MADDM: Multi-Advisor Dynamic Binary Decision-Making by Maximizing the Utility

Zhaori Guo
University of Southampton
United Kingdom
zg2n19@soton.ac.uk

Timothy J. Norman
University of Southampton
United Kingdom
t.j.norman@soton.ac.uk

Enrico H. Gerding
University of Southampton
United Kingdom
eg@ecs.soton.ac.uk

ABSTRACT

Being able to infer ground truth from the responses of multiple imperfect advisors is a problem of crucial importance in many decision-making applications, such as lending, trading, investment, and crowd-sourcing. In practice, however, gathering answers from a set of advisors has a cost. Therefore, finding an advisor selection strategy that retrieves a reliable answer and maximizes the overall utility is a challenging problem. To address this problem, we propose a novel strategy for optimally selecting a set of advisers in a sequential binary decision-making setting, where multiple decisions need to be made over time. Crucially, we assume no access to ground truth and no prior knowledge about the reliability of advisers. Specifically, our approach considers how to simultaneously (1) select advisors by balancing the advisors' costs and the value of making correct decisions, (2) learn the trustworthiness of advisers dynamically without prior information by asking multiple advisers, and (3) make optimal decisions without access to the ground truth, improving this over time. We evaluate our algorithm through several numerical experiments. The results show that our approach outperforms two other methods that combine state-of-the-art models.

KEYWORDS

Trust and Reputation; Crowdsourcing; Truth Inference

ACM Reference Format:
Zhaori Guo, Timothy J. Norman, and Enrico H. Gerding. 2023. MADDM: Multi-Advisor Dynamic Binary Decision-Making by Maximizing the Utility. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Many situations rely on expert advice to make decisions, and often there is no objectively correct answer. Examples are wide-ranging and include crowdsourcing, machine learning ensemble models, or loan approvals. In such settings, and following the principles of the wisdom of the crowd [9, 19], it may be better to rely on the expertise of multiple advisers, especially if the stakes are high. However, asking for multiple advisers comes at a cost, and the reliability or *quality* of advisers can differ. Therefore, knowing how many and who to ask is a challenging task. In addition, typically, multiple sequential decisions need to be made, and the reliability of individual advisers can be learned over time. A good strategy for doing so is not obvious, however, when ground truth is not available. To address these challenges, we design a novel method for maximizing the utility in sequential, binary, multi-advisor decision-making problems for settings with no ground truth.

These types of scenarios are extensively studied in various fields where dynamic pricing for advisors is considered [11, 14, 17]. Tong *et al.* [14] focus on pricing the advisors in different regions and decided by the relationship between supply and demand in spatial crowdsourcing tasks. They do not consider advisors having different qualities, however. Miao *et al.* [11] and Wang *et al.* [17] also assume advisors cost the same but give additional rewards to advisors with more contributions. However, when there is no real-time feedback on the ground truth, it is difficult to determine who should get additional rewards. Therefore, the same price for advisors with different qualities is unrealistic; in contrast, our work considers advisors with different qualities and prices.

Other research considers a fixed budget constraint [2, 15, 16, 18, 20]. However, these papers do not consider that decisions might have different values and costs associated with getting them wrong. Consider the following lending decision scenario. If a $1,000 loan at 9% interest is repaid, it will make a $90 profit, but it can result in a loss of $1,000 if the borrower defaults. Such high-risk decisions require a more reliable assessment, potentially requiring multiple costly advisers, whereas low-value, low-risk decisions may only need a single one. Therefore, we should consider selecting a group of advisors with different qualities and prices to balance potential profits and risks associated with a decision.

Another relevant field of research involves aggregating answers to infer the ground truth [3, 4, 13, 19]. Here, decisions are made, and an advisor's trustworthiness is updated through approaches such as majority voting, weighted voting, and expectation maximization (EM). However, they deal with all decisions simultaneously, whereas we consider the case where decisions are made sequentially. In the sequential decision-making case, initially, the sample size is small. So maximum likelihood estimation methods have a large deviation between the estimated trustworthiness distribution and the real one [10]. This deviation can mislead future decisions and samples. Instead, in our work, we consider how to gradually and steadily establish a trustworthiness model for decisions without prior information.

Research grounded in multi-armed bandit methods is also relevant here [8, 15, 16, 18]. However, these works assume that that the ground truth is available following every decision, which means that advisors' trustworthiness can be reliably updated. Instead, our work infers the reliability of the answer by the decision model, thereby avoiding the need for ground truth. Assessing the reliability of the answer can help us give reasonable update evidence for

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

Table 1: Comparison of MADDM with state of the art. TS = Thompson Sampling; MV = Majority Voting; BWVE = Bayesian Weighted Voting Ensemble; EM = Expectation-Maximization.

Setting	ϵ-First [16]	ZenCrowd [3]	SBB [6]	ACT [8]	DEMV [13]	BAL [4]	MTIRL [5]	MADDM
sequential	✓	×	✓	✓	×	×	✓	✓
truth inference	×	✓	×	×	✓	✓	✓	✓
multi-advisor for one task	×	✓	×	×	✓	✓	✓	✓
budget-limited	✓	×	×	×	×	×	×	✓
different task values	×	×	×	×	×	×	×	✓
different advisor's price	✓	×	×	×	×	×	×	✓
trustworthiness assessment	✓	✓	✓	✓	×	✓	✓	✓
insufficient samples	✓	×	×	✓	×	×	✓	✓
aggregation method	×	EM	Bayesian	×	MV	EM	BWVE	BWVE
advisor selection	ϵ-first	×	×	TS	×	×	×	TS

building models of the trustworthiness of advisors. To summarise, Table 1 provides an overview of the state of the art and how it compares to the problem we are addressing.

In more detail, we present a novel method, called Multi-Advisor Dynamic Decision-Making Method (MADDM), to address the limitations of existing approaches described above. MADDM (see Section 3 for details) integrates and extends several state-of-the-art methods and consists of three interdependent components: trust assessment, advisor selection, and decision-making. Trust assessment builds and maintains models of the trustworthiness of each advisor. For every sequential decision, advisor selection identifies which advisors to consult. This is similar to a multi-armed bandit problem, which requires a balance of exploration and exploitation. We use Thompson Sampling combined with the decision-making model to compute each advisor's expected marginal contribution and select advisers until the marginal contribution is negative. The third component uses the set of answers from the advisor selected to make a decision using the *Bayesian Weighted Voting Ensemble* (BWVE) method proposed in [5]. In addition, we conduct extensive experiments (Section 4) that compare MADDM to a variety of methods that combine state-of-the-art approaches, including budget-limited decision making, ϵ-greedy selection, and expectation maximization, and we benchmark performance against the optimal utility that could be gained with perfect knowledge. The results show that MADDM outperforms the other two methods in almost all environments.

Before presenting MADDM in detail, in what follows we first formalize our problem domain.

2 PROBLEM FORMALIZATION

Let D be the set of decisions and X be a set of advisors. For every decision $d \in D$, the decision-maker needs to choose a unique answer with a binary value, namely $a_d \in \{-1, 1\}$. For simplicity but without loss of generality, we assume that the correct value, i.e. the ground truth, denoted by a_d^*, is positive, i.e. $a_d^* = 1$. Given a decision d, v_d^+ is the value that the decision-maker gets if the answer is correct. We denote with v_d^- the value that the decision-maker pays if the answer it infers is wrong. Therefore, the value of the decision is represented by the tuple $v_d^\pm = (v_d^+, v_d^-)$. Moreover, since we rely on advisors to answer queries to inform decisions, we need to incentive them by introducing a payment system. For each advisor $x \in X$, c_x is its price. For any given $d \in D$, the decision-maker must select a subset of advisors $Y_d \subseteq X$.

The choice of advisors also depends on their trustworthiness. For each advisor, $x \in X$, τ_x is its trustworthiness, which is updated after every decision for which that advisor is consulted. Finally, we denote with $\vec{c}$ and $\vec{\tau}$ the vectors containing all the advisors' prices and trustworthiness values, respectively. We, therefore, describe any possible selection through a function s that, to every tuple $I := (d, \vec{\tau}, v_d^\pm, \vec{c}) \in D \times [0,1]^{|X|} \times [0, +\infty]^2 \times [0, +\infty]^{|X|}$, associates a subset of advisors $Y_d \in \mathcal{P}(X)$, where $|X|$ is the cardinality of X and $\mathcal{P}(X)$ is the power set of X; we call s the *selection function*. Table 2 gives an overview of the main variables and parameters used.

For any given $d \in D$, we denote with $P_{d,s} \subseteq s(I) \subseteq X$ the set of advisors who give positive answers to decision d. Similarly, we denote with $N_{d,s} \subseteq s(I) \subseteq X$ the set of advisors who give a negative answer to decision d. When it is clear from the context, we simplify the notation and use P_d and N_d over $P_{d,s}$ and $N_{d,s}$, respectively. Note that $P_{d,s} \cap N_{d,s} = \emptyset$ and $P_{d,s} \cup N_{d,s} = s(I)$ for every $d \in D$.

We assume that, for any given decision, d, there exists a true answer a_d^*, but this is never revealed to the decision maker. Therefore, we use $a_d = f(P_d, N_d)$ to refer to the decision-making function of our inference model. This is a function of the advisors' responses in P_d and N_d. Let $v_d \in v_d^+, v_d^-$ denote the value that the decision-maker gets from the decision d, and let a_d^* denote the ground truth of the decision. If $a_d = a_d^*$, we say that the answer is correct and $v_d = v_d^+$. Otherwise, we say that the answer is wrong and $v_d = v_d^-$. Accordingly, for every decision, d, the total cost to the decision-maker to hire the advisors in $s(I)$ is $C_d(s) = \sum_{x \in s(I)} c_x$.

Finally, we define the utility that the decision-maker gets for every decision. Given a decision, d, we define its utility to the decision-maker as $u_d(s) = v_d - C_d(s)$. In particular, the sum of the utilities for all the decisions is $u(s) = \sum_{d \in D} u_d(s)$. Since each advisor has a different cost, the final utility depends on the advisor selection function adopted. In this framework, the goal of the decision-maker is to find the selection function, s, that maximizes its payoff:

$$s^* = \arg\max_{s \in \mathcal{S}} u(s), \tag{1}$$

where $\mathcal{S}$ denotes the set of all feasible selection functions.

Table 2: List of additional variables and parameters used in our MADDM system.

Variables and Parameters List	
x	advisor index
d	decision index
s	selection function
f	decision function
α_x	correct estimated evidence of the advisor x
β_x	wrong estimated evidence of the advisor x
θ_x	uncertainty of the advisor x
τ_x	trustworthiness of the advisor x
τ'_x	trustworthiness of the advisor x from Beta Sampling
i_d	confidence value of decision d
c_x	price of the advisor x
P_d	set of the advisors whose answer for decision d is 1
N_d	set of the advisors whose answer for decision d is -1
Y_d	$P_d \cup N_d$
u_d	utility of decision d
a_d	final inferred answer of decision d
a^*_d	ground truth of decision d
v^+_d	profits that if $a_d = a^*_d$
v^-_d	loss that if $a_d \neq a^*_d$
P^{e+}_d	probability that $a_d = 1$ from ensemble model
P^{e-}_d	probability that $a_d = -1$ from ensemble model

3 MULTI-ADVISOR DYNAMIC DECISION-MAKING

The design of MADDM consists of three components. The first is a trust assessment model that determines an advisor's trustworthiness, which can be used as a weight in the decision model and to calculate the contributions of advisors in the advisor selection model. The second component is the advisor selection model, which assigns a set of advisors to every decision. The third is the decision model, which selects an answer after receiving the advisors' opinions. Figure 1 provides a graphical overview of the structure of MADDM.

3.1 Trustworthiness Model

Following Jøsang [7], we build our trustworthiness model using a Beta distribution. Recall that we do not know the ground truth. So, for each advisor, we associate two values, called *advice estimated to be correct* α_x and *advice estimated to be incorrect* β_x. Initially, these values are 1 [7]; i.e. we start with a prior that is Beta$(1, 1)$, or close to uninformative. We update these values whenever the advisor responds to a query. Correct answers to all decisions are *estimated* by our model without ground truth; we use the estimated answer to determine whether the advisor's answer is correct or not (see Section 3.3).

Now, for each advisor $x \in X$, we define its trustworthiness as $\tau_x = \alpha_x/(\beta_x + \alpha_x) \in (0, 1)$. If $\tau_x = 1$, we say that the advisor x is completely trustworthy. If $\tau_x = 0$, we say that the advisor x is completely untrustworthy.

This concept of trustworthiness is insufficient since it does not capture the epistemic uncertainty associated with that assessment. For this reason, each advisor's trustworthiness τ_x is paired with a parameter that quantifies this epistemic uncertainty behind the computation of τ_x. This uncertainty will reduce as we acquire more evidence regarding an advisor x. More specifically, we compute the uncertainty by using *Subjective Logic* [7]. This is commonly-employed method in computational models of trust in multi-agent systems and information fusion. Formally, for each advisor $x \in X$, the uncertainty of x is $\theta_x = 2/(\alpha_x + \beta_x) \in (0, 1]$.

3.2 Advisor Selection

The overall aim of the system is to maximize utility, $u(s)$ (see Equation 1), which requires balancing the trade-off between advisor costs and decision value. Typically, the costs of asking all advisers would exceed the decision value, even if the decision is correct, so it is rarely optimal. For example, for a decision with a value of \$10, it is not worth spending \$100 to hire advisors.

Our method selects the set of advisors according to the value of the problem and estimates their contributions to a decision. We assume their trustworthiness is initially unknown, and all advisers have equal trustworthiness. This knowledge is updated over time but is not reliable at first. Therefore, focusing too early on seemingly good advisors can lead to sub-optimal decisions. To address this, our system solves a multi-armed bandit problem in which it has to balance the exploration of new advisors with the exploitation of the knowledge it has already gathered. Among the many possible algorithms used to solve the multi-armed bandit problem, we use *Thompson Sampling* [1], which samples from a Beta distribution to compute the contribution of each advisor.

In Algorithm 1, we sketch the pseudo-code of our selection function s. Recall that $\vec{\tau}$ denotes the trustworthiness vector that contains the trustworthiness of each advisor, and $\vec{c}$ are their costs. Let $\vec{\alpha}$ and $\vec{\beta}$ denote the estimated evidence vectors, respectively. Given a decision $d \in D$, let P^{e+}_d and P^{e-}_d denote the probability that $a_d = 1$ and $a_d = -1$, respectively.[1] We denote with U_d the vector containing the advisors' utilities.

In more detail, after initializing the answer probabilities P^{e+}_d and P^{e-}_d, the answer sets P_d and N_d, the utility vector U_d, and the trustworthiness vector (Line 2), the model enters a loop for selecting advisors (Line 3). Let V^x_d, u^x_d denote the expected contribution and the marginal utility of the advisor x in decision d. Recall that c_x is the price of advisor x. Their relationship can be expressed as follows:

$$u^x_d = V^x_d - c_x. \tag{2}$$

In each round of advisor selection, we need to compute the marginal utility u^x_d of each advisor and select the one with the best u_{d,x^*}, which is our estimate of the advisor that maximizes the expected profit for the decision-maker (Lines 4-8).

Computing the marginal utility u^x_d is achieved in two steps. First, for each advisor, x, we define a Beta distribution Beta(α_x, β_x) and sample from it to get the Beta trustworthiness τ'_x. We only use it to compute the utility $u_{d,x}$ of the advisor x (Line 5), whereas the model does not use τ'_x for real decision-making. When there is little evidence regarding an advisor, e.g. when $\alpha = 1$ and $\beta = 1$, the

[1] These values are computed by the decision model, as we will see in Section 3.3.

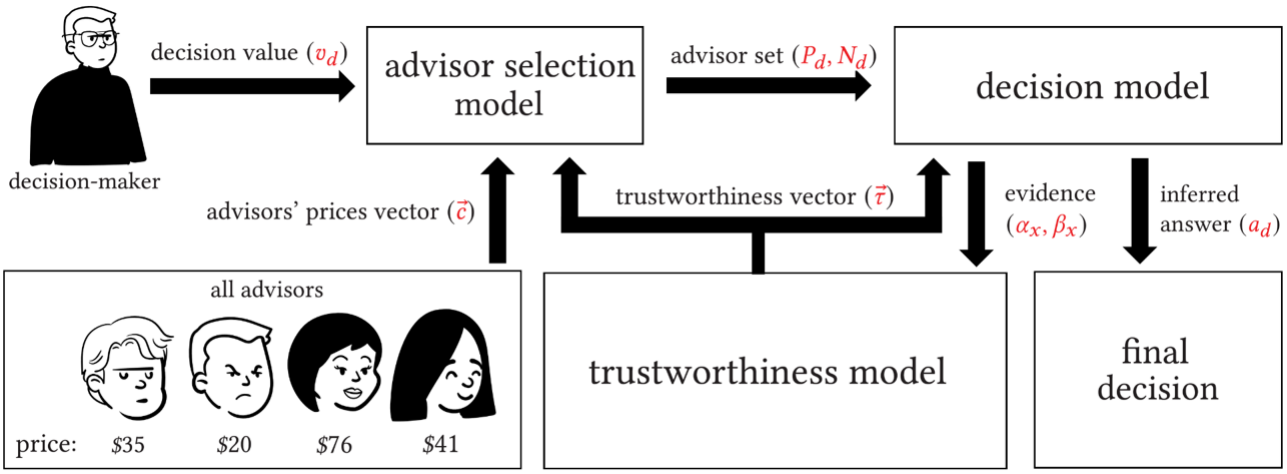


Figure 1: The advisor selection model can select a subset of advisors from all advisors to answer the decision. It needs to consider the decision of value and risk, the advisors' cost, and trustworthiness. The decision model uses advisors' trustworthiness and the answer set to decide the aggregating answer and the estimated evidence for updating the trustworthiness. The trustworthiness model builds and updates the advisors' trustworthiness.

Algorithm 1 Pseudo-code of the Advisor Selection algorithm

1: **Input**: $d, \vec{\tau}, \vec{\alpha}, \vec{\beta}, v_d^+, v_d^-, \vec{c}$
2: *initialize* $P_d^{e+} = P_d^{e-} = 0.5, U_d = P_d = N_d = \emptyset$
3: **while** *true* **do**
4: **for** advisor x in X **do**
5: $\tau'_x \leftarrow ThompsonSampling(\alpha_x, \beta_x)$
6: $u_{d,x} \leftarrow UtilityComputation(\tau'_x, v_d^+, v_d^-, P_d^{e+}, P_d^{e-})$
7: $U_d.append(u_{d,x})$
8: $u_{d,x^*} = Max(U_d)$
9: **if** $u_{d,x^*} > 0$ **do**
10: **if** $a_d^{x^*} = 1$ **do**
11: $P_d.append(x^*)$
12: **if** $a_d^{x^*} = -1$ **do**
13: $N_d.append(x^*)$
14: $P_d^{e+}, P_d^{e-} \leftarrow DecisionModel(P_d, N_d, \vec{\tau})$
15: $U_d = \emptyset$
16: $X.remove(x^*)$
17: **until** $u_{d,x^*} \leq 0$
18: **Output**: P_d, N_d

Beta distribution has a large variance. Consequently, the value τ'_x is subject to large fluctuations, which increases the decision error.

Second, we need to know the contribution V_d^x of each advisor x. Let us now assume that advisor x answered 1 to a decision d; the case in which the advisor answers -1 follows a similar routine. In order to compute its contribution, we first add x to the set P_d and proceed to calculate the probabilities $P_d^{e+'}$ and $P_d^{e-'}$. The value $P_d^{e+'}$ and $P_d^{e-'}$ describes the probability that $a_d = 1$ and $a_d = -1$, respectively. Therefore, the wider the gap between P_d^{e+} and P'_{e+}, the larger the advisor's contribution. We therefore set:

$$\Delta V_{d,+}^x = P_+ |P_d^{e+'} - P_d^{e+}| * (v_d^+ + v_d^-). \quad (3)$$

The values $P_+ := P(a_d^* = 1)$ and $P_- := P(a_d^* = -1)$ are the *a priori* probability that the answer is positive or negative, respectively. Hence the value $|P_d^{e+'} - P_d^{e+}|$ represents the change of the answer probability if advisor x participates in the decision. Similarly, if the advisor answers -1, we set:

$$\Delta V_{d,-}^x = P_- |P_d^{e-'} - P_d^{e-}| * (v_d^+ + v_d^-). \quad (4)$$

After we compute $\Delta V_{d,+}^x$ and $\Delta V_{d,-}^x$, we compute the expected contribution V_d^x as:

$$V_d^x = (\tau'_x - (1 - \tau'_x)) * (\Delta V_{d,+}^x + \Delta V_{d,-}^x). \quad (5)$$

Finally, the algorithm computes the utility u_d^x by Equation 2.

If $u_d^{x^*} > 0$, the advisor x^* is selected, which means that its contribution is greater than its cost. The selected advisor x^* needs to provide the answer for decision d. Depending on the answer from the advisor x^*, it can be added to P_d or N_d (Lines 9–13), which is used to update the answer probability P_d^{e+} and P_d^{e-} (Line 14). After every selection, we need to recalculate the marginal utility of each advisor for selecting the next advisor because their marginal utilities change. For example, if we select an advisor with 90% trustworthiness and give a positive answer to decision d, P_d^{e+} will increase from 50% to 90%. The model repeats Lines 4–16 to select advisors one by one until $u_{d,x^*} \leq 0$ (Line 17), and outputs the final answer set (P_d, N_d) (Line 18).

3.3 Bayesian and Weighted Voting Ensemble Decision Model

We use Bayesian and Weighted Voting Ensemble (BWVE) as the decision function f to make decisions [5]. There are two reasons for choosing BWVE. First, it is a truth inference method without ground truth. It has been shown to outperform the simple weighted voting

method, which considers the advisors' weights, determined by their trustworthiness, to bias majority voting [5]. Second, it returns a probability distribution over the answers, allowing us to evaluate each advisor's contribution, which aligns with our advisor selection model and retrospectively re-calibrates their trustworthiness.

In the following, we detail the BWVE procedure. Essentially, it combines two decision procedures to improve the overall outcome. One is based on a Bayesian model, while the other follows a weighted voting decision method. If we know the real trustworthiness $\vec{\tau}$ of all the advisors, the Bayesian method will obtain higher accuracy than the weighted voting method. However, in the beginning, because the uncertainty of the trustworthiness is large, the Bayesian method is unstable, so BWVE relies more on the weighted voting method for decisions. With the decreasing of the average uncertainty, the Bayesian method has a better performance. So BWVE uses the average uncertainty to control the weights of Bayesian and weighted voting automatically.

3.3.1 Bayesian. For every decision d, the advisor selection function returns a subset $Y_d \subseteq X$ that needs to answer the decision d. We recall that $P_d \subseteq Y_d$ denotes the set of advisors that answered 1 to the decision, while the advisors in $N_d \subseteq Y_d$ answered -1. Given the partition (P_d, N_d) of Y_d, from the Bayesian method, the probability that $a_d^* = 1$ is $P_d^{b+} := P_b(a_d^* = 1|P_d, N_d)$, while $P_d^{b-} := P_b(a_d^* = -1|P_d, N_d)$ is the probability that $a_d^* = -1$. From Bayes theorem, we can then express P_d^{b+} and P_d^{b-} as follows:

$$P_d^{b+} = \frac{P_+ P(P_d, N_d|a_d^* = 1)}{P_+ P(P_d, N_d|a_d^* = 1) + P_- P(P_d, N_d|a_d^* = -1)} \tag{6}$$

$$P_d^{b-} = \frac{P_- P(P_d, N_d|a_d^* = -1)}{P_- P(P_d, N_d|a_d^* = -1) + P_+ P(P_d, N_d|a_d^* = 1)}. \tag{7}$$

We recall that $P_+ := P(a_d^* = 1)$ and $P_- := P(a_d^* = -1)$ is the *a priori* probability that the answer is positive or negative, respectively. Since we do not have any evidence about a_d^*, both P_+ and P_- are equally likely, therefore we set $P_+ = P_- = 0.5$. The quantities $P(P_d, N_d|a_d^* = 1)$ and $P(P_d, N_d|a_d^* = -1)$ describe the probability to observe the partition (P_d, N_d) under the assumption that $a_d^* = 1$ and $a_d^* = -1$, respectively. Both $P(P_d, N_d|a_d^* = 1)$ and $P(P_d, N_d|a_d^* = -1)$ are computed through the trustworthiness τ_x as it follows:

$$P(P_d, N_d|a_d^* = 1) = \prod_{i \in P_d} \prod_{j \in N_d} \tau_i (1 - \tau_j) \tag{8}$$

$$P(P_d, N_d|a_d^* = -1) = \prod_{i \in P_d} \prod_{j \in N_d} \tau_j (1 - \tau_i) \tag{9}$$

3.3.2 Weighted Voting. The Bayesian decision method can only work well when the advisors' trustworthiness is sufficiently high. In the initial phase of the process, the advisors' trustworthiness is unreliable, so the Bayesian method is not stable. Since there is no ground truth, it is easily misled by bad advisors when the mean advisors' accuracy is not high. BWVE deals with this problem by using the weighted voting method, which is more robust than Bayesian at the beginning. Then, during the initialization, the weighted voting method has more influence on the decision than Bayesian. For the weighted voting method, under the answer set (P_d, N_d), the probability that the ground truth a_d^* = 1 and -1 are correct can be denoted as $P_d^{w+} := P_{wv}(a_d^* = 1|P_d, N_d)$ and $P_d^{w-} := P_{wv}(a_d^* = -1|P_d, N_d)$, respectively. The model then uses the sum of the advisors' trustworthiness to calculate them:

$$P_d^{w+} = \frac{\sum_{i \in P_d} \tau_i}{\sum_{j \in P_d \cup N_d} \tau_j} \tag{10}$$

$$P_d^{w-} = \frac{\sum_{i \in N_d} \tau_j}{\sum_{j \in P_d \cup N_d} \tau_j} \tag{11}$$

3.3.3 Ensemble Decision. BWVE uses the average uncertainty $\bar{\theta}_d$ to control the weights of the Bayesian and the weighted voting for decisions. The higher the average uncertainty of the advisors in the answer set Y_d, the lower reliability of trustworthiness and the more weight for the weighted voting method. Let $|Y_d|$ denote the cardinality of Y_d. It can be expressed as:

$$\bar{\theta}_d = \frac{\sum_{i \in Y_d} \theta_i}{|Y_d|} \tag{12}$$

The average uncertainty $\bar{\theta}_d$ gradually decreases as time goes on, and the weight of the Bayesian method needs to increase. For the ensemble decision, given the answer set (P_d, N_d), the probability that $a_d^* = 1$ is $P_d^{e+} := P_b(a_d^* = 1|P_d, N_d)$, while $P_d^{e-} := P_b(a_d^* = -1|P_d, N_d)$is the probability that $a_d^* = -1$. They can be expressed as:

$$P_d^{e+} = (1 - \bar{\theta}_d) P_d^{b+} + \bar{\theta}_d P_d^{w+} \tag{13}$$

$$P_d^{e-} = (1 - \bar{\theta}_d) P_d^{b-} + \bar{\theta}_d P_d^{w-} \tag{14}$$

Their relationship is:

$$P_d^{e+} + P_d^{e-} = 1 \tag{15}$$

After getting P_d^{e+} and P_d^{e-}, the system needs to compare them. If $P_d^{e+} > P_d^{e-}$, the final answer $a_d = 1$. Otherwise, $a_d = -1$.

3.3.4 Trustworthiness Update. BWVE uses the absolute difference of P_d^{e+} and P_d^{e-} as the new estimated evidence to update α and β.

$$i_d = |P_e(a_d^* = 1|P_d, N_d) - P_e(a_d^* = -1|P_d, N_d)| \tag{16}$$

If $a_d = 1$, the update of α_x and β_x can be expressed as:

$$\begin{aligned} \alpha_x &\leftarrow \alpha_x + i_d \quad \forall x \in P_d, \\ \beta_x &\leftarrow \beta_x + i_d \quad \forall x \in N_d, \end{aligned} \tag{17}$$

If $a_d = -1$, the update of α_x and β_x can be expressed as:

$$\begin{aligned} \beta_x &\leftarrow \beta_x + i_d \quad \forall x \in P_d, \\ \alpha_x &\leftarrow \alpha_x + i_d \quad \forall x \in N_d, \end{aligned} \tag{18}$$

3.3.5 Review Update. Recall that MADDM is an online problem without access to ground truth. Moreover, the initial trustworthiness is low. Therefore, the update of the trustworthiness $\vec{\tau}$ relies on the evidence from new decisions. And the decisions, in turn, rely on the trustworthiness $\vec{\tau}$. This dynamic loop is used for building the model to make the trustworthiness and the aggregating answer more accurate. Therefore, similar to the EM method, after every answer, we continuously update the trustworthiness of the advisors through the answers from past decisions.

Algorithm 2 describes how the review update works. Let $\vec{P}_{past}$, $\vec{N}_{past}$ denote the vector that contains the past answer set, and we

recall that $\vec{\tau}$ denote the trustworthiness vector that contains all advisors' trustworthiness. Let $\vec{\tau}_0$ denote the old trustworthiness vector, $\Delta\tau$ the sum of the difference between the old trustworthiness vector $\vec{\tau}_0$ and the new trustworthiness vector $\vec{\tau}$. Furthermore, let V_s denote the threshold of $\Delta\tau$ for terminating the update. V_s usually is set to a small value. Note that $\Delta\tau$ is used to judge the update step size of $\vec{\tau}$. Specifically, when $\Delta\tau$ is smaller than V_s, the model stops updating.

Algorithm 2 Pseudo-code of the review maximization algorithm

Input: $\vec{P}_{past}, \vec{N}_{past}, \vec{\tau}, V_s$
initialize $\Delta\tau = 0$, $\vec{\tau}_0 = \vec{\tau}$
while *true* **do**
  for P_d, N_d in $\vec{P}_{past}, \vec{N}_{past}$ **do**
    $P_d^{e+}, P_d^{e-} \leftarrow f(P_d, N_d, \vec{\tau})$
    $\vec{\tau}_0 = \vec{\tau}$
    $\vec{\tau} \leftarrow TrustworthinessUpdate(P_d^{e+}, P_d^{e-})$
    $\Delta\tau = sum(\vec{\tau} - \vec{\tau}_0)$
until $\Delta\tau \leq V_s$
Output: $\vec{\tau}$

4 EXPERIMENTS

In this section, we present the decision-answer experiments to evaluate our method. Specifically, we compare our method with two cost-constraint-based methods. The first is the Fixed Number of Advisors based method (FNA), which means that the decision-maker selects a fixed number for answering every decision. The second is the Budget-Constraint based method (BC), which means that there is a budget constraint to stop selecting advisors. For both approaches, we combine these with different advisor-selection criteria.

4.1 Setting

To the best of our knowledge, there is no standard environment to run decision experiments. For this reason, we rely on synthetically generated ones. In more detail, the environment we generate includes 1000 decisions with binary answers and different values. The full set of advisors consists of 30 simulated agents with different answer accuracy and costs. An Extended Rectified Gaussian distribution (ERGd) samples both the profits and losses of every decision [12]. We generate each advisor's real accuracy and cost using the same probability distribution. During the experiments, the decision-maker selects a set of advisors to enquire and infers the answers using different methods. After answering 1000 decisions, the decision-maker gets the final utility. Due to the probabilistic nature of the experiments, every experiment is repeated for 100 different runs to obtain statistically significant results. To reduce variance and bias, all methods are run using the same conditions. That is, although the conditions vary between runs, the same set of runs are used to compare the methods (i.e., using the same set of advisor qualities and prices, the same decision sequence, and the same decision profits and losses).

We consider different ratios between the decision's value and the advisor's cost, which leads us to define two sets of experiments.

Table 3: Experiment setting

setting	value
env1: decision profits v_d^+ $mean, std$	100, 100
env1: decision loss v_d^- $mean, std$	100, 100
env2: decision profits v_d^+ $mean, std$	500, 500
env2: decision loss v_d^- $mean, std$	500, 500
advisor cost c_x $mean, std$	0 to 20,10
real trustworthiness $mean, std$	from 0.51 to 1,0.3

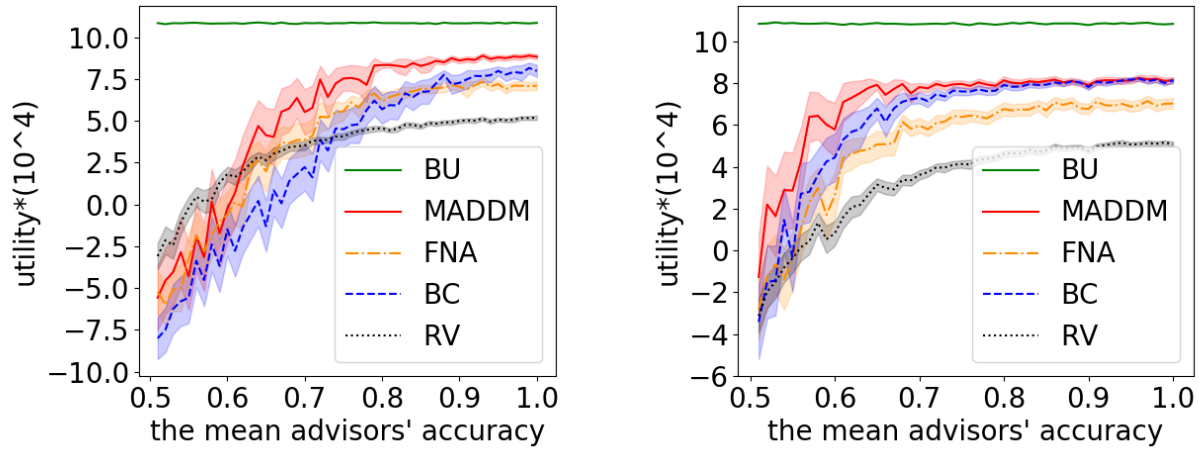
In the first set, both the decision profits and decision losses are sampled from an ERGd whose means and standard deviation are equal to 100. In the second one, the mean and the standard deviation of the ERGd are both changed to 500. Due to the large deviation, the decision values are highly volatile. Hence, some decisions may be worth more than 1000, and some may be worthless.

Furthermore, the real accuracy of advisor x, i.e. τ_x^r, is sampled from an ERGd whose standard deviation is fixed at 0.3 while its mean ranges in the set $\{0.5 + 0.01 * k\}$ where $k = 1, 2, \ldots, 50$. For example, if τ_x^r is 0.8, the advisor x has 80% probability of giving a correct answer. Hence, we consider 50 different frameworks in which the average trustworthiness increases every time. Finally, we assume that the cost of each advisor is proportional to its real trustworthiness. In practice, higher quality often comes at a cost. For example, senior advisors are more costly than junior ones. Similarly, more advanced machine learning algorithms typically require higher computational costs. However, this is only a correlation and not always the case for every instance. To achieve this correlation, the cost of each advisor is sampled from an ERGd whose average is $\tau_x^r * 20$ and whose standard deviation is 10. Note that the correlation makes the problem more challenging since the system has to make trade-offs between cost and quality. Without such correlation, there is a high likelihood of a cheap and reliable advisor which makes the problem easier to solve but also less realistic.

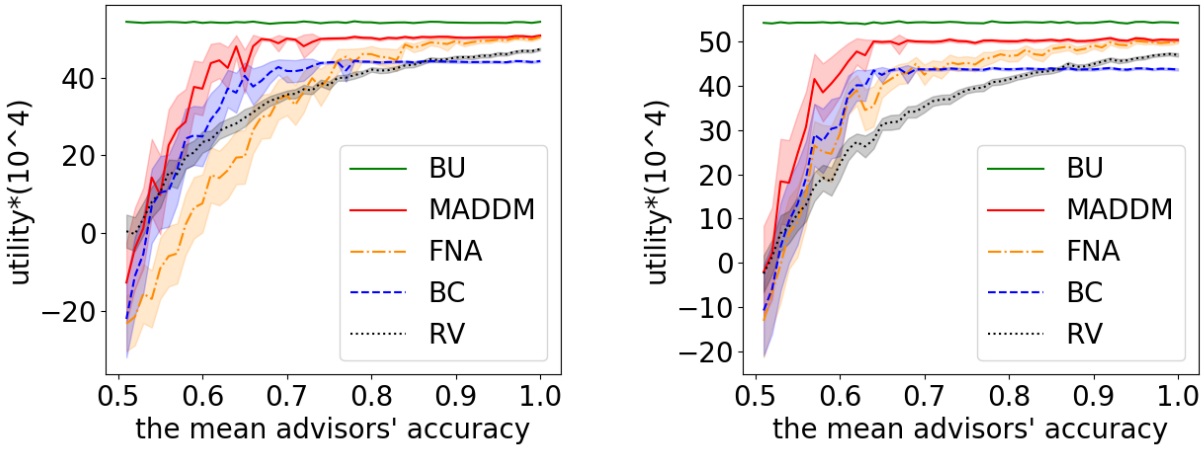
We used three different exploration methods (Upper Confidence Bound (UCB), Thompson Sampling, ϵ-greedy) and two rules of the advisor selection (trustworthiness, cost-effectiveness) to combine with FNA and BC, respectively. The aggregation method of FNA and BC is EM, which can maximize the sample utilization and has been verified multiple times in truth inference [3, 4].

In more detail, in terms of advisor selection strategies, UCB, Thompson Sampling and ϵ-greedy are effective for solving the multi-armed bandit problem. We experimented with a range of values and found that the ϵ-greedy method has the best performance when $\epsilon = 0.1$ (we also tested $\epsilon = 0.05, 0.15, 0.2, 0.25$) for all methods. UCB and Thompson Sampling explore more than ϵ-greedy at the beginning. Since the lack of ground truth, not every exploration can provide correct feedback for updating trustworthiness, especially when the average advisor's accuracy is low.

The criteria for advisor selection contain trustworthiness and cost-effectiveness. For example, if trustworthiness is the rule, the greedy strategy always selects the advisor with bigger trustworthiness. Cost-effectiveness is a method we improved from work [18]. The cost-effectiveness of the advisor x can be expressed by $c_x/(\tau_x - 0.5)$, which means how much cost is the improvement



(a) environment 1: mean acc, std = 100, 100



(b) environment 2: mean acc, std = 500, 500

Figure 2: In four figures, the X-axis represents the mean advisors' accuracy from 0.51 to 1. The Y-axis represents the average utility of 100 experiments. The half-transparent area, along with the curve, is the 95% confidence interval error bar. Figure (a) shows the results of environment 1 (mean, standard deviation = 100, 100). Figure (b) shows the results of environment 2 (mean, standard deviation = 500, 500). In Figures (a) and (b), the left figures represent the standard methods, and the right figures are the exploration-first-based methods. MADDM = multi-advisor dynamic decision-making(ours); FNA = ϵ-greedy fixed number of the advisor EM; BC = ϵ-greedy budget-limited EM; RV = random voting.

of trustworthiness for advisor x. It has a better performance than trustworthiness.

For FNA and BC, we also test their performance under different hyper-parameters. First, we test the performance of FNA by setting the number of advisors from 1 to 10. The results show that five advisors have the best performance. Second, BC, we try 5%, 10%, 15%, 20%, 25% of the value (profit + loss) of every decision as the budget constraint and 10% has the best performance.

To clearly understand the performance of our method, FNA and BC, we selected two other methods for comparison. The first is random voting (RV). It randomly selects three advisors and combines them by majority voting. Another one is the best utility (BU). It describes the maximum utility the decision-maker can get, which means all the decisions are correct, and the advisor cost is 0.

The method with the trustworthiness model is easily misled by malicious advisors when the mean advisors' accuracy is low [5]. In practical applications, the methods for solving the problem include adding some decisions with ground truth, selecting several advisors with high accuracy to participate in decision-making, or considering the prior information of advisors. In this paper, our assumptions are no ground truth and no prior information, so we design the exploration-first model to solve this problem. In the

Table 4: The red colour numbers The meaning of abbreviations are: env1(SD): environment 1 and standard methods, env2(SD): environment 2 and standard methods env1(EF): environment 1 and all methods with exploration-first model, env2(EF): environment 1 and all methods with exploration-first model.

	MADDM	FNA	BC	RV
env1(SD)	5.19 ± 7.68	3.76±5.95	2.85±7.51	3.27±2.84
env2(SD)	42.69 ± 27.38	30.89±33.03	35.46±28.11	34.16±16.31
env1(EF)	7.09 ± 4.43	5.15±4.32	6.26±4.79	3.24±2.93
env2(EF)	44.97 ± 22.84	38.89±24.11	37.58±22.68	34.14±16.20

first few decisions, the model selects all advisors for answering to increase the accuracy of the answer and then back to the method's standard advisor selection strategy. We use this model before rounds 1-15, respectively, and the results show that the three methods perform best when the model is used before the 10 round. Therefore, we added the exploration-first model to our method, FNA and BC and did additional experiments in two environments.

4.2 Results

We now compare the utility obtained by the different methods we considered. Table 4 shows the mean and standard deviation of the utility in every environment. Overall, our MADDM method has the best performance in terms of the average utility in almost all environments. In all the experiments, the average utilities obtained by the exploration-first methods are significantly bigger than the others. Moreover, the standard deviation of the utilities is also reduced, which means that the result is more consistent. We did 600 (3*50*4) pairs of Mann-Whitney Tests between MADDM and FNA, BC, and Random Voting (RV) with 50 different average advisors' accuracy in four different environments. We observe that 527 out of the 600 results have significant differences ($p < 0.05/3$).

Figure 2 describes the utility curves of different methods as the advisors' accuracy increases. In the vast majority of cases, MADDM gets more utility than FNA and BC for all the possible accuracy. In the right graph in Figures 2a and 2b, we compare the utilities when all the methods use the exploration-first based model.

RV is better than the other three methods when the mean advisors' accuracy is low. When there is no ground truth and a significant proportion of bias, the methods with the trustworthiness model are easily misled by malicious advisors. Once the trustworthiness model is misled, then malicious users take the initiative to sabotage future decisions. However, we observe that MADDM is less prone to be sabotaged. This is due to the fact that MADDM selects more advisors at the beginning and decreases as trustworthiness is updated, which means MADDM has stronger robustness to malicious advisors than FNA and BC.

Similarly, we observe that the performances of MADDM are more robust to the manipulation of the bad advisors when the average cost of the advisors and the decision values are bigger. Since the decisions in the environment, 2 are more valuable than the ones in the environment 1, MADDM chooses more advisors to make decisions together at the beginning in environment 2, which helps to increase the reliability of the answer. Therefore, based on this idea, we partially addressed this issue by using the exploration-first-based methods. The disadvantage of the exploration-first is that it uses more cost for building trustworthiness. It does not perform as well as standard methods when the mean advisors' accuracy is high. However, we do not know the real distribution of the mean advisors' accuracy and decision values before asking, so it is worth using some cost at first to improve the method's expected utility.

MADDM automatically selects the advisors by balancing the advisor's cost and the decision values without any hyper-parameters, which makes MADDM less prone to select an insufficient number of advisors or to waste costs. In the two methods based on cost-effectiveness, they need to set the number of advisors and budget proportion to control the advisor cost. If the prior distribution is unknown, the values of these hyper-parameters are difficult to determine. Furthermore, if the advisor cost is too small, the reliability of the output answer is not enough. While if the cost is too high, it causes a waste of advisor costs. For example, in Figure 2b, we observe that FNA does not select enough advisors when the mean advisors' accuracy is less than 0.8, whereas the best performance of BC has a gap with MADDM when the mean advisors' accuracy is higher than 0.65.

5 CONCLUSION

In this paper, we introduce Multi-Advisor Dynamic Decision-Making Method (MADDM), a novel approach for making optimal decisions in sequential decision-making settings with no ground truth. The model takes into account multiple variables, including the decision of profits and loss, advisors' costs, and trustworthiness. It selects advisors by balancing the advisors' costs and the value of making the correct decisions. It also makes decisions by combining the advice from multiple advisors without access to the ground truth and dynamically learns the trustworthiness of advisors without prior information. We test our method through decision-answer experiments in a simulated environment. We also introduce two benchmark methods, one using a fixed number of advisors (FNA) and another one using a fixed budget (BC), which are combined with state-of-the-art sampling and aggregating methods. The results show that MADDM significantly outperforms the benchmark methods.

An interesting direction for future work is moving from binary answers to multiple answers, making our approach applicable to more scenarios. This requires changing the calculations of the probabilities to deal with more than two outcomes. The first challenge in doing so is calculating the confidence value and how to use it for updating the trustworthiness. The second challenge is adjusting the weights of the weighted voting approach and the Bayesian method for making the decision. Another interesting direction is dealing with multiple simultaneous decisions at each point, which requires us to consider the allocation of advisors to each of the decisions.

REFERENCES

[1] Shipra Agrawal and Navin Goyal. 2012. Analysis of thompson sampling for the multi-armed bandit problem. In *Conference on learning theory*. JMLR Workshop and Conference Proceedings, PMLR, Edinburgh, Scotland, 39–1.

[2] Semih Cayci, Atilla Eryilmaz, and Rayadurgam Srikant. 2020. Budget-constrained bandits over general cost and reward distributions. In *International Conference on Artificial Intelligence and Statistics*. PMLR, PMLR, 4388–4398.

[3] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*. 469–478.

[4] Meric Altug Gemalmaz and Ming Yin. 2021. Accounting for Confirmation Bias in Crowdsourced Label Aggregation.. In *IJCAI*. 1729–1735.

[5] Zhaori Guo, Timothy Norman, and Enrico Gerding. 2022. MTIRL: Multi-trainer interactive reinforcement learning system. In *International Conference on Principles and Practice of Multi-Agent Systems*. Springer.

[6] Audun Jøsang. 2016. Generalising Bayes' theorem in subjective logic.. In *MFI*. 462–469.

[7] Audun Jøsang. 2016. *Subjective logic*. Vol. 3. Springer.

[8] Andrey Kurenkov, Ajay Mandlekar, Roberto Martin-Martin, Silvio Savarese, and Animesh Garg. 2020. AC-Teach: A Bayesian Actor-Critic Method for Policy Learning with an Ensemble of Suboptimal Teachers. In *Proceedings of the Conference on Robot Learning (Proceedings of Machine Learning Research, Vol. 100)*. PMLR, 717–734.

[9] Hélène Landemore. 2012. Collective wisdom: Old and new. *Collective wisdom: Principles and mechanisms* (2012), 1–20.

[10] Alexander Ly, Maarten Marsman, Josine Verhagen, Raoul PPP Grasman, and Eric-Jan Wagenmakers. 2017. A tutorial on Fisher information. *Journal of Mathematical Psychology* 80 (2017), 40–55.

[11] Xiaoye Miao, Huanhuan Peng, Yunjun Gao, Zongfu Zhang, and Jianwei Yin. 2022. On Dynamically Pricing Crowdsourcing Tasks. *ACM Transactions on Knowledge Discovery from Data (TKDD)* (2022).

[12] Andrew W Palmer, Andrew J Hill, and Steven J Scheding. 2017. Methods for Stochastic Collection and Replenishment (SCAR) optimisation for persistent autonomy. *Robotics and Autonomous Systems* 87 (2017), 51–65.

[13] Fangna Tao, Liangxiao Jiang, and Chaoqun Li. 2021. Differential evolution-based weighted soft majority voting for crowdsourcing. *Engineering Applications of Artificial Intelligence* 106 (2021), 104474.

[14] Yongxin Tong, Libin Wang, Zimu Zhou, Lei Chen, Bowen Du, and Jieping Ye. 2018. Dynamic pricing in spatial crowdsourcing: A matching-based approach. In *Proceedings of the 2018 International Conference on Management of Data*. 773–788.

[15] Long Tran-Thanh, Archie Chapman, Alex Rogers, and Nicholas Jennings. 2012. Knapsack based optimal policies for budget–limited multi–armed bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 26. 1134–1140.

[16] Long Tran-Thanh, Sebastian Stein, Alex Rogers, and Nicholas R Jennings. 2014. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence* 214 (2014), 89–111.

[17] Huiyang Wang, Diep N Nguyen, Dinh Thai Hoang, Eryk Dutkiewicz, and Qingqing Cheng. 2018. Real-time crowdsourcing incentive for radio environment maps: A dynamic pricing approach. In *2018 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–6.

[18] Yingce Xia, Haifang Li, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2015. Thompson sampling for budgeted multi-armed bandits. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

[19] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.

[20] Datong Zhou and Claire Tomlin. 2018. Budget-constrained multi-armed bandits with multiple plays. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.