previous studies [40], [52] and adopt a multivariate normal distribution to model the correlation between workers' performances on different domains to achieve the goal of cross-domain worker selection.

As correlation is not well-known for the cross-domain problem, we apply a worker learning stage to train and select workers while simultaneously extracting their correlations. During the learning stage, limited *golden questions* are given with accurate labels from experts. Previous approaches [31] treat the assignment of golden questions as a sampling process to get a static estimate of worker quality. However, workers' knowledge of the target domain can be dynamic [19], [21]. For instance, in Figure 1, workers are asked whether the flower is a petunia. Initially, workers may have no idea what a petunia is. However, after we assign multiple golden (learning) questions regarding the petunia and reveal the answers to the workers, they can gradually learn about the characteristics (such as shape and color) of petunia and thus perform better when answering new questions on the same domain.

Unfortunately, previous work has not studied the dynamic worker knowledge change in worker selection. We fill the research gap by simulating workers as trainable to handle the dynamic worker selection instead of a static one. In this paper, we propose to model the learning gain of workers based on the Item Response Theory (IRT) [39] from the Knowledge Tracing field to fully use the golden questions in selecting workers. Our allocation algorithm can select potential workers who can improve quickly during the learning stage, which are filtered out by static methods.

The contributions of this paper are as follows:

- We incorporate the cross-domain knowledge information and propose a novel Median Elimination-based worker selection with training algorithm to find high-quality workers.

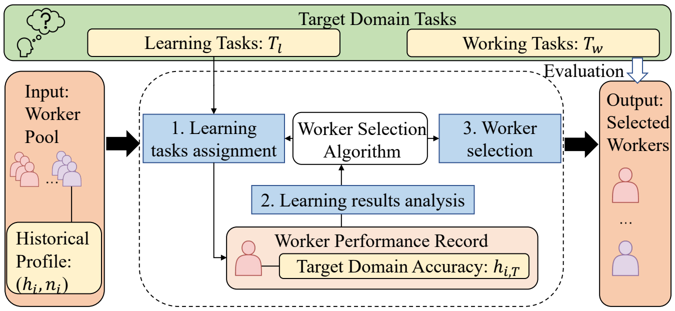

Target Domain Tasks
Learning Tasks: T_l
Working Tasks: T_w
Evaluation
Input: Worker Pool
...
Historical Profile: (h_i, n_i)
1. Learning tasks assignment
Worker Selection Algorithm
3. Worker selection
2. Learning results analysis
Worker Performance Record
Target Domain Accuracy: $h_{i,T}$
Output: Selected Workers
...

Fig. 2. The definition of cross-domain-aware worker selection with training problem. The worker selection algorithm assigns learning tasks to workers, records and analyzes the learning task results, and performs worker selection. The performance of the selected workers is evaluated based on the target domain working tasks.

- We comprehensively consider the learning gain of workers during the learning task worker training process over the new domain to get a better estimate of the dynamic change in worker quality.
- We collect two novel cross-domain worker selection datasets for the crowdsourcing research community to study the problem of cross-domain worker selection with training.
- We conduct extensive experiments on real-world and synthesized datasets to evaluate the performance of our proposed method comprehensively.

The following sections are arranged as follows. We first introduce the related work in Section II. We then discuss the setup and problem formulation in Section III. The methodology is introduced in Section IV. We demonstrate the experiment results in Section V, and finally, we conclude the paper in Section VI.

II. Related Work

The general process of worker selection for quality control requires first estimating worker quality and then designing proper worker elimination algorithms to select the best workers. In this section, we first discuss the related works on worker selection from two aspects: worker quality estimation and worker elimination, which are the core components for worker selection with training task. Then we introduce the related works in Knowledge Tracing, which are related to the learning gain estimation process used for estimating worker quality and performing worker elimination.

A. Worker Quality Estimation

Worker Quality Estimation assesses the abilities and reliability of individual workers participating in tasks on a crowdsourcing platform. Previous studies proposed various worker quality estimation methods based on the information available in different crowdsourcing worker selection scenarios. Liu et al. [31] estimate worker quality based on workers' answers to golden questions. Li et al. [30] proposed a user discovery framework to select reliable workers based on general characteristics such as educational level, gender, and age. However, these characteristics may be too general for us to identify the best workers accurately. For instance, the approach would fail if the recruited workers were all college students with similar backgrounds. Zhao et al. [51], and Wu et al. [47] proposed integrating social network information into the worker quality estimation process. A major limitation of these approaches is that social network information is not necessarily available in many online crowdsourcing platforms, which limits the application domain of these approaches. Yadav et al. [48] proposed constructing a universal skill set with the mapping relationship between the skills and the crowdsourcing tasks. The skills of the workers are estimated based on the historical annotation performance. However, the universal skill set and the mapping relationship should be manually defined. Explicitly constructing the universal skill set and the mapping relationship between the skills and the tasks require domain knowledge, which brings overheads to real-world worker selection applications. Different from existing worker quality estimation approaches, we considered both cross-domain knowledge and the dynamic worker knowledge change to better capture the worker quality during the crowdsourcing process.

B. Worker Elimination

This section introduces the worker elimination algorithms used by existing works. Worker elimination is a process used to select the most qualified workers for a task while filtering out underperforming or unreliable workers. Even-Dar et al. [18] proposed a naive Uniform Sampling algorithm and a Median Elimination algorithm for the top-k selection multi-armed bandits problem, which can be adapted to perform worker elimination. Liu et al. [31] identify the best group of workers by assigning golden questions and selecting the workers that perform the best on those questions. Li et al. [30] selected the best workers based on the estimated performance generated from the general workers' profiles. Cao et al. [10] refined the theoretical bounds of [18]. They introduced budget constraints and proposed a greedy-based heuristic algorithm to sort the workers based on the error rate and the requirements. Zhao et al. [51] proposed forward and backward selection algorithms based on social network connections to gradually identify the best workers. Wu et al. [47] considered the interest similarity between the workers and the tasks based on social network information to identify the best workers for the tasks. Yadav et al. [48] proposed a team formation algorithm to gather the workers with the desired expertise for the target tasks. Building on the Medium Elimination algorithm introduced by [10], we additionally considered the worker learning gain during the worker elimination process, so as to achieve better worker elimination results.

C. Knowledge Tracing

Knowledge tracing is a technique used in education to understand how well a student is learning a particular subject. It involves tracking and predicting a student's knowledge and understanding over time. By analyzing the student's responses to questions or tasks, knowledge tracing models can estimate the student's current level of knowledge, identify areas of strength and weakness, and provide personalized feedback and guidance to enhance learning [12]. As discussed in [5], [34], the knowledge tracing methods can be divided into three categories: Bayesian Knowledge Tracing, Factor Analysis Models, and Deep Knowledge Tracing based on the different types of inputs and application scenarios.

Bayesian Knowledge Tracing (BKT): The BKT model is first introduced by [12], where the skills behind the questions are considered. Four different types of probabilities associated with changes in skill mastery are modeled, and the Bayesian estimation of the final state skill mastery probability is used. Several subsequent works [13], [26], [28], [36], [49] propose to extend the original BKT model with student-specific modeling and inter-skill relationship.

Factor Analysis Models: The simplest and the most widely used Item Response Theory (IRT) model is Rasch's model [39], which defines a one-parameter logistic (1PL) IRT model. The probability that a worker answers a question correctly is modeled as a logistic function based on the worker's learning parameters and the difficulty of the question. Wilson et al. [46] extended the original 1PL IRT model to Hierarchical IRT and Temporal IRT by additionally considering the relatedness of parameters across different questions and times, respectively. Performance Factor Analysis (PFA) [37] is proposed to extend the IRT model by replacing the learning parameter with multiple learning skill parameters to model the relationship of multiple skills.

Deep Knowledge Tracing (DKT): DKT is first proposed in [38], which models the knowledge states (skills) of people with Long Short Term Memory (LSTM) [23]. The LSTM network contains many neurons to represent the hidden states of workers' answer history. The current knowledge states of workers can be learned from the training data. Several works [4], [33], [50] extend the idea of the original DKT model to achieve improved performance.

In our paper, we aim to model the worker learning gain without explicitly defining and modeling the relationship between the skills and the questions, so we adopt Rasch's IRT model [39] to model the learning process of workers while answering learning questions. Note that the focus of this paper is to introduce the knowledge tracing techniques into the crowdsourcing worker selection process to achieve better worker quality estimation and elimination results, instead of developing novel knowledge tracing approaches.

III. Problem Formulation

We present the notations used in this paper in Table I. The general process of cross-domain-aware worker selection with training can be divided into three steps, as shown in Figure 2. We first discuss the tasks and workers considered in our paper, introduce the three steps generally, and formally define the cross-domain-aware worker selection with training problem.

Definition 1: (Tasks). The crowdsourcing tasks on the target domain can be categorized into learning and working tasks. The learning tasks (golden questions) refer to the tasks that have gold labels. The working tasks are the tasks without gold

TABLE I
NOTATIONS.

Notations	Descriptions
T_l, T_w	learning tasks and working tasks sets
W	the worker pool
w_i	the i-th worker in the worker pool W
h_i	the historical profile of the worker w_i
n_i	number of annotation tasks completed by worker w_i on different domains
B	the total budget
k	number of workers we want to select
α_i	the learning parameter of worker w_i
β_d	the domain difficulty parameter for domain d
θ_i	the proficiency parameter of worker w_i
K_j	the cumulative number of learning tasks assigned to each remaining worker till round j
n	the number of elimination rounds
a_t	the initialized annotation accuracy of the target domain
Q	the number of learning tasks per batch

labels. We denote the set of tasks on the target domain as T, the set of learning tasks as T_l, and the set of working tasks as T_w.

For simplicity, we consider Multiple Choice Question tasks in our paper. As suggested by [6], this selection of task type does not influence the generalizability of our approach since our approach is based on the answering accuracy, which can also be computed if other kinds of tasks are used.

Definition 2: (Workers). We denote the worker pool as W. Each worker w_i in W is associated with a historical profile $(h_i, n_i) = (\{h_{i,1}, h_{i,2}, ..., h_{i,D}\}, \{n_{i,1}, n_{i,2}, ..., n_{i,D}\})$ where D is the number of prior domains, $h_{i,j}$ is the annotation accuracy of worker w_i on the j-th prior domain, and $n_{i,j}$ is the number of annotation tasks completed by worker w_i on the j-th prior domain. The annotation accuracy of w_i on the target domain working tasks is denoted as $h_{i,T}$.

Definition 3: (Learning tasks assignment). Learning tasks assignment is the process of assigning learning tasks to the worker and recording the accuracy of each worker. The answers to the learning tasks are revealed to each worker after he/she submits the answers so that he/she can learn the target domain knowledge from the revealed ground truths.

In our paper, we train workers in rounds. Each remaining worker w_i is assigned learning tasks and the annotation accuracy $a_{i,c}$ is recorded in round c.

Definition 4: (Learning results analysis). Learning results analysis is the process of estimating workers' performances based on their results on the learning tasks for the assignment of remaining learning tasks and the selection of workers.

Definition 5: (Worker selection). Considering the cross-domain performances and learning result feedback of workers, the worker selection step is to select the well-trained workers with the best performance in the target domain.

Definition 6: (Cross-domain-aware worker selection with training). Given target domain tasks $T = \{T_l, T_w\}$, the total budget B, and worker pool W with each worker w_i's historical profile h_i. Cross-domain-aware worker selection with training problem is to 1) properly assign no more than B tasks to $|W|$ workers for training based on workers' historical records and learning feedback and 2) select top k workers with the highest possible annotation accuracy on working tasks T_w.

IV. Methodology

In this section, we first introduce our general framework for cross-domain-aware worker selection with training problem (Subsection IV-A). Then we demonstrate the process of worker training (Subsection IV-B) and discuss the details of the two core phases: Worker Quality Estimation (Subsection IV-C) and Worker Selection (Subsection IV-D) with theoretical analysis. A summary of the whole pipeline is presented in Subsection IV-E.

A. Framework

We display our framework in Figure 3. Workers are iteratively trained and selected through a 3-phase pipeline:

- Worker Training. In the target domain, workers answer questions and check answers to renew their knowledge.
- Worker Quality Estimation. The ability estimation of each worker will be updated according to his/her answers during worker training in addition to the historical records on target and other domains.
- Worker Selection. Based on the estimated worker quality, we select the best half of the workers to enter the next round.

Finally, after n rounds, we obtain the selected best k workers and assign the target domain working tasks for them to annotate. In our paper, we fix the budgets in each round and focus on the accuracy of dynamic worker estimation. Mathematically, $t = \lfloor \frac{B}{n} \rfloor$ is the fixed number of learning tasks per round, and $|W_c|$ is the number of remaining workers for the current round c. Then, $\lfloor \frac{t}{|W_c|} \rfloor$ is the number of tasks per worker for round c.

B. Worker Training

We can summarize the worker training as a simple "Answer and Learn" process for workers. To be more specific, after a worker completes one batch of learning tasks, their ground truth answers will be revealed to the worker. For example, as shown in Figure 4, the left shows a learning task completed by a worker, and the right shows the ground truth answer to that task. A worker can learn from the ground truth answers and renew his/her target domain knowledge.

Formally, after each round of task assignment, we denote the answers of worker w_i in the current round c as $a_{i,c}$.

C. Worker Quality Estimation

To achieve high-quality worker selection, two factors are important: *cross-domain correlation* which can help us filter workers according to their performance on other domains; and *worker learning gain* where workers who improve more from training should be preserved and assigned with more training tasks. Thus, we divide the worker quality estimation phase into Cross-domain-aware Performance Estimation (CPE) and Learning Gain Estimation (LGE). CPE focuses on modeling the cross-domain correlation of workers, while LGE focuses on modeling the learning gain in the worker training process.

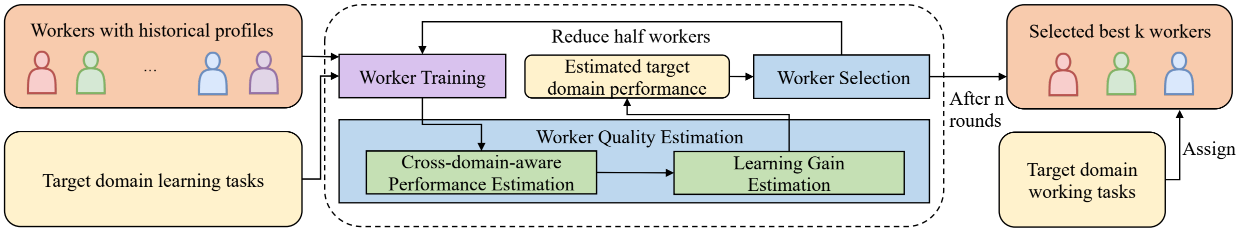

Workers with historical profiles
...
Target domain learning tasks
Reduce half workers
Worker Training
Estimated target domain performance
Worker Selection
Worker Quality Estimation
Cross-domain-aware Performance Estimation
Learning Gain Estimation
After n rounds
Selected best k workers
Target domain working tasks
Assign

Fig. 3. The general pipeline of our cross-domain-aware worker selection with training algorithm.

1) Cross-domain-aware Performance Estimation: As stated in Section I, cross-domain information is important for worker selection. However, no correlation information between the prior domains and the target domain is available before worker training. Instead, worker feedback on learning tasks is accumulatively arriving, which requires a mining algorithm to dynamically capture the cross-domain correlation from scratch. In this subsection, we introduce our CPE estimation scheme, which 1) derives statistically analyzed expectation of accuracy and 2) supports online updates with a Maximum Likelihood Estimation as the base. We present the whole CPE estimation in Algorithm 1.

In order to model the correlation between workers' prior domain knowledge and the target domain knowledge, we adopt the multivariate normal distribution [43]. Precisely, to model the relationship between the D prior domains and the target domain effectively, we adopt a $(D+1)$-dimensional multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$, where $\mu \in \mathbb{R}^{(D+1)}$ and $\Sigma \in \mathbb{R}^{(D+1)\times(D+1)}$:

$$\mu = [\mu_1, \mu_2, ..., \mu_D, \mu_T]^\mathsf{T}, \tag{1}$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho_{1,2}\sigma_1\sigma_2 & \dots & \rho_{1,D}\sigma_1\sigma_D & \rho_{1,T}\sigma_1\sigma_T \\ \rho_{2,1}\sigma_2\sigma_1 & \sigma_2^2 & \dots & \rho_{2,D}\sigma_2\sigma_D & \rho_{2,T}\sigma_2\sigma_T \\ \dots & \dots & \dots & \dots & \dots \\ \rho_{D,1}\sigma_D\sigma_1 & \rho_{D,2}\sigma_D\sigma_2 & \dots & \sigma_D^2 & \rho_{D,T}\sigma_D\sigma_T \\ \rho_{T,1}\sigma_T\sigma_1 & \rho_{T,2}\sigma_T\sigma_2 & \dots & \rho_{T,D}\sigma_T\sigma_D & \sigma_T^2 \end{bmatrix}, \tag{2}$$

the $\mu_i, \sigma_i, \rho_{i,j}$ where $i \neq j$ and $i, j \in \{1, 2, ..., D, T\}$ are the mean and standard deviation of worker accuracy on each domain and the correlation parameters between any pair of domains respectively.

The annotation accuracy for each worker w_i on each domain is modeled as a $(D+1)$-dimensional random vector $v_i = [h_{i,1}, h_{i,2}, ..., h_{i,D}, h_{i,T}]^\mathsf{T} \in \mathbb{R}^{(D+1)}$, where $v_i \sim \mathcal{N}(\mu, \Sigma)$.

As shown in Figure 3, we perform CPE (Algorithm 1) in each elimination round. Specifically, each worker is assigned $(\lfloor \frac{t}{|W_c|} \rfloor)$ learning tasks and we record the answers for each worker w_i: $a_{i,c} = [a_{i,c,1}, a_{i,c,2}, ..., a_{i,c,\lfloor \frac{t}{|W_c|} \rfloor}]$ and store to A_c. For each worker, we compute the number of correct and wrong answers as follows:

$$C_{i,c} = \sum_{j=1}^{\lfloor t/|W_c| \rfloor} \mathbb{1}(a_{i,c,j} = g_{j,c}), \tag{3}$$

$$X_{i,c} = \lfloor (t/|W_c|) \rfloor - C_{i,c}. \tag{4}$$

Given each worker's correct and wrong answers in each round, we adopt Maximum Likelihood Estimation to estimate μ and Σ. The log-likelihood function L is formulated as follows:

$$\begin{aligned} \log L &= \sum_{i=1}^{|W_c|} \log P(h_{i,T}|h_i) \\ &= \sum_{i=1}^{|W_c|} \log \int_0^1 h_{i,T}^{C_{i,c}} (1-h_{i,T})^{X_{i,c}} \frac{e^{-\Psi}}{\sqrt{2\pi|\bar{\Sigma}|}} \mathrm{d}h_{i,T} \\ &= \sum_{i=1}^{|W_c|} \big[\log \int_0^1 h_{i,T}^{C_{i,c}} (1-h_{i,T})^{X_{i,c}} e^{-\Psi} \mathrm{d}h_{i,T} \\ &\quad + \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2}\log|\bar{\Sigma}|\big], \end{aligned} \tag{5}$$

where $\bar{\mu}$ and $\bar{\Sigma}$ are the conditional distribution of the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$:

$$\bar{\mu} = \mu_T + \Sigma_{1\times D}\Sigma_{D\times D}^{-1}(h_i - \mu_{1\sim D}),$$
$$\bar{\Sigma} = \Sigma_{1\times 1} - \Sigma_{1\times D}\Sigma_{D\times D}^{-1}\Sigma_{D\times 1},$$

and $\Psi = \frac{(h_{i,T}-\bar{\mu})^\mathsf{T}(h_{i,T}-\bar{\mu})}{2\bar{\Sigma}}$.

In real-world applications, these parameters are updated under a large amount of streaming data, where directly calculating the optimal parameters is unacceptable. To enable an incremental parameter estimation, we update μ and Σ by maximizing Equation (5) with gradient descent in each round:

$$\mu' = \mu - r_1 \nabla_\mu \log L, \tag{6}$$
$$\Sigma' = \Sigma - r_2 \nabla_\Sigma \log L, \tag{7}$$

where r_1 and r_2 are the learning rates of gradient descent for μ and Σ; μ' and Σ' are the updated μ and Σ at the current gradient descent step. After obtaining the Maximum Likelihood Estimation of the mean $\hat{\mu}$ and standard deviation $\hat{\Sigma}$, we obtain the predicted annotation accuracy for each worker w_i with the updated multivariate normal distribution $\hat{\mathcal{N}}(\hat{\mu}, \hat{\Sigma})$:

$$\begin{aligned} p_{c,i} &= E[h_{i,T}|h_i] \\ &= \int_0^1 h_{i,T} P(h_{i,T}|h_i) \mathrm{d}h_{i,T} \\ &= \int_0^1 h_{i,T} \frac{P(h_i, h_{i,T})}{P(h_i)} \mathrm{d}h_{i,T}, \end{aligned} \tag{8}$$

where $[h_i, h_{i,T}]^\mathsf{T} \sim \hat{\mathcal{N}}(\hat{\mu}, \hat{\Sigma})$ and $[h_i]^\mathsf{T} \sim \hat{\mathcal{N}}(\hat{\mu}_{1\sim D}, \hat{\Sigma}_{D\times D})$. $p_{c,i}$ is used as the estimated worker accuracy in the target domain instead of a coarse observation on $C_{i,c}$ and $X_{i,c}$.

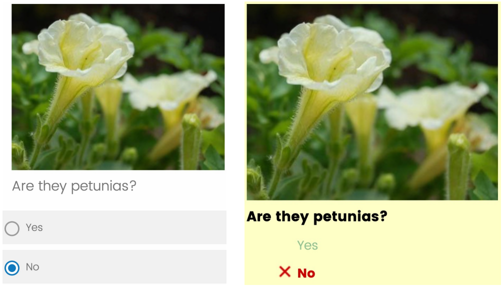


Fig. 4. An illustration of the learning task (left) and its corresponding ground truth answer (right). The learning tasks will be displayed to the workers. After they complete their current round answers, the ground truth will be revealed for them to learn.

```
Algorithm 1 Cross-domain-aware Performance Estimation (CPE)
Input:
    The answers of workers in current round $A_c$
    The learning tasks ground truth in current round $G_c$
    The historical accuracy of workers in current round $H_c$
    The number of workers remaining in current round $|W_c|$
Output:
    The predicted accuracy of remained workers $p_c$
Initialize $p_c$ to be an empty array
Initialize the multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$
Compute the number of correct and wrong tasks of each worker $C_{i,c}, X_{i,c}$ according to Equations (3) and (4)
Compute updated distribution $\hat{\mathcal{N}}(\hat{\mu}, \hat{\Sigma})$, where
    $\hat{\mu}, \hat{\Sigma} = \arg\max_{\mu,\Sigma} \log L(\mathcal{N}, \{C_{i,c}, X_{i,c}\}_{i=1}^{|W_c|})$
for each $h_i$ in $H_c$ do
    Compute $p_{c,i}$ via Equation (8) and append to $p_c$
end for
return $p_c$
```

2) *Learning Gain Estimation:* Through the interactive mode displayed in Figure 4, workers not only provide feedback but also *learn* from the results. This training process plays an important role in crowdsourcing [19], [21], but is hardly studied together with worker selection. This motivates us to enhance worker estimation with training, which aims at capturing the changes in estimated target domain performance for workers. This further enables us to quantize each worker's learning gain after assigning a certain amount of learning tasks to get a more accurate dynamic estimation of workers' performance on T_w.

In order to capture the learning gains, we adopt the item response theory model used for capturing the student learning process from the previous work [34], [46]. In the original IRT model [34], [46], considering a worker w_i, the probability that w_i answers question q from domain d correctly is:

$$p_d(\theta_i) = (1 + e^{-(\theta_i - \beta_d)})^{-1}, \tag{9}$$

where θ_i is the proficiency parameter of the worker and β_d is the difficulty parameter of the question q from domain d. In our setting, $\theta_i = \alpha_i \ln(K_j + 1)$, which is proportional to the logarithm of the cumulative number of learning tasks ($K_j = \frac{(2^j-1)*t}{|W|}$ for the target domain) assigned to worker w_i, while α_i is computed through least square regression and will be discussed in Equation (11). Different difficulty parameters are assigned to tasks in different domains, denoted as $\beta_{1\sim D}$ for tasks in prior domains and β_T for tasks in the target domain. The modified item response theory model for a single worker w_i at the learning stage j on the domain d is:

$$\begin{aligned} \hat{p}_{j,i,d} &= g(\alpha_i, \beta_d, K_j) \\ &= \frac{1}{1 + e^{-(\alpha_i \ln(K_j+1) - \beta_d)}}. \end{aligned} \tag{10}$$

The last step before one can further estimate the dynamic performance after training instead of a static performance is to get the update formula for the intrinsic learning parameter α_i of each worker in each round. We update the learning parameter α_i by minimizing the following least square regression objective:

$$\alpha_i = \arg\min_{\alpha_i} \left[\sum_{d=1}^{D} (\hat{p}_{1,i,d} - h_{i,d})^2 + \sum_{j=1}^{c} (\hat{p}_{j-1,i,t} - p_{j,i})^2 \right], \tag{11}$$

which comprises two parts: the first part minimizes gaps between learning gain estimations and accuracy on D prior domains; the second part minimizes each pair of worker accuracy estimated by Equation (10) in round $j-1$ and CPE in round j. The round index is different because the CPE estimation is based on the workers' performances in round j, where workers are only shown with $j-1$ rounds of answers (trained with $j-1$ rounds). The regression is conducted in each round to update each α_i.

According to the results of each round (e.g., the j^{th} round), we assign tasks to workers and expect the best performance after the training of the next round (e.g., the $j+1^{th}$ round), which can be obtained through Equation (10) (e.g., compute $\hat{p}_{j+1,i,d}$), which is intractable for static methods. We argue that such an estimation is closer to the actual annotation performance on the working tasks after n rounds of training and thus can help us get a more accurate estimate of the actual value of $h_{i,T}$ for each worker w_i. We display the LGE in Algorithm 2.

D. Worker Selection

Based on the above estimations, we propose our algorithm for worker selection in this subsection, where a theoretical guarantee is given. Compared with the intuitive but effective design of the Median Elimination algorithm discussed in [18], we have a fixed amount of budget to allocate tasks, where the original algorithm and theory cannot be directly applied. Here, we propose an adaptation version, displayed in Algorithm 3. To be more specific, ME is called in rounds, where in each round, the worst half of workers are eliminated. The algorithm terminated with k workers left as its output. With a limited budget, we reversely derive the number of rounds needed for

Algorithm 2 Learning Gain Estimation (LGE)

```
Input:
    The workers remained in the current round $W_c$
    The historical accuracy of workers in the current round $H_c$
    The historical task numbers of workers in the current round $N_c$
    The predicted accuracy arrays $p_1, p_2, ..., p_c$ at the current stage
Output:
    The updated predicted accuracy with learning gains $\hat{p}_c$
Initialize $\hat{p}_c$ as an empty array
Initialize the difficulty parameters in the target domain ($\beta_T$) and prior domains ($\beta_1, \beta_2, \cdots, \beta_D$)
for each $w_i \in W_c$ do
    Initialize the learning parameter $\alpha_i$
    for domain $d = 1, 2, ..., D$ do
        Compute the historical accuracy $h_{i,d}$
        Compute the historical task numbers $n_{i,d}$
        Compute the IRT score $\hat{p}_{1,i,d} = g(\alpha_i, \beta_d, n_{i,d})$
    end for
    for stage $j = 1, 2, ..., c$ do
        Compute the predicted accuracy for $w_i$: $p_{j,i}$
        Compute $\hat{p}_{j-1,i,T} = g(\alpha_i, \beta_T, K_{j-1})$
    end for
    Update $\alpha_i$ according to Equation (11)
    Compute $\hat{p}_{c,i,T} = g(\alpha_i, \beta_T, K_c)$
    Append $\hat{p}_{c,i,T}$ to $\hat{p}_c$
end for
return $\hat{p}_c$
```

elimination and allocate the budget. Specifically, given worker pool W, number of k, total budget B, we can get the total round n and the budget allocated in each round t as:

$$n = \lceil \log(|W|/k) \rceil, \tag{12}$$

$$t = \lfloor B/n \rfloor. \tag{13}$$

Unlike the original (ϵ, δ) bound formulation in [18], we constrain the total budget used for the task and prove a theoretical bound over error ϵ_c in each round. Specifically, given a fixed total budget of B, the algorithm aims at finding the top k workers. It satisfies that the best worker outputted by the algorithm in round $c + 1$ is an ϵ_c-optimal worker with respect to the best worker outputted in round c, with a probability of least $1 - \delta_c$. The error at each round c is bounded by $O(\sqrt{(\frac{nk}{B}) \ln(\frac{1}{\delta_c})})$.

Adapting from the proof of Lemma 11 in [18], we have the following theoretical results:

Theorem 1 *By applying our adapted ME algorithm, we have:*

$$P[\max_{w_j \in W_c} h_{j,T} \leq \max_{w_i \in W_{c+1}} h_{i,T} + \epsilon_c] \geq 1 - \delta_c, \tag{14}$$

where each worker is assigned $1/(\frac{2}{\epsilon_c^2}) \ln(\frac{3}{\delta_c})$ tasks in round c.

Proof We present the proof in Appendix A.

Algorithm 3 Median Elimination (ME)

```
Input:
    The predicted accuracy $\hat{p}_c$
    The workers remained in the current round $W_c$
Output:
    The selected workers $W_{c+1}$
$w_1, w_2, ..., w_{|W_c|}$ = the workers sorted in non-increasing order of their predicted accuracy $\hat{p}_c$
$W_{c+1} = \{w_1, w_2, ..., w_{\lceil \frac{|W_c|}{2} \rceil}\}$
return $W_{c+1}$
```

According to the above theorem, we can derive the following bound for our allocation scheme:

Theorem 2 *In each round c of Algorithm 4, the error ϵ_c is bounded by $O(\sqrt{(\frac{nk}{B}) \ln(\frac{1}{\delta_c})})$.*

Proof We present the proof in Appendix B.

E. Summary

The algorithm regarding the whole pipeline is summarized in Algorithm 4. Workers with historical profiles (prior domain performance) are first assigned target domain learning tasks for training purposes (Line 9 of Algorithm 4). The annotation accuracy is recorded, then we enter the worker quality estimation phase (Lines 13-14 of Algorithm 4): we perform Cross-domain-aware Performance Estimation to generate an estimation of the worker accuracy, and we further use Learning Gain Estimation to estimate the performance gains. Finally, we perform worker selection (Line 15 of Algorithm 4) by applying Median Elimination in each round to select the best half of workers. After n rounds of looping, we obtain the selected best k workers on the target domain tasks. As for the time complexity, as shown in Algorithm 4, we have n iterations. Let G be the number of gradient descent epochs performed to maximize Equation 5. In each iteration, we perform CPE, LGE, and ME, which take $O(G|W_c|)$, $O(|W_c| \log(|W_c|/k))$, and $O(|W_c| \log(|W_c|))$, respectively. Therefore, the overall time complexity for the worker quality estimation and selection process is $O(n|W|(G + \log(|W|)))$. We do not consider the time for workers to complete the learning tasks when analyzing the time complexity and we will discuss this in Section V-H.

Note that our solution is not restricted to the case where workers have been working on all the D domains. For each domain d, if worker w_i does not have historical record $h_{i,d}$, we can remove the corresponding d^{th} row and line in μ and $\sum$ of Equation (5) for worker w_i and remove the addition term $(\hat{p}_{1,i,d} - h_{i,d})^2$ in Equation (11), so that our approach can still work if any workers have not been working on all the prior domains. In this way, one can easily adapt our approach to suit the general cases.

V. Experiments

A. Datasets

Currently, no publicly available dataset records both the cross-domain worker historical profiles and the worker training

Algorithm 4 General Algorithm

```
Input:
    A set of workers $w_i \in W$, a set of learning tasks $t_j \in T_l$
    Workers' historical accuracy $h_i = \{h_{i,1}, h_{i,2}, ..., h_{i,D}\}$
    Workers' historical task number $n_i = \{n_{i,1}, n_{i,2}, ..., n_{i,D}\}$
    Total budget $B$
    Probability $\delta$
Output:
    The set of selected top $k$ workers $W_T$
Set $n$, $t$ as Equations (12), (13), $W_1 = W$, and $\delta_1 = \delta$
Initialize the current learning task index $r_1 = 1$
for $c = 1, 2, ..., n$ do
    Set $A_c$ to be an empty set
    Set the ground truth labels of tasks $t_{r_c}$ to $t_{r_c+(\lfloor \frac{t}{|W_c|} \rfloor)}$ as $G_c = [g_{1,c}, g_{2,c}, ..., g_{\lfloor \frac{t}{|W_c|} \rfloor, c}]$
    Set the historical accuracy of $W_c$ as $H_c = \{h_1, h_2, ..., h_{|W_c|}\}$
    Set the historical task number of $W_c$ as $N_c = \{n_1, n_2, ..., n_{|W_c|}\}$
    for each $w_i \in W_c$ do
        Assign learning tasks $t_{r_c}$ to $t_{r_c+(\lfloor \frac{t}{|W_c|} \rfloor)}$ to $w_i$ in batches, reveal the correct answers after $w_i$ submits
        Get the answers $a_{i,c}$ of $w_i$ store to $A_c$
    end for
    $r_{c+1} = r_c + (\lfloor \frac{t}{|W_c|} \rfloor)$
    The predicted accuracy $p_c = \text{CPE}(A_c, G_c, H_c, |W_c|)$
    The updated predicted accuracy with learning gains $\hat{p}_c = \text{LGE}(W_c, H_c, N_c, p_1, p_2, ..., p_c)$
    $W_{c+1} = \text{ME}(\hat{p}_c, W_c)$, $\delta_{c+1} = \frac{\delta_c}{2}$
end for
Set $W_T$ to be the top $k$ workers with highest $\hat{p}_n$ in $W_{n+1}$. If $|W_{n+1}| < k$, set $W_T$ to be the top $k$ workers with highest $\hat{p}_{n-1}$ in $W_n$.
return $W_T$
```

TABLE II
DATASET STATISTICS

Datasets	|W|	Q	k	total # of batches	B
RW-1	27	10	7	3	540
RW-2	35	10	9	3	700
S-1	40	20	5	7	2400
S-2	50	20	5	7	3000
S-3	80	20	5	15	6400
S-4	160	20	5	31	16000

process. Therefore, we have to construct new datasets that cover the two aspects of information to evaluate the performance of our method and the baselines. To this end, we build real-world and synthetic datasets, summarized in Table II. We denote the number of learning tasks per batch as Q. The total budget $B = \lceil \log(\frac{|W|}{k}) \rceil * Q * |W|$, $\#$ of batches $= 2^{\lceil \log \frac{|W|}{k} \rceil} - 1$. Note that Q and k are the independent variables, while # of batches and B are dependent variables. We generate different synthetic datasets with different $|W|$ to study the influence of the size of the worker pool.

Real-world datasets: We invited 27 and 35 workers to complete the Qualtrics survey [24] through volunteer recruitment and gMission [11] and denoted as RW-1 and RW-2 datasets. The tasks are Yes/No questions regarding image classification on three prior domains and one target domain. We chose Yes/No questions since many complex question types such as MCQs can be transformed from them [17], [20]. We present the detailed information of the two real-world datasets in Table III. Specifically, RW-1 examines workers' prior domain knowledge of animals (elephant and clownfish) and machines (plane) and evaluates their performance on plants (petunia). The key features that workers need to distinguish petunias from other flowers are color and shape. We also include RW-2 as a complement to RW-1: RW-2 focuses on finer-grained domains where the Peruvian lily, English marigold, and Lenten rose are all flowers. The key features that workers need to focus on differ: Peruvian lilies can be distinguished based on their color, while English marigolds and Lenten roses require detailed observation of petal and stamen shapes. By conducting experiments on both RW-1 and RW-2, we can comprehensively evaluate the performance and robustness of our approach and obtain interesting insights into cross-domain worker training. On each prior domain, each worker is asked to complete two batches of tasks. Each batch consists of 5 learning tasks and 5 working tasks. The answers are recorded to form the historical profiles of the workers. In the target domain, each worker needs to answer 30 learning tasks and 30 working tasks for us to record the worker training process. The learning and working tasks are assigned to each worker in batches. In each batch, workers are required to complete 10 learning tasks first, check the ground truth answers of the learning tasks, and then complete 10 working tasks. The learning tasks are used to train the workers gradually, while the working tasks are applied to test workers' annotation performance in the target domain. A sample learning task is shown in Figure 4. Only the answers to the learning tasks are used as the algorithm input. The answers to the working tasks are used to evaluate the performance.

TABLE III
DETAILS OF REAL-WORLD DATASETS

Dataset	Domain	Features	Knowledge	Sources
RW-1 prior-1	Elephant	Color, Shape	Animal	[44]
RW-1 prior-2	Clownfish	Color, Shape	Animal	[7], [8], [27]
RW-1 prior-3	Plane	Size	Machine	[32]
RW-1 target	Petunia	Color, Shape	Plant	[35]
RW-2 prior-1	Peruvian lily	Color	Plant	[35]
RW-2 prior-2	Red fox	Shape	Animal	[15]
RW-2 prior-3	English marigold	Shape	Plant	[35]
RW-2 target	Lenten rose	Shape	Plant	[35]

Synthetic datasets: We further constructed synthetic datasets based on the distribution of the RW-1. We considered the synthetic datasets with worker pool sizes of $40, 50, 80,$ and 160 to simulate the different supply conditions. We set the number of learning tasks per batch on primal domains and the target domain to 10 and 20, respectively, on S-1, S-2, S-3, and S-4 datasets. We started by modeling the relationship among the four domains with a truncated multivariate normal distribution $N(\mu, \Sigma)$ within $(0, 1)$, where the mean and standard deviation of the three prior domains are computed based on the learning

TABLE IV
MEAN AND STANDARD DEVIATION OF RW-1 AND SYNTHETIC DATASETS

Dataset	Prior 1	Prior 2	Prior 3	Target
RW-1	(0.70, 0.22)	(0.88, 0.10)	(0.58, 0.25)	(0.55, 0.17)
S-1	(0.72, 0.23)	(0.86, 0.13)	(0.53, 0.29)	(0.49, 0.18)
S-2	(0.64, 0.27)	(0.83, 0.15)	(0.51, 0.25)	(0.51, 0.20)
S-3	(0.66, 0.26)	(0.87, 0.13)	(0.54, 0.27)	(0.50, 0.18)
S-4	(0.68, 0.25)	(0.87, 0.13)	(0.54, 0.27)	(0.50, 0.18)

task result of the workers on the three corresponding domains, while the mean and standard deviation of the target domain is calculated based on the first batch learning task results in the RW-1 dataset. The correlation parameters shown in Equation (2) are uniformly random initialized within $(0, 1)$. Each synthetic worker was sampled from N as $[h_1, h_2, h_3, h_T]^\intercal$, where $h_T \in (0, 1)$ denotes the probability that the worker answers the target domain tasks correctly. We can thus obtain the annotation accuracy on the target domain learning tasks with the following *answering rule: randomly select a number x in $(0, 1)$ if $x < h_T$, then the worker answers correctly. Otherwise, the worker answers wrongly.* We obtained the annotation accuracy of synthetic workers on the first batch of learning tasks and applied the modified IRT model in Equation (10) to get the learning parameter α_i for each worker. Then we updated each worker's h_T after each batch based on the modified IRT model with each worker's α_i and the annotation accuracy generated with the *answering rule*. The top-k high-quality workers were selected based on the value of h_T in the last batch.

Consistency: Notice that the synthetic datasets are generated based on RW-1, we now study the consistency between their distributions. As shown in Table IV, the computed mean and standard deviation of the multivariate normal distributions for RW-1 and synthetic datasets are close in the target and prior domains. Besides, we present the distribution of workers' annotation accuracy on the target domain for RW-1, S-1, S-2, S-3, and S-4 in Appendix C Figure 8. The real-world dataset RW-1 and the four synthetic datasets generated have similar distributions on the target domain. Specifically, we bucket the annotation accuracy, compute the Pearson correlations between RW-1 and each synthetic dataset, and find that all Pearson correlations ρ's are larger than 0.75, which validates the consistency.

B. Baselines

In our experiment, we compared our proposed method with the general worker selection algorithms Median Elimination (ME), Uniform Sampling (US) discussed by [10], [18] and Li et al.'s method [30]. We chose these baselines because they do not require additional social interaction information (required by [47], [51]) and are comparable in terms of tasks and goals (Liu et al. [31] focus on optimizing the number of golden questions used, while the worker selection algorithm used is US [18]; Yadav et al. [48] aim at forming high-performance worker teams, which is not comparable under our problem setting). We introduce the baselines as follows:

- Uniform Sampling (US) [10], [18]: Assign each worker the same amount of learning tasks and select the top-k workers that have the highest accuracy.
- Median Elimination (ME) [10], [18]: Assign each worker $\frac{t}{|W_c|}$ learning tasks in round c, after each round, apply Algorithm 3 to select the best half of workers.
- Li et al. [30]: Adopt linear regression on the multiple features of workers and then select workers based on the regressed values. In our experiment, we use the historical profiles as the features for the regression process.

US and ME focus on selecting high-quality workers based on their annotation performance in the worker training process, while the Li et al. approach focuses on employing the historical profile features to identify high-quality workers.

C. Experiment Setting

To ensure fairness, we allocated the same amount of budget for our method and the baselines. The methods' performances are evaluated with respect to the average annotation accuracy of the selected workers on the target domain working tasks in the last round. For example, in RW-1 and RW-2 datasets, the worker medium elimination process terminates in two rounds. We used the annotation accuracy on the working tasks in the second round as the performance criteria. The difficulty parameters of prior domains β_d were initialized as $\beta_d = \ln(\frac{1}{a_d} - 1)$, where a_d is the averaged annotation accuracy for all the real workers on the domain d. For the target domain, we set $\beta_T = 0$ so that Equation (10) would be 0.5 when $K_j = 0$. Since the tasks are Yes/No questions, we believe that $a_T = 0.5$ is a good choice when no prior knowledge regarding the target domain tasks is given. We further conducted a parameter sensitivity experiment in Section V-F to validate our choice. We adopted the same initialization setting regarding the difficulty parameters for the synthetic datasets. The initial multivariate normal distribution $N(\mu, \Sigma)$ was generated as follows: $\mu_1, \mu_2, ..., \mu_D$ and $\sigma_1, \sigma_2, ..., \sigma_D$ were initialized based on the mean and variance of workers' annotation accuracy on the corresponding prior domains; μ_T was initialized to 0.5, and σ_T was initialized as $\frac{1}{D}\sum_{d=1}^{D}\sigma_d$. The correlation parameters were uniformly random initialized in the $(0, 1)$ range. The gradients with respect to $\bar{\mu}$ and $\bar{\Sigma}$ in the log-likelihood function L (Equation (5)) was computed based on backpropagation [45]. We set the learning rates used for gradient descent as $r_1 = 1e - 7$, $r_2 = 1e - 4$, and the epochs for gradient descent as $G = 50$. The average accuracy of each method was recorded. The implementation and data are available at https://github.com/ysunbp/crowdsourcing.

D. Experiment Results

We present the experiment results regarding the best-k worker average annotation accuracy and the relative improvements of our method over baselines in Table V. The ground truth results are presented in the bottom line of Table V. In general, we observe that our method performs the best baseline approach on real-world and synthetic datasets. Specifically, on RW-1 and RW-2 datasets, our method outperforms US [10],

TABLE V
EXPERIMENT RESULTS

	RW-1	RW-2	S-1	S-2	S-3	S-4
US [10], [18]	0.764 (4.5% ↑)	0.956 (0.5% ↑)	0.765 (8.5% ↑)	0.775 (6.8% ↑)	0.815 (4.3% ↑)	0.865 (2.4% ↑)
ME [10], [18]	0.771 (3.5% ↑)	0.944 (1.8% ↑)	0.720 (15.3% ↑)	0.785 (5.5% ↑)	0.795 (6.9% ↑)	0.880 (0.7% ↑)
Li et al. [30]	0.771 (3.5% ↑)	0.936 (2.7% ↑)	0.780 (6.4% ↑)	0.805 (2.9% ↑)	0.845 (0.6% ↑)	0.870 (1.8% ↑)
ME-CPE	0.781 (2.2% ↑)	0.950 (1.2% ↑)	0.785 (5.7% ↑)	0.790 (4.8% ↑)	0.838 (1.4% ↑)	0.875 (1.3% ↑)
Ours	**0.798**	**0.961**	**0.830**	**0.828**	**0.850**	**0.886**
Ground Truth	0.914	1.000	0.885	0.875	0.915	0.975

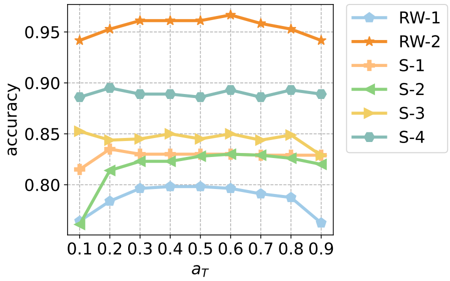


Fig. 5. The sensitivity analysis regarding the initialized annotation accuracy of the target domain: $a_T = \frac{1}{1+e^{\beta_T}}$ on different datasets.

[18] by 4.5% and 0.5%, while boosting the performance of ME [10], [18] by 3.5% and 1.8%, and Li et al. [30] by 3.5% and 2.7%, respectively. On the four synthetic datasets, our method outperforms US, ME, and Li et al. by 5.5%, 7.1%, and 2.9% on average. We attribute the performance uplift of our method over US and ME to the fact that we additionally consider the cross-domain historical profile information of workers and apply proper simulation of the learning gain of workers during the worker training process. The reason why our approach outperforms Li et al. can be attributed to the appropriate worker elimination process applied and the proper simulation of the learning gain of workers. We further notice that the average relative performance uplifts of our approach over the three baselines are 10.1%, 5.1%, 3.9%, and 1.6%, which decrease as the number of workers increases. We attribute this phenomenon to the fact that the number of high-performance workers increases as the size of the worker pool gets large. As a result, the accuracy difference induced by different worker selection strategies of different approaches is likely to decrease, and thus the performance uplift becomes smaller as the size of the worker pool increases. Overall, our method performs persistently well on both real-world and synthetic datasets, which implies the effectiveness and robustness of our approach for cross-domain worker selection.

E. Ablation Study

We conducted an ablation study regarding the following variants to understand the mechanism of different components:

- ME [10], [18]: ME is the backbone of our method, where the CPE and LGE (Worker Quality Estimation) are removed.
- ME-CPE: ME-CPE is a variant of our method where the LGE component is removed.

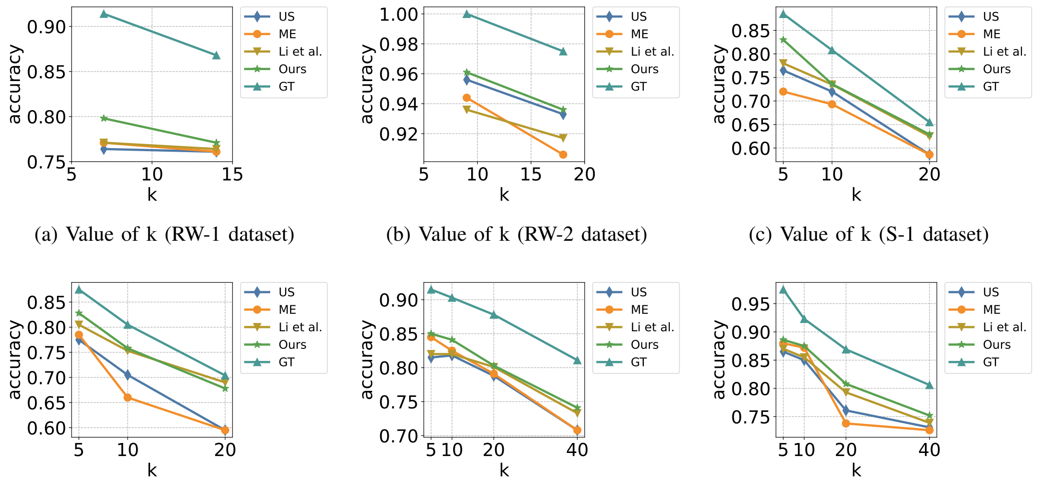
We present the experiment results in Table V. We compare ME [10], [18] with ME-CPE to demonstrate the effect of the Cross-domain-aware Performance Estimation. The comparison between ME-CPE and our method shows the influence of the Learning Gain Estimation. First, we note that ME-CPE outperforms ME by 1.3% and 0.6% and our method further boosts the performance of ME-CPE by 2.2% and 1.2% on RW-1 and RW-2 datasets. This shows the effectiveness brought by CPE and LGE in estimating worker quality. CPE helps the algorithm capture the cross-domain information to estimate the annotation ability of workers, while LGE captures the learning gain of workers during the worker training process. On S-1, S-2, S-3, and S-4 datasets, our method improves over ME-CPE by 5.7%, 4.8%, 1.4%, and 1.3% respectively. We attribute the performance improvement of our method over ME-CPE to the LGE component, which obtains a more accurate estimation of workers' performance in the target domain during training. ME-CPE outperforms ME by relatively 9.0%, 0.6%, and 5.4% on S-1, S-2, and S-3 datasets while performing slightly worse than ME on the S-4 dataset. On average, ME-CPE relatively improves the performance of ME by 3.6% on the four synthetic datasets. In general, ME-CPE can improve or achieve comparable performance as ME. The CPE component can effectively capture the cross-domain information, which is helpful for identifying high-quality workers.

F. Method Parameter Sensitivity

We analyzed the impact of the critical parameter of our method a_T, which is the initialized annotation accuracy of the target domain and is related to the initialization of the difficulty parameter β_T ($a_T = \frac{1}{1+e^{\beta_T}}$). Figure 5 presents the results. We notice that the performance of our method is relatively stable when the value of a_T is set within the range $[0.2, 0.8]$. In practice, we suggest setting a_t according to the difficulty level of the tasks. If the tasks are relatively easy, a large value of a_T can be adopted. Otherwise, a small value of a_T should be considered. If no domain knowledge regarding the difficulty level is available, we can set the value of a_T based on the nature of the tasks. Since our tasks are Yes/No quesetions, a natural choice of a_T would be 0.5. The sensitivity analysis coincides with our selection of a_T in Section V-C: our method achieves a stable and good performance when $a_T = 0.5$.

G. Dataset Parameter Sensitivity

To comprehensively compare the performance of our method and baselines, we further conducted experiments regarding the important dataset parameters: the number of selected workers k and the number of learning tasks per batch Q. Since the number of learning tasks per batch cannot be changed once the RW datasets are collected, we conducted experiments on the four synthetic datasets to analyze its effect.



(a) Value of k (RW-1 dataset)

(b) Value of k (RW-2 dataset)

(c) Value of k (S-1 dataset)

(d) Value of k (S-2 dataset)

(e) Value of k (S-3 dataset)

(f) Value of k (S-4 dataset)

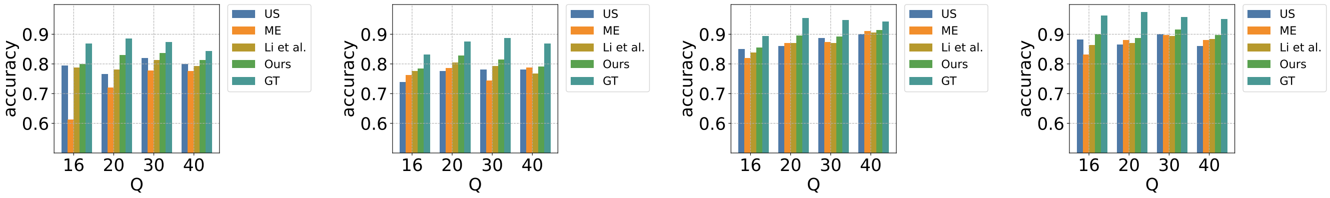
Fig. 6. Parameter sensitivity experiments of the number of selected workers.

The value of k determines the number of rounds n required to obtain the top-k workers. As shown in Table II, the values of k used in our main experiments are 7 and 9 for the RW-1 and RW-2 datasets and 5 for the four synthetic datasets, which leads to 2 rounds for RW-1 and RW-2 datasets, 3 rounds for the S-1 and S-2 datasets, 4 rounds for the S-3 dataset, and 5 rounds for the S-4 dataset. In the parameter sensitivity experiment presented in this section, we further increased the value of k to 14 and 18 for RW-1 and RW-2 datasets, 10 and 20 for the S-1 and S-2 datasets, and 10, 20, and 40 for the S-3 and S-4 datasets. The reason for experimenting with the increased number of k is to present a comprehensive view of the performance of our approach from the beginning stage of worker selection (when the value of k is large) to the ending stage of worker selection (when the value of k is small). For example, when changing the value of k from 14 to 7 on the RW-1 dataset, we can analyze the performance change of our method when conducting one round and two rounds of eliminations. As shown in Figures 6a and 6b, on RW-1 and RW-2 datasets, our method consistently outperforms all the baseline approaches, when we increase the value of k. On the synthetic datasets S-1, S-3, and S-4, as shown in Figures 6c, 6e, and 6f, our approach still outperforms all the baselines when the value of k increases. On the S-2 dataset (Figure 6d), our approach outperforms all the baselines when the value of k is set to 5 and 10, while is slightly worse than the performance of Li et al. [30] when the value of k further increase to 20. We further observe that on the RW-1 dataset, when $k = 14$, our approach and Li et al. have similar performance. Similar phenomena can also be observed on the S-1, S-3, and S-4 datasets when we set the value of k to 20, 40, and 40. We attribute this to the fact that when the value of k is large (i.e., the model is at the beginning stage of elimination), the long-term learning improvement of the workers on the target domain is not yet significant, the linear regression approach introduced by Li et al. [30] can capture the static cross-domain information. However, as the value of k decreases (i.e., the elimination process proceeds), the dynamic cross-domain performance estimation and the learning gain estimation help our approach to outperform Li et al. [30].

As for the number of learning tasks Q, the default value is 20. We changed the value of Q to 16, 30, and 40, while keeping the value of k unchanged with a changing total budget B in this section to evaluate its influence and presented the experimental results in Figure 7. We first notice that our approach consistently outperforms all baselines on four synthetic datasets with different Q. We further observe that on four synthetic datasets, the performance of our approach and baselines tends to get close when Q increases. We attribute this phenomenon to the fact that when the budget is arbitrarily large, the improvement brought by adopting cross-domain information is reduced since the algorithm can get an accurate estimation of workers' target domain knowledge based on a large amount of learning tasks assigned to workers. However, when the total budget is small, our approach efficiently utilizes the cross-domain information to boost the worker selection performance of ME. In real-world applications, selecting workers effectively with a relatively small number of learning tasks is crucial, since in reality, the ground truth answers of golden questions in each domain require manual collection and thus are hard to obtain. In this sense, the performance uplift of our approach over other baselines when Q is small is favored and useful for real-world applications.

H. Discussion

We recorded the running time results on one Intel Xeon Gold 6240 CPU @ 2.60GHz. Our method takes 3.9s, 5.0s, 6.3s, 7.8s, 13.4s, and 28.9s to select the best workers on the RW-1, RW-2, S-1, S-2, S-3, and S-4 datasets. Compared with the median completion times of our two surveys (1185s and 986s), the running time cost of our method is acceptable. As suggested by [53], the average completion time of tasks on AMT usually takes hours; we believe that the running time can facilitate the needs for real-world crowdsourcing applications.



(a) Learning tasks Q (S-1 dataset) (b) Learning tasks Q (S-2 dataset) (c) Learning tasks Q (S-3 dataset) (d) Learning tasks Q (S-4 dataset)

Fig. 7. The parameter sensitivity experiments of the number of learning tasks per batch Q.

The time used for workers to complete learning tasks inside the survey is approximately 500s for the two surveys. We observe that the average accuracy of all workers increases from 0.55 to 0.79 and from 0.65 to 0.85 on RW-1 and RW-2 respectively after a single round of worker training with 10 questions. Despite the additional training time required, we believe that by introducing worker training into the worker selection process of crowdsourcing, we can significantly improve the overall worker performance in the target domain. As for the cost of the worker training process: denote the number of learning tasks and working tasks assigned as $|T_l|$ and $|T_w|$, the annotation accuracy before and after the worker training as a_t and a'_t. For simplicity, we consider the effect on one single worker with one round of training, assume the worker's accuracy is the same as the average accuracy of all workers, and consider the same monetary cost for completing each learning and working task. If $|T_w|/|T_l| > \frac{a_t}{a'_t - a_t}$, under the same total worker learning and working budget, the number of correctly annotated samples in T_w for the worker with worker training process would be greater than that without worker training. In our case, once $|T_w|/|T_l|$ is greater than 2.3 and 3.3 for RW-1 and RW-2 respectively, then the additional monetary cost can be counteracted. Furthermore, our worker training phase interacts with the workers by revealing the correct answers to learning tasks to workers promptly. Throughout multiple worker training rounds, workers can learn about their overall improved performance in the target domain, have a sense of accomplishment, and obtain new skills related to the target domain. As discussed by previous works [14], [16], [22], [42], with timely feedback received and new skills learned, workers tend to have improved engagement and performance. Therefore, we believe the worker training phase improves the engagement of workers, and stimulates workers to explore more useful features in the target domain.

We further report the estimated correlation between domains on the RW-1 and RW-2 datasets. Specifically, the correlation parameters estimated by our method are 0.50, 0.69, and 0.65 for Plane-Flower (P-F), Fish-Flower (F-F), and Elephant-Flower (E-F) on RW-1 and 0.23, 0.10, and 0.68 for Peruvian lily-Lenten rose (P-L), Red fox-Lenten rose (R-L), and English marigold-Lenten rose (E-L) on RW-2. The correlation parameters for F-F and E-F are larger than that for P-F, which coincides with our intuition that workers sensitive to color and shape differences (good at fish and elephant domains) are likely to perform well in distinguishing flowers. The correlation parameters for P-L and E-L are larger than that for R-L, which means the workers who are good at distinguishing other flowers are good at distinguishing Lenten roses. Besides, the correlation for E-L is larger than P-L. To distinguish English marigold from its counter-parts, workers need to notice the small differences in petals and stamen (shape), which is close to the requirement of distinguishing Lenten rose; while to identify the Peruvian lily from its counter-parts, workers only need to pay attention to color difference. As a result, the correlation for E-L is larger than P-L. More details of the prior and target domains are presented in Table III.

VI. Conclusion

In this paper, we formulated the cross-domain-aware worker selection with training problem and proposed a novel algorithm based on Medium Elimination to resolve it. Specifically, two estimation components CPE and LGE are designed to incorporate cross-domain knowledge information and capture the learning gains during worker training. Real-world and synthetic cross-domain-aware worker selection with training datasets were collected to evaluate different approaches. We conducted extensive experiments on real-world and synthetic datasets to show that our method outperforms all the state-of-the-art baselines on real-world and synthetic datasets. We confirm that applying CPE and LGE can capture cross-domain knowledge information and estimate the learning gain during the worker training process. As a future direction, we aim to construct a unified multi-domain taxonomy that optimizes the worker training and selection process.

Acknowledgment

Jiachuan Wang and Libin Zheng are the corresponding authors. Peng Cheng's work is supported by the National Natural Science Foundation of China under Grant No. 62102149. Libin Zheng is supported by the National Natural Science Foundation of China No. 62102463 and the Natural Science Foundation of Guangdong Province of China No. 2022A1515011135. Lei Chen's work is partially supported by National Key Research and Development Program of China Grant No. 2023YFF0725100, National Science Foundation of China (NSFC) under Grant No. U22B2060, the Hong Kong RGC GRF Project 16213620, RIF Project R6020-19, AOE Project AoE/E-603/18, Theme-based project TRS T41-603/20R, CRF Project C2004-21G, Guangdong Province Science and Technology Plan Project 2023A0505030011, Hong Kong ITC ITF grants MHX/078/21 and PRP/004/22FX, Zhujiang scholar program 2021JC02X170, Microsoft Research Asia Collaborative Research Grant and HKUST-Webank joint research lab grants.

REFERENCES

[1] "Alibaba-platform," 2023. [Online]. Available: https://zhongbao.aliyun.com/

[2] "Baidu-platform," 2023. [Online]. Available: https://zhongbao.baidu.com/

[3] "Jd-platform," 2023. [Online]. Available: https://biao.jd.com/

[4] G. Abdelrahman and Q. Wang, "Knowledge tracing with sequential key-value memory networks," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2019, pp. 175–184.

[5] G. Abdelrahman, Q. Wang, and B. P. Nunes, "Knowledge tracing: A survey," *ACM Computing Surveys*, 2022, just Accepted.

[6] T. Awwad, N. Bennani, K. Ziegler, V. Sonigo, L. Brunie, and H. Kosch, "Efficient worker selection through history-based learning in crowdsourcing," in *2017 IEEE 41st Annual Computer Software and Applications Conference*, vol. 1. IEEE, 2017, pp. 923–928.

[7] B. J. Boom, P. X. Huang, C. Beyan, C. Spampinato, S. Palazzo, J. He, E. Beauxis-Aussalet, S.-I. Lin, H.-M. Chou, G. Nadarajan *et al.*, "Long-term underwater camera surveillance for monitoring and analysis of fish populations," in *Proceedings of Workshop on Visual Observation and Analysis of Animal and Insect Behavior*. Curran Associates Inc., 2012.

[8] B. J. Boom, P. X. Huang, J. He, and R. B. Fisher, "Supporting ground-truth annotation of image datasets using clustering," in *Proceedings of the 21st International Conference on Pattern Recognition*. IEEE, 2012, pp. 1542–1545.

[9] C. C. Cao, J. She, Y. Tong, and L. Chen, "Whom to ask? jury selection for decision making tasks on micro-blog services," *Proceedings of the VLDB Endowment*, vol. 5, no. 11, pp. 1495–1506, 2012.

[10] W. Cao, J. Li, Y. Tao, and Z. Li, "On top-k selection in multi-armed bandits and hidden bipartite graphs," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1. MIT Press, 2015, pp. 1036–1044.

[11] Z. Chen, R. Fu, Z. Zhao, Z. Liu, L. Xia, L. Chen, P. Cheng, C. C. Cao, Y. Tong, and C. J. Zhang, "gmission: A general spatial crowdsourcing platform," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1629–1632, 2014.

[12] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modeling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[13] R. S. d Baker, A. T. Corbett, and V. Aleven, "More accurate student modeling through contextual estimation of slip and guess probabilities in bayesian knowledge tracing," in *International Conference on Intelligent Tutoring Systems*. Springer, 2008, pp. 406–415.

[14] R. de Leon Pereira, A. Tan, A. Bunt, and O. Tremblay-Savard, "Increasing player engagement, retention and performance through the inclusion of educational content in a citizen science game," in *Proceedings of the 16th International Conference on the Foundations of Digital Games*, 2021, pp. 1–12.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.

[16] M. Dontcheva, R. R. Morris, J. R. Brandt, and E. M. Gerber, "Combining crowdsourcing and learning to improve engagement and performance," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, pp. 3379–3388.

[17] A. Dudley, "Multiple dichotomous-scored items in second language testing: Investigating the multiple true-false item type under norm-referenced conditions," *Language Testing*, vol. 23, no. 2, pp. 198–228, 2006.

[18] E. Even-Dar, S. Mannor, Y. Mansour, and S. Mahadevan, "Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems," *Journal of Machine Learning Research*, vol. 7, no. 6, pp. 1079–1105, 2006.

[19] U. Gadiraju, B. Fetahu, and R. Kawase, "Training workers for improving performance in crowdsourcing microtasks," in *European Conference on Technology Enhanced Learning*. Springer, 2015, pp. 100–114.

[20] K. Green, "Multiple choice and true-false: reliability and validity compared," *The Journal of Experimental Education*, vol. 48, no. 1, pp. 42–44, 1979.

[21] D. Haas, J. Ansel, L. Gu, and A. Marcus, "Argonaut: Macrotask crowdsourcing for complex data processing," *Proceedings of the VLDB Endowment*, vol. 8, no. 12, pp. 1642–1653, 2015.

[22] J. R. Hackman and G. R. Oldham, "Motivation through the design of work: Test of a theory," *Organizational behavior and human performance*, vol. 16, no. 2, pp. 250–279, 1976.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] Q. Inc., "Qualtrics," 2023, retrieved on Jan 1, 2023. [Online]. Available: https://www.qualtrics.com

[25] A. Jain, H. Patel, L. Nagalapatti, N. Gupta, S. Mehta, S. Guttula, S. Mujumdar, S. Afzal, R. Sharma Mittal, and V. Munigala, "Overview and importance of data quality for machine learning tasks," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2020, pp. 3561–3562.

[26] M. Khajah, R. Wing, R. V. Lindsey, and M. Mozer, "Integrating latent-factor and knowledge-tracing models to predict individual differences in learning," in *Proceedings of the 7th International Conference on Educational Data Mining*. IEDMS, 2014, pp. 99–106.

[27] Koto, "Heywhale marine life data," 2020, retrieved on Jan 1, 2023. [Online]. Available: https://www.heywhale.com/mw/dataset/5e55f7960e2b66002c245df5

[28] J. I. Lee and E. Brunskill, "The impact on individualizing student models on necessary practice opportunities," in *Proceedings of the 5th International Conference on Educational Data Mining*. IEDMS, 2012, pp. 118–125.

[29] H. Li and Q. Liu, "Cheaper and better: Selecting good workers for crowdsourcing," in *Proceedings of the AAAI Conference on Human Computation and Crowdsourcing*, vol. 3. AAAI, 2015, pp. 20–21.

[30] H. Li, B. Zhao, and A. Fuxman, "The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 165–176.

[31] Q. Liu, A. T. Ihler, and M. Steyvers, "Scoring workers in crowdsourcing: How many control questions are enough?" in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, vol. 2. Curran Associates Inc., 2013, pp. 1914–1922.

[32] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi, "Fine-grained visual classification of aircraft," *Arxiv Preprint*, 2013.

[33] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. ACL, 2016, pp. 1400–1409.

[34] S. Minn, Y. Yu, M. C. Desmarais, F. Zhu, and J.-J. Vie, "Deep knowledge tracing and dynamic student classification for knowledge tracing," in *2018 IEEE International Conference on Data Mining*. IEEE, 2018, pp. 1182–1187.

[35] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*. IEEE, 2008, pp. 722–729.

[36] Z. A. Pardos and N. T. Heffernan, "Kt-idem: Introducing item difficulty to the knowledge tracing model," in *International Conference on User Modeling, Adaptation, and Personalization*. Springer, 2011, pp. 243–254.

[37] P. I. Pavlik, H. Cen, and K. R. Koedinger, "Performance factors analysis–a new alternative to knowledge tracing," in *Proceedings of the 2009 conference on Artificial Intelligence in Education: Building Learning Systems that Care: From Knowledge Representation to Affective Modelling*. IOS Press, 2009, pp. 531–538.

[38] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein, "Deep knowledge tracing," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, vol. 1. MIT Press, 2015, pp. 505–513.

[39] G. Rasch, *Probabilistic models for some intelligence and attainment tests*. ERIC, 1993.

[40] C. Shan, N. Mamoulis, G. Li, R. Cheng, Z. Huang, and Y. Zheng, "A crowdsourcing framework for collecting tabular data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2060–2074, 2019.

[41] S. Shen, M. Ji, Z. Wu, and X. Yang, "An optimization approach for worker selection in crowdsourcing systems," *Computers & Industrial Engineering*, vol. 173, 2022.

[42] T. L.-P. Tang and L. Sarsfield-Baldwin, "The effects of self-esteem, task label, and performance feedback on task liking and intrinsic motivation," *The Journal of Social Psychology*, vol. 131, no. 4, pp. 567–572, 1991.

[43] Y. L. Tong, *The multivariate normal distribution*. Springer Science & Business Media, 2012.

[44] Vivek, "Kaggle elephant data," 2022, retrieved on Jan 1, 2023. [Online]. Available: https://www.kaggle.com/datasets/vivmankar/asian-vs-african-elephant-image-classification

[45] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[46] K. H. Wilson, Y. Karklin, B. Han, and C. Ekanadham, "Back to the basics: Bayesian extensions of irt outperform neural networks for proficiency estimation," in *Proceedings of the 9th International Conference on Educational Data Mining*. ERIC, 2016, pp. 539–544.

[47] G. Wu, Z. Chen, J. Liu, D. Han, and B. Qiao, "Task assignment for social-oriented crowdsourcing," *Frontiers of Computer Science*, vol. 15, no. 2, 2021.

[48] A. Yadav, S. Mishra, and A. S. Sairam, "A multi-objective worker selection scheme in crowdsourced platforms using nsga-ii," *Expert Systems with Applications*, vol. 201, 2022.

[49] M. V. Yudelson, K. R. Koedinger, and G. J. Gordon, "Individualized bayesian knowledge tracing models," in *International Conference on Artificial Intelligence in Education*. Springer, 2013, pp. 171–180.

[50] J. Zhang, X. Shi, I. King, and D.-Y. Yeung, "Dynamic key-value memory networks for knowledge tracing," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 765–774.

[51] Y. Zhao, G. Liu, K. Zheng, A. Liu, Z. Li, and X. Zhou, "A context-aware approach for trustworthy worker selection in social crowd," *World Wide Web*, vol. 20, no. 6, pp. 1211–1235, 2017.

[52] Z. Zhao, F. Wei, M. Zhou, W. Chen, and W. Ng, "Crowd-selection query processing in crowdsourcing databases: A task-driven approach," in *Proceedings 26th International Conference on Extending Database Technology*. OpenProceedings.org, 2015, pp. 397–408.

[53] L. Zheng and L. Chen, "Dlta: A framework for dynamic crowdsourcing classification tasks," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 5, pp. 867–879, 2018.

Appendix A
Proof of Theorem 1

Theorem 1 *By applying our adapted ME algorithm, we have:*

$$P[\max_{w_j\in W_c} h_{j,T} \le \max_{w_i\in W_{c+1}} h_{i,T} + \epsilon_c] \ge 1-\delta_c, \tag{15}$$

where each worker is assigned $(\frac{2}{\epsilon_c^2})\ln(\frac{3}{\delta_c})$ tasks in round c.

Proof Our proof is transformed from the proof of Lemma 11 in [18]. Without loss of generality, we look at the c-th round and let h_0 be the historical performance of the best worker. Let event $E_c = \{\hat{h}_0 < h_0 - \frac{\epsilon_c}{2}\}$, which means the empirical estimate of the best worker is pessimistic. According to Theorem 6 in [18]:

$$P[\hat{h}_0 < h_0 - \epsilon/2] \le \exp[-2(\epsilon_c/2)^2 l], \tag{16}$$

where $1/l$ is the number of tasks assigned to each worker in round c. Plug in $l = (\frac{2}{\epsilon_c^2})\ln(\frac{3}{\delta_c})$ we have $P[E_c] \le \frac{\delta_c}{3}$.

In the case when E_c does not hold, according to [18], the probability that a worker w_j which is not ϵ_c-optimal yet is empirical better than the best worker is:

$$\begin{aligned}
&P[\hat{h}_{j,T} \ge \hat{h}_0 | \hat{h}_0 \ge h_0 - \epsilon_c/2] \\
&\le P[\hat{h}_{j,T} \ge h_j + \epsilon_c/2 | \hat{h}_0 \ge h_0 - \epsilon_c/2] \\
&\le \delta_c/3.
\end{aligned} \tag{17}$$

Following the formulation of *bad arms*, according to Lemma 11 of [18], we have $\mathbb{E}[\#\text{bad} | \hat{h}_0 \ge h_0 - \frac{\epsilon_c}{2}] \le n\frac{\delta_c}{3}$. Following the proof in [18], we apply Markov inequality to obtain:

$$P[\#\text{bad} \ge n/2 | \hat{h}_0 \ge h_0 - \epsilon_c/2] \le 2\delta_c/3. \tag{18}$$

Apply union bound as done by [18], we have:

$$P[\max_{w_j\in W_c} h_{j,T} > \max_{w_i\in W_{c+1}} h_{i,T} + \epsilon_c] \le \delta_c, \tag{19}$$

which is equivalent to the result we desired.

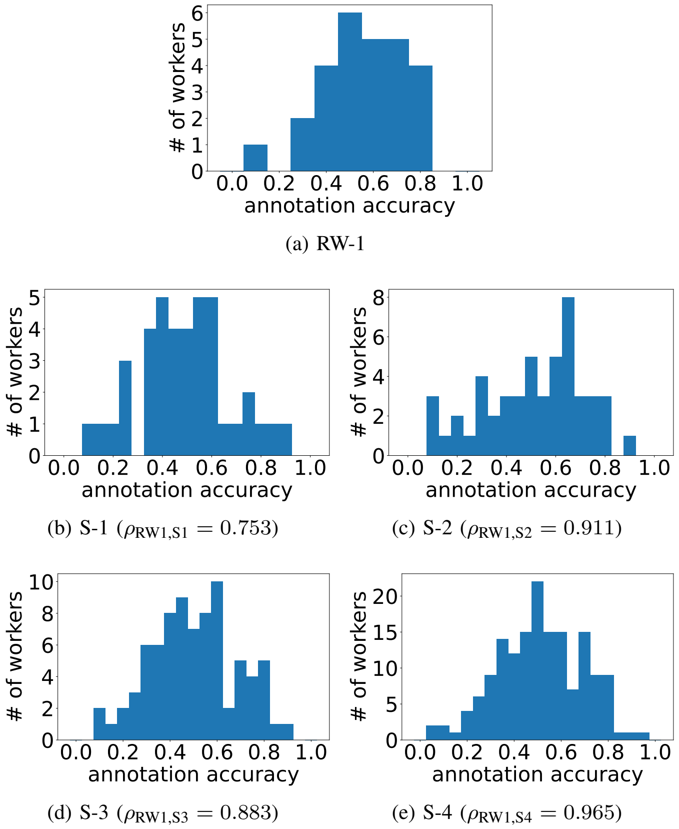
Appendix B
Proof of Theorem 2

Theorem 2 *In each round c of Algorithm 4, the error ϵ_c is bounded by $O(\sqrt{(\frac{nk}{B})\ln(\frac{1}{\delta_c})})$.*

Proof Note that each worker is assigned $\frac{2}{\epsilon_c^2}\ln\frac{3}{\delta_c}$ tasks in round c, therefore:

$$\frac{2}{\epsilon_c^2}\ln\frac{3}{\delta_c} = \frac{B}{n\frac{|W|}{2^{c-1}}} \tag{20}$$

$$\begin{aligned}
\epsilon_c &= \sqrt{\frac{2n|W|}{2^{c-1}B}\ln\frac{3}{\delta_c}} \\
&= \sqrt{\frac{4n|W|}{2^{\log_2\frac{|W|}{k}}B}\ln\frac{3}{\delta_c}} \\
&= O\left(\sqrt{\frac{nk}{B}\ln\frac{1}{\delta_c}}\right).
\end{aligned} \tag{21}$$



(a) RW-1

(b) S-1 ($\rho_{\text{RW1,S1}} = 0.753$)

(c) S-2 ($\rho_{\text{RW1,S2}} = 0.911$)

(d) S-3 ($\rho_{\text{RW1,S3}} = 0.883$)

(e) S-4 ($\rho_{\text{RW1,S4}} = 0.965$)

Fig. 8. The target domain accuracy distribution of datasets before worker training. We also present the Pearson correlations ρ for each synthetic dataset.

Appendix C
Correlations between Datasets

We present the detailed distribution of the RW1 and the synthetic datasets in Figure 8. The real-world dataset RW-1 and the four synthetic datasets generated have similar distributions on the target domain. Specifically, we bucket the annotation accuracy, compute the Pearson correlations between RW-1 and each synthetic dataset, and find that all Pearson correlations ρ's are larger than 0.75, which validates the consistency.