# Incentivizing High Quality Crowdsourcing Information using Bayesian Inference and Reinforcement Learning

**Anonymous Authors**[1]


## Abstract

This document provides a basic paper template and submission guidelines. Abstracts must be a single paragraph, ideally between 4–6 sentences long. Gross violations will trigger corrections at the camera-ready phase.


## 1. Introduction

123

## 2. Related Work

123

## 3. Problem Formulation

Suppose one data requester assigns $M$ tasks with binary answer space $\{1, 2\}$ to $N \geq 3$ candidate workers at each time step $t$. We denote the tasks and workers by $\mathcal{T}^t = \{1, 2, \ldots, M\}$ and $\mathcal{C} = \{1, 2, \ldots, N\}$, respectively. Meanwhile, we use $L_i^t(j)$ to denote the label generated by worker $i \in \mathcal{C}$ for task $j \in \mathcal{T}^t$. If $L_i^t(j) = 0$, we mean that task $j$ is not assigned to worker $i$ at step $t$.

The generated label $L_i^t(j)$ depends both on the ground-truth label $L^t(j)$ and worker $i$'s effort level $e_i^t$ and reporting strategy $r_i^t$. Any worker $i$ can potentially have two effort levels, High ($e_i^t = 1$) and Low ($e_i^t = 0$). Also, he/she can decide either to truthfully report his observation $r_i^t = 1$ or to revert the answer $r_i^t = 0$. Workers may act differently for different tasks. We thus define $e_i^t \in [0, 1]$ and $r_i^t \in [0, 1]$ as worker $i$'s probability of exerting high efforts and being truthful, respectively. In this case, worker $i$'s probability of being correct (PoBC) can be computed as

$$
\begin{aligned}
&p_i^t = r_i^t e_i^t p_{i,H} + r_i^t (1 - e_i^t) p_{i,L} + \\
&(1 - r_i^t) e_i^t (1 - p_{i,H}) + (1 - r_i^t)(1 - e_i^t)(1 - p_{i,L})
\end{aligned}
\tag{1}
$$

where $p_{i,H}$ and $p_{i,L}$ denote worker $i$'s probability of observing the correct label when exerting high and low efforts, respectively. Following (**??**), we assume that the tasks are homogeneous and the workers share the same set of $p_{i,H}, p_{i,L}$, denoting by $p_H, p_L$, and $p_H > p_L = 0.5$. Here, $p_i^t = 0.5$ means that worker $i$ randomly selects a label to report.

The data requester needs to pay each worker some money as the incentive for providing labels. We denote the payment for worker $i$ at step $t$ as $P_i^t$. At the beginning of each time step, the data requester promises the workers a certain rule of payment determination which acts the contract between two sides and cannot be changed until the next time step. The workers are self-interested and may change their reporting strategies ($e_i^t$ and $r_i^t$) according to the payment rule. Workers' different reporting strategies will lead to the different values of workers' PoBCs and finally different levels of label quality. After collecting the labels from the workers, the data requester will infer the true labels by using a certain inference algorithm, and (**?**) provide a good survey of existing inference algorithms. Denote the the inferred true label of task $j$ by $\tilde{L}^t(j)$. Then, the label accuracy $A^t$ and the utility $u^t$ of the data requester satisfy

$$
\begin{aligned}
A^t &= \frac{1}{M} \sum\nolimits_{j=1}^{M} 1 \left[ \tilde{L}^t(j) = L^t(j) \right] \\
u^t &= F(A^t) - \eta \sum\nolimits_{i=1}^{N} P_i^t
\end{aligned}
\tag{2}
$$

where $F(\cdot)$ is a non-decreasing monotonic function mapping accuracy to utility and $\eta$ is a tunable parameter balancing label quality and costs. Intuitively, the $F(\cdot)$ function needs to be non-deceasing as higher accuracy is preferred.

The number of tasks in crowdsourcing is often very large, and the interaction between tasks and workers may last for hundreds of time steps. Thus, we introduce the cumulative utility $U(t)$ of the data requester from the current step $t$ as

$$
U(t) = \sum\nolimits_{k=t}^{\infty} \rho^{k-t} u^t \tag{3}
$$

where $0 \leq \rho < 1$ is the discount factor which determines the importance of future utilities. The objective of our study is to maximize $U(t)$ by optimally designing the payment rule and the ex-post adjustment algorithm of the payment rule, which we call as the incentive mechanism.

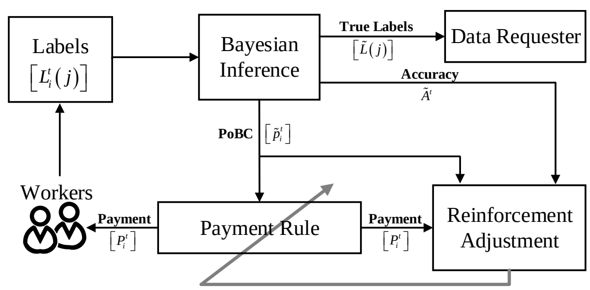



*Figure 1.* Layout of our incentive mechanism.

## 4. Incentive Mechanism for Crowdsourcing

We present the layout of our incentive mechanism in Figure **??**, where the estimate of a variable is denoted by adding an over-tilde. In our incentive mechanism, the Bayesian inference algorithm is responsible for estimating the true labels, workers' PoBCs and the label accuracy based on the collected labels at each time step. The payment rule is designed to ensure that reporting truthfully ($r_i^t = 0$) and exerting high efforts ($e_i^t = 1$) is the payment-maximizing strategy for all workers at any time step. By doing so, we wish to induce workers to generate high-quality labels and thus improve the label accuracy. The reinforcement learning algorithm adjusts the payment rule based on the historical data of payments, workers' PoBCs and the accuracy of aggregate labels. In this way, we can optimally balance the utility got from the labels and lost in the payments, which corresponds to $F(A^t)$ and $\sum_i P_i^t$ in Equation 2, respectively. Besides, our incentive mechanism can ensure that always reporting truthfully ($r_i^t \equiv 0$) and exerting high efforts ($e_i^t \equiv 1$) is the payment-maximizing strategy for workers in the long term. This property prevents the clever manipulation which earns higher long-term benefits by sacrificing short-term ones.

Nevertheless, there are three challenges to achieve our design. Firstly, our empirical studies reveal that popular inference algorithms may be heavily biased towards overestimating the accuracy when the quality of labels is very low. For example, when there are 10 workers and $q_i^t = 0.55$, the estimated label accuracy of the EM estimator (**??**) stays at around 0.9 while the real accuracy is only around 0.5. This heavy bias will cause the utility to be miscalculated and thus mislead our reinforcement adjustment. To reduce the inference bias, we develop our Bayesian inference algorithm by introducing the soft Dirichlet priors to both the true labels and workers' PoBCs. In this case, the posterior distribution cannot be expressed as any known distributions, which motivates us to derive the explicit posterior distribution at first and then employ Gibbs sampling to conduct inference. Secondly, the reinforcement adjustment expects the utility to be accurately calculated so that the direction of adjustment is clear. However, both the label accuracy and workers' PoBCs in our incentive mechanism are corrupted by noise. Considering that these estimates are calculated as an average over $M$ tasks, the central limit theorem ensures that the inference noise approaches the Gaussian distribution. Therefore, to overcome the inference noise, we develop our reinforcement adjustment algorithm based on the Gaussian process. Lastly, the biggest challenge of our study is to prove that our incentive mechanism can ensure that reporting truthfully and exerting high efforts is the payment-maximizing strategy for workers in not only each time step and but also the long term. For clarity, we put the theoretical analysis in the next section. In this section, we focus on the first two challenges.

### 4.1. Payment Rule

Suppose, at time step $t$, worker $i$ finishes $M_i^t$ tasks. Then, the payment for worker $i$ should be

$$P_i^t = M_i^t \cdot (a^t r_i^t + b) \;\;,\;\; \phi_i^t = \tilde{p}_i^t - 0.5 \tag{4}$$

where we call $\phi_i^t$ as worker $i$'s score and $\tilde{p}_i^t$ will be calculated by our Bayesian inference algorithm. $a^t$ is the scaling factor. It is determined by our reinforcement adjustment algorithm at the beginning of step $t$. We denote all the available values of $a^t$ as set $\mathcal{A}$. Besides, $b \geq 0$ is the fixed base payment.

### 4.2. Bayesian Inference

Now, we present the details of our inference algorithm. For the simplicity of notations, we omit the superscript $t$ in this subsection. The joint distribution of the collected labels $\mathcal{L} = [L_i(j)]$ and the true labels $\boldsymbol{L} = [L(j)]$ satisfies

$$P(\mathcal{L}, \boldsymbol{L}|\boldsymbol{p}, \boldsymbol{\tau}) = \prod\nolimits_{j=1}^{M}\prod\nolimits_{k=1}^{K}\left\{\tau_k \prod_{i=1}^{N} p_i^{\delta_{ijk}}(1-p_i)^{\delta_{ij(3-k)}}\right\}^{\xi_{jk}} \tag{5}$$

where $\boldsymbol{p} = [p_i]_N$ and $\boldsymbol{\tau} = [\tau_1, \tau_2]$. $\tau_1$ and $\tau_2$ denote the distribution of answer 1 and 2 among all tasks, respectively. Besides, $\delta_{ijk} = \mathbb{1}(L_i(j) = k)$ and $\xi_{jk} = \mathbb{1}(L(j) = k)$. Here, we assume Dirichlet priors Dir$(\cdot)$ for $p_i$ and $\boldsymbol{\tau}$ as

$$[p_i, 1-p_i] \sim \text{Dir}(\alpha_1, \alpha_2) \;,\; \boldsymbol{\tau} \sim \text{Dir}(\beta_1, \beta_2). \tag{6}$$

Then, the joint distribution of $\mathcal{L}$, $\boldsymbol{L}$, $\boldsymbol{p}$ and $\boldsymbol{\tau}$ satisfies

$$P(\mathcal{L}, \boldsymbol{L}, \boldsymbol{p}, \boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = P(\mathcal{L}, \boldsymbol{L}|\boldsymbol{p}, \boldsymbol{\tau}) \cdot P(\boldsymbol{p}, \boldsymbol{\tau}|\boldsymbol{\alpha}, \boldsymbol{\beta})$$
$$= \frac{1}{B(\boldsymbol{\beta})}\prod_{k=1}^{K}\tau_k^{\hat{\beta}_k - 1} \cdot \prod_{i=1}^{N}\frac{1}{B(\boldsymbol{\alpha})}p_i^{\hat{\alpha}_{i1}-1}(1-p_i)^{\hat{\alpha}_{i2}-1} \tag{7}$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]$, $\boldsymbol{\beta} = [\beta_1, \beta_2]$ and

$$\begin{aligned}\hat{\alpha}_{i1} &= \sum\nolimits_{j=1}^{M}\sum\nolimits_{k=1}^{K}\delta_{ijk}\xi_{jk} + \alpha_1 \\ \hat{\alpha}_{i2} &= \sum\nolimits_{j=1}^{M}\sum\nolimits_{k=1}^{K}\delta_{ij(3-k)}\xi_{jk} + \alpha_2 \\ \hat{\beta}_k &= \sum\nolimits_{j=1}^{M}\xi_{jk} + \beta_k.\end{aligned} \tag{8}$$

**Algorithm 1** Gibbs sampling for crowdsourcing

```
1: Input: the collected labels $\mathcal{L}$, the number of samples $W$
2: Output: the sample sequence $\mathcal{S}$
3: $\mathcal{S} \leftarrow \emptyset$, Initialize $\boldsymbol{L} = [L(j)]_M$ with the uniform distribution
4: for $s = 1$ to $W$ do
5:   for $j = 1$ to $M$ do
6:     Set $L(j) = 1$ and compute $x_1 = B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i)$
7:     Set $L(j) = 2$ and compute $x_2 = B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i)$
8:     $L(j) \leftarrow$ Sample $\{1, 2\}$ with $P(1) = x_1/(x_1 + x_2)$
9:   end for
10:  Append $\boldsymbol{L}$ to the sample sequence $\mathcal{S}$
11: end for
```

Besides, $B(x, y) = (x-1)!(y-1)!/(x+y-1)!$ denotes the beta function. The convergence of our inference algorithm requires $\alpha_1 > \alpha_2$. To simplify the theoretical analysis, we set $\alpha_1 = 1.5$ and $\alpha_2 = 1$ in this paper. Meanwhile, we employ the uniform distribution for $\boldsymbol{\tau}$ by setting $\beta_1 = \beta_2 = 1$. In this case, we can conduct marginalization via integrating Equation 7 over $\boldsymbol{p}$ and $\boldsymbol{\tau}$ as

$$P(\mathcal{L}, \boldsymbol{L}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{B(\hat{\boldsymbol{\beta}})}{B(\boldsymbol{\beta})} \cdot \prod\nolimits_{i=1}^{N} \frac{B(\hat{\boldsymbol{\alpha}}_i^*)}{[B(\boldsymbol{\alpha})]^2} \tag{9}$$

where $\hat{\boldsymbol{\alpha}}_i^* = [\hat{\alpha}_{i1} + 0.5, \hat{\alpha}_{i2}]$ and $\hat{\boldsymbol{\beta}} = [\hat{\beta}_1, \hat{\beta}_2]$. Following Bayes' theorem, we can know that

$$P(\boldsymbol{L}|\mathcal{L}) = \frac{P(\mathcal{L}, \boldsymbol{L}|\boldsymbol{\alpha}, \boldsymbol{\beta})}{P(\mathcal{L}|\boldsymbol{\alpha}, \boldsymbol{\beta})} \propto B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i^*). \tag{10}$$

Based on the joint posterior distribution $P(\boldsymbol{L}|\mathcal{L})$, we cannot derive an explicit formulation for the true label distribution of task $j$. Hence, we resort to Gibbs sampling for the inference based on $P(\boldsymbol{L}|\mathcal{L})$. More specifically, according to Bayes' theorem, we can know the conditional distribution of the true label of task $j$ satisfies $P[L(j)|\mathcal{L}, \boldsymbol{L}(-j)] \propto P(\boldsymbol{L}|\mathcal{L})$. In this case, we can generate the samples of the true label vector $\boldsymbol{L}$ by using Algorithm 1. At each step of sampling (line 6-8), Algorithm 1 calculates the conditional distribution and generate a new sample of $L(j)$ to replace the old one in $\boldsymbol{L}$. Through traversing all tasks, Algorithm 1 generates a new sample of the true label vector $\boldsymbol{L}$. Repeating this process for $W$ times, we can get the required posterior samples of $\boldsymbol{L}$, which is sequentially recorded in $\mathcal{S}$. Here, we write the $s$-th sample as $\boldsymbol{L}^{(s)}$. Since Gibbs sampling requires a burn-in process, we need to discard the first $b$ samples in $\mathcal{S}$. Thus, we can estimate worker $i$'s PoBC $p_i$ as

$$\tilde{p}_i = \frac{\sum_{s=b+1}^{W} \left[\alpha_1 + \sum_{j=1}^{M} \mathbb{1}(L^{(s)}(j) = L_i(j))\right]}{(W-b) \cdot (\alpha_1 + \alpha_2 + M)} \tag{11}$$

and the distribution of true labels $\boldsymbol{\tau}$ as

$$\tilde{\tau}_k = \frac{\sum_{s=b+1}^{W} \left[\beta_1 + \sum_{j=1}^{M} \mathbb{1}(L^{(s)}(j) = k)\right]}{(W-b) \cdot (\beta_1 + \beta_2 + M)}. \tag{12}$$

Furthermore, we define the log-ratio of task $j$ as

$$\tilde{\sigma}_j = \log \frac{P[L(j) = 1]}{P[L(j) = 2]} = \log \left( \frac{\tilde{\tau}_1}{\tilde{\tau}_2} \prod_{i=1}^{N} \tilde{\lambda}_i^{\delta_{ij1} - \delta_{ij2}} \right) \tag{13}$$

where $\tilde{\lambda}_i = \tilde{p}_i/(1 - \tilde{p}_i)$. Then, we decide the true label estimate $\tilde{L}(j)$ as 1 if $\tilde{\sigma}_j > 0$ and as 2 if $\tilde{\sigma}_j < 0$. Correspondingly, the label accuracy $A$ can be estimated as

$$\tilde{A} = \mathbb{E}A = \frac{1}{M} \sum\nolimits_{j=1}^{M} e^{|\tilde{\sigma}_j|} \left(1 + e^{|\tilde{\sigma}_j|}\right)^{-1}. \tag{14}$$

Note that, both $W$ and $b$ should be large values, and in this paper, we set $W = 1000$ and $b = 100$.

### 4.3. Reinforcement Incentive Adjustment

In this subsection, we formally introduce our reinforcement learning algorithm, which adjusts the incentive scaling level at each time step $t$. Stepping back and viewing it under the large picture, the reinforcement learning serves as the glue to connect each other component in our framework. As Figure 1 shows, the reinforcement learning algorithm takes as input workers' PoBC, reward signal, and internally its action history, and outputs the current incentive scaling level. Recall, the set consisting of all available scaling levels $\mathcal{A}$, is the action space for our reinforcement learning algorithm, The newly determined incentive scaling level (i.e. the most recent action) gets plugged back into the payment rule, and by following formula (4) payments to each worker are decided. **[Yitao: what is the reward here, and how it is calculated]**

**State Representation**

**Q-function Approximation**

**Policy Construction**

## 5. Game-Theoretic Analysis

In this section, we present the game-theoretic analysis on our incentive mechanism. Our main results are as follows:

**Proposition 1.** *When $M \gg 1$ and $(2p_H)^{2(N-1)} \geq M$, in any time step $t$, reporting truthfully ($r_i^t = 0$) and exerting high efforts ($e_i^t = 1$) is the payment-maximizing strategy for any worker $i$ if the other workers all follow this strategy. In other words, reporting truthfully and exerting high efforts is a Nash equilibrium for all workers in any time step.*

**Proposition 2.** *Suppose the conditions in Proposition 1 are satisfied. In our reinforcement learning algorithm, when $\tilde{Q}(s, a)$ approaches the real $Q(s, a)$ and*

$$\eta\zeta \cdot \min_{a,b \in \mathcal{A}} |a - b| > \frac{F(1) - F(1-\psi)}{1-\rho} \tag{15}$$

*always reporting truthfully ($r_i^t \equiv 0$) and exerting high efforts ($e_i^t \equiv 1$) is the payment-maximizing strategy for any worker $i$ in the long term if the other workers all follow this strategy. In other words, always reporting truthfully and exerting high efforts is a Nash equilibrium for all workers.*

The proof of Proposition 1 relies on the convergence of our Bayesian inference algorithm, namely $\tilde{p}_i^t \to p_i^t$. Proposition 2 provides a novel idea about the game-theoretic analysis of the reinforcement learning algorithm. More specifically, in the right-hand side Equation 15,

$$\psi = 2(\tau_1\tau_2^{-1} + \tau_1^{-1}\tau_2)[4p_H(1-p_H)]^{\frac{N-1}{2}} \tag{16}$$

is the upper bound of the label accuracy increment brought by a single worker. Thus, the right-hand side Equation 15 indicates the upper bound of the long-term utility increment that a single worker can bring. On the other hand, $\zeta = M(N-1)p_H$ and $\min_{a,b\in\mathcal{A}}|a-b|$ denotes the minimal gap between two available values of the scaling factor $a^t$. Thus, the left-hand side of Equation 15 is actually the lower bound of the payment increment if our reinforcement adjustment algorithm increases the scaling factor. Thereby, if Equation 15 is satisfied, a single worker will always be unable to cause our reinforcement adjustment algorithm to change $a^t$. This property ensures always reporting truthfully and exerting high efforts to be a Nash equilibrium, and also prevents the clever manipulation that a worker scarifies short-term benefits for higher payments in the future. In the remaining parts of this section, we will provide the details of our proof. It is also worth noting that we prove over 10 lemmas as the foundation of our proof. Due to the space limitation, we put them all in the supplementary file.

### 5.1. Proof for Proposition 1

After the workers report their labels, the payment in our incentive mechanism is only decided by $\tilde{p}_i^t$ which only depends on the labels in the current step. Thus, in this subsection, we focus on analyzing our Bayesian inference algorithm and omit the superscript $t$ in all equations for the simplicity of notations. From Equation 10, we can know the posterior distribution of the true labels satisfies

$$P(\boldsymbol{L}|\mathcal{L},\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{B(\hat{\boldsymbol{\beta}})\prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i^*)}{C_p \cdot P(\mathcal{L}|\boldsymbol{\alpha},\boldsymbol{\beta})} \tag{17}$$

where $C_p$ is the nomalization constant. Denote the labels generated by $N$ workers for one task as vector $\boldsymbol{x}$. Then, we can compute the distribution of $\boldsymbol{x}$ as

$$P_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{k=1}^{2}\tau_k\prod_{i=1}^{N}p_i^{1(x_i=k)}(1-p_i)^{1(x_i=3-k)} \tag{18}$$

where $\boldsymbol{\theta} = [\tau_1, p_1, \ldots, p_N]$ denotes all the parameters. For the denominator in Equation 17, we can have

**Proposition 3.** *When $M \to \infty$,*

$$P(\mathcal{L}|\boldsymbol{\alpha},\boldsymbol{\beta}) \to C_L(M)\cdot\prod_{\boldsymbol{x}}[P_{\boldsymbol{\theta}}(\boldsymbol{x})]^{M\cdot P_{\boldsymbol{\theta}}(\boldsymbol{x})} \tag{19}$$

*where $C_L(M)$ denotes a constant that depends on $M$.*

*Proof.* Denote the prior distribution of $\boldsymbol{\theta}$ by $\pi$. Then,

$$P(\mathcal{L}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{j=1}^{M}P_{\boldsymbol{\theta}}(\boldsymbol{x}_j)\int e^{[-M\cdot d_{KL}]}\mathrm{d}\pi(\hat{\boldsymbol{\theta}}) \tag{20}$$

$$d_{KL} = \frac{1}{M}\sum_{j=1}^{M}\log\frac{P_{\boldsymbol{\theta}}(\boldsymbol{x}_j)}{P_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_j)} \to \mathrm{KL}[P_{\boldsymbol{\theta}}(\boldsymbol{x}), P_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x})] \tag{21}$$

where $\boldsymbol{x}_j$ denotes the labels generated for task $j$. The KL divergence $\mathrm{KL}[\cdot,\cdot]$, which denotes the expectation of the log-ratio between two probability distributions, is a constant for the given $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$. Thus, $\int e^{[-M\cdot d_{KL}]}\mathrm{d}\pi(\hat{\boldsymbol{\theta}}) = C_L(M)$. In addition, when $M\to\infty$, we can also have $\sum 1(\boldsymbol{x}_j = \boldsymbol{x}) \to M\cdot P_{\boldsymbol{\theta}}(\boldsymbol{x})$, which concludes Proposition 3. □

Then, we move our focus to the posterior true label vector $\boldsymbol{L}$ generated by $P(\boldsymbol{L}|\mathcal{L},\boldsymbol{\alpha},\boldsymbol{\beta})$. We introduce $n$ and $m$ to denote the number of tasks of which the posterior true label is correct and wrong, respectively. Besides, for the simplicity of notations, we employ the convention that $\bar{p} = 1-p$, $\hat{p} = \max\{p,\bar{p}\}$ and $p_0 = \tau_1$. Hence, we can have

**Proposition 4.** *When $M \gg 1$,*

$$\mathbb{E}[m/M] \lesssim (1+e^{\delta})^{-1}(\varepsilon+e^{\delta})(1+\varepsilon)^{M-1} \tag{22}$$

$$\mathbb{E}[m/M]^2 \lesssim (1+e^{\delta})^{-1}(\varepsilon^2+e^{\delta})(1+\varepsilon)^{M-2} \tag{23}$$

*where $\varepsilon^{-1} = \prod_{i=0}^{N}(2\hat{p}_i)^2$, $\delta = O[\Delta\cdot\log(M)]$ and*

$$\Delta = \sum_{i=1}^{N}[1(p_i<0.5) - 1(p_i>0.5)].$$

*Proof.* Firstly, we introduce a set of variables to describe the real true labels and the collected labels. Among the $n$ tasks of which the posterior true label is correct,

- $x_0$ and $y_0$ denote the number of tasks of which the real true label is 1 and 2, respectively.
- $x_i$ and $y_i$ denote the number of tasks of which worker $i$'s label is correct and wrong, respectively.

Also, among the remaining $m = M-n$ tasks,

- $w_0$ and $z_0$ denote the number of tasks of which the real true label is 1 and 2, respectively.
- $w_i$ and $z_i$ denote the number of tasks of which worker $i$'s label is correct and wrong, respectively.

Thus, we can have $x_i + y_i = n$ and $w_i + z_i = m$. Besides, we use $\xi_i$ to denote the combination $(x_i, y_i, w_i, z_i)$.

To compute the expectation of $m/M$, we need to analyze the probability distribution of $m$. According to Equation 10, we can know that $P(m)$ satisfies

$$P(m) \approx \frac{C_M^m}{Z} \sum_{\xi_0,\ldots,\xi_N} \prod_{i=0}^{N} P(\xi_i|m) B(\hat{\boldsymbol{\beta}}) \prod_{i=1}^{N} B(\hat{\boldsymbol{\alpha}}_i^*) \quad (24)$$

where $Z = C_p C_L \prod_{\boldsymbol{x}} [P_{\boldsymbol{\theta}}(\boldsymbol{x})]^{M \cdot P_{\boldsymbol{\theta}}(\boldsymbol{x})}$ is independent of $\xi_i$ and $m$. Meanwhile, $\hat{\beta}_1 = x_0 + z_0 + 1$, $\hat{\beta}_2 = y_0 + w_0 + 1$, $\hat{\alpha}_{i1}^* = x_i + z_i + 2$ and $\hat{\alpha}_{i2}^* = x_i + z_i + 1$. When the $m$ tasks of which the posterior true label is wrong are given, we can know that $x_i \sim \mathrm{Bin}(n, p_i)$ and $w_i \sim \mathrm{Bin}(m, p_i)$, where $\mathrm{Bin}(\cdot)$ denotes the binomial distribution. In addition, $x_i$ and $y_i$ are independent of $w_i$, $z_i$ and $\xi_{k \neq i}$. Also, $w_i$ and $z_i$ are independent of $x_i$ and $y_i$ and $\xi_{k \neq i}$. Thus, we can further obtain $P(m) \approx 2^{-(N+1)(M+1)} Z^{-1} \cdot C_M^m Y(m)$, where

$$\begin{aligned} & Y(m) = e^{\log H(m,p_0;M,0) + \sum_{i=1}^{N} \log H(m,p_i;M,1)} \\ & H(m,p;M,t) = \sum\nolimits_{x=0}^{n} \sum\nolimits_{w=0}^{m} 2^{M+1} C_n^x C_m^w \times \\ & \quad p^{x+w}(1-p)^{y+z} B(x+z+1+t, y+w+1). \end{aligned} \quad (25)$$

Besides, considering $\sum_{m=1}^{M} P(m) = 1$, we can know that

$$2^{-(N+1)(M+1)} \cdot Z \approx \sum\nolimits_{m=1}^{M} C_M^m Y(m). \quad (26)$$

The biggest challenge of computing $P(m)$ exists in the analysis of function $H(m,p;M,t)$ which we put in the supplementary file because of the space limitation. Here, we directly use the obtained lower and upper bounds of the $H$ function (Lemmas **??** and **??**) and can have

$$\begin{cases} e^{C - K_l m} \lesssim Y(m) \lesssim e^{C - K_u m} & 2m \leq M \\ e^{C + \delta - K_l n} \lesssim Y(m) \lesssim e^{C + \delta - K_u n} & 2m > M \end{cases} \quad (27)$$

where $C = H(0, p_0; M, 0) + \sum_{i=1}^{N} H(0, p_i; M, 1)$ and

$$\begin{aligned} & K_l = \sum\nolimits_{i=0}^{N} \log \hat{\lambda}_i \ , \ K_u = 2 \sum\nolimits_{i=0}^{N} \log (2\hat{p}_i) \\ & \delta = \Delta \cdot \log(M) + \sum\nolimits_{i=1}^{N} (-1)^{1(p_i > 0.5)} \phi(\hat{p}_i) \\ & \hat{\lambda}_i = \max \left\{ \frac{p_i}{\bar{p}_i + \frac{1}{M}}, \frac{\bar{p}_i}{p_i + \frac{1}{M}} \right\} \ , \ \phi(p) = \log \frac{2p-1}{p}. \end{aligned}$$

Besides, we set a convention that $\phi(p) = 0$ when $p = 0.5$. Thereby, the expectations of $m$ and $m^2$ satisfy

$$\mathbb{E}[m] \lesssim \frac{\sum_{m=0}^{M} m e^{-K_u m} + \sum_{m=0}^{M} m e^{\delta - K_u n}}{\sum_{m=0}^{k} e^{-K_l m} + \sum_{m=k+1}^{M} e^{\delta - K_l n}} \quad (28)$$

$$\mathbb{E}[m^2] \lesssim \frac{\sum_{m=0}^{M} m^2 e^{-K_u m} + \sum_{m=0}^{M} m^2 e^{\delta - K_u n}}{\sum_{m=0}^{k} e^{-K_l m} + \sum_{m=k+1}^{M} e^{\delta - K_l n}} \quad (29)$$

where $k = \lfloor M/2 \rfloor$. By using Lemmas **??**, **??**, **??** and **??**, we can know the upper bounds of the numerator in Equations 28 and 29 are $M(\varepsilon + e^{\delta})(1+\varepsilon)^{M-1}$ and $[M^2\varepsilon^2 + M\varepsilon + e^{\delta}(M^2 + M\varepsilon)](1+\varepsilon)^{M-2}$, respectively, where $\varepsilon = e^{-K_u}$. On the other hand, by using Lemma **??**, we can obtain the lower bound of the denominator as $(1+e^{\delta})[1 - e^{-c(\omega)M}](1+\omega)^M$, where $\omega = e^{-K_l}$ and $c(\omega) = 0.5(1-\omega)^2(1+\omega)^{-2}$. Considering $M \gg 1$, we can make the approximation that $e^{-c(\omega)M} \approx 0$ and $(1+e^{\delta})\varepsilon/M \approx 0$. Besides, $(1+\omega)^M \geq 1$ holds because $\omega \geq 0$. In this case, Proposition 4 can be concluded by combining the upper bound of the numerator and the lower bound of the denominator. □

Lastly, focusing on worker $i$, we calculate the difference between the estimated PoBC $\tilde{p}_i$ and the real PoBC $p_i$ when the other workers all exert high efforts and report truthfully. When $M \gg 1$, according to Equation 11, we can know that $\tilde{p}_i \approx \mathbb{E}_{\boldsymbol{L}}(x_i + z_i)/M$, where $\mathbb{E}_{\boldsymbol{L}}$ denotes the expectation based on the posterior distribution $P(\boldsymbol{L}|\mathcal{L})$. Meanwhile, in the proof of Proposition 4, according to the law of large numbers, $p_i \approx (x_i + w_i)/M$. Thus, we can have

$$|\tilde{p}_i - p_i| \approx \mathbb{E}_{\boldsymbol{L}} |w_i - z_i|/M \leq \mathbb{E}_{\boldsymbol{L}} [m/M]. \quad (30)$$

If workers except for worker $i$ all report truthfully and exert high efforts, then $\Delta \leq -1$ in Proposition 4 because we require $N \geq 3$ in Section 3. Considering $M \gg 1$, we can make the approximation that $e^{\delta} \approx 0$. In addition, considering $2\hat{p}_i \geq 1$,we can have $\varepsilon^{-1} \geq (2p_H)^{2(N-1)}$. When $(2p_H)^{2(N-1)} \geq M$, $\varepsilon \leq M^{-1}$. Thus, the upper bound in Proposition 4 can be further calculated as

$$\mathbb{E}\left[\frac{m}{M}\right] \lesssim \frac{C_1}{M \cdot C_2} \ , \ \mathbb{E}\left[\frac{m}{M}\right]^2 \lesssim \frac{C_1}{M^2 \cdot C_2^2} \quad (31)$$

where $C_1 = (1 + M^{-1})^M \approx e$ and $C_2 = 1 + M^{-1} \approx 1$. Then, $m/M \approx 0$ because $\mathbb{E}[m/M] \approx 0$ and $\mathrm{Var}[m/M] = \mathbb{E}[m/M]^2 - (\mathbb{E}[m/M])^2 \approx 0$. In this case, $\tilde{p}_i \approx p_i$. Thereby, worker $i$ can only get the maximal payment when reporting truthfully and exerting high efforts, namely, when $p_i = p_H$, which concludes Proposition 1.

### 5.2. Proof for Proposition 2

$$Q(s_t, a^t) = \sum\nolimits_{i=0}^{\infty} \rho^i u_{t+i}. \quad (32)$$

$$Q(s,a) = \boldsymbol{k}_t(s,a)^T (\boldsymbol{K}_t + \sigma^2 \boldsymbol{I}_t)^{-1} H_t^{-1} \boldsymbol{u}_t \quad (33)$$

where $\boldsymbol{k}_t(s,a) = [k(x,x_0), \ldots, k(x,x_t)]^T$, $\boldsymbol{K}_t = [\boldsymbol{k}_t(s_0,a_0), \ldots, \boldsymbol{k}_t(s_t,a_t)]$ and $x = (s,a)$. Besides,

$$H_t = \begin{bmatrix} 1 & \rho & \rho^2 & \ldots & \rho^{t+1} \\ 0 & 1 & \rho & \ldots & \rho^t \\ 0 & 0 & 1 & \ldots & \rho^{t-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix} \quad (34)$$

To prove Proposition 2, we need to analyze worker $i$'s effects on our reinforcement learning algorithm. If worker $i$ wishes to get higher payments in the long term, he/she must push our reinforcement learning algorithm to at least increase the scaling factor from $a$ to $b > a$ at a certian state $s$. In the $\epsilon$-greedy strategy used by our reinforcement learning algorithm, the random selection part is independent of worker $i$. Thus, worker $i$ must mislead the greedy part by letting $\tilde{Q}(s,a) \leq \tilde{Q}(s,b)$. In this proof, we will show that, under the condition defined in Equation 15, there does not exist $b \in \mathcal{A}$ that can achieve this objective. In other words, our reinforcement learning algorithm will never increase the scaling factor to please a single worker. On the other hand, in any time step $t$, worker $i$ will loss some money if $p_i^t < p_H$. Thereby, the payment-maximizing strategy for worker $i$ is to report truthfully and exert high efforts in all time steps, which concludes Proposition 2.

Since Proposition 2 requires $\tilde{Q}(s,a) \approx Q(s,a)$ as one of the conditions, we now focus on proving that $Q(s,a) - Q(s,b) > 0$ always holds. Suppose all workers except for worker $i$ report truthfully and exert high efforts in all time steps. According to Equations 2 and 32, we can have $Q(s,a) - Q(s,b) \geq X(a) - X(b) + Y$, where

$$X(a) = \sum\nolimits_{k=0}^{\infty} \rho^k \cdot \mathbb{E}F(\tilde{A}^{k+t} | s_t = s^*, a^t = a) \quad (35)$$

denotes the expected long-term utility that we get from the labels. $Y = \eta M(N-1)p_H(b-a) > 0$ denotes the payment increment for workers except worker $i$. To attract our reinforcement learning algorithm to increase the scaling factor, worker $i$ must increase $p_i^t$ when $a^t$ is increased from $a$ to $b$. Otherwise, we will get less accurate labels with higher payments, which is impossible for the greedy strategy used in our reinforcement learning algorithm. In this case, the payment for worker $i$ will also increase. However, we do not know $p_i$. Thus, we regard the payment increment as 0 when deriving the lower bound of $Q(s,a) - Q(s,b)$.

Here, to bound $X(a) - X(b)$, we analyze the effects of worker $i$ on the estimated accuracy $\tilde{A}$. Since our analysis is satisfied in all time steps, we omit the time step $t$ for the simplicity of notations. From Equation 14, we can know that, when $M \gg 1$, the estimated accuracy $\tilde{A}$ satisfies

$$\tilde{A} \approx 1 - \mathbb{E}g(\tilde{\sigma}_j) \ \ , \ \ g(\tilde{\sigma}_j) = 1/(1 + e^{|\tilde{\sigma}_j|}). \quad (36)$$

From the proof of Proposition 1, we can know that $\tilde{p}_i^t \approx p_i^t$. In this case, according to Equation 13, we can have

$$\tilde{\sigma}_j(p_i) \approx \log\left(\frac{\tau_1}{\tau_2} \lambda_i^{\delta_{ij1} - \delta_{ij2}} \prod\nolimits_{k \neq i} \lambda_H^{\delta_{kj1} - \delta_{kj2}}\right). \quad (37)$$

where $\lambda_i = p_i/(1-p_i)$ and $\lambda_H = p_H/(1-p_H)$.

Considering the case that worker $i$ exert low efforts and reports randomly, namely $p_i = 0.5$, we can eliminate $\lambda_i$ from Equation 37 because $\lambda_i = 1$. Furthermore, according to Lemma **??** in the supplementary file, we can know that $g(\tilde{\sigma}_j) < e^{\tilde{\sigma}_j}$ and $g(\tilde{\sigma}_j) < e^{-\tilde{\sigma}_j}$ both hold. Thus, we build a more tight upper bound of $g(\tilde{\sigma}_j)$ by dividing all the combinations of $\delta_{kj1}$ and $\delta_{kj2}$ in Equation 37 into two sets and using the smaller one of $e^{\tilde{\sigma}_j}$ and $e^{-\tilde{\sigma}_j}$ in each set. By using this method, if the true label is 1, we can have $\mathbb{E}_{[L(j)=1]} g(\tilde{\sigma}_j) < q_1 + q_2$, where

$$q_1 = \frac{\tau_2}{\tau_1} \sum\nolimits_{n=K+1}^{N-1} C_{N-1}^n \left(\frac{1}{\lambda_H}\right)^{n-m} p_H^n (1-p_H)^m$$
$$q_2 = \frac{\tau_1}{\tau_2} \sum\nolimits_{n=0}^{K} C_{N-1}^n \lambda_H{}^{n-m} p_H^n (1-p_H)^m$$
$$n = \sum\nolimits_{k \neq i} \delta_{kj1} \ , \ m = \sum\nolimits_{k \neq i} \delta_{kj2} \ , \ K = \lfloor (N-1)/2 \rfloor.$$

Here, we use $e^{-\tilde{\sigma}_j}$ and $e^{\tilde{\sigma}_j}$ as the upper bound of $g(\tilde{\sigma}_j)$ when $n \in (K, N-1]$ and $n \in [0, K]$, respectively. By using Lemma **??** in the supplementary file, we can thus get

$$\mathbb{E}_{[L(j)=1]} g(\tilde{\sigma}_j) < c_\tau [4p_H(1-p_H)]^{\frac{N-1}{2}}. \quad (38)$$

where $c_\tau = \tau_1 \tau_2^{-1} + \tau_1^{-1} \tau_2$. Similarly,

$$\mathbb{E}_{[L(j)=2]} g(\tilde{\sigma}_j) < c_\tau [4p_H(1-p_H)]^{\frac{N-1}{2}}. \quad (39)$$

Thereby, $\tilde{A} > 1 - 2c_\tau [4p_H(1-p_H)]^{\frac{N-1}{2}} = 1 - \psi$.

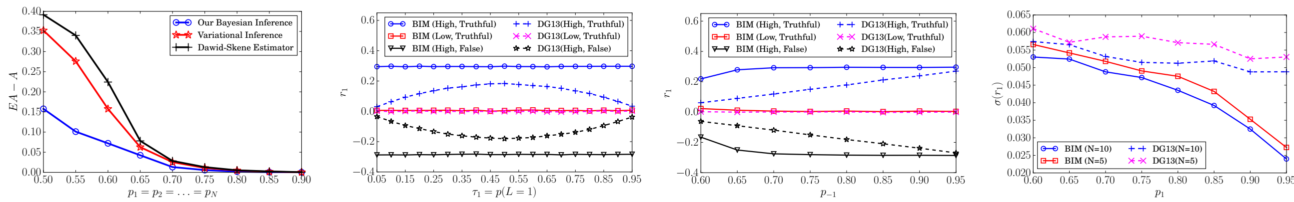
We then consider another case where worker $i$ exerts high efforts but reports falsely, namely $p_i = 1 - p_H$. In this case, we can rewrite Equation 37 as

$$\tilde{\sigma}_j(1-p_H) \approx \log\left(\frac{\tau_1}{\tau_2} \lambda_H^{x-y} \prod\nolimits_{k \neq i} \lambda_H^{\delta_{kj1} - \delta_{kj2}}\right). \quad (40)$$

where $x = \delta_{ij2}$ and $y = \delta_{ij1}$. Since $p_i = 1 - p_H$, $x$ and $y$ actually has the same distribution as $\delta_{kj1}$ and $\delta_{kj2}$. Thus, the distribution of $\tilde{\sigma}_j(1-p_H)$ is actually the same as $\tilde{\sigma}_j(p_H)$. In other words, since Proposition 1 ensures $p_i$ to be accurately estimated, our Bayesian inference algorithm uses the information provided by worker $i$ via flipping the label when $p_i < 0.5$. Thus, $p_i = 0.5$ actually lowers $\tilde{A}$ to the utmost because worker $i$ provides no information about the true label in this case. Thus, $\tilde{A} \geq 1 - \psi$ always holds. On the other hand, $\tilde{A} \leq 1.0$ also always holds. Considering $F(\cdot)$ is a non-decreasing monotonic function, we can get $X(a) \geq (1-\rho)^{-1} F(1-\psi)$ while $X(b) \leq (1-\rho)^{-1} F(1)$. Thereby, when Equation 15 is satisfied, $X(a) - X(b) + Y > 0$ always holds, which concludes Proposition 2.

# 6. Empirical Experiments

In this section, we conduct experiments on our Bayesian incentive mechanism at first to verify its advantages to lower the inference bias and improve the fairness and stability of the rewards for workers. Then, to verify the advantage of




(a) (b) (c) (d)

*Figure 2.* Empirical analysis on our one-step Bayesian incentive mechanism (BIM) (a) the inference bias (b) the reward variation as the distribution of true labels (c) the reward variation as the score of other workers (d) the standard variance of the reward for worker 1

our reinforcement incentive mechanism to boost the utility of the data requester, we empirically test our mechanism by using three representative worker models, including fully rational, bounded rational and self-learning agents.

### 6.1. Bayesian incentive mechanism experiments

In the literature of crowdsourcing, the Dawid-Skene estimator is the most popular method used to infer the true labels (**??**). The variational inference estimator, which has the similar Bayesian model to our inference algorithm, is also widely-adopted in the existing studies of crowdsourcing (**??**). To compare different estimators, we set $M = 100$ and $N = 10$ in Figure 2a. Also, we let the score of all workers be equal, namely $p_1 = \ldots = p_N$, and increase the value of $p_i$ from $0.5$ to $0.9$. Meanwhile, we set the true label distribution as the uniform distribution, namely $\tau_1 = \tau_2 = 0.5$. For a given $p_i$, we firstly generate the true labels and then the labels of all workers both by the Bernoulli distribution. For each value of $p_i$, we run the experiments for $1000$ rounds. To show the bias of inference, we calculate the average value differences between the posterior expected accuracy $\mathbb{E}A$ and the real accuracy $A$. From the figure, we can find that, when workers can provide not-so-bad labels ($p_i > 0.75$), both the two above estimators and our inference algorithm have very small bias, which agrees with the good performance of these estimators in the literature (**??**). However, if workers can only provide low-quality labels, the bias of the Dawid-Skene and variational inference estimators will become unacceptable, because the difference can be larger than $0.3$ while both $\mathbb{E}A$ and $A$ belong to $[0.5, 1.0]$. In this case, we cannot use $\mathbb{E}A$ to calculate the utility of the data requester as Equation 2. By contrast, the bias of our Bayesian inference algorithm is much smaller, which is the foundation of our reinforcement incentive mechanism.

In Figures 2b-d, we focus on $r_1$, namely, the per-task-reward received by worker $1$. Here, DG13 (**??**), which is the state-of-the-art incentive mechanism for binary labels, is employed as the benchmark. DG13 decides the reward for a worker by comparing his labels with the labels provided by another randomly selected worker. By elaborately designing the reward rules, it can also ensure reporting truthfully and exert high efforts to be a Nash equilibrium for all workers. In all these experiments, we set $p_H = 0.8$, $p_L = 0.5$, and keep the other settings the same as those in Figure 2a.

In Figure 2b, we let $p_{-1} = p_H$, where the subscript $-1$ denotes all the workers except for worker $1$. We change the distribution of true labels by increasing $\tau_1$ from $0.05$ to $0.95$ and compare the average values of $r_1$ corresponding to the different strategies of worker $1$. In Figure 2c, we fix the distribution of true labels to be the uniform distribution, namely, $\tau_1 = \tau_2 = 0.5$, and increase $p_{-1}$ from $0.6$ to $0.95$. From these two figures, we can find that the rewards provided by our mechanism are almost not affected by the variation of the distribution of true labels and the strategies of the other workers. This observation reveals that $\mathbb{E}\tilde{p}_1$ converges to $p_1$ in most cases. The only exception is $p_{-1} < 0.7$ in Figure 2c where the low-quality labels will lead to a remarkable bias of inference. Even in this case, worker $1$ can only get the maximal reward when $p_1 = p_H$, which shows the attracting ability of our mechanism to induce truthful reports and high efforts. By contrast, $r_1$ in DG13 is severely affected by the distribution of true labels and the strategies of other workers. For example, in Figure 2c, if the other workers lower their efforts, the reward received by worker $1$ will also decrease, although worker $1$ never changes his strategies. Thereby, for worker $1$, our Bayesian incentive mechanism is much fairer than DG13.

In Figure 2d, we set $\tau_1 = \tau_2 = 0.5$ and $p_{-1} = p_H$. We change worker $1$'s strategies by increasing $p_1$ from $0.6$ to $0.95$. Under these settings, the average values of $r_1$ corresponding to our mechanism and DG13 both can reflect the variation of $p_1$ very well. Thus, we focus on the standard variance comparison of $r_i$ in Figure 2d. If the variance is very large, the reward received by worker $1$ when $p_1 = p_H$ may become lower than the reward when $p_1 < p_H$. If this case happens, it will significantly discourage worker $1$. For example, in Figure 2b, when $\tau_1 = 0.05$, for DG13, the difference between $r_1(p_1 = p_H)$ and $r_1(p_1 = 0.5)$ is around $0.06$. On the other hand, from Figure 2d, the standard variance of $r_1$ is around $0.052$, which means there is a quite high probability for $r_1(p_1 = p_H) < r_1(p_1 = 0.5)$.

From Figure 2d, we can find that our Bayesian incentive mechanism has a lower variance than DG13. If we take the fairness of our mechanism into consideration, we can conclude that our mechanism is more stable than DG13 in inducing truthful reports and high efforts from workers.