Peer Prediction with Heterogeneous Users

ARPIT AGARWAL, University of Pennsylvania
DEBMALYA MANDAL and DAVID C. PARKES, Harvard University
NISARG SHAH, University of Toronto

Peer prediction mechanisms incentivize agents to truthfully report their signals, in the absence of a verification mechanism, by comparing their reports with those of their peers. Prior work in this area is essentially restricted to the case of homogeneous agents, whose signal distributions are identical. This is limiting in many domains, where we would expect agents to differ in taste, judgment, and reliability. Although the Correlated Agreement (CA) mechanism [Shnayder et al. 2016a] can be extended to handle heterogeneous agents, there is a new challenge of efficiently estimating agent signal types. We solve this problem by clustering agents based on their reporting behavior, proposing a mechanism that works with clusters of agents, and designing algorithms that learn such a clustering. In this way, we also connect peer prediction with the Dawid and Skene [1979] literature on latent types. We retain the robustness against coordinated misreports of the CA mechanism, achieving an approximate incentive guarantee of ε-informed truthfulness. We show on real data that this incentive approximation is reasonable in practice, even with a small number of clusters.

CCS Concepts: • **Theory of computation** → **Algorithmic mechanism design**; • **Computing methodologies** → **Artificial intelligence**;

Additional Key Words and Phrases: Peer prediction, information elicitation, clustering, tensor decompsition

ACM Reference format:
Arpit Agarwal, Debmalya Mandal, David C. Parkes, and Nisarg Shah. 2020. Peer Prediction with Heterogeneous Users. *ACM Trans. Econ. Comput.* 8, 1, Article 2 (March 2020), 34 pages.
https://doi.org/10.1145/3381519

1 INTRODUCTION

Peer prediction is the technique of eliciting truthful information in the absence of verification by comparing an agent's response with those of their peers. Peer prediction mechanisms incentivize users to provide honest reports when the reports cannot be verified, either because there is no objective ground truth or because it is costly to acquire the ground truth. Peer prediction mechanisms leverage correlation in the reports of peers to score contributions. The main challenge of peer prediction is to incentivize agents to put in effort to obtain a signal or form an opinion and then honestly report to the system. In recent years, peer prediction has been studied in several

This work is supported in part by Google, the SEAS TomKat fund, and NSF grant CCF-1301976.
Authors' addresses: A. Agarwal, University of Pennsylvania; email: aarpit@seas.upenn.edu; D. Mandal, Columbia University; email: dm3557@columbia.edu; D. C. Parkes, Harvard University; email: parkes@eecs.harvard.edu; N. Shah, University of Toronto; email: nisarg@cs.toronto.edu.

2167-8375/2020/03-ART2 $15.00
https://doi.org/10.1145/3381519

domains, including peer assessment in massively open online courses (MOOCs) [Gao et al. 2016; Shnayder and Parkes 2016], for feedback on local places in a city [Mandal et al. 2016], and in the context of collaborative sensing platforms [Radanovic and Faltings 2015a].

The simplest peer prediction mechanism is *output agreement*, which pairs up two users and rewards them in the event that their reports agree (the ESP game [Von Ahn and Dabbish 2004] can be interpreted this way). However, output agreement is not incentive aligned for reports of *a priori* unlikely signals. As a result, there has been a lot of attention in recent years on finding methods that work more generally and provide robustness to coordinated misreports.

All existing, general methods are essentially restricted to settings with homogeneous participants, whose signal distributions are identical. This is a poor fit with many suggested applications of peer prediction. Consider, for example, the problem of peer assessment in MOOCs. DeBoer et al. [2013] and Wilkowski et al. [2014] observe that students differ based on their geographical locations, educational backgrounds, and level of commitment, and indeed the heterogeneity of assessment is clear from a study of Coursera data [Kulkarni et al. 2015]. Simpson et al. [2013] observed that the users participating in a *citizen science* project can be categorized into five distinct groups based on their behavioral patterns in classifying an image as a Supernovae or not. A similar problem occurs in determining whether a news headline is offensive or not. Depending on which social community a user belongs to, we should expect to get different opinions [Zafar et al. 2016]. Moreover, Allcott and Gentzkow [2017] report that leading to the 2016 U.S. presidential election, people were more likely to believe the stories that favored their preferred candidate; Fourney et al. [2017] find that there is very low connectivity among Trump and Clinton supporters on social networks, which leads to confirmation bias among the two groups and clear heterogeneity about how they believe whether a piece of news is “fake” or not.

One obstacle to designing peer prediction mechanisms for heterogeneous agents is an impossibility result. No mechanism can provide strict incentives for truth-telling to a population of heterogeneous agents without knowledge of their signal distributions [Radanovic and Faltings 2015c]. This negative result holds for minimal mechanisms, which only elicit signals and not beliefs from agents. One way to alleviate this problem, without going to non-minimal mechanisms, is to use reports from the agents across multiple tasks to estimate their signal distributions. This is our goal: We want to design minimal peer prediction mechanisms for heterogeneous agents that use reports from the agents for both learning and scoring. We also want to provide robustness against coordinated misreports.

As a starting point, we consider the *correlated agreement* (CA) mechanism proposed by Shnayder et al. [2016a]. If the agents are homogeneous and the designer has knowledge of their joint signal distribution, then the CA mechanism is *informed truthful*, i.e., no strategy profile, even if coordinated, can provide more expected payment than truth-telling, and the expected payment under an uninformed strategy (where an agent’s report is independent of her signal) is strictly less than the expected payment under truth-telling. These two properties remove any incentive for coordinated deviations and strictly incentivize the agents to put effort into acquiring signals, respectively. In a detail-free variation, in which the designer learns the signal distribution from reports, approximate incentive alignment is provided (still maintaining the second property as a strict guarantee). The detail-free CA mechanism can be extended to handle agent heterogeneity, but a naïve approach would require learning the joint signal distributions between every pair of agents, and the total number of reports that need to be collected would be prohibitive for many settings.

Our contributions: We design the first minimal and detail-free mechanism for peer prediction with heterogeneous agents, where the learning component has sample complexity that is only linear in the number of agents, while providing an incentive guarantee of approximate informed truthfulness. Like the CA mechanism, this is a multi-task mechanism in that each agent makes reports across multiple tasks. Our mechanism is robust to any coordination between agents as long as the task assignments are such that from an agent's perspective every other agent is equally likely to be her peer. Hence, our mechanism is robust to any coordination between agents that happens prior to task assignment. Our mechanism will also be robust to coordinations after task assignments as long as the agents are not able to figure out which agents are more likely to be their peers based on the identity of the tasks they are assigned. For example, in the context of a MOOC, the organizer can anonymize the homeworks to be graded, and hence, it will require a lot of effort for students to figure out whose homeworks they are grading even after the homeworks have been assigned for grading. Since our mechanism has a learning component, the task assignments to agents should also be such that both the goals of incentive alignment and learning are simultaneously achieved. We consider two assignment schemes under which these goals can be achieved and analyze the sample complexity of our methods for these schemes.

The mechanism clusters the agents based on their reported behavior[1] and learns the pairwise correlations between these clusters. The clustering introduces one component of the incentive approximation, and it could be problematic in the absence of a good clustering such that agents within a cluster behave similarly. Using eight real-world datasets, which contain reports of users on crowdsourcing platforms for multiple labeling tasks, we show that the clustering error is small in practice even when using a relatively small number of clusters. The second component of the incentive approximation stems from the need to learn the pairwise correlations between clusters; this component can be made arbitrarily small using a sufficient number of signal reports.

Another contribution of this work is to connect, we believe for the first time, the peer prediction literature with the extensive and influential literature on latent, confusion matrix models of label aggregation [Dawid and Skene 1979]. The Dawid-Skene model assumes that signals are generated independently, conditional on a latent attribute of a task and according to an agent's confusion matrix. We cluster the agents based on their confusion matrices and then estimate the average confusion matrices within clusters using recent developments in tensor decomposition algorithms [Anandkumar et al. 2014; Zhang et al. 2016]. These average confusion matrices are then used to learn the pairwise correlations between clusters and design reward schemes to achieve approximate informed truthfulness.

In effect, the mechanism learns how to map one agent's signal reports onto the signal reports of the other agents. For example, consider the context of a MOOC, in which an agent in the "accurate" cluster accurately provides grades, an agent in the "extremal" cluster only uses grades "A" and "E," and an agent in the "contrarian" cluster flips good grades for bad grades and vice versa. The mechanism might learn to positively score an "A" report from an "extremal" agent matched with a "B" report from an "accurate" agent, or matched with an "E" report from a "contrarian" agent for the same essay. In practice, our mechanism will train on the data collected during a semester of peer assessment reports and then cluster the students, estimate the pairwise signal distributions between clusters, and accordingly score the students (i.e., the scoring is done retroactively).

1.1 Related Work

We provide a brief review of the related work in peer prediction and suggest Faltings and Radanovic [2017] for a detailed discussion. We focus our discussion on related work about minimal mechanisms but remark that we are not aware of any non-minimal mechanisms (following from the work

[1] One could also consider clustering the agents based on their observable covariates as long as agents with similar covariates have similar "signal type." However, in the applications that we consider in this article, for example MOOCs, such covariates may not be observable, and hence, we only rely on agent reports for clustering.

of Prelec [2004]) that handle agent heterogeneity. Miller et al. [2005] introduce the peer prediction problem and propose an incentive-aligned mechanism for the single-task setting. However, their mechanism requires knowledge of the joint signal distribution and is vulnerable to coordinated misreports. In regard to coordinated misreports, Jurca et al. [2009] show how to eliminate uninformative, pure-strategy equilibria through a three-peer mechanism, and Kong et al. [2016] provide a method to design robust, single-task, binary signal mechanisms (but need knowledge of the joint signal distribution). Frongillo and Witkowski [2017] provide a characterization of minimal (single task) peer prediction mechanisms.

Witkowski and Parkes [2013] introduce the combination of learning and peer prediction, coupling the estimation of the signal prior together with the shadowing mechanism. Some results make use of reports from a large population. Radanovic and Faltings [2015b], for example, establish robust incentive properties in a large-market limit where both the number of tasks and the number of agents assigned to each task grow without bound. Radanovic et al. [2016] provide complementary theoretical results, giving a mechanism in which truthfulness is the equilibrium with the highest payoff in the asymptote of a large population and with a structural property on the signal distribution.

Dasgupta and Ghosh [2013] show that robustness to coordinated misreports can be achieved for binary signals in a small population by using a multi-task mechanism. The idea is to reward agents if they provide the same signal on the same task, but punish them if one agent's report on one task is the same as another's on a different task. The Correlated Agreement (CA) mechanism [Shnayder et al. 2016a] generalizes this mechanism to handle multiple signals and uses reports to estimate the correlation structure on pairs of signals without compromising incentives. In related work, Kong and Schoenebeck [2016] show that many peer prediction mechanisms can be derived within a single information-theoretic framework. Their results use different technical tools than those used by Shnayder et al. [2016a] and also include a different multi-signal generalization of the Dasgupta-Ghosh mechanism that provides robustness against coordinated misreports in the limit of a large number of tasks. Shnayder et al. [2016b] adopt replicator dynamics as a model of population learning in peer prediction and confirm that these multi-task mechanisms (including the mechanism by Kamble et al. [2015]) are successful at avoiding uninformed equilibria.

There are very few results on handling agent heterogeneity in peer prediction. For binary signals, the method of Dasgupta and Ghosh [2013] is likely to be an effective solution, because their assumption on correlation structure will tend to hold for most reasonable models of heterogeneity. But it will break down for more than two signals, as explained by Shnayder et al. [2016a]. Moreover, although the CA mechanism can in principle be extended to handle heterogeneity, it is not clear how the required statistical information about joint signal distributions can be efficiently learned and coupled with an analysis of approximate incentives. For a setting with binary signals and where each task has one of a fixed number of latent types, Kamble et al. [2015] design a mechanism that provides strict incentive compatibility for a suitably large number of heterogeneous agents and when the number of tasks grows without bound (while allowing each agent to only provide reports on a bounded number of tasks). Their result is restricted to binary signals and requires a strong regularity assumption on the generative model of signals. Kong and Schoenebeck [2016] design an information theoretic framework for peer prediction. Their mechanism pays each agent the mutual information between her report and her peer's report. This mechanism can be extended to the heterogeneous agents setting as long as we can measure the mutual information between all pairs of agents. However, such a mechanism would require the agents to provide reports on a large number of tasks.

Finally, we consider only binary effort of a user, i.e., the agent either invests effort and receives an informed signal or does not invest effort and receives an uninformed signal. Shnayder et al. [2016a]

work with the binary effort setting and provide strict incentive for being truthful. Therefore, as long as the mechanism designer is aware of the cost of investing effort, the payments can be scaled to cover the cost of investing effort. The importance of motivating effort in the context of peer prediction has also been considered by Liu and Chen [2017b] and Witkowski et al. [2013].[2] See Mandal et al. [2016] for a setting with heterogeneous tasks but homogeneous agents. Liu and Chen [2017a] also designed single-task peer prediction mechanism for the same setting but only when each task is associated with a latent ground truth.

2 MODEL

Let notation $[t]$ denote $\{1, \ldots, t\}$ for $t \in \mathbb{N}$. We consider a population of agents $P = [\ell]$ and use indices such as p and q to refer to agents from this population. There is a set of tasks $M = [m]$. For example, a task can be either grading an essay or answering a question in an online rating sytem. When an agent performs a task, she receives a signal from $N = [n]$. Such a signal usually indicates the quality of the task, i.e., the number of points assigned to the essay or how good the food is at a restaurant. The agents need to put in some effort to get an informative signal about the task. As mentioned before, we assume that the effort of an agent is binary, i.e., either the agent puts full effort and receives an informative signal or the agent puts no effort and receives a signal drawn uniformly at random. We also assume that the tasks are *ex ante* identical; that is, the signals of an agent for different tasks are sampled i.i.d. For example, in the essay-grading scenario, if the essays assigned to any student are drawn uniformly at random from a large population of essays, the student's signal distribution for an assigned essay is *ex ante* almost identical to any other assigned essay.

Each agent is assigned a set of tasks and she decides, for each task, whether to put in effort and receive an informative signal or put in no effort and receive a random signal. This provides the agent with a set of signals, one for each task. Then the agent reports back to mechanism designer a set of signals, one for each assigned task. Before putting in any effort to receive informative signals, the agents have no knowledge about the tasks apart from the fact they are *ex ante* identical. Once the agents receive their signals, their reports are determined completely by these signals. In other words, the agents do not use any additional information to determine their reports. We will assume that, for each task, the message space and the signal space are the same. Since the payments made to the agents depend on their reported signals (*messages*), the reported signals can be very different than the observed signals. The goal of a peer prediction mechanism is to ensure that the agents put effort into all the tasks and report their signals truthfully. For the MOOC setting, a student spends some amount of time to figure out the grade of each of her assigned essays. She might also decide to not look at an essay and report an arbitrary grade. The goal of our mechanism is to ensure that the students put in some effort to determine the grades of the essays and report them truthfully back to the platform. We work in the setting where the agents are heterogeneous, i.e., the distribution of signals can be different for different agents. These differences are captured by the agents' types, and we say that the agents vary by *signal type*. In peer prediction, we compare the reports of an agent to the reports of their peers on the same tasks, and hence, we also need to talk about joint signal distribution of pairs of agents in addition to the signal distribution of an individual agent. In our setting, these joint signal distributions can be different for different pairs of agents.

Let S_p, S_q denote random variables for the signal observed by agents p and q on some task. Let $D_{p,q}(i, j)$ denote the joint probability that agent p receives signal i while agent q receives

[2] Cai et al. [2015] work in a different model, showing how to achieve optimal statistical estimation from data provided by rational agents. They only focus on the cost of effort. They do not consider possible misreports, and thus their mechanism is also vulnerable to coordinated misreports.

signal j on a task, i.e., $D_{p,q}(i,j) = \Pr(S_p = i, S_q = j)$. Let $D_p(i)$ and $D_q(j)$ denote the corresponding marginal probabilities, i.e., $D_p(i) = \Pr(S_p = i)$ and, $D_q(j) = \Pr(S_q = j)$. An important part of our mechanisms are the *delta matrices* that are defined as follows. We define the *Delta matrix* $\Delta_{p,q}$ between agents p and q as

$$\Delta_{p,q}(i,j) = D_{p,q}(i,j) - D_p(i) \cdot D_q(j), \ \forall i,j \in [n]. \tag{1}$$

The delta matrices capture the correlation between pairs of realized signals. For example, if $\Delta_{p,q}(1,2) = D_{p,q}(1,2) - D_p(1)D_q(2) > 0$. This implies that $\Pr[S_p = 1|S_q = 2] > \Pr[S_p = 1]$. Therefore, the event of agent p observing signal 1 is positively correlated with the event of agent q observing signal 2. This would also mean that the event that agent p receives signal 1 and agent q receives signal 2 is more likely when these signals are for the same task than when they are for different tasks. Our mechanism will use these correlations to decide the score for an agent given the reports of the agent and her peers. The *correlated agreement* (CA) mechanism [Shnayder et al. 2016a] also uses these delta matrices to construct a scoring mechanism for agent reports; however, they work in a setting where agents are *exchangeable*, i.e., the delta matrix $\Delta_{p,q}$ is the same for all pairs p, q of agents.

Example 2.1. For two agents p and q, consider the following joint signal distribution $D_{p,q}$ is

$$D_{p,q} = \begin{bmatrix} 0.2 & 0.3 \\ 0.1 & 0.4 \end{bmatrix}$$

with marginal distributions $D_p = [0.5 \ \ 0.5]$ and $D_q = [0.3 \ \ 0.7]$, the Delta matrix $\Delta_{p,q}$ is

$$\Delta_{p,q} = \begin{bmatrix} 0.2 & 0.3 \\ 0.1 & 0.4 \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \cdot \begin{bmatrix} 0.3 & 0.7 \end{bmatrix} = \begin{bmatrix} 0.05 & -0.05 \\ -0.05 & 0.05 \end{bmatrix}.$$

An agent's *strategy* defines, for every signal it may receive and each task it is assigned, a probability distribution over signals it will report. Shnayder et al. [2016a] show that it is without loss of generality for the class of mechanisms we study in this article to assume that an agent's strategy is uniform across different tasks. Hence, we will make the assumption that an agent's strategy is uniform across tasks. Formally, let R_p denote the random variable for the report of agent p for a given task. The strategy of agent p, denoted F^p, defines the distribution of reports for each possible signal i, with $F^p_{ir} = \Pr(R_p = r|S_p = i)$. Therefore, if there are n signals, then the strategy $F^p : [n] \to \mathcal{P}_n$, where $\mathcal{P}_n$ is the set of all possible distributions with support in $[n]$. The collection of agent strategies, denoted $\{F^p\}_{p \in P}$, is the *strategy profile*. A strategy of agent p is *informed* if there exist distinct $i, j \in [n]$ and $r \in [n]$ such that $F^p_{ir} \neq F^p_{jr}$, i.e., if not all rows of F^p are identical. We say that a strategy is *uninformed* otherwise.

2.1 Multi-task Peer Prediction

In this article, we consider *multi-task peer prediction* mechanisms defined in Shnayder et al. [2016a] and extend them to the setting of heterogeneous agents. In these mechanisms, each agent performs multiple tasks, and the score of an agent depends on its reports and the reports of its peers. For each agent, a random subset of her tasks is designated as *bonus tasks*, and its complement is designated as *penalty tasks*, without the knowledge of the agent. These mechanisms are characterized by *scoring matrices* for each pair of agents, which are used to score agents' reports. In our mechanism, the scoring matrix $S_{p,q} : [n] \times [n] \to \{0, 1\}$ for agent pair p and q will be such that $S_{p,q}(i,j) = 1$ when the event that agent p receives signal i is positively correlated with the event that agent q receives signal j on the same task, otherwise $S_{p,q}(i,j) = 0$. We will thus use the delta matrices (which will be learnt from agent reports) to design these scoring matrices.

For signals i and j, if $S_{p,q}(i,j) = 1$, then, for each bonus task of an agent p, we will add 1 to her score for reporting i when the report of its peer agent q on the same task is j, otherwise, we will not add anything. Additionally, for each bonus task of agent p, we randomly select a penalty task and subtract some score her total score based on her report on the penalty task. For signals i and j, if $S_{p,q}(i,j) = 1$, then, we will subtract 1 from her score for reporting i on the penalty task when the report of its peer agent q on a different task is j, otherwise, we will not subtract anything. The penalty is included in the score to avoid "uninformative equilibria," where agents agree to report the same signal on every task without investing effort in gathering the signals. The total score of an agent will be the sum of all the scores over all bonus tasks calculated this way.

In our mechanism, the score of an agent on a bonus task will be "+1" when its report is positively correlated with the report of its peer agent on the same task. The score of an agent on a penalty task will be "−1" when its report is positively correlated with the report of its peer on a different task. The intuition behind our mechanism is that when signals i and j of agents p and q are correlated, then it will be more likely that agents receive this pair of signals on tasks they share than on tasks they do not share. Hence, the overall score will be positive in expectation when agents are truthful. Whenever the agents use any uninformed strategy then the event that "the report of agent p is i and the report of agent q is j" is as likely to happen when they perform the same task as it is when they perform different tasks. Hence, the expected payment of any uninformed strategy will be zero. The *correlated agreement* (CA) mechanism [Shnayder et al. 2016a] also uses a scoring matrix for scoring agent. However, in their homogeneous setting, only one scoring matrix is required, because the delta matrices are the same for each pair of agents. In our heterogeneous setting, we have to use different scoring matrices for different pairs of agents.

Formally, for agent p, we denote the set of her bonus tasks by M_1^p and the set of her penalty tasks by M_2^p. To calculate the payment to an agent p for a bonus task $t \in M_1^p$, we do the following:

(1) Randomly select an agent $q \in P \setminus \{p\}$ such that $t \in M_1^q$, and the set $M_2^p \cup M_2^q$ has at least 2 distinct tasks, and call q the *peer* of p.
(2) Pick tasks $t' \in M_2^p$ and $t'' \in M_2^q$ randomly such that $t' \neq t''$ (t' and t'' are the penalty tasks for agents p and q, respectively).
(3) Let the reports of agent p on tasks t and t' be r_p^t and $r_p^{t'}$, respectively, and the reports of agent q on tasks t and t'' be r_q^t and $r_q^{t''}$, respectively.
(4) The payment of agent p for task t is then $S_{p,q}(r_p^t, r_q^t) - S_{p,q}(r_p^{t'}, r_q^{t''})$.

The total payment to an agent is the sum of payments for the agent's bonus tasks.

2.2 Task Assignments

Since we work in the setting where agents perform multiple tasks, it is important to address how these tasks are assigned to agents. Our mechanism has two requirements from any task assignment.

(1) From an agent's perspective, every other agent is equally likely to be her peer. This requires agents not to know each other's task assignments before deciding a strategy. For example, if agents of one "type" are more likely to be peers with agents of another "type" based on their task assignments, then they can coordinate amongst themselves to decide a more profitable strategy than truth-telling. Our mechanism will be robust to coordinations that happen before the task assignments. Our mechanism will also be robust to coordinations after task assignments as long as the agents are not able to figure out which agents are more likely to be their peers based on the identity of the tasks they are assigned.

(2) We should always be able to find a peer agent q for any agent p. Precisely, the tasks are assigned in a way that for every agent p, we can find a peer agent q such that q has performed at least one bonus task that p has performed, and we have reports from p and q for two different tasks that are not the same as the bonus task.

In addition, our mechanism has a learning component, where we learn about the correlation between agents' signals and also cluster agents into groups. Hence, to learn these quantities, we need to collect sufficient reports from each agent. This imposes some other requirements for the task assignment. In Section 4, we propose two task assignment schemes that a principal can use that satisfy all these requirements.

2.3 Expected Payments

The expected payment to agent p under strategy profile $\{F^q\}_{q \in P}$ for any bonus task performed by her, equal across all bonus tasks as the tasks are *ex ante* identical, is given as

$$\begin{aligned} u_p(F^p, \{F^q\}_{q \neq p}) &= \frac{1}{\ell - 1} \sum_{q \neq p} \left\{ \sum_{i,j} D_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q) \right. \\ &\quad \left. - \sum_i D_p(i) \sum_{r_p} F^p_{ir_p} \sum_j D_q(j) \sum_{r_q} F^q_{jr_q} S_{p,q}(r_p, r_q) \right\} \\ &= \frac{1}{\ell - 1} \sum_{q \neq p} \left\{ \sum_{i,j} \Big(D_{p,q}(i,j) - D_p(i) D_q(j) \Big) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q) \right\} \\ &= \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q). \end{aligned} \tag{2}$$

2.4 Informed Truthfulness

Following Shnayder et al. [2016a], we define the notion of approximate informed truthfulness for a multi-task peer prediction mechanism.

Definition 2.2 (ε-informed truthfulness). We say that a multi-task peer prediction mechanism is ε-informed truthful, for some $\varepsilon \geqslant 0$, if and only if for every strategy profile $\{F^q\}_{q \in P}$ and every agent $p \in P$, we have $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant u_p(F^p, \{F^q\}_{q \neq p}) - \varepsilon$, where $\mathbb{I}$ is the truthful strategy, and $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > u_p(F^p_0, \{F^q\}_{q \neq p})$ where F^p_0 is an uninformed strategy.

An ε-informed truthful mechanism ensures that every agent prefers (up to ε) the truthful strategy profile over any other strategy profile and strictly prefers the truthful strategy profile over any uninformed strategy. Moreover, no coordinated strategy profile provides more expected utility than the truthful strategy profile (up to ε). For a small ε, this is responsive to the main concerns about incentives in peer prediction: a minimal opportunity for coordinated manipulations and a strict incentive to invest effort in collecting and reporting an informative signal.[3]

2.5 Learning and Agent Clustering

Suppose that one knows $\Delta_{p,q}$ for every pair of agents, then one can calculate the scoring matrices $S_{p,q}$ according to these delta matrices and use these scoring matrices to score the agents. It is not

[3] We do not model the cost of effort explicitly in this article, but a binary cost model (effort → signal, no-effort → no signal) can be handled in a straightforward way. See Shnayder et al. [2016a].

hard to prove (see Lemma 3.4 for a proof) that such an extension of the CA mechanism will be informed truthful. However, we seek to design a detail-free mechanism where one does not have the knowledge of delta matrices, and one needs to learn them from agent reports. However, it would require $\Omega(\ell^2)$ samples to learn the delta matrices between every pair of agents, which will often be impractical. Rather, the number of reports in a practical mechanism should scale closer to linearly in the number of agents.

In response, we will assume that agents can be (approximately) clustered into a bounded number K of agent signal types, such that agents of the same type have similar signal distributions. Hence, a cluster of agents will be treated as a meta-agent, and we will work with signal distributions of these meta-agents. Formally, let $G_1, \ldots, G_K$ denote a partitioning of agents into K clusters. With a slight abuse of notation, we also use $G(p)$ to denote the cluster to which agent p belongs.

To reduce the sample complexity of our mechanism, we want that the clustering of agents to be such that for each pair p, q of agents, the signals of meta-agents (clusters) $G(p)$ and $G(q)$ are correlated in a similar manner as the signals of agents p and q. With this in mind, for $s, t \in [K]$, let us define the cluster Delta matrix between clusters G_s and G_t to be the average signal correlation taken over all pairs of agents $p \in G_s$ and $q \in G_t$, i.e.,

$$\Delta_{G_s, G_t} = \begin{cases} \frac{1}{|G_s| \times |G_t|} \sum_{p \in G_s, q \in G_t} \Delta_{p,q} & \text{if } s \neq t \\ \frac{1}{|G_s|^2 - G_s} \sum_{p, q \in G_s, q \neq p} \Delta_{p,q} & \text{if } s = t \end{cases}.$$

Now, the clustering of agents should be such that for each pair of agents p, q, we should be able to use $\Delta_{G(p), G(q)}$ as a proxy for $\Delta_{p,q}$. This allows us to learn the Delta matrices for every cluster pair, instead of learning Delta matrices for every agent pair. This intuition results in the following definition of an ε_1-*accurate* clustering:

Definition 2.3. We say that clustering $G_1, \ldots, G_K$ is ε_1-*accurate*, for some $\varepsilon_1 \geqslant 0$, if for every pair of agents $p, q \in P$,

$$\|\Delta_{p,q} - \Delta_{G(p), G(q)}\|_1 \leqslant \varepsilon_1, \tag{3}$$

where $\Delta_{G(p), G(q)}$ is the *cluster Delta matrix* between clusters $G(p)$ and $G(q)$.

Example 2.4. Let there be four agents p, q, r, and s. Let the pairwise Delta matrices be the following:

$$\Delta_{p,q} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \Delta_{p,r} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \Delta_{p,s} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix}$$
$$\Delta_{q,r} = \begin{bmatrix} -0.05 & 0.05 \\ 0.05 & -0.05 \end{bmatrix}, \Delta_{q,s} = \begin{bmatrix} -0.15 & 0.15 \\ 0.15 & -0.15 \end{bmatrix}, \Delta_{r,s} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}.$$

In this example, agents p and q tend to agree with each other, while agents r and s tend to agree with each other while disagreeing with p and q. Let the clustering be G_1, G_2 where p, q belong to G_1 and r, s belong to G_2. Then the cluster Delta matrices are the following:

$$\Delta_{G_1, G_1} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}, \Delta_{G_1, G_2} = \begin{bmatrix} -0.1 & 0.1 \\ 0.1 & -0.1 \end{bmatrix}, \Delta_{G_2, G_2} = \begin{bmatrix} 0.15 & -0.15 \\ -0.15 & 0.15 \end{bmatrix}.$$

It is easy to observe that G_1, G_2 is a 0.2-accurate clustering.

Our mechanism will use an estimate of $\Delta_{G(p), G(q)}$ (instead of $\Delta_{p,q}$) to define the scoring matrix $S_{p,q}$. Thus, the incentive approximation will directly depend on the accuracy of the clustering as well as how good the estimate of $\Delta_{G(p), G(q)}$ is.

There is an inverse relationship between the number of clusters K and the clustering accuracy ε_1: The higher the K, the lower the ε_1. In the extreme, we can let every agent be a separate cluster

($K = \ell$), which results in $\varepsilon_1 = 0$. But a small number of clusters is essential for a reasonable sample complexity, as we need to learn $O(K^2)$ cluster Delta matrices. For instance, in Example 2.4, we need to learn three Delta matrices with clustering, as opposed to six without clustering. In Section 4, we give a learning algorithm that can learn all the pairwise cluster Delta matrices with $\tilde{O}(K)$ samples given a clustering of the agents. In Section 5, we show using real-world data that a reasonably small clustering error can be achieved with relatively few clusters.

3 CORRELATED AGREEMENT FOR HETEROGENEOUS AGENTS

In this section, we define our Correlated Agreement for Heterogeneous Agents (CAHU) mechanism, presented as Algorithm 1. Our mechanism builds upon the multi-task Correlated Agreement (CA) mechanism of Shnayder et al. [2016a], which uses the correlation between signals of different agents to design a scoring matrix to score the agents. However, since we work in a heterogeneous setting, we will need to design different scoring matrices for different pairs of agents based on the different correlations between different pairs.

For intuition, consider the case when one has knowledge of the Delta matrices for all pairs of agents. In this case, in the multi-task peer prediction framework defined in Section 2.1, the scoring matrices $S_{p,q}$ can be defined such that $S_{p,q}(i,j) = 1$ when $\Delta_{p,q} > 0$, and $S_{p,q}(i,j) = 0$ otherwise. Such a mechanism will be 0-informed truthful, as we prove in Lemma 3.4.

However, to design a detail-free mechanism with low sample complexity, we will assume that we have a clustering of agents such that the average cluster Delta matrices can be used as a proxy for agent Delta matrices. Hence, our mechanism works with a clustering of agents and uses the cluster Delta matrices to design scoring matrices for pairs of agents. Here, we will describe our mechanism when a clustering as well as estimates of cluster Delta matrices are given as inputs to the mechanism. In Section 4, we will see how one can learn such a clustering and estimates of Delta matrices from agents' reports.

Specifically, CAHU takes as input a clustering $G_1, \ldots, G_K$ of agents. It also takes as input matrices $\{\overline{\Delta}_{G_s,G_t}\}_{s,t\in[K]}$, which are estimates of the cluster Delta matrices $\{\Delta_{G_s,G_t}\}_{s,t\in[K]}$ defined in Section 2.5. The scoring matrix $S_{p,q}$ for agent pair p and q is then defined such that $S_{p,q}(i,j) = 1$ when $\Delta_{G(p),G(q)} > 0$, and $S_{p,q}(i,j) = 0$ otherwise, where $G(p)$ and $G(q)$ denote the clusters that p and q belong to, respectively. The CAHU mechanism then calculates the reward of an agent according to the framework of multi-task peer prediction discussed in Section 2.1. This would mean that an agent p gets a positive score whenever her report and her peer q's report on a bonus task is such that there is positive correlation between the corresponding signals of clusters $G(p)$ and $G(q)$. However, we also include a penalty when this happens on different tasks. The idea is that if the clustering is ε_1-accurate and the estimates of cluster Delta matrices are accurate, then the mechanism should retain its truthfulness properties. With this in mind, we define an $(\varepsilon_1, \varepsilon_2)$-accurate input to the algorithm as follows:

Definition 3.1. We say that a clustering $\{G_s\}_{s\in[K]}$ and the estimates $\{\overline{\Delta}_{G_s,G_t}\}_{s,t\in[K]}$ are $(\varepsilon_1, \varepsilon_2)$-accurate if

- $\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \varepsilon_1$ for all agents $p, q \in P$, i.e., the clustering is ε_1-accurate, and
- $\|\Delta_{G_s,G_t} - \overline{\Delta}_{G_s,G_t}\|_1 \leqslant \varepsilon_2$ for all clusters $s, t \in [K]$, i.e., the cluster Delta matrix estimates are ε_2-accurate.

An ε_1 clustering intuitively means that if we pick one agent from cluster G_s and another agent from cluster G_t, then their signal correlation is determined by the pair of clusters up to an error ε_1 and is independent of the identities of the agents. However, ε_2-accurate clustering simplly means that we can estimate the cluster delta matrices up to an error ε_2. When we have a clustering and estimates of the delta matrices that are $(\varepsilon_1, \varepsilon_2)$-accurate, we prove that the CAHU mechanism is $(\varepsilon_1 + \varepsilon_2)$-informed truthful. In Section 4, we present algorithms that can learn an ε_1-accurate clustering and ε_2-accurate estimates of cluster Delta matrices.

Throughout the rest of this section, we will use ε_1 to denote the clustering error and ε_2 to denote the learning error. We remark that the clustering error ε_1 is determined by the level of similarity present in agent signal-report behavior, as well as the number of clusters K used, whereas the learning error ε_2 depends on how many samples the learning algorithm sees.

ALGORITHM 1: Mechanism CAHU

```
Require:
    A clustering $G_1, \ldots, G_K$ such that $\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant \epsilon_1$ for all $p, q \in P$;
    estimates $\{\overline{\Delta}_{G_s,G_t}\}_{s,t \in [K]}$ such that $\|\overline{\Delta}_{G_s,G_t} - \Delta_{G_s,G_t}\|_1 \leqslant \epsilon_2$ for all $s, t \in [K]$; and
    for each agent $p \in P$, her bonus tasks $M_1^p$, penalty tasks $M_2^p$, and responses $\{r_b^p\}_{b \in M_1^p \cup M_2^p}$.
Ensure:
for every agent $p \in P$ do
    for every task $b \in M_1^p$ do                                          ▷ Reward response $r_b^p$
        $q \leftarrow$ uniformly at random conditioned on $b \in M_1^q \cup M_2^q$ and (either $|M_2^q| \geqslant 2, |M_2^p| \geqslant 2$
          or $M_2^q \neq M_2^p$)                                               ▷ Peer agent
        Pick tasks $b' \in M_2^p$ and $b'' \in M_2^q$ randomly such that $b' \neq b''$    ▷ Penalty tasks
        $S_{p,q} \leftarrow \text{Sign}(\overline{\Delta}_{G(p),G(q)})^\dagger$
        Reward to agent $p$ for task $b$ is $S_{p,q}\left(r_b^p, r_b^q\right) - S_{p,q}\left(r_{b'}^p, r_{b''}^q\right)$
    end for
end for
```

[†] Sign(x) = 1 if $x > 0$, and 0 otherwise.

3.1 Analysis of CAHU

In this section, we will prove the incentive properties of the CAHU mechanism. We will first present an overview of the proof before presenting it formally. Recall that the expected payment of an agent in this setting is the following:

$$u_p(F^p, \{F^q\}_{q \neq p}) = \frac{1}{\ell - 1} \sum_{q \neq p} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q) .$$

One can think of the expected payment to an agent p to be the average over all other agents q, the expected payment when q is p's peer agents. The expected payment when q is p's peer agent is given by the quantity $\sum_{i,j} \Delta_{p,q}(i,j) \cdot \sum_{r_p,r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q)$.

For intuition, let us only consider deterministic strategies in this discussion. Our proof covers general randomized strategies. For deterministic strategies, we have that

$$\sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} S_{p,q}(r_p, r_q) = S_{p,q}\left(F^p_i, F^q_j\right),$$

where F^p_i and F^q_j denote (deterministic) reports of agents p and q given signals i and j, respectively. In this case, the expected payment for p when q is her peer is $\sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F^p_i, F^q_j)$. Suppose that $\Delta_{p,q}$ has positive diagonals, and negative non-diagonals, and the scoring matrix $S_{p,q}$ is the identity matrix; then, it is not hard to see that the maximum value of $\sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(F^p_i, F^q_j)$ for any deterministic F^p and F^q is the trace of the matrix $\Delta_{p,q}$. Moreover, this maximum is achieved when F^p and F^q are truthful. Also, suppose that agents p and q adopt an uniformed strategy—say, reporting "1" for every task—then the expected payment is $\sum_{i,j} \Delta_{p,q}(i,j) \cdot S_{p,q}(1,1)$, which is zero, since the sum of the entries of the Delta matrices is always zero. For the general case, we will show that the maximum expected payment to p when agent q is her peer is given by $\sum_{i,j} \Delta_{p,q}(i,j) \cdot \text{Sign}(\Delta_{p,q}(i,j))$. Hence, when $S_{p,q} = \text{Sign}(\Delta_{p,q}(i,j))$, then this maximum is achieved when the agents are truthful. Also, the payment of any uninformed strategy is 0. Since this holds for any peer agent q, this would imply informed truthfulness of the mechanism where $S_{p,q} = \text{Sign}(\Delta_{p,q}(i,j))$. A similar argument also follows for any mixed strategies. A formal proof is presented in Lemma 3.4 and is very similar to the proof of informed truthfulness of the CA mechanism [Shnayder et al. 2016a].

However, we use approximate cluster Delta matrices instead of agent Delta matrices to design the scoring matrices. Hence, we need to additionally worry about the effect of approximations due to clustering and learning on the incentive properties of our mechanisms. We will show that even under these approximations a truthful strategy will attain an expected reward that is close to the maximum possible expected reward. Precisely, we will show that when the clustering is ε_1-accurate and the cluster Delta matrix estimates are ε_2-accurate, then the expected reward of a truthful strategy is at most $(\varepsilon_1 + \varepsilon_2)$ away from the maximum reward under any strategy and scoring matrices. Also, the expected reward of any uninformed strategy will always be zero. This will imply that CAHU is $(\varepsilon_1 + \varepsilon_2)$-informed truthful.

We will first need the following technical lemmas before proceeding to the main proof.

Lemma 3.2. *For any matrix $\widehat{S} \in \{0,1\}^{n\times n}$, and any probability distributions $\psi \in \mathcal{P}_n$ and $\phi \in \mathcal{P}_n$, where $\mathcal{P}_n$ is the set of all probability distributions over $[n]$, we have that*

$$0 \leqslant \sum_{r_1, r_2 \in [n]} \psi_{r_1} \widehat{S}(r_1, r_2) \phi_{r_2} \leqslant 1.$$

Proof. The fact that $\sum_{r_1, r_2 \in [n]} \psi_{r_1} \widehat{S}(r_1, r_2) \phi_{r_2} \geqslant 0$ follows easily from the fact that $\psi_{r_1} \geqslant 0$, $\phi_{r_2} \geqslant 0$, and $\widehat{S}(r_1, r_2) \geqslant 0$ for all r_1 and r_2. The other direction follows from the following:

$$\begin{aligned} \sum_{r_1, r_2 \in [n]} \psi_{r_1} \widehat{S}(r_1, r_2) \phi_{r_2} &= \sum_{r_1 \in [n]} \psi_{r_1} \sum_{r_2 \in [n]} \widehat{S}(r_1, r_2) \phi_{r_2} \\ &\leqslant \sum_{r_1 \in [n]} \psi_{r_1} \sum_{r_2 \in [n]} 1 \cdot \phi_{r_2} && (\widehat{S}(r_1, r_2) \leqslant 1) \\ &= \sum_{r_1 \in [n]} \psi_{r_1} \cdot 1 && (\textstyle\sum_{r_2 \in [n]} \phi_{r_2} = 1) \\ &= 1 && (\textstyle\sum_{r_1 \in [n]} \psi_{r_1} = 1) \end{aligned}$$

□

We now prove another technical lemma that gives an upper bound on the maximum payoff to an agent p under any scoring matrix.

Lemma 3.3. *Let $\{\widehat{S}_{p,q}\}_{p,q \in P}$ be an arbitrary set of scoring matrices where $\widehat{S}_{p,q} \in \{0,1\}^{n\times n}$ denotes the score matrix for agent p and agent q. Then for every strategy profile $\{F^q\}_{q \in P}$, we have that*

$$\sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{ir_p} F^q_{jr_q} \widehat{S}_{p,q}(r_p, r_q) \leqslant \sum_{i,j:\Delta_{p,q}(i,j)>0} \Delta_{p,q}(i,j)\,.$$

Proof. We have that

$$\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q) = \sum_{(i,j):\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q) + \sum_{(i,j):\Delta_{p,q}(i,j)\leqslant 0}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q). \tag{4}$$

Now, we make two observations. First,

$$\sum_{i,j:\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j) \geqslant \sum_{(i,j):\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q),$$

which follows from Lemma 3.2 as $\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q) \leqslant 1$. Second,

$$\sum_{(i,j):\Delta_{p,q}(i,j)\leqslant 0}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q) \leqslant 0,$$

which again follows from Lemma 3.2 as $\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}\widehat{S}_{p,q}(r_p,r_q) \geqslant 0$.

Now, the desired bound follows from Equation (4) and the two observations above. □

We will now analyze our mechanism formally using the above lemmas. The derivation of the following result closely follows a similar analysis due to Shnayder et al. [2016a]. We use $u^*_p(\cdot)$ to denote the utility of agent p when the scoring matrices are Sign($\Delta_{p,q}(i,j)$), for all pairs p,q.

Lemma 3.4. *For a strategy profile $\{F^q\}_{q\in P}$ and an agent $p \in P$, define*

$$u^*_p(F^p,\{F^q\}_{q\neq p}) = \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}S^*_{p,q}(r_p,r_q),$$

where $S^*_{p,q}(i,j) = Sign(\Delta_{p,q}(i,j))$ for all $i,j \in [n]$. Then, $u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p}) \geqslant u^*_p(F^p,\{F^q\}_{q\neq p})$. Moreover, for any uninformed strategy F^p, $u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p}) > u^*_p(r,\{F^q\}_{q\neq p})$. This implies informed-truthfulness of the mechanism where $S^*_{p,q}$ is used for scoring agents p and q.

Proof. Let $\mathbf{1}[\cdot]$ denote the indicator function. Then the utility of the truthful strategy profile $\{\mathbb{I},\{\mathbb{I}\}_{q\neq p}\}$ is given by

$$\begin{aligned} u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\in P\setminus\{p\}}) &= \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}\mathbf{1}[i=r_p]\cdot\mathbf{1}[j=r_q]\cdot S^*_{p,q}(r_p,r_q) \\ &= \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}\Delta_{p,q}(i,j)\cdot S^*_{p,q}(i,j) \\ &= \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j:\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j). \end{aligned}$$

The utility of any other strategy profile $\{F^p,\{F^q\}_{q\neq p}\}$ is given by

$$u^*_p(F^p,\{F^q\}_{q\in P\setminus\{p\}}) = \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}S^*_{p,q}(r_p,r_q).$$

From Lemma 3.3, we then have

$$u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\in P\setminus\{p\}}) \geqslant u^*_p(F^p,\{F^q\}_{q\in P\setminus\{p\}}).$$

For an uninformed strategy F^p such that all the rows of F^p are the same, i.e., $F^p_{i\cdot} = \psi$ for all i where ψ is a probability distribution, we have

$$\begin{aligned} u^*_p(F^p, \{F^q\}_{q\neq p}) &= \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q} F^p_{ir_p}F^q_{jr_q}S^*_{p,q}(r_p,r_q) \\ &= \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q} \psi_{r_p}F^q_{jr_q}S^*_{p,q}(r_p,r_q) \\ &= \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{j}\sum_{r_p,r_q}\psi_{r_p}F^q_{jr_q}S^*_{p,q}(r_p,r_q)\left(\sum_i \Delta_{p,q}(i,j)\right) = 0. \end{aligned}$$

The last equality follows, since the row/column sum of delta matrices is zero. However, $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})$, being a sum of only positive entries, is strictly greater than 0. □

We now prove our main theorem that $(\varepsilon_1 + \varepsilon_2)$-informed truthfulness holds when $(\varepsilon_1, \varepsilon_2)$-accurate clustering and learning holds.

Theorem 3.5. *With $(\varepsilon_1, \varepsilon_2)$-accurate clustering and learning, mechanism CAHU is $(\varepsilon_1 + \varepsilon_2)$-informed truthful if* $\min_p u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > \varepsilon_1 + \varepsilon_2$. *In particular,*

(1) For every profile $\{F^q\}_{q\in P}$ and agent $p \in P$, we have $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$.

(2) For any uninformed strategy F^p_0, $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) > u_p(F^p_0, \{F^q\}_{q\neq p})$.

Proof. Fix a strategy profile $\{F^q\}_{q\in P}$. We first show that $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p})$ and then show that $|u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| \leqslant \varepsilon_1 + \varepsilon_2$. These together imply that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) \geqslant u_p(F^p, \{F^q\}_{q\neq p}) - \varepsilon_1 - \varepsilon_2$. For the former, we first observe (similarly, as in proof of Lemma 3.4) that the utility of truthful reporting when the scoring matrix $S^*_{p,q}(i,j) = \mathrm{Sign}(\Delta_{p,q}(i,j))$, is given by

$$u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\in P\setminus\{p\}}) = \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\ \sum_{i,j:\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j).$$

The utility $u_p(F^p, \{F^q\}_{q\in P\setminus\{p\}})$ of an agent p for any strategy profile $\{F^p, \{F^q\}_{q\in P\setminus\{p\}}\}$ under our mechanism, when the scoring matrix $S_{p,q} = \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})$, is given by

$$u_p(F^p, \{F^q\}_{q\in P\setminus\{p\}}) = \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q).$$

Now, using Lemma 3.3 and the expressions for $u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\in P\setminus\{p\}})$ and $u_p(F^p, \{F^q\}_{q\in P\setminus\{p\}})$, we have that

$$u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\in P\setminus\{p\}}) \geqslant u_p(F^p, \{F^q\}_{q\in P\setminus\{p\}}).$$

For the latter, we have

$$\begin{aligned} |u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| &= \left|\frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}\Delta_{p,q}(i,j)\left(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\right)\right| \\ &\leqslant \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}|\Delta_{p,q}(i,j)\left(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j}\right)| \end{aligned} \tag{5}$$

$$
\begin{aligned}
&\leqslant \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{i,j}|\Delta_{p,q}(i,j)-\overline{\Delta}_{G(p),G(q)}(i,j)| \\
&= \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\|\Delta_{p,q}-\overline{\Delta}_{G(p),G(q)}\|_1 \\
&\leqslant \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\|\Delta_{p,q}-\Delta_{G(p),G(q)}\|_1+\|\Delta_{G(p),G(q)}-\overline{\Delta}_{G(p),G(q)}\|_1 \\
&\leqslant \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\varepsilon_1+\varepsilon_2=\varepsilon_1+\varepsilon_2.
\end{aligned}
$$

To show that the third transition holds, we show that $|a\cdot(\text{Sign}(a)-\text{Sign}(b))|\leqslant|a-b|$ for all real numbers $a,b\in\mathbb{R}$. When $\text{Sign}(a)=\text{Sign}(b)$, this holds trivially. When $\text{Sign}(a)\neq\text{Sign}(b)$, note that the RHS becomes $|a|+|b|$, which is an upper bound on the LHS, which becomes $|a|$. The penultimate transition holds by ε_1-accurate clustering and ε_2-accurate estimates of cluster Delta matrices. This proves the first part of the theorem.

Now, we prove the second part of the theorem. For an uninformed strategy F^p such that all the rows of F^p are the same, i.e., $F^p_i=\psi$ for all i where ψ is a probability distribution, we have

$$
\begin{aligned}
u_p(F^p,\{F^q\}_{q\neq p}) &= \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}F^p_{ir_p}F^q_{jr_q}S_{p,q}(r_p,r_q) \\
&= \frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j}\Delta_{p,q}(i,j)\sum_{r_p,r_q}\psi_{r_p}F^q_{jr_q}S_{p,q}(r_p,r_q) \\
&= \frac{1}{\ell-1}\sum_{q\in P\setminus\{p\}}\sum_{j}\sum_{r_p,r_q}\psi_{r_p}F^q_{jr_q}S_{p,q}(r_p,r_q)\left(\sum_i\Delta_{p,q}(i,j)\right)=0,
\end{aligned}
$$

where the last equality follows, because the rows and columns of $\Delta_{p,q}$ sum to zero. Since $|u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})-u_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})|\leqslant\varepsilon_1+\varepsilon_2$, we have

$$u_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})\geqslant u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})-\varepsilon_1-\varepsilon_2>0,$$

as $u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})>\varepsilon_1+\varepsilon_2$ for any p. □

The CAHU mechanism always ensures that there is no strategy profile that gives an expected utility more than $\varepsilon_1+\varepsilon_2$ above truthful reporting. The condition $\min_p u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})>\varepsilon_1+\varepsilon_2$ is required to ensure that any uninformed strategy gives strictly less than the truth-telling equilibrium. This is important to promote effort in collecting and reporting an informative signal. Note that the learning error ε_2 can be made if we have sufficient amount of data. Therefore, we need to guarantee that $\min_p u^*_p(\mathbb{I},\{\mathbb{I}\}_{q\neq p})>\varepsilon_1$ to ensure that any uninformed strategy gives strictly less than the truth-telling. Writing it out, this condition requires that for each agent p the following holds:

$$\frac{1}{\ell-1}\sum_{q\neq p}\sum_{i,j:\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j)>\varepsilon_1. \tag{6}$$

In particular, a sufficient condition for this property is that for every pair of agents, the expected reward on a bonus task in the CA mechanism when making truthful reports is at least ε_1, i.e., for every pair of agents p and q,

$$\sum_{i,j:\Delta_{p,q}(i,j)>0}\Delta_{p,q}(i,j)>\varepsilon_1. \tag{7}$$

In turn, as pointed out by Shnayder et al. [2016a], the LHS in Equation (7) quantity can be interpreted as a measure of how much positive correlation there is in the joint distribution on signals between a pair of agents. Note that it is not important that this is same-signal correlation. For example, this quantity would be large between an accurate and an always-wrong agent in a binary-signal domain, since the positive correlation would be between one agent's report and the flipped report from the other agent.

The incentive properties of the mechanism are retained when used together with learning the cluster structure and cluster Delta matrices. However, we do assume that the agents do not reveal their task assignments to each other. If the agents were aware of the identities of the tasks they are assigned, then they could coordinate on the task identifiers to arrive at a profitable coordinated strategy. This is reasonable in practical settings, as the number of tasks is often large. The next theorem shows that even if the agents could set the scoring matrices to be an arbitrary function $\widehat{S}$ through any possible deviating strategies, it is still beneficial to use the scoring matrices estimated from the truthful strategies. Let $\widehat{S}$ be an arbitrary scoring function, i.e., $\widehat{S}_{p,q}$ specifies the score matrix for two agents from p and q. We will write $\hat{u}_p(F^p, \{F^q\}_{q \neq p})$ to denote the expected utility of agent p under the CAHU mechanism with the reward function $\widehat{S}$ and strategy profile $(F^p, \{F^q\}_{q \neq p})$.

Theorem 3.6. *Let $\{\widehat{S}_{p,q}\}_{p,q \in P}$ be an arbitrary set of scoring matrices where $\widehat{S}_{p,q} \in \{0,1\}^{n \times n}$ denotes the score matrix for agent p and agent q. Then for every profile $\{F^q\}_{q \in P}$ and agent $p \in P$, we have*

(1) $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant \hat{u}_p(F^p, \{F^q\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$.

(2) If $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \varepsilon_1$, *then for any uninformed strategy* F_0^p, $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \hat{u}_p(F_0^p, \{F^q\}_{q \neq p})$.

Proof. Similar to the proof of Lemma 3.4, the utility of truthful reporting when the scoring matrix $S^*_{p,q}(i,j) = \mathrm{Sign}(\Delta_{p,q}(i,j))$, is given by

$$u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j : \Delta_{p,q}(i,j) > 0} \Delta_{p,q}(i,j).$$

The utility $\hat{u}_p(F^p, \{F^q\}_{q \in P \setminus \{p\}})$ of an agent p for any strategy profile $\{F^p, \{F^q\}_{q \in P \setminus \{p\}}\}$ when the scoring matrix is $\widehat{S}_{p,q}$, is given by

$$\hat{u}_p(F^p, \{F^q\}_{q \in P \setminus \{p\}}) = \frac{1}{\ell - 1} \sum_{q \in P \setminus \{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \sum_{r_p, r_q} F^p_{i r_p} F^q_{j r_q} \widehat{S}_{p,q}(r_p, r_q).$$

Now, using Lemma 3.3 and the expressions for $u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \in P \setminus \{p\}})$ and $\hat{u}_p(F^p, \{F^q\}_{q \in P \setminus \{p\}})$, we have that

$$u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant \hat{u}_p(F^p, \{F^q\}_{q \neq p}).$$

Now the proof of Theorem 3.5 shows that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. Using the result above, we get $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) \geqslant \hat{u}_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) - \varepsilon_1 - \varepsilon_2$. Similar to the proof of Theorem 3.5, it can be shown that $\hat{u}_p(F_0^p, \{F^q\}_{q \neq p}) = 0$ for any uninformed strategy F_0^p. The proof of Theorem 3.5 also shows that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})$ can be made positive whenever $\min_p u_p^*(\mathbb{I}, \{\mathbb{I}\}_{q \neq p}) > \varepsilon_1$. □

The above theorem implies that the incentive properties of our mechanism hold even when agents are allowed to coordinate their strategies and the mechanism is learned using reports from these coordinated strategies. To be precise, recall that $u_p(\mathbb{I}, \{\mathbb{I}\}_{q \neq p})$ is the expected payment to agent p when the mechanism learns the true Delta matrix and the agent reports truthfully. This

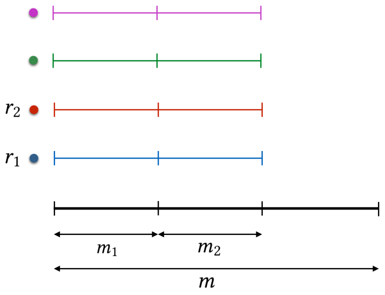


Fig. 1. Fixed task assignment.

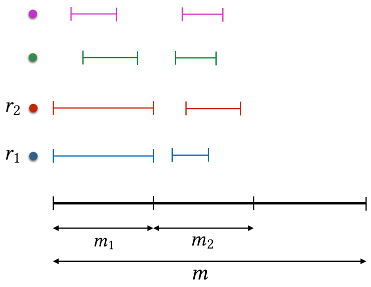


Fig. 2. Uniform task assignment.

is no less than the expected payment minus $\varepsilon_1 + \varepsilon_2$ when the mechanism learns any other delta matrices and the agents misreport in any arbitrary way.

4 LEARNING THE AGENT SIGNAL TYPES

In this section, we provide algorithms for learning a clustering of agent signal types from reports, and further, for learning the cluster pairwise Δ matrices. The estimates of the Δ matrices can then be used to give an approximate-informed truthful mechanism. Along the way, we couple our methods with the latent "confusion matrix" methods of Dawid and Skene [1979].

Recall that m is the total number of tasks about which reports are collected. Reports on m_1 of these tasks will also be used for clustering, and reports on a further m_2 of these tasks will be used for learning the cluster pairwise Δ matrices. We consider two different schemes for assigning agents to tasks for the purpose of clustering and learning (see Figures 1 and 2):

(1) **Fixed Task Assignment:** Each agent is assigned to the same, random subset of tasks of size $m_1 + m_2$ of the given m tasks.
(2) **Uniform Task Assignment:** For clustering, we select two agents r_1 and r_2, uniformly at random, to be *reference agents*. These agents are assigned to a subset of tasks of size $m_1(<m)$. For all other agents, we then assign a required number of tasks, s_1, uniformly at random from the set of m_1 tasks. For learning the cluster pairwise Δ-matrices, we also assign one agent from each cluster to some subset of tasks of size s_2, selected uniformly at random from a second set of $m_2(<m - m_1)$ tasks.

For each assignment scheme, the analysis establishes that there are enough agents who have done a sufficient number of joint tasks. Table 1 summarizes the sample complexity results, stating them under two different assumptions about the way in which signals are generated.

4.1 Clustering

We proceed by presenting and analyzing a simple clustering algorithm.

Definition 4.1. A clustering $G_1, \ldots, G_K$ is ε-good if for some $\gamma > 0$

$$G(q) = G(r) \Rightarrow \|\Delta_{pq} - \Delta_{pr}\|_1 \leqslant \varepsilon - 4\gamma \ \forall p \in [\ell] \setminus \{q, r\}, \tag{8}$$

$$G(q) \neq G(r) \Rightarrow \|\Delta_{pq} - \Delta_{pr}\|_1 > \varepsilon \ \forall p \in [\ell] \setminus \{q, r\}. \tag{9}$$

We first show that an ε-good clustering, if exists, must be unique.

[†] For an arbitrary m_2, this bound is Km_2 as long as m_2 is $\Omega(n^7/(\varepsilon')^2)$.
[‡] In the no assumption approach (respectively, Dawid-Skene Model), ε' is the error in the estimation of the joint probability distribution (respectively, aggregate confusion matrix).

Table 1. Sample Complexity for the CAHU Mechanism

	No Assumption	Dawid-Skene
Fixed Assignment	Clustering: $\tilde{O}\left(\frac{\ell n^2}{\gamma^2}\right)$ Learning: $\tilde{O}\left(\frac{Kn^2}{(\varepsilon')^2}\right)$	Clustering: $\tilde{O}\left(\frac{\ell n^2}{\gamma^2}\right)$ Learning: $\tilde{O}\left(\frac{\ell n^7}{(\varepsilon')^2}\right)$
Uniform Assignment	Clustering: $\tilde{O}\left(\frac{\ell n^2}{\gamma^2} + m_1\right)$ Learning: $\tilde{O}\left(K m_2^{7/8} \sqrt{\frac{n^2}{(\varepsilon')^2}}\right)$	Clustering: $\tilde{O}\left(\frac{\ell n^2}{\gamma^2} + m_1\right)$ Learning: $\tilde{O}\left(\frac{Kn^7}{(\varepsilon')^2}\right)$[†]

The rows indicate the assignment scheme and the columns indicate the modeling assumption. Here, ℓ is the number of agents, n is the number of signals, ε' is a parameter that controls learning accuracy[‡], γ is a clustering parameter, K is the number of clusters, and m_1 (resp. m_2) is the size of the set of tasks from which the tasks used for clustering (respectively, learning) are sampled.

Theorem 4.2. *Suppose there exist two clustering $\{G_j\}_{j \in [K]}$ and $\{T_i\}_{i \in [K']}$ that are ε-good. Then $K' = K$ and $G_j = T_{\pi(j)}$ for some permutation π over $[K]$.*

Proof. Suppose Equations (8) and (9) hold with parameters γ_1 and γ_2, respectively, for the clusterings $\{G_j\}_{j \in [K]}$ and $\{T_i\}_{i \in [K']}$. If possible, assume there exist T_i and G_j such that $T_i \setminus G_j \neq \emptyset$, $G_j \setminus T_i \neq \emptyset$, and $T_i \cap G_j \neq \emptyset$. Pick $s \in T_i \cap G_j$ and $r \in G_j \setminus T_i$. Then, we must have, for any $p \notin \{q, s, r\}$,

(1) $\|\Delta_{pr} - \Delta_{ps}\|_1 > \varepsilon$ (inter-cluster distance in $\{T_i\}_{i \in [K']}$).
(2) $\|\Delta_{pr} - \Delta_{ps}\|_1 \leqslant \varepsilon - 4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j \in [K]}$).

This is a contradiction. Now suppose $K' > K$. Then there must exist T_i and T_k such that $T_i \cup T_k \subseteq G_j$ for some j. Pick $q \in T_i$ and $r \in T_k$. Then, for any $p \notin \{q, r\}$,

(1) $\|\Delta_{pq} - \Delta_{pr}\|_1 > \varepsilon$ (inter-cluster distance in $\{T_i\}_{i \in [K']}$).
(2) $\|\Delta_{pq} - \Delta_{pr}\|_1 \leqslant \varepsilon - 4\gamma_1$ (intra-cluster distance in $\{G_j\}_{j \in [K]}$).

This leads to a contradiction and proves that $K' \leqslant K$. Similarly, we can prove $K \leqslant K'$. Therefore, we have shown that for each each G_j there exists i such that $G_j = T_i$. □

Since there is a unique ε-good clustering (up to a permutation), we will refer to this clustering as the *correct clustering*. The assumption that there exists an ε-good clustering is stronger than Equation (3) introduced earlier. In particular, identifying the correct clustering needs to satisfy Equation (9), i.e., the Δ-matrices of two agents belonging to two different clusters are different with respect to every other agent. So, we need low inter-cluster similarities in addition to high intra-cluster similarities. The pseudo-code for the clustering algorithm is presented in Algorithm 2. This algorithm iterates over the agents and forms clusters in a greedy manner. In particular, as Figure 3 shows, it checks whether i and q_t are in the same cluster by estimating Δ_{p_t,q_t} and $\Delta_{p_t,i}$. First, we prove that as long as we can find an agent p_t that has $\Omega(\frac{n^2 \log(\ell/\delta)}{\gamma^2})$ tasks in common with both q_t and i, then the clustering produced by Algorithm 2 is correct with probability at least $1 - \delta$.

Theorem 4.3. *If for all $i \in P$ and $q_t \in G(i)$ there exists p_t, which has $\Omega(\frac{n^2 \log(\ell/\delta)}{\gamma^2})$ tasks in common with both q_t and i, then Algorithm 2 recovers the correct clustering, i.e., $\hat{G}_t = G_t$ for $t = 1, \ldots, K$ with probability at least $1 - \delta$.*

We need two key technical lemmas to prove Theorem 4.3. The first lemma shows that to estimate $\Delta_{p,q}$ with an L1 distance of at most γ, it is sufficient to estimate the joint probability distribution $D_{p,q}$ with an L1 distance of at most $\gamma/3$. With this, we can estimate the delta matrices of agent pairs from the joint empirical distributions of their reports.

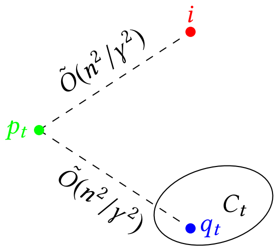


Fig. 3. Algorithm 2 checks whether i and q_t are in the same cluster by estimating Δ_{p_t,q_t} and $\Delta_{p_t,i}$.

ALGORITHM 2: Clustering

```
Require: $\epsilon, \gamma$ such that there exists an $\epsilon$-good clustering with parameter $\gamma$.
Ensure: A clustering $\{\hat{G}_t\}_{t=1}^{\hat{K}}$
$\hat{G} \leftarrow \emptyset$, $\hat{K} \leftarrow 0$                ▷ $\hat{G}$ is the list of clusters, $\hat{K} = |\hat{G}|$
Make a new cluster $\hat{G}_1$ and add agent 1
Add $\hat{G}_1$ to $\hat{G}$, $\hat{K} \leftarrow \hat{K} + 1$
for $i = 2, \ldots, \ell$ do
    for $t \in [\hat{K}]$ do
        Pick an arbitrary agent $q_t \in \hat{G}_t$
        Pick $p_t \in [l] \setminus \{i, q_t\}$ (Fixed) or $p_t \in \{r_1, r_2\} \setminus \{i, q_t\}$(Uniform), such that $p_t$ has at least $\Omega(\frac{n^2 \log(K\ell/\delta)}{\gamma^2})$ tasks in common with both $q_t$ and $i$
        Let $\bar{\Delta}_{p_t,q_t}$ be the empirical Delta matrix from reports of agents $p_t$ and $q_t$
        Let $\bar{\Delta}_{p_t,i}$ be the empirical Delta matrix from reports of agents $p_t$ and $i$
    end for
    if $\exists t \in [\hat{K}] : \|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leq \epsilon - 2\gamma$ then
        add $i$ to $\hat{G}_t$ (with ties broken arbitrarily for $t$)
    else
        Make a new cluster $\hat{G}_{\hat{K}+1}$ and add agent $i$ to it
        Add $\hat{G}_{\hat{K}+1}$ to $\hat{G}$, $\hat{K} \leftarrow \hat{K} + 1$
    end if
end for
```

Lemma 4.4. *For all* $p, q \in P$, $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \gamma/3 \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leqslant \gamma$.

Proof.

$$\begin{aligned}
\|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 &= \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - \bar{D}_p(i)\bar{D}_q(j) - \left(D_{p,q}(i,j) - D_p(i)D_q(j)\right)\right| \\
&= \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| + \sum_{i,j} \left|\bar{D}_p(i)\bar{D}_q(j) - \bar{D}_p(i)D_q(j) + \bar{D}_p(i)D_q(j) - D_p(i)D_q(j)\right| \\
&\leqslant \gamma/3 + \sum_i \bar{D}_p(i) \sum_j \left|\bar{D}_q(j) - D_q(j)\right| + \sum_j D_q(j) \sum_i \left|\bar{D}_p(i) - D_p(i)\right| \\
&\leqslant \gamma/3 + \sum_j \left|\bar{D}_q(j) - D_q(j)\right| + \sum_i \left|\bar{D}_p(i) - D_p(i)\right| \\
&\leqslant \gamma/3 + \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| + \sum_{i,j} \left|\bar{D}_{p,q}(i,j) - D_{p,q}(i,j)\right| \\
&\leqslant \gamma,
\end{aligned}$$

as required. □

The second lemma is about learning the empirical distributions of reports of pairs of agents. This can be proved using Theorems 3.1 and 2.2 from the work of Devroye and Lugosi [2012].

Lemma 4.5. *Any distribution over a finite domain Ω is learnable within a L1 distance of d with probability at least $1 - \delta$ by observing $O(\frac{|\Omega|}{d^2}\log(1/\delta))$ samples from the distribution.*

We can use the above lemma to show that the joint distributions of reports of agents can be learned to within an $L1$ distance γ with probability at least $1 - \delta/K\ell$ by observing $O(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ reports on joint tasks.

Corollary 4.6. *For any agent pair $p, q \in P$, the joint distribution of their reports $D_{p,q}$ is learnable within an L1 distance of γ using $O(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ reports on joint tasks with probability at least $1 - \delta/K\ell$.*

We are now ready to prove Theorem 4.3.

Proof of Theorem 4.3. The proof is by induction on the number of agents ℓ. Suppose all the agents up to and including $i - 1$ have been clustered correctly. Consider the ith agent and suppose i belongs to the cluster G_t. Suppose $\hat{G}_t \neq \emptyset$. Then, using the triangle inequality, we have

$$\|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \|\bar{\Delta}_{p_t,q_t} - \Delta_{p_t,q_t}\|_1 + \|\Delta_{p_t,q_t} - \Delta_{p_t,i}\|_1 + \|\bar{\Delta}_{p_t,i} - \Delta_{p_t,i}\|_1.$$

Since $q_t \in G_t$, we have $\|\Delta_{p_t,q_t} - \Delta_{p_t,i}\|_1 \leqslant \varepsilon/2 - 4\gamma$. Moreover, using Lemma 4.4 and Corollary 4.6, we have that, with probability at least $1 - \delta/K\ell$, $\|\bar{\Delta}_{p_t,q_t} - \Delta_{p_t,q_t}\|_1 \leqslant \gamma$ and $\|\bar{\Delta}_{p_t,i} - \Delta_{p_t,i}\| \leqslant \gamma$. This ensures that $\|\bar{\Delta}_{p_t,q_t} - \bar{\Delta}_{p_t,i}\|_1 \leqslant \varepsilon/2 - 2\gamma$. However, pick any cluster G_s such that $s \neq t$ and $\hat{G}_s \neq \emptyset$. Then

$$\|\bar{\Delta}_{p_s,q_s} - \bar{\Delta}_{p_s,i}\|_1 \geqslant \|\Delta_{p_s,q_s} - \Delta_{p_s,i}\| - \|\bar{\Delta}_{p_s,q_s} - \Delta_{p_s,q_s}\|_1 - \|\bar{\Delta}_{p_s,i} - \Delta_{p_s,i}\|_1.$$

Since $i \notin G_s$, we have $\|\Delta_{p_s,q_s} - \Delta_{p_s,i}\|_1 > \varepsilon/2$. Again, with probability at least $1 - \delta/K\ell$, we have $\|\bar{\Delta}_{p_s,q_s} - \Delta_{p_s,q_s}\|_1 \leqslant \gamma$ and $\|\bar{\Delta}_{p_s,i} - \Delta_{p_s,i}\|_1 \leqslant \gamma$. This ensures that $\|\bar{\Delta}_{p_s,q_s} - \bar{\Delta}_{p_s,i}\|_1 > \varepsilon/2 - 2\gamma$. This ensures that condition on line (11) is violated for all clusters $s \neq t$. If $\hat{G}_t \neq \emptyset$, then this condition is satisfied and agent i is added to cluster $\hat{G}_t$, otherwise the algorithm makes a new cluster with agent i. Now note that the algorithm makes a new cluster only when it sees an agent belonging to a new cluster. This implies that $\hat{K} = K$. Taking a union bound over the K choices of q_s for the K clusters, we see that agent i is assigned to its correct cluster with probability at least $1 - \delta/\ell$. Finally, taking a union bound over all the ℓ agents, we get the desired result. □

Next, we show how the assumption in regard to task overlap is satisfied under each assignment scheme and characterize the sample complexity of learning the clusterings under each scheme. In the fixed assignment scheme, all the agents are assigned to the same set of $m_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks. Thus, for each agent pair q_t and i, any other agent in the population can act as p_t. The total number of tasks performed is $O(\frac{\ell n^2}{\gamma^2}\log(K\ell/\delta))$.

In the uniform assignment scheme, we select two agents r_1 and r_2 uniformly at random to be reference agents and assign these agents to each of $m_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks. For all other agents, we then assign $s_1 = \Omega(\frac{n^2}{\gamma^2}\log(K\ell/\delta))$ tasks uniformly at random from this set of m_1 tasks. If $m_1 = s_1$, then the uniform task assignment is the same as fixed task assignment. However, in applications (e.g., Karger et al. [2011]) where one wants the task assignments to be more uniform across tasks, it will make sense to use a larger value of m_1. The reference agent r_1 can act as p_t for all agent pairs q_t and i other than r_1. Similarly, reference r_2 can act as p_t for all agent pairs q_t and i other than

r_2. If $q_t = r_1$ and $i = r_2$ or $q_t = r_2$ and $i = r_1$, then any other agent can act as p_t. The total number of tasks performed is $\Omega(\frac{\ell n^2}{\gamma^2}\log(K\ell/\delta) + m_1)$, which is sufficient for the high-probability result.

4.2 Learning the Cluster Pairwise Δ Matrices

We proceed now under the assumption that the agents are clustered into K groups, $G_1, \ldots, G_K$. Our goal is to estimate the cluster-pairwise delta matrices Δ_{G_s,G_t} as required by Algorithm 1. We estimate the Δ_{G_s,G_t} under two different settings: when we have no model of the signal distribution and in the Dawid-Skene latent attribute model.

ALGORITHM 3: Learning-Δ-No-Assumption

```
for $t = 1, \ldots, K$ do
    Choose agent $q_t \in G_t$ arbitrarily.
end for
for each pair of clusters $G_s, G_t$ do
    Let $q_s$ and $q_t$ be the chosen agents for $G_s$ and $G_t$, respectively.
    Let $\bar{D}_{q_s,q_t}$ be the empirical estimate of $D_{q_s,q_t}$ such that $\|\bar{D}_{q_s,q_t} - D_{q_s,q_t}\|_1 \le \epsilon'$ with probability at
  least $1 - \delta/K^2$
    Let $\bar{\Delta}_{q_s,q_t}$ be the empirical Delta matrix computed using $\bar{D}_{q_s,q_t}$
    Set $\bar{\Delta}_{G_s,G_t} = \bar{\Delta}_{q_s,q_t}$
end for
```

4.2.1 *Learning the Δ-Matrices with No Assumption.*

We first characterize the sample complexity of learning the Δ-matrices in the absence of any modeling assumptions. To estimate $\bar{\Delta}_{G_s,G_t}$, Algorithm 3 first picks agent q_s from cluster G_s, estimates $\bar{\Delta}_{q_s,q_t}$, and uses this estimate in place of $\bar{\Delta}_{G_s,G_t}$. For the fixed assignment scheme, we assign the agents q_s to the same set of tasks of size $O(\frac{n^2}{(\varepsilon')^2}\log(K/\delta))$. For the uniform assignment scheme, we assign the agents to subsets of tasks of an appropriate size among the pool of m_2 tasks.

Theorem 4.7. *Given an ε-good clustering $\{G_s\}_{s=1}^K$, if the number of shared tasks between any pair of agents q_s, q_t is $O(\frac{n^2}{(\varepsilon')^2}\log(K/\delta))$, then Algorithm 3 guarantees that for all s, t, $\|\bar{\Delta}_{G_s,G_t} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon$ with probability at least $1-\delta$. The total number of samples collected by the algorithm is $O(\frac{Kn^2}{(\varepsilon')^2}\log(K/\delta))$ (respectively, $O(Km_2^{7/8}\sqrt{\frac{n^2}{(\varepsilon')^2}\log(K/\delta)})$ w.h.p.) under the fixed (respectively, uniform) assignment scheme.*

We first prove a sequence of lemmas that will be used to prove the result.

Lemma 4.8. *For every pair of agents p, q, we have*

$$\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 \leqslant 2 \cdot \max_{a,b,c \in P: G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1.$$

Proof. Let $\Delta_{p,G(q)} = \frac{1}{|G(q)|}\sum_{r\in G(q)} \Delta_{p,r}$, then using the property of clusters, we have

$$\begin{aligned}\|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1 &= \left\|\Delta_{p,q} - \frac{1}{|G(p)|\,|G(q)|}\textstyle\sum_{u\in G(p),v\in G(q)} \Delta_{u,v}\right\|_1 \\ &= \left\|\frac{1}{|G(p)|\,|G(q)|}\textstyle\sum_{u\in G(p),v\in G(q)} \left(\Delta_{p,q} - \Delta_{u,v}\right)\right\|_1 \\ &\leqslant \frac{1}{|G(p)|\,|G(q)|}\sum_{u\in G(p),v\in G(q)} \|\Delta_{p,q} - \Delta_{u,v}\|_1\end{aligned}$$

$$\leqslant \frac{1}{|G(p)|\,|G(q)|} \sum_{u \in G(p), v \in G(q)} \|\Delta_{p,q} - \Delta_{u,q}\|_1 + \|\Delta_{u,q} - \Delta_{u,v}\|_1$$
$$\leqslant \frac{1}{|G(p)|\,|G(q)|} \sum_{u \in G(p), v \in G(q)} 2 \max_{a,b,c \in P: G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1$$
$$= 2 \max_{a,b,c \in P: G(a)=G(b)} \|\Delta_{a,c} - \Delta_{b,c}\|_1,$$

as required. □

The next lemma characterizes the error made by Algorithm 3 in estimating the Δ_{G_s,G_t}-matrices.

Lemma 4.9. *For any two agents $p \in G_s$ and $q \in G_t$, $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \varepsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon$.*

Proof. Lemma 4.4 shows that $\|\bar{D}_{p,q} - D_{p,q}\|_1 \leqslant \varepsilon' \Rightarrow \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 \leqslant 3\varepsilon'$.
Now,

$$\|\bar{\Delta}_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant \|\bar{\Delta}_{p,q} - \Delta_{p,q}\|_1 + \|\Delta_{p,q} - \Delta_{G_s,G_t}\|_1 \leqslant 3\varepsilon' + 2\varepsilon.$$

The last inequality uses Lemma 4.8 □

Proof (Theorem 4.7). By Lemma 4.5, to estimate $D_{p,q}$ within a distance of ε' with probability at least $1 - \delta/K^2$, we need $O(\frac{n^2}{(\varepsilon')^2}\log(K^2/\delta))$. By a union bound over the K^2 pairs of clusters, we see that with probability at least $1-\delta$, we have $\|\bar{D}_{q_s,q_t} - D_{q_s,q_t}\|_1 \leqslant \varepsilon'$. This proves the first part of the theorem. When the assignment scheme is fixed, we can assign all the same tasks to K agents $\{q_t\}_{t=1}^K$, and hence the total number of samples is multiplied by K.

However, under the uniform assignment scheme, suppose each agent $\{q_t\}_{t=1}^K$ is assigned to a subset of s_2 tasks selected uniformly at random from the pool of m_2 tasks. Now consider any two agents q_s and q_t. Let X_i be an indicator random variable that is 1 when $i \in [m_2]$ is included in tasks of q_s, and 0 otherwise. Also, let Y_i be a similar random variable for the tasks of q_t. Let $Z_i = X_i \times Y_i$. The probability that both agents are assigned to a particular task i, $\Pr(Z_i = 1) = (s_2/m_2)^2$. Therefore, the expected number of overlapping tasks among the two agents is $m_2 \cdot (\frac{s_2}{m_2})^2 = \frac{s_2^2}{m_2}$, i.e., $\mathrm{E}[\sum_i Z_i] = \frac{s_2^2}{m_2}$. Now, we want to bound the deviations from this expectation. Let $R_j = \mathrm{E}[\sum_{i=1}^{m_2} Z_i | X_1, \ldots, X_j, Y_1, \ldots, Y_j]$, then R_j is a Doob martingale sequence for $\sum_{i=1}^{j} Z_i$. Also, it is easy to see that this martingale sequence is bounded by 1, i.e., $|R_{j+1} - R_j| \leqslant 1$. Therefore, we apply the Azuma-Hoeffding bound (Lemma 4.10) as

$$\Pr\left[\left|\sum_i Z_i\right| > \frac{s_2^2}{2m_2}\right] \leqslant 2\exp\left\{-\frac{s_2^4}{8m_2^3}\right\}.$$

Now substituting $s_2 = m_2^{7/8} \cdot L^{1/2}$ where $L = O\left(\frac{n^2}{(\varepsilon')^2}\log(K^2/\delta)\right)$, we get

$$\Pr\left[\sum_i Z_i < m_2^{3/4} L/2\right] \leqslant 2\exp\left\{-\sqrt{m_2}L^2\right\}.$$

Taking a union bound over K^2 pairs of agents, if each agent completes $m_2^{7/8} \cdot L^{1/2}$ tasks selected uniformly at random from the pool of m_2 tasks, then the probability that any pair of agents has a number of shared tasks L is at least $1 - K^2\exp\{-\sqrt{m_2}L^2\}$, which is exponentially small in m_2. □

Lemma 4.10. *Suppose $X_n, n \geqslant 1$ is a martingale such that $X_0 = 0$ and $|X_i - X_{i-1}| \leqslant 1$ for each $1 \leqslant i \leqslant n$. Then for every $t > 0$*

$$\Pr[|X_n| > t] \leqslant 2\exp\{-t^2/2n\}.$$

4.2.2 Learning the Δ-matrices under the Dawid-Skene Model. In this section, we assume that the agents receive signals according to the Dawid and Skene [1979] model. Here, each task has a latent attribute and each agent has a confusion matrix to parameterize its signal distribution conditioned on this latent value. Recall two notations from the introduction: $D_p(i)$ is the marginal probability of observing signal i for agent p, and $D_{p,q}(i,j)$ is the joint probability that the agents p and q observe signals i and j, respectively. Then the Dawid-Skene Model is formally defined as:

- Let $\{\pi_k\}_{k=1}^n$ denote the prior probability over n latent values.
- Agent p has *confusion matrix* $C^p \in \mathbb{R}^{n\times n}$, such that $C^p_{ij} = D_p(S_p = j|T = i)$ where T is the latent value. Given this, the joint signal distribution for a pair of agents p and q is

$$D_{p,q}(S_p = i, S_q = j) = \sum_{k=1}^{n} \pi_k C^p_{ki} C^q_{kj}, \tag{10}$$

  and the marginal signal distribution for agent p is

$$D_p(S_p = i) = \sum_{k=1}^{n} \pi_k C^p_{ki}. \tag{11}$$

For cluster G_t, we write $C^t = \frac{1}{|G_t|}\sum_{p\in G_t} C^p$ to denote the aggregate confusion matrix of G_t. As before, we assume that we are given an ε-good clustering, $G_1,\ldots,G_K$, of the agents. Our goal is to provide an estimate of the Δ_{G_s,G_t}-matrices.

Lemma 4.11 proves that to estimate Δ_{G_s,G_t} within an L1 distance of ε', it is enough to estimate the aggregate confusion matrices within an L1 distance of $\varepsilon'/4$. So, to learn the pairwise delta matrices between clusters, we first ensure that for each cluster G_t, we have $\|\bar{C}^t - C^t\|_1 \leqslant \varepsilon'/4$ with probability at least $1-\delta/K$ and then use the following formula to compute the delta matrices:

$$\Delta_{G_s,G_t}(i,j) = \sum_{k=1}^{n} \pi_k \bar{C}^s_{ki}\bar{C}^t_{kj} - \sum_{k=1}^{n} \pi_k \bar{C}^s_{ki} \sum_{k=1}^{n} \pi_k \bar{C}^t_{kj}. \tag{12}$$

Lemma 4.11. *For all* G_a, G_b, $\|\bar{C}^a - C^a\|_1 \leqslant \varepsilon'/4$ *and* $\|\bar{C}^b - C^b\|_1 \leqslant \varepsilon'/4 \;\Rightarrow\; \|\bar{\Delta}_{G_a,G_b} - \Delta_{G_a,G_b}\| \leqslant \varepsilon'$.

Proof.

$$\begin{aligned}\Delta_{G_a,G_b}(i,j) &= \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a, q\in G_b} \Delta_{p,q}(i,j) = \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a, q\in G_b} D_{p,q}(i,j) - D_p(i)D_q(j)\\ &= \frac{1}{|G_a|\,|G_b|}\sum_{p\in G_a, q\in G_b} \sum_k \pi_k C^p_{ki}C^q_{kj} - \sum_k \pi_k C^p_{ki}\sum_k C^q_{kj}\\ &= \sum_k \pi_k \left(\frac{1}{|G_a|}\sum_{p\in G_a} C^p_{ki}\right)\left(\frac{1}{|G_b|}\sum_{q\in G_b} C^q_{kj}\right)\end{aligned}$$

$$
\begin{aligned}
&- \sum_k \pi_k \left(\frac{1}{|G_a|} \sum_{p \in G_a} C^p_{ki} \right) \sum_k \pi_k \left(\frac{1}{|G_b|} \sum_{q \in G_b} C^q_{kj} \right) \\
&= \sum_k \pi_k C^a_{ki} C^b_{kj} - \sum_k \pi_k C^a_{ki} \sum_k \pi_k C^b_{kj}.
\end{aligned}
$$

Now

$$
\begin{aligned}
&\|\bar{\Delta}_{G_a,G_b} - \Delta_{G_a,G_b}\|_1 = \sum_{i,j} \left| \bar{\Delta}_{G_a,G_b}(i,j) - \Delta_{G_a,G_b}(i,j) \right| \\
&= \sum_{i,j} \left| \sum_k \pi_k \bar{C}^a_{ki} \bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki} \sum_k \pi_k \bar{C}^b_{kj} - \left(\sum_k \pi_k C^a_{ki} C^b_{kj} - \sum_k \pi_k C^a_{ki} \sum_k \pi_k C^b_{kj} \right) \right| \\
&\leqslant \sum_{i,j} \left| \sum_k \pi_k \bar{C}^a_{ki} \bar{C}^b_{kj} - \sum_k \pi_k C^a_{ki} C^b_{kj} \right| + \sum_{i,j} \left| \sum_k \pi_k \bar{C}^a_{ki} \sum_k \pi_k \bar{C}^b_{kj} - \sum_k \pi_k C^a_{ki} \sum_k \pi_k C^b_{kj} \right| \\
&= \sum_{i,j} \left| \sum_k \pi_k \bar{C}^a_{ki} \bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki} C^b_{kj} + \sum_k \pi_k \bar{C}^a_{ki} C^b_{kj} - \sum_k \pi_k C^a_{ki} C^b_{kj} \right| \\
&\quad + \sum_{i,j} \left| \sum_k \pi_k \bar{C}^a_{ki} \sum_k \pi_k \bar{C}^b_{kj} - \sum_k \pi_k \bar{C}^a_{ki} \sum_k \pi_k C^b_{kj} + \sum_k \pi_k \bar{C}^a_{ki} \sum_k \pi_k C^b_{kj} - \sum_k \pi_k C^a_{ki} \sum_k \pi_k C^b_{kj} \right| \\
&= \sum_k \pi_k \sum_j \left| \bar{C}^b_{kj} - C^b_{kj} \right| \sum_i \bar{C}^a_{ki} + \sum_k \pi_k \sum_i \left| \bar{C}^a_{ki} - C^a_{ki} \right| \sum_j C^b_{kj} \\
&\quad + \sum_k \pi_k \sum_i \bar{C}^a_{ki} \sum_{k'} \pi_{k'} \sum_j \left| \bar{C}^b_{k'j} - C^b_{k'j} \right| + \sum_k \pi_k \sum_j C^b_{kj} \sum_{k'} \pi_{k'} \sum_i \left| \bar{C}^a_{k'i} - C^a_{k'i} \right| \\
&= 2 \sum_k \pi_k \sum_j \left| \bar{C}^b_{kj} - C^b_{kj} \right| + 2 \sum_k \pi_k \sum_i \left| \bar{C}^a_{ki} - C^b_{ki} \right| \\
&\leqslant 2\|\bar{C}^a - C^a\|_1 + 2\|\bar{C}^b - C^b\|_1 \leqslant 4 \times \varepsilon'/4 = \varepsilon'. \qquad \square
\end{aligned}
$$

We now turn to the estimation of the aggregate confusion matrix of each cluster. Let us assume for now that the agents are assigned to the tasks according to the uniform assignment scheme, i.e., agent p belonging to cluster G_a is assigned to a subset of B_a tasks selected uniformly at random from a pool of m_2 tasks. For cluster G_a, we choose $B_a = \frac{m_2}{|G_a|} \ln(\frac{m_2 K}{\beta})$. This implies:

(1) For each $j \in [m_2]$, Pr[agent $p \in G_a$ completes task j] $= \frac{\log(m_2 K/\beta)}{|G_a|}$, i.e., each agent p in G_a is equally likely to complete every task j.

(2) Pr[task j is unlabeled by G_a] $= (1 - \frac{\log(m_2 K/\beta)}{|G_a|})^{|G_a|} \leqslant \frac{\beta}{m_2 K}$. Taking a union bound over the m_2 tasks and K clusters, we get the probability that any task is unlabeled is at most β. Now if we choose $\beta = 1/\text{poly}(m_2)$, we observe that with probability at least $1 - 1/\text{poly}(m_2)$, each task j is labeled by some agent in each cluster when $B_a = \tilde{O}(\frac{m_2}{|G_a|})$.

All that is left to do is to provide an algorithm and sample complexity for learning the aggregate confusion matrices. For this, we will use n dimensional unit vectors to denote the reports of the agents (recall that there are n possible signals). In particular, agent p's report on task j, $r_{pj} \in \{0,1\}^n$. If p's report on task j is c, then the cth coordinate of r_{pj} is 1, and all the other coordinates are 0. The expected value of agent p's report on the jth task is $\mathrm{E}[r_{pj}] = \sum_{k=1}^n \pi_k C^p_k$ The aggregated report for a cluster G_t is given as $R_{tj} = \frac{1}{|G_t|} \sum_{p \in G_t} r_{pj}$.

Suppose we want to estimate the aggregate confusion matrix C^1 of some cluster G_1. To do so, we first pick three clusters G_1, G_2, and G_3 and write down the corresponding cross-moments. Let (a, b, c) be a permutation of the set $\{1, 2, 3\}$. We have:

$$E[R_{aj}] = \sum_k \pi_k C_k^a, \tag{13}$$

$$E[R_{aj} \otimes R_{bj}] = \sum_k \pi_k C_k^a \otimes C_k^b, \tag{14}$$

$$E[R_{aj} \otimes R_{bj} \otimes R_{cj}] = \sum_k \pi_k C_k^a \otimes C_k^b \otimes C_k^c. \tag{15}$$

The cross moments are asymmetric; however, using Theorem 3.6 in the work by Anandkumar et al. [2014], we can write the cross-moments in a symmetric form.

Lemma 4.12. *Assume that the vectors $\{C_1^t, \ldots, C_n^t\}$ are linearly independent for each $t \in \{1, 2, 3\}$. For any permutation (a, b, c) of the set $\{1, 2, 3\}$ define*

$$\begin{aligned} R'_{aj} &= \mathrm{E}[R_{cj} \otimes R_{bj}](\mathrm{E}[R_{aj} \otimes R_{bj}])^{-1} R_{aj}, \\ R'_{bj} &= \mathrm{E}[R_{cj} \otimes R_{aj}](\mathrm{E}[R_{bj} \otimes R_{aj}])^{-1} R_{bj}, \\ M_2 &= \mathrm{E}[R'_{aj} \otimes R'_{bj}] \text{ and } M_3 = \mathrm{E}[R'_{aj} \otimes R'_{bj} \otimes R_{cj}]. \end{aligned}$$

$$\text{Then } M_2 = \sum_{k=1}^{n} \pi_k C_k^c \otimes C_k^c \ \text{ and } \ M_3 = \sum_{k=1}^{n} \pi_k C_k^c \otimes C_k^c \otimes C_k^c.$$

We cannot compute the moments exactly, but rather estimate the moments from samples observed from different tasks. Furthermore, for a given task j, instead of exactly computing the aggregate label R_{gj}, we select one agent p uniformly at random from G_g and use agent p's report on task j as a proxy for R_{gj}. We will denote the corresponding report as $\tilde{R}_{gj}$. The next lemma proves that the cross-moments of $\{\tilde{R}_{gj}\}_{g=1}^K$ and $\{R_{gj}\}_{g=1}^K$ are the same.

Lemma 4.13.

(1) For any group G_a, $\mathrm{E}[\tilde{R}_{aj}] = \mathrm{E}[R_{aj}]$.
(2) For any pair of groups G_a and G_b, $\mathrm{E}[\tilde{R}_{aj} \otimes \tilde{R}_{bj}] = \mathrm{E}[R_{aj} \otimes R_{bj}]$.
(3) For any three groups G_a, G_b and G_c, $\mathrm{E}[\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj}] = \mathrm{E}[R_{aj} \otimes R_{bj} \otimes R_{cj}]$.

Proof.

1. First moments of $\{\tilde{R}_{gj}\}_{g=1}^K$ and $\{R_{gj}\}_{g=1}^K$ are equal:

$$\mathrm{E}[\tilde{R}_{aj}] = \frac{1}{|G_a|} \sum_{p \in G_a} \mathrm{E}[r_{pj}] = E[R_{aj}].$$

2. Second order cross-moments of $\{\tilde{R}_{gj}\}_{g=1}^K$ and $\{R_{gj}\}_{g=1}^K$ are equal:

$$\mathrm{E}[\tilde{R}_{aj} \otimes \tilde{R}_{bj}] = \sum_k \pi_k \mathrm{E}[\tilde{R}_{aj} \otimes \tilde{R}_{bj} | y_j = k] = \sum_k \pi_k \mathrm{E}[\tilde{R}_{aj} | y_j = k] \otimes \mathrm{E}[\tilde{R}_{bj} | y_j = k]$$

$$= \sum_k \pi_k \left(\frac{1}{|G_a|} \sum_{p \in G_a} C_k^p \right) \otimes \left(\frac{1}{|G_b|} \sum_{q \in G_b} C_k^q \right) = \sum_k \pi_k C_k^a \otimes C_k^b = \mathrm{E}[R_{aj} \otimes R_{bj}].$$

3. Third order cross-moments of $\{\tilde{R}_{gj}\}_{g=1}^{K}$ and $\{R_{gj}\}_{g=1}^{K}$ are equal:

$$
\begin{aligned}
\mathrm{E}[\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj}] &= \sum_{k} \pi_k \mathrm{E}[\tilde{R}_{aj} \otimes \tilde{R}_{bj} \otimes \tilde{R}_{cj} | y_j = k] \\
&= \sum_{k} \pi_k \mathrm{E}[\tilde{R}_{aj} | y_j = k] \otimes \mathrm{E}[\tilde{R}_{bj} | y_j = k] \otimes \mathrm{E}[\tilde{R}_{cj} | y_j = k] \\
&= \sum_{k} \pi_k \left(\frac{1}{|G_a|} \sum_{p \in G_a} C_k^p \right) \otimes \left(\frac{1}{|G_b|} \sum_{q \in G_b} C_k^q \right) \otimes \left(\frac{1}{|G_c|} \sum_{r \in G_c} C_k^r \right) \\
&= \sum_{k} \pi_k C_k^a \otimes C_k^b \otimes C_k^c = \mathrm{E}[R_{aj} \otimes R_{bj} \otimes R_{cj}]. \qquad \square
\end{aligned}
$$

The next set of equations shows how to approximate the moments M_2 and M_3:

$$
\hat{R}'_{aj} = \left(\frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{cj'} \otimes \tilde{R}_{bj'} \right) \left(\frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{aj'} \otimes \tilde{R}_{bj'} \right)^{-1} \tilde{R}_{aj}, \tag{16}
$$

$$
\hat{R}'_{bj} = \left(\frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{cj'} \otimes \tilde{R}_{aj'} \right) \left(\frac{1}{m_2} \sum_{j'=1}^{m_2} \tilde{R}_{bj'} \otimes \tilde{R}_{aj'} \right)^{-1} \tilde{R}_{bj}, \tag{17}
$$

$$
\hat{M}_2 = \frac{1}{m_2} \sum_{j'=1}^{m_2} \hat{R}'_{aj'} \otimes \hat{R}'_{bj'} \quad \text{and} \quad \hat{M}_3 = \frac{1}{m_2} \sum_{j'=1}^{m_2} \hat{R}'_{aj'} \otimes \hat{R}'_{bj'} \otimes \tilde{R}_{cj'}. \tag{18}
$$

ALGORITHM 4: Estimating Aggregate Confusion Matrix

```
Require: $K$ clusters of agents $G_1, G_2, \ldots, G_K$ and the reports $\tilde{R}_{gj} \in \{0,1\}^n$ for $j \in [m]$ and $g \in [K]$
Ensure: Estimate of the aggregate confusion matrices $\bar{C}^g$ for all $g \in [K]$
Partition the $K$ clusters into groups of three
for Each group of three clusters $\{g_a, g_b, g_c\}$ do
    for $(a, b, c) \in \{(g_b, g_c, g_a), (g_c, g_a, g_b), (g_a, g_b, g_c)\}$ do
        Compute the second and the third order moments $\hat{M}_2 \in \mathbb{R}^{n \times n}$, $\hat{M}_3 \in \mathbb{R}^{n \times n \times n}$.   ▷ Compute $\bar{C}^g$ and $\bar{\Pi}$ by tensor decomposition
        Compute whitening matrix $\hat{Q} \in \mathbb{R}^{n \times n}$ such that $\hat{Q}^T \hat{M}_2 \hat{Q} = I$
        Compute eigenvalue-eigenvector pairs $(\hat{\alpha}_k, \hat{v}_k)_{k=1}^n$ of the whitened tensor $\hat{M}_3(\hat{Q}, \hat{Q}, \hat{Q})$ by using the robust tensor power method
        Compute $\hat{w}_k = \hat{\alpha}_k^{-2}$ and $\hat{\mu}_k = (\hat{Q}^T)^{-1} \hat{\alpha} \hat{v}_k$
        For $k = 1, \ldots, n$ set the $k$th column of $\bar{C}^c$ by some $\hat{\mu}_k$ whose $k$th coordinate has the greatest component, then set the $k$th diagonal entry of $\bar{\Pi}$ by $\hat{w}_k$
    end for
end for
```

We use the tensor decomposition algorithm (4) on $\hat{M}_2$ and $\hat{M}_3$ to recover the aggregate confusion matrix $\bar{C}^c$ and $\bar{\Pi}$, where $\bar{\Pi}$ is a diagonal matrix whose kth component is $\bar{\pi}_k$, an estimate of π_k. To analyze the sample complexity of Algorithm 4, we need to make some mild assumptions about the problem instance. For any two clusters G_a and G_b, define $S_{ab} = \mathrm{E}[R_{aj} \otimes R_{bj}] = \sum_{k=1}^{n} \pi_k C_k^a \otimes C_k^b$. We make the following assumptions:

(1) There exists $\sigma_L > 0$ such that $\sigma_n(S_{ab}) \geqslant \sigma_L$ for each pair of clusters a and b, where $\sigma_n(M)$ is the smallest eigenvalue of M.
(2) $\kappa = \min_{t \in [k]} \min_{s \in [n]} \min_{r \neq s} \{C_{rr}^t - C_{rs}^t\} > 0$.

The first assumption implies that the matrices S_{ab} are non-singular. The smallest eigenvalue of S_{ab} controls how many samples we need to approximate S_{ab} from its sample mean. The second assumption implies that within a group, the probability of assigning the correct label is always higher than the probability of assigning any incorrect label. Note that this assumption might be false for an individual confusion matrix. However, we are averaging over all the users within a cluster to get the cluster average confusion matrix, and unless a large fraction of individuals within a cluster has the propensity to mislabel, i.e., assign large probability on incorrect labels, this assumption is usually satisfied. The following theorem gives the number of tasks each agent needs to complete to get an ε'-estimate of the aggregate confusion matrices. We will use the following two lemmas due to Zhang et al. [2016]:

Lemma 4.14. *For any $\hat{\varepsilon} \leqslant \sigma_L/2$, the second and the third empirical moments are bounded as*

$$\max\{\|\hat{M}_2 - M_2\|_{op}, \|\hat{M}_3 - M_3\|_{op}\} \leqslant 31\hat{\varepsilon}/\sigma_L^3$$

with probability at least $1 - \delta$ where $\delta = 6\exp(-(\sqrt{m_2}\hat{\varepsilon} - 1)^2) + n\exp(-(\sqrt{m_2/n}\hat{\varepsilon} - 1)^2)$.

Lemma 4.15. *For any $\hat{\varepsilon} \leqslant \kappa/2$, if the empirical moments satisfy*

$$\max\{\|\hat{M}_2 - M_2\|_{op}, \|\hat{M}_3 - M_3\|_{op}\} \leqslant \hat{\varepsilon}H$$

$$\textit{for } H := \min\left\{\frac{1}{2}, \frac{2\sigma_L^{3/2}}{15n(24\sigma_L^{-1} + 2\sqrt{2})}, \frac{\sigma_L^{3/2}}{4\sqrt{3/2}\sigma_L^{1/2} + 8n(24/\sigma_L + 2\sqrt{2})}\right\},$$

then $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\hat{\varepsilon}$, $\|\bar{\Pi} - \Pi\|_{op} \leqslant \hat{\varepsilon}$ with probability at least $1 - \delta$ where δ is defined in Lemma 4.14.

Zhang et al. [2016] prove Lemma 4.14 when $\hat{M}_2$ is defined using the aggregate labels R_{gj}. However, this lemma holds even if one uses the labels $\tilde{R}_{gj}$. The proof is similar if one uses Lemma 4.13. We now characterize the sample complexity of learning the aggregate confusion matrices.

Theorem 4.16. *For any $\varepsilon' \leqslant \min\{\frac{31}{\sigma_L^2}, \frac{\kappa}{2}\}n^2$ and $\delta > 0$, if the size of the universe of shared tasks m_2 is at least $O(\frac{n^7}{(\varepsilon')^2\sigma_L^{11}}\log(\frac{nK}{\delta}))$, then we have $\|\bar{C}^t - C^t\|_1 \leqslant \varepsilon'$ for each cluster G_t. The total number of samples collected by Algorithm 4 is $\tilde{O}(Km_2)$ under the uniform assignment scheme.*

Proof. Substituting $\hat{\varepsilon} = \hat{\varepsilon}_1 H\sigma_L^3/31$ in Lemma 4.14, we get

$$\max\{\|\hat{M}_2 - M_2\|_{op}, \|\hat{M}_3 - M_3\|_{op}\} \leqslant \hat{\varepsilon}_1 H$$

with probability at least $1 - (6+n)\exp(-(\frac{m_2^{1/2}\hat{\varepsilon}_1 H\sigma_L^3}{31n^{1/2}} - 1)^2)$. This substitution requires $\hat{\varepsilon}_1 H\sigma_L^3/31 \leqslant \sigma_L/2$. Since $H \leqslant 1/2$, it is sufficient to have

$$\hat{\varepsilon}_1 \leqslant 31/\sigma_L^2. \tag{19}$$

Now using Lemma 4.15, we see that $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\hat{\varepsilon}_1$ and $\|\bar{\Pi} - \Pi\|_{op} \leqslant \hat{\varepsilon}_1$ with the above probability. It can be checked that $H \geqslant \frac{\sigma_L^{5/2}}{230n}$. This implies that the bounds hold with probability at least $1 - (6+n)\exp(-(\frac{m_2^{1/2}\sigma_L^{11/2}\hat{\varepsilon}_1}{7130n^{3/2}} - 1)^2)$. The second substitution requires

$$\hat{\varepsilon}_1 \leqslant \kappa/2. \tag{20}$$

Therefore, to achieve a probability of at least $1 - \delta$, we need

$$m_2 \geqslant \frac{7130^2 n^3}{\hat{\varepsilon}_1^2\sigma_L^{11}}\left(1 + \sqrt{\log\left(\frac{6+n}{\delta}\right)}\right)^2.$$

It is sufficient that

$$m_2 \geqslant \Omega\left(\frac{n^3}{\hat{\varepsilon}_1^2 \sigma_L^{11}} \log\left(\frac{n}{\delta}\right)\right)$$

to ensure $\|\bar{C}^c - C\|_{op} \leqslant \sqrt{n}\hat{\varepsilon}_1$. For each k, $\|\bar{C}_k^c - C_k\|_1 \leqslant \sqrt{n}\|\bar{C}_k^c - C_k\|_2 \leqslant \sqrt{n}\|\bar{C}^c - C\|_{op} \leqslant n\hat{\varepsilon}_1$. Substituting $\hat{\varepsilon}_1 = \hat{\varepsilon}'/n^2$, we get $\|\bar{C}^c - C\|_1 = \sum_{k=1}^n \|\bar{C}_k^c - C_k\|_1 \leqslant n^2\hat{\varepsilon}_1 = \hat{\varepsilon}'$ when $m_2 = \Omega(\frac{n^7}{(\hat{\varepsilon}')^2 \sigma_L^{11}} \log(\frac{n}{\delta}))$. By a union bound, the result holds for all the clusters simultaneously with probability at least $1 - \delta K$. Substituting δ/K instead of δ gives the bound on the number of samples. Substituting $\hat{\varepsilon}' = \hat{\varepsilon}_1/n^2$ in Equations (19) and (20), we get the desired bound on $\hat{\varepsilon}'$.

Now to compute the total number of samples collected by the algorithm, note that each agent in cluster G_a provides $\frac{m_2}{|G_a|} \log(\frac{Km_2}{\beta})$ samples. Therefore, total number of samples collected from cluster G_a is $m_2 \log(\frac{Km_2}{\beta})$ and the total number of samples collected over all the clusters is $Km_2 \log(\frac{Km_2}{\beta})$. □

Discussion. If the algorithm chooses $m_2 = \tilde{O}(\frac{n^7}{(\varepsilon')^2 \sigma_L^{11}})$, then the total number of samples collected under the uniform assignment scheme is at most $\tilde{O}(\frac{n^7}{(\varepsilon')^2 \sigma_L^{11}})$. So far, we have analyzed the Dawid-Skene model under the uniform assignment scheme. When the assignment scheme is fixed, the moments of R_{aj} and $\tilde{R}_{aj}$ need not be the same. In this case, we will have to run Algorithm 4 with respect to the actual aggregate labels $\{R_{gj}\}_{g=1}^K$. This requires collecting samples from every member of a cluster, leading to a sample complexity of $O(\frac{\ell n^7}{(\varepsilon')^2 \sigma_L^{11}} \log(\frac{nK}{\delta}))$.

To estimate the confusion matrices, Zhang et al. [2016] require each agent to provide at least $O(n^5 \log((\ell + n)/\delta)/(\varepsilon')^2)$ samples. Our algorithm requires $O(n^7 \log(nK/\delta)/(\varepsilon')^2)$ samples from each cluster. The increase of n^2 in the sample complexity comes about, because we are estimating the aggregate confusion matrices in L1 norm instead of the infinity norm. Moreover, when the number of clusters is small ($K \ll \ell$), the number of samples required from each cluster does not grow with ℓ. This improvement is due to the fact that, unlike Zhang et al. [2016], we do not have to recover individual confusion matrices from the aggregate confusion matrices.

Note that the approach based on the work of Dawid and Skene [1979], for the uniform assignment scheme, does not require all agents to provide reports on the same set of shared tasks. Rather, we need that for each group of three clusters (as partitioned by Algorithm 4 on line 1) and each task, there should exist one agent from those three clusters who completes the same task. In particular, the reports for different tasks can be acquired from different agents within the same cluster. The assignment scheme makes sure that this property holds with high probability.

We now briefly compare the learning algorithms under the no-assumptions and model-based approach. When it is difficult to assign agents to the same tasks, and when the number of signals is small (which is often true in practice), the Dawid-Skene method has a strong advantage. Another advantage of the Dawid-Skene method is that the learning error ε' can be made arbitrarily small, since each aggregate confusion matrix can be learned with arbitrary accuracy, whereas the true learning error of the no-assumption approach is at least 2ε (see Theorem 4.7) and depends on the problem instance.

5 CLUSTERING EXPERIMENTS

Our goal in this section is to empirically evaluate the incentive that an agent has to use a non-truthful strategy under the CAHU mechanism in real-world scenarios. Recall that this *incentive error* comes from two sources:

- The clustering error. This represents how "clusterable" the agents are. From theory, we have the upper bound $\varepsilon_1 = \max_{p,q\in[\ell]} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1$.
- The learning error. This represents how accurate our estimates for the cluster Delta matrices are. From theory, we have the upper bound $\varepsilon_2 = \max_{i,j\in[K]} \|\Delta_{G_i,G_j} - \overline{\Delta}_{G_i,G_j}\|_1$.

Given this, the CAHU mechanism is $(\varepsilon_1 + \varepsilon_2)$-informed truthful (Theorem 3.5).

In our experiments, we focus solely on the clustering error due to two reasons. First, the available real-world datasets have little overlap between the tasks performed by different agents, making it harder for us to learn their true pairwise Δ-matrices up to a reasonable accuracy and evaluate the error in our estimation. Note that the overlap is only needed to be able to *evaluate* the learning error of our approach; under the Dawid-Skene model, we do not require any overlap when using our approach in practice.

More importantly, the clustering error and the learning error differ in a key sense. Even with the best possible clustering, the clustering error ε_1 cannot be made arbitrarily small with a fixed number of clusters, because it depends on how close the signal distributions of the agents really are. In contrast, the learning error ε_2 of the no-assumption approach is $3\varepsilon' + 2\varepsilon_1$ (Theorem 4.7), from which the part that does not depend on clustering (ε') can be made arbitrarily small by simply acquiring a sufficient amount of data about agents' behavior. Similarly, the learning error ε_2 in the Dawid-Skene approach—which we use in this experiment—can be made arbitrarily small, too (Theorem 4.16). Hence, given a sufficient amount of data from the agents, the total error would be dominated by the clustering error ε_1. In particular, we show that in practice even a relatively small number of clusters leads to a small clustering error.

We use eight real-world crowdsourcing datasets. Six of these datasets are from the *SQUARE* benchmark [Sheshadri and Lease 2013], selected to ensure a sufficient density of worker labels across different latent attributes as well as the availability of latent attributes for sufficiently many tasks. In addition, we also use the Stanford *Dogs* dataset [Khosla et al. 2011] and the *Expressions* dataset [Mozafari et al. 2012, 2014]. Below, we briefly describe the format of tasks, the number of agents ℓ, and the number of signals n for each dataset.[4]

- Adult: Rating websites for their appropriateness, $\ell = 269$, $n = 4$.
- BM: Sentiment analysis for tweets, $\ell = 83$, $n = 2$.
- CI: Assessing websites for copyright infringement, $\ell = 10$, $n = 3$.
- Dogs: Identifying species from images of dogs, $\ell = 109$, $n = 4$.
- Expressions: Classifying images of human faces by expression, $\ell = 27$, $n = 4$.
- HCB: Assessing relevance of web search results, $\ell = 766$, $n = 4$.
- SpamCF: Assessing whether response to a crowdsourcing task was spam, $\ell = 150$, $n = 2$.
- WB: Identifying whether the waterbird in the image is a duck, $\ell = 53$, $n = 2$.

Since all datasets specify the latent value of the tasks, we adopt the Dawid-Skene model and estimate the confusion matrices from the frequency with which each agent p reports each label j in the case of each latent attribute i.

We first use a clustering algorithm to cluster the estimated confusion matrices. Typical clustering algorithms take a distance metric over the space of data points and attempt to minimize the maximum cluster diameter, which is the maximum distance between any two data points in a cluster. In contrast, our objective function (Equation (21)) is a complex function of the underlying confusion matrices. We therefore compare two approaches:

[4] We filter each dataset to remove tasks for which the latent attribute is unknown and remove workers who only perform such tasks. ℓ is the number of agents that remain after filtering.

(1) In this approach, we cluster the confusion matrices using the standard k-means++ algorithm with the $L2$ norm distance (available in Matlab) and hope that resulting clustering leads to a small error.[5]

(2) In the following lemma, we derive a distance metric over confusion matrices for which the maximum cluster diameter is provably an upper bound on the clustering error and use k-means++ with this metric (implemented in Matlab).[6] Note that computing this metric requires knowledge of the prior over the latent attribute (e.g., in the WB dataset, this would require knowing the probability that a random image of a waterbird is a duck), which can be estimated easily from a small amount of ground truth data.

Lemma 5.1. *For all agents p, q, r, we have $\|\Delta_{p,q} - \Delta_{p,r}\|_1 \leqslant 2 \cdot \sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}|$.*

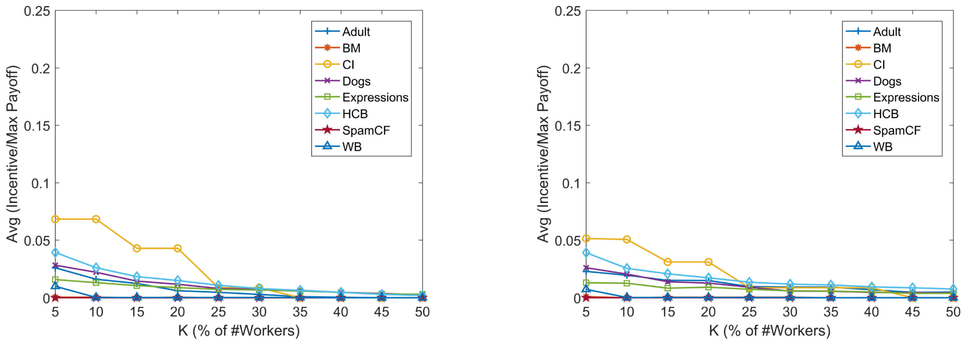
Proof. We have

$$
\begin{aligned}
&\|\Delta_{p,q} - \Delta_{p,r}\|_1 = \sum_{i,j} \left|\Delta_{p,q}(i,j) - \Delta_{p,r}(i,j)\right| \\
&= \sum_{i,j} \left|D_{p,q}(i,j) - D_p(i)D_q(j) - D_{p,r}(i,j) + D_p(i)D_r(j)\right| \\
&= \sum_{i,j} \left|D_{p,q}(i,j) - D_{p,r}(i,j) - D_p(i)(D_q(j) - D_r(j))\right| \\
&= \sum_{i,j} \left|\sum_k \pi_k C^p_{ki} C^q_{kj} - \sum_k \pi_k C^p_{ki} C^r_{kj} - \sum_k \pi_k C^p_{ki} \left(\sum_l \pi_l C^q_{lj} - \sum_l \pi_l C^r_{lj}\right)\right| \\
&= \sum_{i,j} \left|\sum_k \pi_k C^p_{ki} \left(C^q_{kj} - C^r_{kj}\right) - \sum_k \pi_k C^p_{ki} \left(\sum_l \pi_l \left(C^q_{lj} - C^r_{lj}\right)\right)\right| \\
&\leqslant \sum_j \sum_k \pi_k \left|C^q_{kj} - C^r_{kj}\right| \sum_i C^p_{ki} + \sum_j \sum_k \pi_k \sum_l \pi_l \left|C^q_{lj} - C^r_{lj}\right| \sum_i C^p_{ki} \\
&= \sum_j \sum_k \pi_k \left|C^q_{kj} - C^r_{kj}\right| + \sum_j \sum_k \pi_k \sum_l \pi_l \left|C^q_{lj} - C^r_{lj}\right| \qquad \left[\text{Using} \sum_i C^p_{ki} = 1\right] \\
&\leqslant \sum_k \pi_k \sum_j \left|C^q_{kj} - C^r_{kj}\right| + \sum_k \pi_k \sum_l \pi_l \sum_j \left|C^q_{lj} - C^r_{lj}\right| \\
&= \sum_k \pi_k \sum_j \left|C^q_{kj} - C^r_{kj}\right| + \sum_l \pi_l \sum_j \left|C^q_{lj} - C^r_{lj}\right| \qquad \left[\text{Using} \sum_k \pi_k = 1\right] \\
&= 2 \cdot \sum_k \pi_k \sum_j \left|C^q_{kj} - C^r_{kj}\right|,
\end{aligned}
$$

as required. □

[5] We use $L2$ norm rather than $L1$ norm, because the standard k-means++ implementation uses as the centroid of a cluster the confusion matrix that minimizes the sum of distances from the confusion matrices of the agents in the cluster. For $L2$ norm, this amounts to averaging over the confusion matrices, which is precisely what we want. For $L1$ norm, this amounts to taking a pointwise median, which does not even result in a valid confusion matrix. Perhaps for this reason, we observe that using the $L1$ norm performs worse.

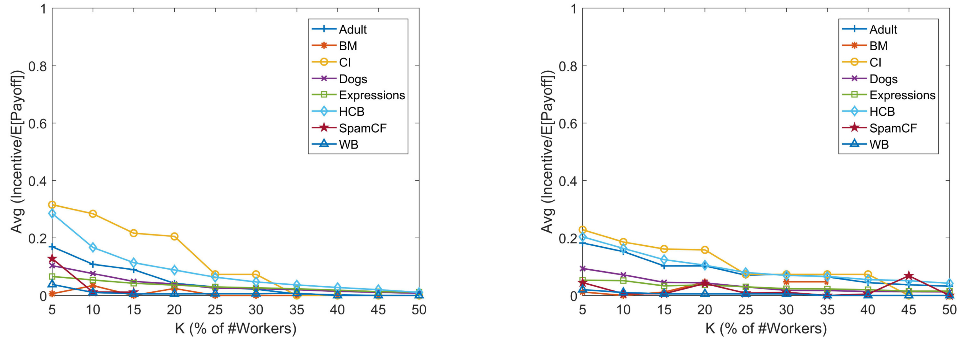
[6] For computing the centroid of a cluster, we still average over the confusion matrices of the agents in the cluster. Also, since the algorithm is no longer guaranteed to converge (indeed, we observe cycles), we restart the algorithm when a cycle is detected, at most 10 times.



(a) Standard $L2$ metric

(b) Our custom metric

Fig. 4. The incentive error as a fraction of the maximum payoff of an agent, averaged over agents, on eight different datasets when using k-means++ with the $L2$ metric and with our custom metric.



(a) Standard $L2$ metric

(b) Our custom metric

Fig. 5. The incentive error as a fraction of the expected payoff of an agent, averaged over agents, on eight different datasets when using k-means++ with the $L2$ metric and with our custom metric.

Note that $\sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}| \leqslant \|C^q - C^r\|_1$, because $\sum_j |C^q_{lj} - C^r_{lj}| \leqslant \|C^q - C^r\|_1$. Lemma 5.1, along with Lemma 4.8, shows that the incentive error due to clustering is upper bounded by four times the maximum cluster diameter under our metric, which defines the distance between C^q and C^r as $\sum_k \pi_k \sum_j |C^q_{kj} - C^r_{kj}|$.

For each dataset, we vary the number of clusters K from 5% to 15% of the number of agents in the dataset. Within the k-means++ algorithm, we use 20 random seeds and choose the best clustering produced.

Next, we compute the clustering error. Instead of using the weak bound $\max_{p,q\in[\ell]} \|\Delta_{p,q} - \Delta_{G(p),G(q)}\|_1$ on the clustering error (which is nevertheless helpful for our theoretical results), we use the following tighter bound from the proof of Theorem 3.5:

$$|u^*_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p}) - u_p(\mathbb{I}, \{\mathbb{I}\}_{q\neq p})| = \left| \frac{1}{(\ell - 1)} \sum_{q\in P\setminus\{p\}} \sum_{i,j} \Delta_{p,q}(i,j) \Big(\mathrm{Sign}(\Delta_{p,q})_{i,j} - \mathrm{Sign}(\overline{\Delta}_{G(p),G(q)})_{i,j} \Big) \right|. \tag{21}$$

Assuming no learning error, this would be an upper bound on the incentive that agent p has to use a non-truthful strategy under the CAHU mechanism. We compare this bound to both the maximum payoff that agent p can receive and the expected payoff that agent p would receive under our mechanism and plot the result averaged over p. Figures 4(a) and 4(b) similarly show the incentive of an average agent as a fraction of her *maximum* payoff with the standard $L2$ metric and with our custom metric, respectively. Figures 5(a) and 5(b) show the incentive of an average agent as a fraction of her *expected* payoff with standard $L2$ metric and with our custom metric, respectively. We note that the expected payoff is a stronger and more realistic benchmark than the maximum payoff.

In comparison to both the maximum and the expected payoffs, the incentive error is small—less than 20% of the expected payoff and less than 5% of the maximum payoff—even with the number of clusters K as small as 15% of the number of workers. The number of agents does not seem to significantly affect this bound as long as the number of clusters is a fixed percentage of the number of agents. We also note that using our custom metric leads to a somewhat smaller error than using the standard $L2$ norm.

6 CONCLUSION

We have provided the first general solution to the problem of peer prediction with heterogeneous agents. This is a compelling research direction, where new theory and algorithms can help to guide practice. In particular, heterogeneity is likely to be quite ubiquitous due to differences in taste, context, judgment, and reliability across users. Beyond testing these methods in a real-world application such as marketing surveys, there remain interesting directions for ongoing research. For example, is it possible to solve this problem with a similar sample complexity but without a clustering approach? Is it possible to couple methods of peer prediction with optimal methods for inference in crowdsourced classification [Ok et al. 2016] and with methods for task assignment in budgeted settings [Karger et al. 2014]? This should include attention to adaptive assignment schemes [Khetan and Oh 2016] that leverage generalized Dawid-Skene models [Zhou et al. 2015] and could connect to the recent progress on task heterogeneity within peer prediction [Mandal et al. 2016]. Finally, it is worth investigating if we can cluster the agents based on some observable characteristics such as demographics, reputation scores, and so on, and reduce the sample complexity of the original mechanism.

ACKNOWLEDGMENTS

We thank Adish Singla and Dengyong Zhou for providing us the Stanford Dogs (Dogs) and the Waterbird dataset (WB). We also acknowledge the useful feedback from anonymous reviewers.

REFERENCES

Hunt Allcott and Matthew Gentzkow. 2017. *Social Media and Fake News in the 2016 Election.* Technical Report. National Bureau of Economic Research.

Animashree Anandkumar, Rong Ge, Daniel J. Hsu, Sham M. Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.* 15, 1 (2014), 2773–2832.

Yang Cai, Constantinos Daskalakis, and Christos Papadimitriou. 2015. Optimum statistical estimation with strategic data sources. In *Proceedings of the 28th Conference on Learning Theory.* 280–296.

Anirban Dasgupta and Arpita Ghosh. 2013. Crowdsourced judgement elicitation with endogenous proficiency. In *Proceedings of the 22nd International Conference on World Wide Web.* ACM, 319–330.

Philip A. Dawid and Allan M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Appl. Stat.* 28, 1 (1979), 20–28.

Jennifer DeBoer, Glenda S. Stump, Daniel Seaton, and Lori Breslow. 2013. Diversity in MOOC students' backgrounds and behaviors in relationship to performance in 6.002x. In *Proceedings of the 6th Learning International Networks Consortium Conference*, Vol. 4.

Luc Devroye and Gábor Lugosi. 2012. *Combinatorial Methods in Density Estimation.* Springer Science & Business Media.

Boi Faltings and Goran Radanovic. 2017. Game theory for data science: Eliciting truthful information. *Synth. Lect. Artif. Intell. Mach. Learn.* 11, 2 (2017), 1–151.

Adam Fourney, Miklos Z. Racz, Gireeja Ranade, Markus Mobius, and Eric Horvitz. 2017. Geographic and temporal trends in fake news consumption during the 2016 US presidential election. In *Conference on Information and Knowledge Management*, Vol. 17. 6–10.

Rafael Frongillo and Jens Witkowski. 2017. A geometric perspective on minimal peer prediction. *ACM Trans. Econ. Comput.* 5, 3 (2017), 1–27.

Alice Gao, James R. Wright, and Kevin Leyton-Brown. 2016. Incentivizing evaluation via limited access to ground truth: Peer-prediction makes things worse. In *Proceedings of the Workshop on Algorithmic Game Theory and Data Science (EC'16).*

Radu Jurca, Boi Faltings et al. 2009. Mechanisms for making crowds truthful. *J. Artif. Intell. Res.* 34, 1 (2009), 209.

Vijay Kamble, David Marn, Nihar Shah, Abhay Parekh, and Kannan Ramachandran. 2015. Truth serums for massively crowdsourced evaluation tasks. In *Proceedings of the 5th Workshop on Social Computing and User-Generated Content.*

David R. Karger, Sewoong Oh, and Devavrat Shah. 2011. Budget-optimal task allocation for reliable crowdsourcing systems. *CoRR* abs/1110.3564 (2011). http://arxiv.org/abs/1110.3564.

David R. Karger, Sewoong Oh, and Devavrat Shah. 2014. Budget-optimal task allocation for reliable crowdsourcing systems. *Op. Res.* 62, 1 (2014), 1–24. DOI: https://doi.org/10.1287/opre.2013.1235

Ashish Khetan and Sewoong Oh. 2016. Achieving budget-optimality with adaptive schemes in crowdsourcing. In *Proceedings of the International Conference on Neural Information Processing Systems.* 4844–4852.

Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Li Fei-Fei. 2011. Novel dataset for fine-grained image categorization. In *Proceedings of the 1st CVPR Workshop on Fine-Grained Visual Categorization.*

Yuqing Kong, Katrina Ligett, and Grant Schoenebeck. 2016. Putting peer prediction under the micro (economic) scope and making truth-telling focal. In *Proceedings of the International Conference on Web and Internet Economics.* Springer, 251–264.

Yuqing Kong and Grant Schoenebeck. 2016. A framework for designing information elicitation mechanism that rewards truth-telling. Retrieved from http://arxiv.org/abs/1605.01021.

Chinmay Kulkarni, Koh Pang Wei, Huy Le, Daniel Chia, Kathryn Papadopoulos, Justin Cheng, Daphne Koller, and Scott R. Klemmer. 2015. Peer and self assessment in massive online classes. In *Design Thinking Research.* Springer, 131–168.

Yang Liu and Yiling Chen. 2017a. Machine-learning aided peer prediction. In *Proceedings of the ACM Conference on Economics and Computation.* ACM, 63–80.

Yang Liu and Yiling Chen. 2017b. Sequential peer prediction: Learning to elicit effort using posted prices. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence.* 607–613.

Debmalya Mandal, Matthew Leifer, David C. Parkes, Galen Pickard, and Victor Shnayder. 2016. Peer prediction with heterogeneous tasks. In *Proceedings of the NIPS 2016 Workshop on Crowdsourcing and Machine Learning.*

Nolan Miller, Paul Resnick, and Richard Zeckhauser. 2005. Eliciting informative feedback: The peer-prediction method. *Manag. Sci.* 51 (2005), 1359–1373.

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. 2012. Active learning for crowd-sourced databases. *CoRR* abs/1209.3686 (2012).

Barzan Mozafari, Purnamrita Sarkar, Michael J. Franklin, Michael I. Jordan, and Samuel Madden. 2014. Scaling up crowdsourcing to very large datasets: A case for active learning. *PVLDB* 8, 2 (2014), 125–136.

Jungseul Ok, Sewoong Oh, Jinwoo Shin, and Yung Yi. 2016. Optimality of belief propagation for crowdsourced classification. In *Proceedings of the 33rd International Conference on Machine Learning (ICML'16).* 535–544.

Drazen Prelec. 2004. A Bayesian truth serum for subjective data. *Science* 306, 5695 (2004), 462.

Goran Radanovic and Boi Faltings. 2015a. Incentive schemes for participatory sensing. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15).* 1081–1089.

Goran Radanovic and Boi Faltings. 2015b. Incentive schemes for participatory sensing. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS'15).*

Goran Radanovic and Boi Faltings. 2015c. Incentives for subjective evaluations with private beliefs. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI'15).* 1014–1020.

Goran Radanovic, Boi Faltings, and Radu Jurca. 2016. Incentives for effort in crowdsourcing using the peer truth serum. *ACM Trans. Intell. Syst. Technol.* 7, 4 (2016), 48.

Aashish Sheshadri and Matthew Lease. 2013. SQUARE: A benchmark for research on computing crowd consensus. In *Proceedings of the 1st AAAI Conference on Human Computation (HCOMP'13).* 156–164.

Victor Shnayder, Arpit Agarwal, Rafael Frongillo, and David C. Parkes. 2016a. Informed truthfulness in multi-task peer prediction. In *Proceedings of the ACM Conference on Economics and Computation.* ACM, 179–196.

Victor Shnayder, Rafael Frongillo, and David C. Parkes. 2016b. Measuring performance of peer prediction mechanisms using replicator dynamics. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI'16).* 2611–2617.

Victor Shnayder and David C. Parkes. 2016. Practical peer prediction for peer assessment. In *Proceedings of the 4th AAAI Conference on Human Computation and Crowdsourcing*.

Edwin Simpson, Stephen J. Roberts, Ioannis Psorakis, and Arfon Smith. 2013. Dynamic Bayesian combination of multiple imperfect classifiers. *Dec. Making Imperf.* 474 (2013), 1–35.

Luis Von Ahn and Laura Dabbish. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'04)*. 319–326.

Julia Wilkowski, Amit Deutsch, and Daniel M. Russell. 2014. Student skill and goal achievement in the mapping with Google MOOC. In *Proceedings of the 1st ACM Conference on Learning@ Scale*. ACM, 3–10.

Jens Witkowski, Yoram Bachrach, Peter Key, and David Parkes. 2013. Dwelling on the negative: Incentivizing effort in peer prediction. In *Proceedings of the 1st AAAI Conference on Human Computation and Crowdsourcing*.

Jens Witkowski and David C. Parkes. 2013. Learning the prior in minimal peer prediction. In *Proceedings of the 3rd Workshop on Social Computing and User Generated Content at the ACM Conference on Electronic Commerce*. 14.

Muhammad Bilal Zafar, Krishna P. Gummadi, and Cristian Danescu-Niculescu-Mizil. 2016. Message impartiality in social media discussions. In *Proceedings of the International Conference on Web and Social Media (ICWSM'16)*. 466–475.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I. Jordan. 2016. Spectral methods meet EM: A provably optimal algorithm for crowdsourcing. *J. Mach. Learn. Res.* 17, 102 (2016), 1–44.

Dengyong Zhou, Qiang Liu, John C. Platt, Christopher Meek, and Nihar B. Shah. 2015. Regularized minimax conditional entropy for crowdsourcing. *CoRR* abs/1503.07240 (2015). http://arxiv.org/abs/1503.07240.

Received November 2017; revised November 2018; accepted May 2019