No Oops, You Won't Do It Again: Mechanisms for Self-correction in Crowdsourcing

Nihar B. Shah NIHAR@EECS.BERKELEY.EDU
Dept. of EECS, University of California, Berkeley

Dengyong Zhou DENGYONG.ZHOU@MICROSOFT.COM
Microsoft Research, Redmond

Abstract

Crowdsourcing is a very popular means of obtaining the large amounts of labeled data that modern machine learning methods require. Although cheap and fast to obtain, crowdsourced labels suffer from significant amounts of error, thereby degrading the performance of downstream machine learning tasks. With the goal of improving the quality of the labeled data, we seek to mitigate the many errors that occur due to silly mistakes or inadvertent errors by crowdsourcing workers. We propose a two-stage setting for crowdsourcing where the worker first answers the questions, and is then allowed to change her answers after looking at a (noisy) reference answer. We mathematically formulate this process and develop mechanisms to incentivize workers to act appropriately. Our mathematical guarantees show that our mechanism incentivizes the workers to answer honestly in both stages, and refrain from answering randomly in the first stage or simply copying in the second. Numerical experiments reveal a significant boost in performance that such "self-correction" can provide when using crowdsourcing to train machine learning algorithms.

1. Introduction

The emergence of deep learning and other complex machine learning tools have resulted in a need for huge amounts of labeled data (Raykar et al., 2010; Deng et al., 2009; Carlson et al., 2010). One of the most popular means of obtaining labeled data is crowdsourcing, where data is labeled by crowds of semi-skilled workers through the Internet typically in exchange from some monetary payments. Crowdsourcing is widely used in many real-world applications, and is particularly popular for collecting training labels for machine learning powered systems like web search engines (Burges et al., 2005; Alonso & Mizzaro, 2009; Kazai, 2011) or to supplement automated algorithms (Khatib et al., 2011; Lang & Rio-Ross, 2011; Von Ahn et al., 2008). The labels obtained from crowdsourcing, however, have significant amounts of error (Kazai et al., 2011; Vuurens et al., 2011; Wais et al., 2010), thereby degrading the performance of the machine learning algorithms that use this data downstream. Consequently, there is much emphasis on gathering higher quality labels, since a lower noise implies requirement of fewer labels for obtaining the same accuracy in practice.

In a study from a few years back, Kahneman & Frederick (2002) asked the following question to many participants: *"A bat and ball cost a dollar and ten cents. The bat costs a dollar more than the ball. How much does the ball cost?"* (See also The New Yorker (2012).) A large number of respondents gave an incorrect answer of "10 cents", including a majority of the students surveyed at Harvard University, Princeton University and MIT. Indeed, making silly mistakes is a part and parcel of being human. In several domains of science and technology that deal with humans, there are special provisions to mitigate the effects of such inadvertent errors (Dijkstra, 1979; Ayewah & Pugh, 2009; Aggarwal et al., 2013). In this work, we consider the problem of mitigating silly mistakes in crowdsourcing.

Unsurprisingly, the data obtained from crowdsourcing also suffers from several forms of inadvertent errors. Examples of such errors include those resulting from not following instructions properly (Gupta et al., 2012), misreading questions (Chros & Sundell, 2011), mistakes when entering solutions (Gupta et al., 2012), incorrect recollection (Lasecki et al., 2013), framing effects (Levin et al., 1998), satisficing (Krosnick, 1991), and many others (e.g., see Tversky & Kahneman 1974; Fleurbaay & Eveleigh 2012).

Proceedings of the 33^{rd} International Conference on Machine Learning, New York, NY, USA, 2016. JMLR: W&CP volume 48.

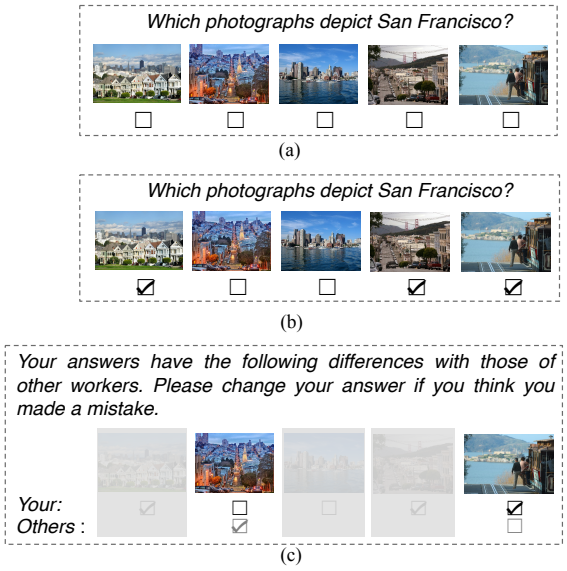

Figure 1: Illustration of self-correction in crowdsourcing. (a) Interface at the start of the task, comprising a set of 5 questions in this example. (b) In the first stage, the worker gives her answers to all the questions. (c) In the second stage, the worker is intimated of all the questions where her answers mismatched with a reference set of answers, and is allowed to change her answer to any of these questions

One approach towards mitigating these errors is to hire more workers to *independently* perform the same task and then aggregate their responses. This approach is the topic of several recent papers using of statistical aggregation algorithms to aggregate this data (e.g., see Raykar et al. (2010); Karger et al. (2011); Liu et al. (2012a); Zhou et al. (2015) and references therein). Most such existing work on crowdsourcing focuses on independent workers. Our approach is complementary to this line of work, and in fact, can nicely supplement these algorithms by providing them higher quality data at lower costs.

Specifically, we suggest a two stage "self-correction" setting. In the first stage, all workers are required to independently accomplish the crowdsourcing task which consists of a set of questions (see Figure 1a and Figure 1b). Standard crowdsourcing setups stop at this stage; we will term these settings as "single-stage" or "without self-correction" settings. Moving on, our self-correction setting is associated to a second stage in which the worker's answers are compared with a reference set of answers. For instance, the reference could be an aggregate of the responses of other workers who may have performed this task.[1] Alternatively, the reference could be the output of a (potentially noisy) machine learning algorithm. For every question that the worker answered differently from the reference (Figure 1), we offer her a second chance, and allow her to change her answer if she wishes to (see Figure 1c). In this paper we will often refer to the proposed self-correction setting as a "two-stage" setting.

The self-correction setting exploits the well-understood fact that reviewing a task for mistakes takes much less time than processing a new task (Gu, 2015; Haas et al., 2015). Moreover, incentivizing the workers to look at the feedback has an additional positive consequence of improving their understanding and performance in subsequent tasks (Jiang & Matsubara, 2012; Dow et al., 2011).

Contributions of the paper With a broad goal of designing ways to improve the quality of labeled data for machine learning algorithms, the specific contributions of this paper are three-fold. First, we introduce and mathematically formulate such a two-stage crowdsourcing setting for self-correction to mitigate inadvertent errors. Second are our primary contributions — theoretical results on mechanism design for the two-stage setting for self-correction. In particular, we consider tasks involving binary-choice questions. We design payment mechanisms to ensure that the workers are indeed incentivized to report truthfully in both stages of the task. (Any such mechanism is called "incentive compatible.") The problem of designing incentive-compatible mechanisms in this setting is challenging since on one hand we must ensure that the worker doesn't simply copy the reference answer, while on the other, we must also ensure that the worker cannot earn greater amounts by deliberately providing a false report in the first round and then changing her answer in the second round. The mechanism must accommodate all possible beliefs of the worker regarding the distribution of the true and the reference answers. We also theoretically prove attractive additional guarantees offered by our mechanism such as minimum slack (to be defined later) and uniqueness of the mechanism. Third, we conduct extensive numerical experiments that reveal how our self-correction setting can result in a significant improvement in the end-to-end accuracy of machine learning systems that use crowdsourced training data.

Related literature Our proposal to use other workers' answers as a reference is inspired from the benefits of communication studied in the literature on psychology. Several papers in the field show that interaction in a group can improve overall group performance in decision making (Kerr & Tindale, 2004; Kozlowski & Ilgen, 2006). However, the amount of shared information has to be limited and controlled, for example, by the so-called Delphi technique (Clayton, 1997; Rowe & Wright, 1999; Hasson et al., 2000). Otherwise, if the interaction in a group is rich, such as face-to-face discussions, it could lead to social bias

[1] The first worker who does this task will operate under the single-stage setting.

(Muchnik et al., 2013). The relationships among social pressure, attention to the stimulus, doubt about one's own judgment, and conformity have been thoroughly explored in psychology (Tesser et al., 1983), political science (Gerber et al., 2008), and consumer research (Bearden & Rose, 1990). These observations influenced the design of the proposed two-stage setting.

Our self-correction setting is related to but fundamentally different from the examination-verification methods in the crowdsourcing literature (Bernstein et al., 2010; Gao et al., 2011; Miller & Steyvers, 2011; Liu et al., 2012b; Su et al., 2012; Hara et al., 2013). Both allow limited information sharing among crowdsourcing workers rather than letting them work independently. However, in the examination-verification approaches, workers sequentially work on a task. Every worker examines the results from her predecessor and revises them when she disagrees. Consequently, workers do not have a chance for self-correction. Moreover, to the best of our our best knowledge, there is no incentive mechanism proposed for these examination-verification approaches. A worker may thus be incentivized to simply approve all the answers from her predecessor.

Several other works in the literature focus on design mechanisms for crowdsourcing (e.g., see Prelec 2004; Miller et al. 2005; Ranade & Varshney 2012; Shah & Zhou 2015; Shah et al. 2015 and references therein), a subset of which share our focus on the aspect of better labels for machine learning algorithms. However, these works all consider various forms of the single-stage setup. While the variants of the single-stage setup analyzed in these works are indeed non-trivial, as we will see in the sequel, the proposed two-stage self-correction setting on the other hand, comes with a set of very unique challenges.

Strictly proper scoring rules (Brier, 1950; Savage, 1971; Gneiting & Raftery, 2007) provide a general theory of mechanism design for eliciting private beliefs about the prediction of an event. Our setting of the design of payment mechanisms falls into the broad framework of strictly proper scoring rules.

2. Problem formulation

We begin with a formal description of the problem setting.

2.1. The task interface

There are N questions asked to a worker. We focus on binary-valued questions. The questions are objective, that is, for every question exactly one of the two options is correct. We will denote the two options for any questions as "A" and "B". The task proceeds in two stages:

- *Stage 1:* The worker is shown N questions. For every question, the worker selects either A or B as her answer.
- *Stage 2:* The worker's answers to all the questions are matched to a reference set of answers. For each question whose answer does not match, the worker is alerted about this mismatch and is given an option to either retain her own answer or copy the reference answer.

In order to evaluate the worker's performance, it is a common practice to include some "gold standard" questions in the task, that is, questions to which the answers known apriori to the mechanism designer. Specifically, we assume that the set of N questions contain G "gold standard" questions ($1 \leq G \leq N$), mixed uniformly at random in the entire set of questions. The worker does not know the identities of the gold standard questions. It is important to note that the gold standard questions are used only for evaluating the worker's performance at the end of the entire task, and are separate from the reference answer.

2.2. Beliefs of the worker

The worker has her own subjective probabilities with respect to the true answer and the reference answer for every question. During the first stage, from the *point of view of the worker*, for any question $i \in [N]$, let[2]

- $p_{A,i}$ be the probability that the correct answer is A
- $p_{B,i}$ $(= 1 - p_{A,i})$ be the probability that the correct answer is B
- $q_{A,i}$ be the probability that the reference answer is A
- $q_{B,i}$ $(= 1 - q_{A,i})$ be the probability that the reference answer is B.

In the second stage, the questions for which the worker's answers do not match the reference are displayed to the worker. The worker updates her subjective probabilities accordingly as, for any question $i \in [N]$ displayed in the second stage,

- $p'_{A|B,i}$ $(\leq p_{A,i})$ be the probability that the correct answer is A given that the reference answer was B
- $p'_{B|A,i}$ $(\leq p_{B,i})$ be the probability that the correct answer is B given that the reference answer was A.

We also define $p'_{A|A,i} = 1 - p'_{B|A,i}$ and $p'_{B|B,i} = 1 - p'_{A|B,i}$.

We make the standard game theoretic assumptions that the workers aim to maximize their expected payment, and that her beliefs about the different questions are independent. With respect to further rationality, we consider two types of workers:

[2] We adopt the standard notation of letting $[N]$ denote the set $\{1, \ldots, N\}$ for any positive integer N.

- *Fully rational:* The worker ensures her beliefs are restricted to obey the law of total probability

$$p_{A,i} = q_{A,i}p'_{A|A,i} + q_{B,i}p'_{A|B,i}, \tag{1a}$$
$$p_{B,i} = q_{A,i}p'_{B|A,i} + q_{B,i}p'_{B|B,i}, \tag{1b}$$

  for all i.

- *Partially rational:* The worker may only have a "bounded" view of the probabilities involved, in which case the worker may assume values of $p_{A,i}$, $q_{A,i}$, $p'_{A|B,i}$ and $p'_{B|A,i}$ without the restriction imposed in (1).

In this paper we will support both fully and partially rational workers. The mechanisms designed subsequently will be incentive-compatible for both these types of workers.

Finally note that the values of the worker's beliefs are, of course, unknown to us. The goal is to design mechanisms that are incentive compatible for arbitrary values of these beliefs, as formalized below.

2.3. Requirements

The goal is to design a payment mechanism that incentivizes the worker to act as follows. Consider any choice of a fixed threshold $T \in \left[\frac{1}{2}, 1\right)$. The choice of the threshold T is made by the system designer based on the application at hand, and in this paper we will assume that the threshold is given to us. For any question $i \in [N]$, for arbitrary values of the worker's beliefs, the worker should be incentivized to select her answers in the following manner.

- First stage: For every question $i \in [N]$, the worker should be incentivized to select the option that she thinks is most likely to be correct, namely

$$\text{select} \begin{cases} \text{option "A"} & \text{if } p_{A,i} > \frac{1}{2} \\ \text{option "B"} & \text{if } p_{A,i} < \frac{1}{2}. \end{cases}$$

- Second stage: For every question $i \in [N]$ that had a mismatch in the first stage, the worker should copy the reference answer if and only if she is really sure about the reference answer. Formally, if the worker selected option "A" in the first stage, then she should

$$\text{select} \begin{cases} \text{"Copy"} & \text{if } p'_{B|B,i} > T \\ \text{"Retain"} & \text{if } p'_{B|B,i} < T \end{cases},$$

  and if the worker selected option "B" in the first stage, then she should

$$\text{select} \begin{cases} \text{"Copy"} & \text{if } p'_{A|A,i} > T \\ \text{"Retain"} & \text{if } p'_{A|A,i} < T. \end{cases}$$

Observe that in our model, we have restricted T to take a value of $\frac{1}{2}$ or more.[3] When $T = \frac{1}{2}$, the setting reduces to the conventional setting requiring the worker to select the option she thinks is most likely to be correct. When T is chosen to be strictly greater than a half, the worker should copy the reference answer only if she is really sure. This choice helps avoid the bias of simply believing in the reference and copying it.

The worker's final performance is evaluated based on her responses to the G gold standard questions. The worker's selection for any question in the gold standard may get evaluated to one of six possibilities, denoted by $\{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}$, and defined as:

- $+\mathfrak{M}$: Match in the first round, and correct
- $-\mathfrak{M}$: Match in the first round, and incorrect
- $+\mathfrak{R}$: Mismatch in the first round, retained in the second round, and correct
- $-\mathfrak{R}$: Mismatch in the first round, retained in the second round, and incorrect
- $+\mathfrak{C}$: Mismatch in the first round, copied in the second round, and correct
- $-\mathfrak{C}$: Mismatch in the first round, copied in the second round, and incorrect.

Here "match" and "mismatch" respectively stand for whether the answer to a question given by a worker is same as the answer to that question in the reference or not. The terms "correct" and "incorrect" respectively refer to whether the option selected by the worker was correct (that is, matched the gold standard) or not.

Let μ denote the maximum pay a worker can receive in this task. The value of μ should be chosen based on application-specific conditions such as the recommended hourly wage for the worker; in this paper, we assume that the value of μ is given to us. In accordance with the requirements of crowdsourcing platforms, we will also assume that the payments made to the workers are non-negative.

Given the notation introduced thus far, we can mathematically represent any payment mechanism as a function $f : \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \to [0, \mu]$. Then by definition of the parameter μ, we have $\max f(\cdot) = \mu$.

As mentioned earlier in Section 2.2, we assume that the worker aims to maximize her expected payment. The expectation of the payment f is taken over the random distribution of the G gold standard questions among the N questions, and over the worker's uncertainties $\{p_{A,i}, p_{B,i}, q_{A,i}, q_{B,i}, p'_{A|B,i}, p'_{B|A,i}\}_{i\in[N]}$ about the correctness of her own answers and of the reference answers.

[3] Our results also extend to the case of $T < \frac{1}{2}$. However, we choose to omit this case since we are interested in eliminating the bias towards simply copying the reference answer, and hence restrict attention to only $T \geq \frac{1}{2}$ in the narrative.

The goal is to design a mechanism f such that its expected value (from the point of view of the worker) is *strictly* maximized when in both stages, the worker answers as per the requirements stated above. Any such mechanism is termed an "incentive-compatible" mechanism.

3. One Stage: Trivial Mechanism

To set the ball rolling, let us first consider the standard setting of a single stage, which is typical of the crowdsourcing setups of today. Under such a setting, the worker must answer all the questions, and the payment is made to the worker based on these answers (to the gold standard questions). The condition of incentive compatibility requires that for all the questions, the worker must be incentivized to select the option which she thinks is most likely to be correct, i.e., to incentivize the worker to choose option A if $p_{A,i} > p_{B,i}$ and B if $p_{A,i} < p_{B,i}$ for any question $i \in [N]$. Of course, the mechanism designer does not know the values of $\{p_{A,i}, p_{B,i}\}_{i\in[N]}$.

Proposition 1 (trivial)**.** *Consider any values M_+ and M_- such that $M_+ > M_- \geq 0$ and $GM_+ = \mu$. Letting C denote the number of questions in the gold standard answered correctly, the following mechanism is incentive compatible:*

$$\text{Payment} = (M_+C + M_-(G - C)).$$

Proposition 1 presents just one of the many mechanisms that can be constructed for the single stage setting, and it is trivial to construct mechanisms that are incentive compatible if there was only one stage. The situation, however, changes dramatically upon introduction of the second stage, as is discussed in the rest of this paper.

4. Two stages: Where Things Get Interesting

We now consider the two-stage setting of Section 2.

4.1. Impossibility of incentive compatible mechanisms

Unlike the multitude of mechanisms available in the single-stage setting (Section 3), we are hit with an immediate roadblock in the two-stage case.

Theorem 1. *For any values of $N \geq G \geq 1$ and $T \in (\frac{1}{2}, 1)$, there is no mechanism that is incentive compatible.*

In order to circumvent this impossibility theorem[4], we will make a mild relaxation to our requirements.

[4]The theorem considers $T > \frac{1}{2}$. When $T = \frac{1}{2}$, the mechanism in the proof of Theorem 2 below (with the associated parameter $\xi = 0$) is incentive compatible.

4.2. Relax: Incentive compatibility with margins

Given the impossibility result of Theorem 1, in this section, we make a relaxation to the requirements outlined earlier in Section 2. Recall that the aforementioned setting requires that in the first stage, for every question $i \in [N]$, the worker must be incentivized to

$$\text{select} \begin{cases} \text{option "A"} & \text{if } p_{A,i} > \frac{1}{2} \\ \text{option "B"} & \text{if } p_{A,i} < \frac{1}{2}. \end{cases}$$

We relax this requirement as: in the first stage, for every question $i \in [N]$, the worker must be incentivized to

$$\text{select} \begin{cases} \text{option "A"} & \text{if } p_{A,i} > \frac{1}{2} + \xi \\ \text{option "B"} & \text{if } p_{A,i} < \frac{1}{2} - \xi, \end{cases}$$

for some parameter $\xi > 0$ whose value will be specified later. Thus, the incentivization for the first stage is changed from a hard threshold at $\frac{1}{2}$ to an interval between $\frac{1}{2} - \xi$ and $\frac{1}{2} + \xi$. The new formulation does not impose any requirements in the first stage when the confidence of the worker is in the range $[\frac{1}{2} - \xi, \frac{1}{2} + \xi]$. The incentivization requirement in the second stage remains the same as before.

It turns out that with this relaxation, perhaps surprisingly, for every value of $\xi > 0$ there exist infinitely many incentive compatible mechanisms.

Theorem 2. *For every value of $N \geq G$, $T \in [\frac{1}{2}, 1)$ and $\xi > 0$, there exists a mechanism that is incentive compatible. Moreover, there exist infinitely many mechanisms and the number of degrees of freedom in choosing any mechanism grows exponentially in G.*

The proof of Theorem 2 is constructive, that is, it provides explicit constructions of incentive-compatible mechanisms for every value of ξ.

The parameter ξ represents the amount by which a mechanism is allowed to slack as compared to the guarantees required in Section 2. Consequently, we would like to keep the value of ξ small. But Theorem 2 guarantees the existence of incentive compatible mechanisms for any positive value of ξ, and furthermore, points to the existence of infinitely many mechanisms. This result thus raises the following two questions:

- What value of ξ should be chosen?
- For the chosen ξ, what mechanism should be used?

Given that the proof of Theorem 2 constructs an explicit class of mechanisms for use, one may then be tempted to simply pick an arbitrary value of ξ and an arbitrary mechanism from that class. In this paper, however, we will take a principled approach towards this choice.

4.3. No-free-lunch axiom and a unique mechanism

In this section, we identify a simple and naturally desirable condition for any mechanism, that will help us answer the two questions raised above. Specifically, we impose the following requirement on the payment mechanism, which we term the 'no-free-lunch' axiom.

Definition 1 (No-free-lunch axiom)**.** *If all the answers (in the gold standard) given by a worker are either wrong or copied then the worker should get a zero payment, unless all answers given by the worker are correct. More formally, we require* $f(\mathbf{x}) = 0 \quad \forall\ \mathbf{x} \in \{-\mathfrak{M}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{C}\}^G$.

The axiom is quite intuitive since if a worker gives only wrong answers or copies them from the reference, then these answers do not provide any new information to the mechanism designer.[5]

The no-free-lunch axiom stated above is a variant of the no-free-lunch axioms for other settings proposed in Shah & Zhou (2015); Shah et al. (2015). It is important to note that if the "zero payment" appears harsh, then one can replace the "zero" with any fixed positive value and all the results of this paper will continue to hold.

Given the natural requirement of the no-free-lunch axiom, in what follows, we will investigate the effects of this requirement under our self-correction setting.

Theorem 3. *For any values of* $N \geq G \geq 1$*, and any* $T \in \left[\frac{1}{2}, 1\right)$*, it is impossible to construct an incentive compatible mechanism satisfying the no-free-lunch axiom if* $\xi < \xi_{\min}$*, where* $\xi_{\min} \in \left(0, \frac{1}{2}\right)$ *is given by*

$$\xi_{\min} = \begin{cases} \frac{1}{2}\frac{1-T}{1+T} & \text{if } T \leq \frac{1}{\sqrt{2}} \\ \frac{1}{2}\left((2-T) - \sqrt{(5-T)(1-T)}\right) & \text{if } T \geq \frac{1}{\sqrt{2}} \end{cases}.$$

Theorem 3 thus prohibits the choice of any ξ below $\xi_{\min}$.

We now show that a slack of $\xi_{\min}$ is indeed feasible, i.e., it allows for incentive compatible mechanism(s) satisfying no-free-lunch. This helps *answer our first question on how to choose ξ: it is desirable to choose the smallest permissible value of the slack parameter, which turns out to be $\xi_{\min}$.* The rest of this section thus considers $\xi = \xi_{\min}$.

Consider the payment mechanism given in Algorithm 1.

As the following theorem shows, the proposed algorithm indeed works as desired.

Theorem 4. *For any choice of* $N \geq G \geq 1$*,* $T \in \left[\frac{1}{2}, 1\right)$ *and* $\xi = \xi_{\min}$*, the mechanism of Algorithm 1 satisfies the no-free-lunch axiom and is incentive compatible.*

[5]The exception of the case where all answers are correct is discussed subsequently in Section 4.4.

Algorithm 1 Incentive mechanism for self-correction

- Define function $\alpha : \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\} \to \mathbb{R}_+$ as $\alpha(+\mathfrak{M}) = 1$, $\alpha(-\mathfrak{M}) = 0$, $\alpha(+\mathfrak{R}) = \frac{\frac{1}{2} - \xi_{\min}}{1-T}$, $\alpha(-\mathfrak{R}) = 0$, $\alpha(+\mathfrak{C}) = \frac{\frac{1}{2} - \xi_{\min}}{T}$ and $\alpha(-\mathfrak{C}) = 0$.
- If $(x_1, \ldots, x_G) \in \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G$ are the evaluations of the answers to the G questions in the gold standard, then the payment is

$$\text{Payment}(x_1, \ldots, x_G) = \kappa \prod_{i=1}^{G} \alpha(x_i)$$

where $\kappa = \mu \left(\max\left\{1, \frac{\frac{1}{2} - \xi_{\min}}{1-T}\right\}\right)^{-G}$.

It turns out that this mechanism is unique in the following sense.

Theorem 5. *For any* $N \geq G \geq 1$*,* $T \in \left[\frac{1}{2}, 1\right)$ *and* $\xi = \xi_{\min}$*, there is only one incentive-compatible mechanism satisfying the no-free-lunch axiom, and that is the mechanism of Algorithm 1.*

The uniqueness result of Theorem 5 thus *answers our second question about deciding which mechanism to choose.*

4.4. No stronger than no-free-lunch

The reader may have wondered about the "unless" clause in the definition of the no-free-lunch axiom (Definition 1). This section will investigate the implications of removing that clause. To this end, let us define a marginally stronger version of the no-free-lunch axiom.

Definition 2 (Strong no-free-lunch)**.** *If all the answers (in the gold standard) given by a worker are either wrong or copied, i.e, when the worker gives no correct answer on her own, then the worker should get a zero payment. More formally, we require* $f(\mathbf{x}) = 0\ \forall\ \mathbf{x} \in \{-\mathfrak{M}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^N$.

Intuitively, if a worker's answers are all either wrong or simply copied then she is not contributing any new information. The strong no-free-lunch axiom is precisely the no-free-lunch axiom but without the 'unless' clause. The following theorem investigates this stronger requirement.

Theorem 6. *For any choice of* $N \geq G \geq 1$*,* $T \in \left[\frac{1}{2}, 1\right)$ *and* $\xi \in [0, \frac{1}{2})$*, there is no incentive-compatible mechanism satisfying the strong no-free-lunch condition.*

The result of this theorem thus justifies the inclusion of the 'unless' clause in the no-free-lunch axiom.

5. Numerical Experiments

In Section 4 we analytically proved the working and the optimality of our proposed mechanism, Algorithm 1, for the

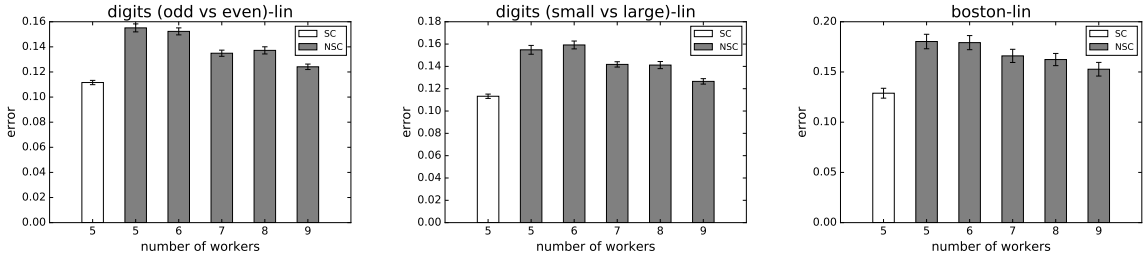

Figure 2: Error incurred by SVM with a linear kernel under the self-correction (SC) setting with 5 workers, compared to the error incurred under the standard setting with no self correction (NSC) with 5 to 9 workers.

two-stage setting. In this section we return to our primary hypothesis of the benefits of the two-stage self-correction setting, and via extensive numerical experiments, investigate the possible benefits of using a two-stage setting as compared to the standard one-stage setting without any self-correction. Such an examination is worthwhile since while the second stage would help to eliminate inadvertent errors and improve the quality of the data, it would also require each worker to spend more time on the task. In other words, for a fixed budget (under a fixed expected hourly wage), our two-stage setting trades off cleaner data with allowing for a slightly smaller number of workers. It turns out, as we will see below, that in machine learning systems that use crowdsourcing for labeled data, the self-correction setting results in a significant reduction in the end-to-end error rates as compared to the standard single-stage settings employed today.

Data We consider the labeling of the following two popular data sets: (a) UCI digits dataset (Lichman, 2013): Contains images of handwritten numeric digits from 0 to 9. We investigate two binary classification versions of this dataset: odd vs. even digits, and small values $0-4$ vs. large values $5-9$; (b) Boston housing dataset (Harrison & Rubinfeld; 1978): Contains information regarding housing in the area of Boston, USA. The binary classification problem is to predict whether the price of a house is greater than a certain value.

We then simulate the crowdsourced labeling procedure in the following manner. The number of workers hired in the two-stage self-correction setting and the standard single-stage setting may be different. In our simulations, the collection of workers are associated to a first-stage reliability parameter p and a second-stage improvement parameter q as follows. The workers have a reliability of p in the first stage, meaning that each worker, for each question, makes an error independently with probability $(1-p)$ in the first stage. In the second stage, the quality is assumed to improve by q due to self correction by the workers, that is, the reliability is $(p+q)$ at the end of the second stage. In the simulations, we investigate the effects of different values of p, q and the number of workers.

Machine learning algorithms We study the performance of two popular binary classification algorithms:

- support vector machine (SVM) with a linear kernel, and
- SVM with a radial basis function (RBF) kernel.

We perform the following operations separately for each of the two classification algorithms, for each of the three classification problems mentioned above, and for the two settings of with and without self-correction. The data is split into two equal halves, which are used for training and testing respectively. The labels for the training data are noisy, where the noise comes from the crowdsourced labelling described above. The test set is used to measure and compare the final performance of the classification algorithms, and is hence free of errors. The hyperparameters of the algorithms, including the regularization parameter and the kernel bandwidth, are chosen via 5-fold cross-validation.

Results In each of the plots to follow, each data point is averaged over 50 runs. We plot the results for SVM with linear kernel here in the main text, and noting that the results for the RBF kernel are almost identical to that for the linear kernel, we relegate the plots of SVM with RBF kernel to Appendix B.

We first investigate how the error under the self-correction setting compares with the error in the setting with no self-correction, for various amounts of redundancies in the task. More specifically, we fix the number of workers per question as 5 in the self-correction setting and vary the number of workers per question from 5 to 9 in the setting with no self-correction. For this set of experiments, we set $p = 0.6$ and $q = 0.15$. In each case, we use the aggregate of the worker's answers as training data for the two classification algorithms described earlier. Figure 2 plots the amount of error incurred by the SVM algorithm with the linear kernel. In each case, the performance of the algorithms when supplied the data from the self-correction setting outperforms the performance when data comes from the standard crowdsourcing setup with no self-correction. It is notewor-

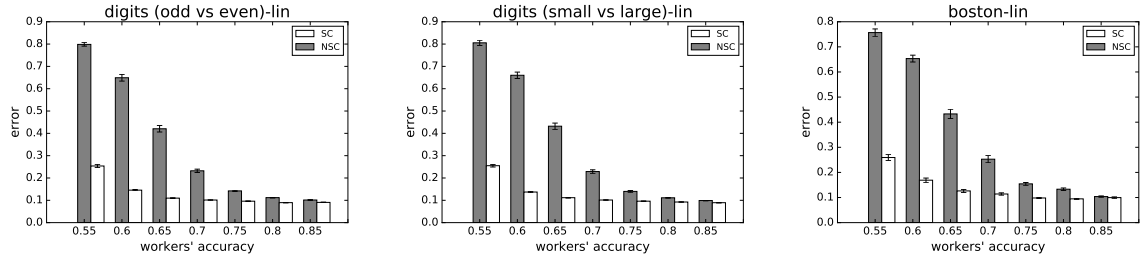

Figure 3: Error incurred by SVM with a linear kernel for different reliabilities (p) of the worker in the first stage. The no-self-correction (NSC) setting has 7 workers whereas the self-correction (SC) setting has only 5 workers.

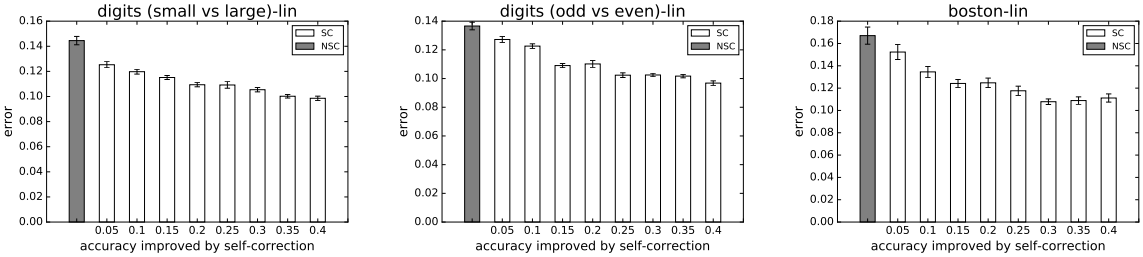

Figure 4: Error incurred by SVM with a linear kernel for different values of the improvement in accuracy (q) via self-correction. The no-self-correction (NSC) setting has 7 workers whereas the self-correction (SC) setting has only 5 workers.

thy that the self-correction setting shows an improved performance even when the number of workers in the standard setup is almost twice that in the self-correction setup.

Next, we compare the performance of the two settings for various values of the first-stage reliability p of the worker. To this end, we consider the self-correction setting with 5 workers per question and the setting with no self-correction having 7 workers per question. We vary the reliability p of each worker in the range 0.55 to 0.85, fixing $q = 0.15$. As before, the data obtained is employed to train an SVM algorithm with a linear kernel for the three datasets. The accuracy of the algorithm is shown in Figure 3. Observe that the self-correction setting consistently outperforms the standard setting with no self-correction. The improvement is particularly striking in high-noise conditions (i.e., when p is small).

Finally, we now compare the performance in these two settings when the second-stage accuracy parameter q is varied, keeping p fixed. In particular, we again consider the self-correction setting with 5 workers per question and the setting with no self-correction having 7 workers per question; we set $p = 0.6$ and vary q from 0.05 to 0.4. We observe (Figure 4) that even if the second stage offers marginal improvements (such as $q \leq 0.1$), we can still get significant gains from the two-stage setting as compared to a one-stage setting, despite the one stage setting having more workers.

All in all, the numerical experiments indicate significant improvements in the quality of the labels due to self-correction, and a corresponding increase in the accuracy of machine learning algorithms. Such improvements arise even in cases when the amount of self-correction may be quite small and when the setting without self correction has more workers than the setting with self correction.

6. Discussions

In this paper we proposed a two-stage setting for self-correction to overcome the various inadvertent errors that are observed widely in crowdsourcing. We showed the potential of such a self-correction setting via numerical experiments where we observed significant gains in the end-to-end performance of machine learning algorithms based on crowdsourced data. On the theoretical front, we investigated incentive mechanisms to ensure that workers report truthfully in both stages. (The modeling choices underlying the theory are discussed further in Appendix A.)

Our work leads to a number of interesting directions for future work. We addressed crowdsourcing tasks with binary-choice problems – such tasks are very popular in practice and quite challenging to analyze theoretically. We hope to use our results as building blocks for addressing more complex tasks. Second, our numerical experiments reveal that our proposed two-stage setup can offer significant gains as compared to standard single-stage setups. It remains to evaluate these mechanisms in real crowdsourcing platforms, which however, will necessitate sufficient training and exposure of the workers to this new setting.

References

Aggarwal, V., , Srikant, S., and Shashidhar, V. Principles for using machine learning in the assessment of open response items: Programming assessment as a case study. In *NIPS Workshop on Data Driven Education*, 2013.

Alonso, O. and Mizzaro, S. Can we get rid of TREC assessors? Using Mechanical Turk for relevance assessment. In *SIGIR Workshop on the Future of IR Evaluation*, pp. 15–16, 2009.

Ayewah, N. and Pugh, W. Using checklists to review static analysis warnings. In *Workshop on Defects in Large Software Systems (at ACM ISSTA)*, 2009.

Bearden, W. and Rose, R. Attention to social comparison information: An individual difference factor affecting consumer conformity. *Journal of Consumer Research*, 1990.

Bernstein, M. S., Little, G., Miller, R. C., Hartmann, B., Ackerman, M. S., Karger, D. R., Crowell, D., and Panovich, K. Soylent: a word processor with a crowd inside. In *UIST*, 2010.

Brier, G. W. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., and Hullender, G. Learning to rank using gradient descent. In *ICML*, 2005.

Carlson, A., Betteridge, J., Wang, R.and Hruschka Jr, E., and Mitchell, T. Coupled semi-supervised learning for information extraction. In *WSDM*, 2010.

Chros, O. and Sundell, S. Digitalkoot: Making old archives accessible using crowdsourcing. In *Human Computation*, 2011.

Clayton, M. J. Delphi: a technique to harness expert opinion for critical decision-making tasks in education. *Educational Psychology*, 17(4):373–386, 1997.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *IEEE CVPR*, 2009.

Dijkstra, E. W. On the foolishness of "natural language programming". In *Program Construction*. 1979.

Dow, S., Kulkarni, A., Bunge, B., Nguyen, T., Klemmer, S., and Hartmann, B. Shepherding the crowd: managing and providing feedback to crowd workers. In *CHI*, 2011.

Fleurbaay, E. and Eveleigh, A. Crowdsourcing: Prone to error? In *International Council on Archives*, 2012.

Gao, H., Barbier, G., and Goolsby, R. Harnessing the crowdsourcing power of social media for disaster relief. *IEEE Intelligent Systems*, 2011.

Gerber, A., Green, D., and Larimer, C. Social pressure and voter turnout: Evidence from a large-scale field experiment. *American Political Science Review*, 2008.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 2007.

Gu, L. Crowdsourcing for complex tasks: How to ensure quality output, September 2015. http://engineering.godaddy.com/crowdsourcing-for-complex-tasks-how-to- ensure-quality-output/.

Gupta, A., Thies, W., Cutrell, E., and Balakrishnan, R. mclerk: enabling mobile crowdsourcing in developing regions. In *SIGCHI*, 2012.

Haas, D., Ansel, J., Gu, L., and Marcus, A. Argonaut: macrotask crowdsourcing for complex data processing. *VLDB*, 2015.

Hara, K., Le, V., and Froehlich, J. Combining crowdsourcing and google street view to identify street-level accessibility problems. In *SIGCHI*, 2013.

Harrison, D. and Rubinfeld, D. Boston housing dataset. `http://www.cs.toronto.edu/~delve/data/boston/bostonDetail.html`. Retrieved: October 13, 2015.

Harrison, D. and Rubinfeld, D. L. Hedonic housing prices and the demand for clean air. *Journal of environmental economics and management*, 5(1):81–102, 1978.

Hasson, F., Keeney, S., and McKenna, H. Research guidelines for the delphi survey technique. *Journal of advanced Nursing*, 32(4):1008–1015, 2000.

Jiang, H. and Matsubara, S. Improving crowdsourcing efficiency based on division strategy. In *IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology*, 2012.

Kahneman, D. and Frederick, S. Representativeness revisited: Attribute substitution in intuitive judgment. *Heuristics and biases: The psychology of intuitive judgment*, 49, 2002.

Karger, D. R., Oh, S., and Shah, D. Iterative learning for reliable crowdsourcing systems. In *NIPS*, 2011.

Kazai, G. In search of quality in crowdsourcing for search engine evaluation. In *Advances in information retrieval*, 2011.

Kazai, G., Kamps, J., Koolen, M., and Milic-Frayling, N. Crowdsourcing for book search evaluation: impact of HIT design on comparative system ranking. In *ACM SIGIR*, 2011.

Kerr, N. and Tindale, R. Group performance and decision making. *Annual Review of Psychology*, 2004.

Khatib et al., F. Crystal structure of a monomeric retroviral protease solved by protein folding game players. *Nature structural & molecular biology*, 2011.

Kozlowski, S. W. and Ilgen, D. R. Enhancing the effectiveness of work groups and teams. *Psychological science in the public interest*, 7(3):77–124, 2006.

Krosnick, J. A. Response strategies for coping with the cognitive demands of attitude measures in surveys. *Applied cognitive psychology*, 5(3):213–236, 1991.

Lang, A. and Rio-Ross, J. Using Amazon Mechanical Turk to transcribe historical handwritten documents. *The Code4Lib Journal*, 2011.

Lasecki, W., Miller, C., and Bigham, J. Warping time for more effective real-time crowdsourcing. In *SIGCHI*, 2013.

Levin, I., Schneider, S., and Gaeth, G. All frames are not created equal: A typology and critical analysis of framing effects. *Organizational behavior and human decision processes*, 1998.

Lichman, M. UCI digits dataset (UCI machine learning repository). `http://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits`, 2013. Retrieved: October 13, 2015.

Liu, Q., Peng, J., and Ihler, A. Variational inference for crowdsourcing. In *NIPS*, 2012a.

Liu, X., Lu, M., Ooi, B. C., Shen, Y., Wu, S., and Zhang, M. Cdas: a crowdsourcing data analytics system. In *VLDB*, 2012b.

Miller, B. and Steyvers, M. The wisdom of crowds with communication. In *Conference of the Cognitive Science Society*, 2011.

Miller, N., Resnick, P., and Zeckhauser, R. Eliciting informative feedback: The peer-prediction method. *Management Science*, 51(9):1359–1373, 2005.

Muchnik, L., Aral, S., and Taylor, S. J. Social influence bias: A randomized experiment. *Science*, 2013.

Prelec, D. A Bayesian truth serum for subjective data. *Science*, 306(5695):462–466, 2004.

Ranade, G. and Varshney, L. To crowdsource or not to crowdsource. In *HCOMP workshop at AAAI*, 2012.

Raykar, V., Yu, S., Zhao, L., Valadez, G., Florin, C., Bogoni, L., and Moy, L. Learning from crowds. *JMLR*, 2010.

Rowe, G. and Wright, G. The delphi technique as a forecasting tool: issues and analysis. *International Journal of Forecasting*, 15(4):353–375, 1999.

Savage, L. J. Elicitation of personal probabilities and expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971.

Shah, N. B. and Zhou, D. Double or nothing: Multiplicative incentive mechanisms for crowdsourcing. In *NIPS*, 2015.

Shah, N. B., Zhou, D., and Peres, Y. Approval voting and incentives in crowdsourcing. In *ICML*, 2015.

Su, H., Deng, J., and Fei-Fei, L. Crowdsourcing annotations for visual object detection. In *Workshops at AAAI*, 2012.

Tesser, A., Campbell, J., and Mickler, S. The role of social pressure, attention to the stimulus, and selfdoubt in conformity. *European Journal of Social Psychology*, 1983.

The New Yorker. Why smart people are stupid. `http://www.newyorker.com/tech/frontal-cortex/why-smart-people-are-stupid`, June 2012.

Tversky, A. and Kahneman, D. Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157):1124–1131, 1974.

Von Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M. reCAPTCHA: Human-based character recognition via web security measures. *Science*, 2008.

Vuurens, J., de Vries, A. P., and Eickhoff, C. How much spam can you take? An analysis of crowdsourcing results to increase accuracy. In *SIGIR Workshop on Crowdsourcing for Information Retrieval*, 2011.

Wais, P., Lingamneni, S., Cook, D., Fennell, J., Goldenberg, B., Lubarov, D., Marin, D., and Simons, H. Towards building a high-quality workforce with Mechanical Turk. *NIPS workshop on computational social science and the wisdom of crowds*, 2010.

Zhou, D., Liu, Q., Platt, J. C., Meek, C., and Shah, N. B. Regularized minimax conditional entropy for crowdsourcing. *arXiv preprint arXiv:1503.07240*, 2015.

A. Discussion on modeling assumptions

We begin this section with a discussion on our rationale behind the modelling assumptions made in this paper.

- Workers maximize their expected payment: In the literature on game theory, this assumption is a standard, albeit highly debated, assumption. We argue that this assumption is quite reasonable in our setting. In standard labeling tasks in crowdsourcing, workers typically spend only about a few minutes for each task, and participate in hundreds of tasks every week. As a consequence of the law of large numbers, their earning per hour quickly converges to its expected value. Assuming that workers aim to maximize their hourly wages, the expected payment is the correct quantity to consider.

- Cost-of-effort: This choice of not explicitly modelling a "cost-for-effort" of each worker was guided by the principle of Occam's razor. The cost-for-effort is a highly complex quantity and is not very well understood. (For instance, what is the monetary cost for the effort in writing or reading this paper?). Hence, instead, we consider the parameter μ to be a surrogate for the cost-for-effort: the parameter must be scaled in a fashion that ensures a expected fair pay to any worker who does a reasonable job.

- Workers perfectly know their beliefs: We admit this is a mathematical idealization, but is somewhat necessary to enable a principled game-theoretic analysis of the setting, and is quite a standard assumption in the literature.

- Non-negative payments: To the best of our knowledge, all crowdsourcing platforms today (such as Amazon mechanical turk, Clickworker, Mobileworks, etc.) require the payment to be non-negative.

- Rational workers: We do not require workers to be rational; rationality is a standard game theoretic assumption employed to guard against the worst case of workers exploiting the payment mechanism. From a practical standpoint, workers exposed to any mechanism for long enough durations may eventually "rationalize" and identify loopholes (if any) in the mechanism.

B. Simulations for SVM with RBF Kernel

In this section, we plot the results of the simulations for the SVM algorithm with the RBF kernel. To begin, Figure 5 plots the error incurred when the number of workers in the setting with no self correction is varied from 5 to 9, keeping the number of workers in the setting with self correction at 5. Next, Figure 6 compares the error in the two settings when q is fixed at 0.15 for various values of parameter p. Finally, Figure 7 compares the error in the two settings when p is fixed at 0.6 for various values of parameter q. We observe that as in the case of the linear kernel studied earlier, the two-stage setting with self correction offers significant advantages over the single-stage setting with no self correction.

C. Proofs

In this section, we will present the proofs of the various theoretical claims made in the main text. We begin with the claim for the single-stage setting followed by the proofs of the main two-stage setting considered in the paper. Towards the latter, in Section C.2, we introduce some notation and a lemma that will subsequently be used in several other proofs.

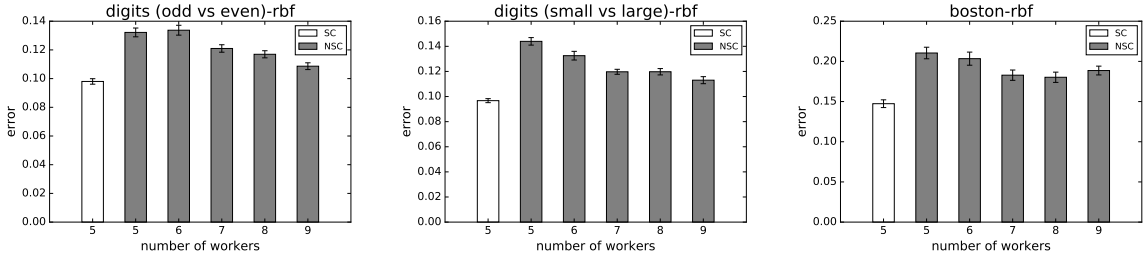

Figure 5: Error incurred by SVM with an RBF kernel under the self-correction (SC) setting with 5 workers, compared to the error incurred under the standard setting with no self correction (NSC) with 5 to 9 workers.

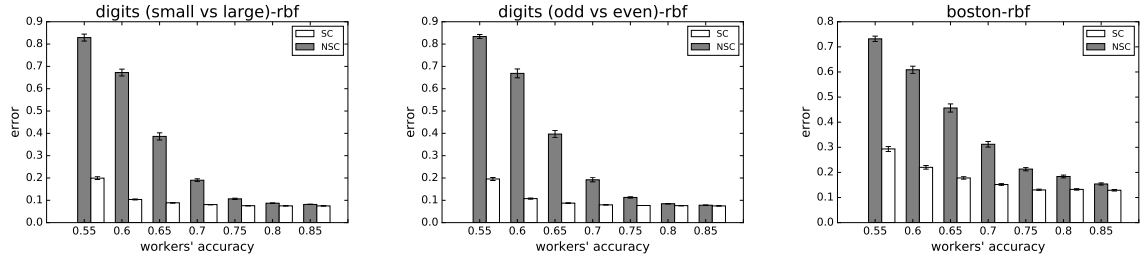

Figure 6: Error incurred by SVM with an RBF kernel for different reliabilities (p) of the worker in the first stage. The no-self-correction (NSC) setting has 7 workers whereas the self-correction (SC) setting has only 5 workers.

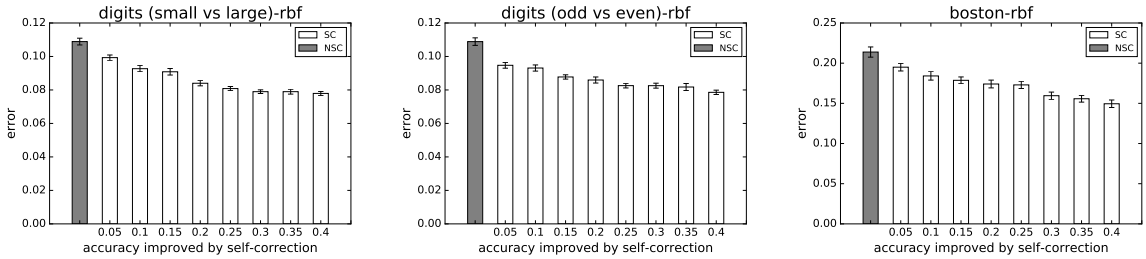

Figure 7: Error incurred by SVM with an RBF kernel for different values of the improvement in accuracy (q) via self-correction. The no-self-correction (NSC) setting has 7 workers whereas the self-correction (SC) setting has only 5 workers.

C.1. Proof of Proposition 1: One stage is easy

The proof is straightforward, but is included for completeness. Let p_A and p_B $(= 1 - p_A)$ be the worker's subjective probabilities of A or B respectively being correct. If the worker selects A then her expected reward is

$$R_A := p_A M_+ + p_B M_- \ .$$

On the other hand, if the worker selects B then her expected reward is

$$R_B := p_B M_+ + p_A M_- \ .$$

Noting that $p_A + p_B = 1$, one can easily verify that

$$M_+ > M_- \Rightarrow R_A \underset{p_A > \frac{1}{2} > p_B}{\overset{p_A < \frac{1}{2} < p_B}{\lesseqgtr}} R_B$$

which implies incentive compatibility.

C.2. Necessary and sufficient condition for incentive compatibility when $N = G = 1$

In this section, we establish a key result on necessary and sufficient conditions for incentive compatibility when $N = G = 1$, which will be useful in subsequent proofs. The reader interested in only the proof of Theorem 1 may directly read that proof in the next subsection without loss in continuity.

Under the special case of $N = G = 1$, any mechanism $f : \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\} \to [0, \mu]$ can be defined using six values in the interval $[0, \mu]$, namely $M_+ := f(+\mathfrak{M})$, $M_- := f(-\mathfrak{M})$, $R_+ := f(+\mathfrak{R})$, $R_- := f(-\mathfrak{R})$, $C_+ := f(+\mathfrak{C})$ and $C_- := f(-\mathfrak{C})$.

We will also use the following two functions $R_R, R_C : [0, 1] \to [0, \mu]$:

$$R_R(p') := p' R_+ + (1 - p') R_-, \tag{2a}$$

$$R_C(p') := (1 - p') C_+ + p' C_- \ . \tag{2b}$$

In words, $R_R(p')$ and $R_C(p')$ represent the expected reward of a worker (from her point of view) who has a belief of p' in the option she chose in the first stage and who either retains her answer or copies the reference answer respectively. In this section, since we consider only one question, we will drop the subscripts "i" in the notation of the worker's beliefs.

The following lemma establishes necessary and sufficient conditions for incentive compatibility.

Lemma 1. *When $N = G = 1$, a necessary and sufficient condition for a mechanism to be incentive compatible is that it satisfies the following conditions:*

$$(1-T)R_+ + TR_- = TC_+ + (1-T)C_-, \tag{3a}$$

$$\max\left\{C_+, \frac{C_+ + C_-}{2} + 2\xi\frac{C_+ - C_-}{2}, \frac{R_+ + R_-}{2} - 2\xi\frac{R_+ - R_-}{2}\right\} \leq \frac{M_+ + M_-}{2} + 2\xi\frac{M_+ - M_-}{2}, \tag{3b}$$

$$\frac{M_+ + M_-}{2} - 2\xi\frac{M_+ - M_-}{2} \leq \min\left\{C_+, TC_+ + (1-T)C_-, \max\{\frac{C_+ + C_-}{2} + 2\xi\frac{C_+ - C_-}{2}, \frac{R_+ + R_-}{2} - 2\xi\frac{R_+ - R_-}{2}\}\right\}, \tag{3c}$$

$$M_+ > M_-, \qquad R_+ > C_-, \qquad C_+ > R_-, \qquad R_+ > M_-. \tag{3d}$$

The remainder of this subsection is devoted to the proof of this lemma.

Proof of Lemma 1 We will prove this lemma by first identifying the basic conditions necessary and sufficient for incentive compatibility, and then showing the equivalence of the conditions to those stated in the lemma.

Recall the conditions for incentive compatibility in the second stage (Section 2.3). One can verify that equivalently, necessary and sufficient conditions for incentive compatibility in the second stage are $R_R(1-T) = R_C(1-T)$ and $R_+ > C_-,\ C_+ > R_-$. The first condition is identical to (3a).

For the first stage, by definition, a necessary and sufficient condition for incentive compatibility is

$$\begin{aligned} &q_A(p_A M_+ + p_B M_-) + q_B \max\{R_R(p'_{A|B}), R_C(p'_{A|B})\} \\ &\underset{p_A > \frac{1}{2}+\xi}{\overset{p_A < \frac{1}{2}-\xi}{\lessgtr}} q_B(p_B M_+ + p_A M_-) + q_A \max\{R_R(p'_{B|A}), R_C(p'_{B|A})\}, \end{aligned} \tag{4}$$

for all $p_A \in [0,1], p'_{A|B} \in [0, p_A], p'_{B|A} \in [0, p_A], p'_{B|A} \in [0, p_B]$, and $q_A \in [0,1]$.

Setting $p'_{B|A} = p'_{A|B} = 0$ and $q_A = q_B = \frac{1}{2}$ in (4) results in the necessity of the condition $M_+ > M_-$. Let us now investigate the conditions (3b) and (3c).

Consider the case of $p_A < \frac{1}{2} - \xi$. Here, the worst case is when the left hand side of (4) is maximized and the right hand side is minimized. Satisfying (4) when $p_A < \frac{1}{2} - \xi$ is thus equivalent to satisfying the inequality

$$\begin{aligned} &q_A(p_A M_+ + p_B M_-) + q_B \max_{p'_{A|B} \in [0, p_A]} \max\{R_R(p'_{A|B}), R_C(p'_{A|B})\} \\ &< q_B(p_B M_+ + p_A M_-) + q_A \min_{p'_{B|A} \in [0, p_B]} \max\{R_R(p'_{B|A}), R_C(p'_{B|A})\}. \end{aligned} \tag{5}$$

Recall that $q_A = 1 - q_B$. Observe that the inequality (5) is linear in q_A. As a result, a necessary and sufficient for (5) to be satisfied for all values of $q_A \in [0,1]$ is that the inequality (5) is satisfied for the two extreme values of q_A, namely $q_A \in \{0,1\}$. Setting $q_A = 0$ in (5) gives

$$\max_{p'_{A|B} \in [0, p_A]} \max\{R_R(p'_{A|B}), R_C(p'_{A|B})\} < (p_B M_+ + p_A M_-). \tag{6}$$

The 'maximum' term in the left hand side of (6) is a maximum over two linear functions, and hence the term is maximized when $p'_{A|B}$ is either 0 or p_A. Thus (6) reduces to

$$\max\{R_-, C_+, R_R(p_A), R_C(p_A)\} < (p_B M_+ + p_A M_-),$$

for all $p_A \in [0, \frac{1}{2} - \xi)$. Using the condition $C_+ > R_-$ from (3d), we obtain the equivalent condition

$$\max\{C_+ - (p_B M_+ + p_A M_-), R_R(p_A) - (p_B M_+ + p_A M_-), R_C(p_A) - (p_B M_+ + p_A M_-)\} < 0. \tag{7}$$

Each of the three expressions in the maximum on the left hand side of (7) are linear expressions in terms of the variable p_A. Consequently, the maximum is attained at one of the end-points of the permitted values of p_A, that is, when $p_A = 0$ or when p_A approaches $\frac{1}{2} - \xi$. Substituting these two values of p_A into (7) yields the necessary and sufficient condition of (3b), for the setting of $p_A < \frac{1}{2} - \xi$ and $q_A = 0$.

Next we move to the case of $q_A = 1$. Setting $q_A = 1$ in (5) gives

$$(p_A M_+ + p_B M_-) < \min_{p'_{B|A} \in [0, p_B]} \max\{R_R(p'_{B|A}), R_C(p'_{B|A})\}. \tag{8}$$

The term "$\max\{R_R(p'_{B|A}), R_C(p'_{B|A})\}$" in the right hand side of (8) is a maximum over two linear functions, and hence the term is necessarily minimized in one of the following three cases: (i) At $R_R(p'_{B|A}) = R_C(p'_{B|A})$ if one of the two functions $R_R(p'_{B|A})$ and $R_C(p'_{B|A})$ is increasing and one decreasing in $p'_{B|A}$. As a consequence of (3a), the two functions are equal when $p'_{B|A} = 1 - T$. Note that this value of $p'_{B|A}$ is a valid value because $1 - T \leq \frac{1}{2} \leq p_B$. (ii) At $p'_{B|A} = 0$, which is a minimizer when both functions increase with an increase $p'_{B|A}$. (iii) At $p'_{B|A} = p_B$, which is a minimizer when both functions decrease with an increase $p'_{B|A}$. Putting the three cases together, we get the equivalent condition

$$(1 - p_B) M_+ + p_B M_- < \min\{C_+, TC_+ + (1 - T) C_-, \max\{R_R(p_B), R_C(p_B)\}\}, \tag{9}$$

for all $p_B \in [0, \frac{1}{2} - \xi)$. One can verify that due to linearity (in p_B) of the various constituents of (9), it is necessary and sufficient that the inequality (9) be satisfied for the extreme values of p_B. Setting $p_B = 1$ and $p_B = \frac{1}{2} + \xi$ and performing some algebraic simplifications yields the condition (3c).

The case of $p_B < \frac{1}{2} - \xi$ gives the same result by symmetry. This completes the proof of the necessity and sufficiency of (3) for incentive compatibility.

C.3. Proof of Theorem 1: Impossibility

We first prove the claimed impossibility result for the case of a single question $N = G = 1$. The proof for the case of $N = G = 1$ proceeds via a contradiction-based argument, and uses the notation of Section C.2.[6] Suppose there is an incentive compatible mechanism, i.e., there exist values of $M_+, M_-, R_+, R_-, C_+, C_-$ that ensure that in both stages the worker selects the answer she thinks is most likely to be correct.

Incentive compatibility then necessitates:

- Second stage:
  - if worker answered A in the first stage and reference answer was B:

$$R_R(p'_{A|B}) \underset{p'_{A|B} > 1-T}{\overset{p'_{A|B} < 1-T}{\lessgtr}} R_C(p'_{A|B}), \tag{10}$$

  - if worker answered B in the first stage and reference answer was A:

$$R_R(p'_{B|A}) \underset{p'_{B|A} > 1-T}{\overset{p'_{B|A} < 1-T}{\lessgtr}} R_C(p'_{B|A}). \tag{11}$$

- First stage:

$$\begin{aligned} &q_A(p_A M_+ + p_B M_-) + q_B \max\{R_R(p'_{A|B}), R_C(p'_{A|B})\} \\ &\underset{p_A > \frac{1}{2} > p_B}{\overset{p_A < \frac{1}{2} < p_B}{\lessgtr}} q_B(p_B M_+ + p_A M_-) + q_A \max\{R_R(p'_{B|A}), R_C(p'_{B|A})\}. \end{aligned} \tag{12}$$

[6] While one could use Lemma 1 to prove this result, we opt for a different proof here for its significantly greater simplicity.

We now show that the requirements (10), (11) and (12) cannot be met simultaneously. To this end, consider some value $p' \in [0, \frac{1}{2}]$, and consider a worker who has subjective probabilities $p_A = p_B = \frac{1}{2}$, $p'_{A|B} = p'_{B|A} = p' \leq \frac{1}{2}$, and $q_A \neq q_B$. Observe that both the left and right hand sides of (12) are continuous in (p_A, p_B). As a result, when $p_A = p_B = \frac{1}{2}$ we must have

$$\begin{aligned} &q_A(\frac{1}{2}M_+ + \frac{1}{2}M_-) + q_B \max\{R_R(p'_{A|B}), R_C(p'_{A|B})\} \\ &\quad = q_B(\frac{1}{2}M_+ + \frac{1}{2}M_-) + q_A \max\{R_R(p'_{B|A}), R_C(p'_{B|A})\}. \end{aligned}$$

Some simple algebraic manipulations yield

$$\frac{M_+ + M_-}{2} = \max\{R_R(p'), R_C(p')\}, \tag{13}$$

for every $p' \leq \frac{1}{2}$. In the two sets of inequalities (10) and (11), the left hand sides are greater than the right hand sides for certain values of $p' \leq \frac{1}{2}$, and vice versa for certain other values of $p' \leq \frac{1}{2}$, whenever $T > \frac{1}{2}$. It follows that the term $\max\{R_R(p'), R_C(p')\}$ in the right hand side of (13) must depend on the value of p' and cannot be a constant. On the other hand, the left hand side of (13) is a constant, independent of p'. This argument thus yields a contradiction.

Given that the worker cannot be incentivized for even one question, the impossibility easily extends to the more general case of $N \geq G \geq 1$ as follows. Assume that for questions $2, \ldots, N$, the worker is sure that the answer is option A in both stages, is sure that the reference answer will be option A, and the reference answer as well as the correct answer actually turn out to equal option A. In this setting, the incentivization requirements reduce to incentivizing the worker for only the first question, which is shown to be impossible in the proof for the $N = G = 1$ setting below.

C.4. Proof of Theorem 2: Many mechanisms for every slack

We begin with the case of $N = G = 1$ which will convey many of the key ideas of the proof. We will adopt the notation introduced in Section C.2. Let $M_+ = 1, M_- = 0, R_+ = 1, R_- = 0, C_+ = (1 - T), C_- = (1 - T)$. It is easy to verify that this choice satisfies the conditions (3a) and (3d). If these payments satisfy the inequalities (3b) and (3c), then we are done. If not then the values will result in the left hand side being greater than the right hand side in (3b) and/or (3c). In that case, compute the difference between the left and right hand sides of (3b) and (3c), and let $\zeta > 0$ denote the larger of the two values. Perform the following modifications to the values: $M_+ \to M_+ + \frac{\zeta+1}{\xi}$, $M_- \to M_-$, $R_+ \to R_+ + \frac{\zeta+1}{2\xi}$, $R_- \to R_- + \frac{\zeta+1}{2\xi}$, $C_+ \to C_+ + \frac{\zeta+1}{2\xi}$, and $C_- \to C_- + \frac{\zeta+1}{2\xi}$. At this point, we would like to remind the reader that $\zeta > 0$ and $\xi > 0$.

One can verify that with the changes described above, the payment values continue to satisfy the conditions (3a) and (3d). However, importantly, with these changes, the left hand side of (3b) increases by at most $\frac{\zeta+1}{2\xi}$ while the right hand side increases by $(1 + 2\xi)\frac{\zeta+1}{2\xi}$, and the left hand side of (3b) increases by $(1 - 2\xi)\frac{\zeta+1}{2\xi}$ while its right hand side increases by $\frac{\zeta+1}{2\xi}$. It follows that in both inequalities, the difference between the right and left hand sides increases by at least $(\zeta + 1)$. Thus with the updated values, both (3b) and (3c) are satisfied. Finally, scaling all payments by $\frac{\mu}{M_+}$ also ensures that the mechanism abides by the constraint of the maximum allowable payment. We have thus proved the existence of an incentive-compatible mechanism when $\xi > 0$, for the case of $N = G = 1$.

For the more general case of $N \geq G \geq 1$, consider a mechanism that first considers each gold standard question separately and allots a score equaling the payment that would have been made in the case of $N = G = 1$. The net payment across all questions is the sum of the scores across all questions (normalized by a positive factor to satisfy the budget constraint of μ). From the worker's point of view, due to linearity of expectation, the expected payment for any choice of answers is the sum of the expected scores for the N individual questions (normalized by a positive constant factor). Incentive compatibility of the individual scores for $N = G = 1$ implies incentive compatibility for the general mechanism as well.

Observe that in the proof above, we started out with one particular choice of the parameters M_+, M_-, R_+, R_-, C_+, C_- that satisfied (3a) and (3d). There are however infinitely many choices of these parameters that satisfy these two conditions. The rest of the proof for $G = 1$ above demonstrated a procedure to construct an incentive-compatible mechanism starting from any such choice. It is not hard to see that the set of resulting mechanisms also form an infinite set. When $G > 1$, one can choose separate mechanisms for each individual question and combine them in one of an exponentially large number of ways, e.g., multiplying or adding any of the mechanisms for the individual questions. The number of degrees of freedom thus grows exponentially in G.

C.5. Proof of Theorem 3: Minimum slack needed

We begin with the case of $N = G = 1$ which will convey many of the key ideas of the proof. We will adopt the notation introduced in Section C.2 for this case.

It is straightforward to see that when $N = G = 1$, a necessary and sufficient condition to satisfy the no-free-lunch axiom is that $M_- = R_- = C_- = 0$. Substituting these conditions in Lemma 1 gives that a necessary and sufficient condition under any ξ and T for the existence of an incentive-compatible mechanism satisfying the no-free-lunch axiom is

$$\frac{\frac{1}{2}-\xi}{T} \leq \frac{C_+}{M_+} \leq \frac{\frac{1}{2}+\xi}{T} \min\left\{\frac{1-T}{\frac{1}{2}-\xi}, T\right\} \tag{14a}$$

$$(1-T)R_+ = TC_+ > 0. \tag{14b}$$

Observe the following four properties of (14a): (i) the leftmost side strictly decreases with an increase in ξ while its rightmost side increases strictly, (ii) when $\xi = 0$, the leftmost side is strictly greater than its rightmost side (using the fact that $T < 1$), (iii) when $\xi = \frac{1}{2}$, the leftmost side is zero whereas the rightmost side is one, and (iv) both the leftmost and rightmost sides are continuous in ξ. It follows that the leftmost and rightmost sides of (14a) meet each other at exactly one point in $\xi \in (0, \frac{1}{2})$. Solving (14a) for ξ, with the inequalities are replaced by equalities, gives precisely the value denoted by $\xi_{\min}$ in the statement of the theorem. For any $\xi < \xi_{\min}$, the aforementioned arguments imply a violation of (14a).

Let us now consider the more general case of $N \geq G \geq 1$. Suppose there exists some value $\xi < \xi_{\min}$ for which there exists an incentive compatible mechanism satisfying the no-free-lunch axiom. Then we have

$$\frac{1}{\frac{1}{2}-\xi} = \frac{1}{\frac{1}{2}+\xi} \max\left\{\frac{\frac{1}{2}-\xi}{1-T}, \frac{1}{T}\right\} - \delta_{\xi,T}, \tag{15}$$

for some value $\delta_{\xi,T} > 0$ that depends on the values of ξ and T. We will now call upon the proof of Theorem 5 to complete our proof. The proof of Theorem 5 shows that under the no-free-lunch axiom, there is only one mechanism that can be incentive compatible when $\xi = \xi_{\min}$. In the proof of Theorem 5, the steps till Equation (16) are applicable to all values of $\xi \in (0, \frac{1}{2})$; ξ is set as $\xi_{\min}$ in (16) to obtain (17) and in subsequent steps. If $\xi_{\min}$ is replaced by ξ, then the inequality (20) becomes

$$\left(\frac{1}{2}+\xi\right)\bar{h}(-\mathfrak{M}) + \frac{\frac{1}{2}-\xi}{\frac{1}{2}+\xi} T\bar{h}(+\mathfrak{C})\delta_{\xi,T} \leq \frac{\left(\frac{1}{2}-\xi\right)^2}{\frac{1}{2}+\xi}\bar{h}(-\mathfrak{M}),$$

where $\bar{h}(-\mathfrak{M}) \geq 0$, $\bar{h}(+\mathfrak{C}) > 0$. One can see that when $\xi > 0$, a necessary condition for this inequality to be satisfied (and consequently for any mechanism to be incentive compatible) is to have $\delta_{\xi,T} = 0$. This assignment, in turn, necessitates $\xi = \xi_{\min}$ for existence of any incentive compatible mechanism, as claimed.

C.6. Proof of Theorem 4: The algorithm works

First consider the case of $N = G = 1$. One can verify that when $\xi = \xi_{\min}$, the proposed payment mechanism satisfies the necessary and sufficient conditions (3) derived earlier in Lemma 1.

For the case of $N \geq G \geq 1$, observe that the mechanism assigns a non-negative value to the worker for each question in the gold standard, and the final payment is the product of these values (scaled by a positive constant). We recall our assumption that the worker's beliefs are independent across questions. Consequently, in either stage, the net expected payment from the worker's point of view equals the product of the expected values for each question (where the value is 1 if the question is not in the gold standard). Since for every individual question, the expectation is maximized when the worker answers as desired, the overall expected payment is also maximized when the worker answers as desired. The mechanism is thus incentive compatible.

Finally, one can verify that the payment is always non-negative, and the maximum payment equals μ.

C.7. Proof of Theorem 5: One and only mechanism

Let us first consider the much simpler case of $N = G = 1$. Recall the necessary and sufficient conditions (3) for incentive compatibility with the no free lunch condition. When $\xi = \xi_{\min}$, the two inequalities in (14a) get tightly sandwiched and

transform into equalities. Thus, the parameters M_+, R_+ and C_+ now have a unique relation between them. Moreover, $M_- = R_- = C_- = 0$ are also fixed. Setting $\max\{M_+, R_+, C_+\} = \mu$ now fixes the entire mechanism to be identical to Algorithm 1.

We now move on to the case of general values of the parameters (N, G). We begin with two lemmas that derive properties that any mechanism must necessarily satisfy. The proofs of these two lemmas are provided at the end of this section. The first of the two lemmas applies to any incentive compatible mechanism, that may or may not satisfy no free lunch.

Lemma 2. *For any $i \in [G]$, any $(y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_G) \in \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^{G-1}$, any incentive compatible mechanism must satisfy*

$$\begin{aligned}(1-T)f(y_1, \ldots, y_{i-1}, +\mathfrak{R}, y_{i+1}, \ldots, y_G) + Tf(y_1, \ldots, y_{i-1}, -\mathfrak{R}, y_{i+1}, \ldots, y_G) \\ = Tf(y_1, \ldots, y_{i-1}, +\mathfrak{C}, y_{i+1}, \ldots, y_G) + (1-T)f(y_1, \ldots, y_{i-1}, -\mathfrak{C}, y_{i+1}, \ldots, y_G).\end{aligned}$$

The proof of Theorem 5 inducts on the number of entries in $\mathbf{y}$ that take values in the set $\{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}$. The hypothesis of this induction is that the payment mechanism must be of the form given in Algorithm 1 up to a constant positive scaling. The base case of $\mathbf{y} \in \{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{R}, +\mathfrak{C}\}^G$ is handled in Lemma 3 below.

Lemma 3. *Any incentive-compatible mechanism satisfying the no-free-lunch axiom must satisfy $f(\mathbf{y}) = 0 \ \forall \ \mathbf{y} \in \{-\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{R}, +\mathfrak{C}\}^G$.*

From Lemmas 2 and 3, we obtain the base case of the induction that the mechanism must be identical to that of Algorithm 1 whenever $\mathbf{y} \in \{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{R}, +\mathfrak{C}\}^G$.

Moving on, let us now suppose that the induction hypothesis is true whenever $\mathbf{y} \in \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{M}, +\mathfrak{R}, +\mathfrak{C}\}^G$ and $\sum_{i=1}^{G} \mathbf{1}\{y_i \in \{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}\} \geq G - \gamma + 1$, for some $\gamma \in [G]$. We now prove that the induction hypothesis remains true whenever $\mathbf{y} \in \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{M}, +\mathfrak{R}, +\mathfrak{C}\}^G$ and $\sum_{i=1}^{G} \mathbf{1}\{y_i \in \{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}\} = G - \gamma$.

Suppose that without loss of generality that $y_1, \ldots, y_{\gamma-1} \in i \in \{+\mathfrak{M}, -\mathfrak{M}\}$ and $y_{\gamma+1}, \ldots, y_G \in \{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}$. In the total set of N questions, suppose that for every $i \leq \gamma - 1$, we have $q_{A,i} = 1, p_{A,i} > \frac{1}{2} + \xi$, and for every $i \geq \gamma + 1$, we have $q_{A,i} = 0, p_{A,i} > \max\{\frac{1}{2} + \xi, T\}$. Suppose that for all questions $[N]\backslash\{\gamma\}$, the worker decides to act precisely as what the mechanism wishes her to do. Thus in the first stage, she will select option A for all questions $[N]\backslash\{\gamma\}$. Furthermore, the worker believes that questions $1, \ldots, \gamma - 1$ will surely match, whereas questions $\gamma + 1, \ldots, G$ will surely mismatch and go into the second stage.

Let $h : \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\} \to [0, \mu]$ be a function defined as follows: $h(y_\gamma)$ is the expected payment, from the point of view of the worker, conditioned on the γ^{th} question evaluating to y_γ. (Note that since $q_{A,i} \in \{0, 1\}$ for every $i \neq \gamma$, and since the evaluation of question γ is fixed at y_γ, the expected pay is identical in both stages.) The expectation is over the randomness in the choice of the gold standard questions as well as over the worker's uncertainty about the correctness of her answers to the remaining $N - 1$ questions. One can see that for any value of y_γ, the function $h(y_\gamma)$ is composed of a convex combination of two parts: the first part is for the case when the γ^{th} question is in the gold standard and the second part is when the γ^{th} question is not in the gold standard. Consequently, the first part depends on y_γ and the second part is independent of it. Letting $\bar{h}$ denote the first part, we can write $h(y_\gamma) = \theta\bar{h}(y_\gamma) + (1-\theta)c$ for some constants $c \geq 0$ and $\theta \in (0, 1)$.

The function $\bar{h}$ is a convex combination of the function f evaluated at various points. In particular, when $y_\gamma \in \{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}$, each component of this convex combination is the function f evaluated at a vector with at least $(G - \gamma + 1)$ of its entries taking values in the set $\{+\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}$. Hence applying Lemma 2 we get that $(1-T)\bar{h}(+\mathfrak{R}) + T\bar{h}(-\mathfrak{R}) = T\bar{h}(+\mathfrak{C}) + (1-T)\bar{h}(-\mathfrak{C})$. Furthermore, from our induction hypothesis above, we have $\bar{h}(y_\gamma) = 0$ when $y_\gamma \in \{-\mathfrak{R}, -\mathfrak{C}\}$. Consequently, we also have $\bar{h}(+\mathfrak{R}) = \frac{T}{1-T}\bar{h}(+\mathfrak{C})$.

Let $p_A, p_B = 1 - p_A, q_A, q_B = 1 - q_A, p'_{A|B}, p'_{B|A}$ be the confidences of the worker for question γ. In order to incentivize the worker appropriately for question γ in the first stage, it must be that

$$\begin{aligned}&q_A(p_A\bar{h}(+\mathfrak{M}) + p_B\bar{h}(-\mathfrak{M})) + q_B \max\{p'_{A|B}\bar{h}(+\mathfrak{R}), (1 - p'_{A|B})\bar{h}(+\mathfrak{C})\} \\ &\underset{p_A > \frac{1}{2} + \xi}{\overset{p_A < \frac{1}{2} - \xi}{\lessgtr}} q_B(p_B\bar{h}(+\mathfrak{M}) + p_A\bar{h}(-\mathfrak{M})) + \max\{p'_{B|A}\bar{h}(+\mathfrak{R}), (1 - p'_{B|A})\bar{h}(+\mathfrak{C})\}.\end{aligned}$$

Substituting $\bar{h}(+\mathfrak{R}) = \frac{T}{1-T}\bar{h}(+\mathfrak{C})$ we get

$$q_A(p_A\bar{h}(+\mathfrak{M}) + p_B\bar{h}(-\mathfrak{M})) + q_B \max\{p'_{A|B}\frac{T}{1-T}, (1-p'_{A|B})\}\bar{h}(+\mathfrak{C})$$
$$\underset{p_A>\frac{1}{2}+\xi}{\overset{p_A<\frac{1}{2}-\xi}{\lesseqgtr}} q_B(p_B\bar{h}(+\mathfrak{M}) + p_A\bar{h}(-\mathfrak{M})) + q_A \max\{p'_{B|A}\frac{T}{1-T}, (1-p'_{B|A})\}\bar{h}(+\mathfrak{C}) \ .$$

Let $p_A = \frac{1}{2} - \xi$. Setting $p'_{B|A} = 1 - T$ and allowing $p'_{A|B}$ to be 0 or p_A gives

$$q_A\left(\frac{1}{2}-\xi\right)\bar{h}(+\mathfrak{M}) + q_A\left(\frac{1}{2}+\xi\right)\bar{h}(-\mathfrak{M}) + q_B T \max\left\{\left(\frac{1}{2}-\xi\right)\frac{1}{1-T}, \frac{1}{T}\right\}\bar{h}(+\mathfrak{C})$$
$$\leq q_B\left(\frac{1}{2}+\xi\right)\bar{h}(+\mathfrak{M}) + q_B\left(\frac{1}{2}-\xi\right)\bar{h}(-\mathfrak{M}) + q_A T\bar{h}(+\mathfrak{C}).$$

From the definition of the minimum slack (Theorem 3), when $\xi = \xi_{\min}$, we have

$$\frac{\frac{1}{2}-\xi}{T} = \frac{\frac{1}{2}+\xi}{T}\min\left\{\frac{1-T}{\frac{1}{2}-\xi}, T\right\} \ , \tag{16}$$

and hence

$$q_A\left(\frac{1}{2}-\xi\right)\bar{h}(+\mathfrak{M}) + q_A\left(\frac{1}{2}+\xi\right)\bar{h}(-\mathfrak{M}) + q_B T\frac{\frac{1}{2}+\xi}{\frac{1}{2}-\xi}\bar{h}(+\mathfrak{C})$$
$$\leq q_B\left(\frac{1}{2}+\xi\right)\bar{h}(+\mathfrak{M}) + q_B\left(\frac{1}{2}-\xi\right)\bar{h}(-\mathfrak{M}) + q_A T\bar{h}(+\mathfrak{C}). \tag{17}$$

Setting $q_A = 1$ gives

$$\left(\frac{1}{2}-\xi\right)\bar{h}(+\mathfrak{M}) + \left(\frac{1}{2}+\xi\right)\bar{h}(-\mathfrak{M}) \leq T\bar{h}(+\mathfrak{C}), \tag{18}$$

and setting $q_A = 0$ gives

$$T\frac{\frac{1}{2}+\xi}{\frac{1}{2}-\xi}\bar{h}(+\mathfrak{C}) \leq \left(\frac{1}{2}+\xi\right)\bar{h}(+\mathfrak{M}) + \left(\frac{1}{2}-\xi\right)\bar{h}(-\mathfrak{M}). \tag{19}$$

Combining the inequalities (18) and (19) yields the bound

$$\left(\frac{1}{2}+\xi\right)\bar{h}(-\mathfrak{M}) \leq \frac{\left(\frac{1}{2}-\xi\right)^2}{\frac{1}{2}+\xi}\bar{h}(-\mathfrak{M}). \tag{20}$$

Since $\xi \in (0, \frac{1}{2})$, the inequality (20) can be satisfied only if $\bar{h}(-\mathfrak{M}) = 0$. The function $\bar{h}(-\mathfrak{M})$ is a convex combination of various evaluations of the non-negative function f including $f(y_1, \ldots, y_G)$. It follows that these evaluations of f must also be zero. We have thus proved that

$$f(\mathbf{y}) = 0 \quad \forall\, \mathbf{y} \in \{+\mathfrak{M}, -\mathfrak{M}, +\mathfrak{R}, -\mathfrak{R}, +\mathfrak{C}, -\mathfrak{C}\}^G \backslash \{+\mathfrak{M}, +\mathfrak{R}, +\mathfrak{C}\}^G. \tag{21}$$

Continuing on, substituting the result of (21) in (18) and (19) yields the relation

$$T\bar{h}(+\mathfrak{C}) = \left(\frac{1}{2}-\xi\right)\bar{h}(+\mathfrak{M}). \tag{22}$$

We now convert this relation of the function $\bar{h}$ to an analogous relation of the function f. Suppose that for every question $i \in \{G+1, \ldots, N\}$, the worker has beliefs $p_{A,i} = 1,\ p'_{A|B,i} = 1,\ q_{A,i} = 0$, and that every question in this set actually results in a mismatch. Recall that the function $\bar{h}$ is a convex combination of the function f evaluated at various points corresponding to the various choices of the G gold standard questions out of the N total questions, where the choice necessarily includes question 1. Applying this observation to the relation (22) yields

$$\sum_{\substack{j\in\{0,\ldots,G-1\}\\ i_1,\ldots,i_j\subseteq\{2,\ldots,G\}}} \alpha_{i_1,\ldots,i_j}\Big\{Tf(+\mathfrak{C}, y_{i_1}, \ldots, y_{i_j}, +\mathfrak{R}, \ldots, +\mathfrak{R}) - (\frac{1}{2}-\xi)f(+\mathfrak{M}, y_{i_1}, \ldots, y_{i_j}, +\mathfrak{R}, \ldots, +\mathfrak{R})\Big\},$$

where $\left\{\alpha_{i_1,\ldots,i_j}\right\}$ are all positive constants. An inductive argument on the values of $(y_2,\ldots,y_G)$, starting with $y_2=\ldots=y_G=+\mathfrak{R}$ as the base case and further inducting on the number of values in $y_2,\ldots,y_G$ equalling $+\mathfrak{R}$ yields the result

$$Tf(+\mathfrak{C},y_2,\ldots,y_G)=\left(\frac{1}{2}-\xi\right)f(+\mathfrak{M},y_2,\ldots,y_G), \tag{23}$$

for every value of $y_2,\ldots,y_G$. Calling upon Lemma 2 and using (21) also yields

$$Tf(+\mathfrak{C},y_2,\ldots,y_G)=(1-T)f(+\mathfrak{R},y_2,\ldots,y_G). \tag{24}$$

From the relations (21), (23) and (24) and using the fact that all arguments above apply to any permutation of the G gold standard questions yield the claimed result that f must be identical to the mechanism of Algorithm 1.

The only remaining detail is to prove Lemma 2 and Lemma 3 which we do below.

Proof of Lemma 2 We begin by introducing some additional notation that will aid in subsequent discussion. Define a function $g:\{+\mathfrak{M},-\mathfrak{M},+\mathfrak{R},-\mathfrak{R},+\mathfrak{C},-\mathfrak{C}\}^N\to[0,\mu]$ as the expected payment (across the randomness in the choice of the gold standard questions) given the evaluations to all the N questions, that is,

$$g(y_1,\ldots,y_N)=\frac{1}{\binom{N}{G}}\sum_{(i_1,\ldots,i_G)\subseteq\{1,\ldots,N\}}f(y_{i_1},\ldots,y_{i_G}). \tag{25}$$

We first show that the function g must satisfy the relation

$$(1-T)g(y_1,\ldots,y_{N-1},+\mathfrak{R})+Tg(y_1,\ldots,y_{N-1},-\mathfrak{R})=Tg(y_1,\ldots,y_{N-1},+\mathfrak{C})+(1-T)g(y_1,\ldots,y_{N-1},-\mathfrak{C}), \tag{26}$$

for every value of $(y_1,\ldots,y_{N-1})$. To this end, suppose the worker is presently in the second stage. Suppose that the worker's beliefs regarding the various questions are unaffected by the results of matching or mismatching at the end of the first stage. Letting $\mathcal{S}:=\{i\in[N-1]\mid y_i\in\{+\mathfrak{R},-\mathfrak{R},+\mathfrak{C},-\mathfrak{C}\}$, suppose that questions $\mathcal{S}\cup\{N\}$ make it to the second stage. For every $i\in[N]$, let p_i' be the confidence of the worker for the answer that she marked under event y_i. For every $i\in[N-1]$, let $r_i=p_i'$ if $y_i<0$ and $r_i=(1-p_i')$ if $y_i>0$.[7] Let $E=[\epsilon_1\ \cdots\ \epsilon_{N-1}]\in\{-1,1\}^{N-1}$.

Since the mechanism is incentive compatible, it must be able to appropriately incentivize the worker for the N^{th} question. This condition necessitates

$$\begin{aligned}
&p'\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1y_1,\ldots,-\epsilon_{N-1}y_{N-1},+\mathfrak{R})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)\\
&+(1-p')\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1y_1,\ldots,-\epsilon_{N-1}y_{N-1},-\mathfrak{R})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)\\
&\underset{p'>1-T}{\overset{p'<1-T}{\lessgtr}}(1-p')\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1y_1,\ldots,-\epsilon_{N-1}y_{N-1},+\mathfrak{C})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)\\
&+p'\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1y_1,\ldots,-\epsilon_{N-1}y_{N-1},-\mathfrak{C})\prod_{j\in[N]\setminus\{\gamma\}}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right).
\end{aligned} \tag{27}$$

The left hand side of (27) is the expected payment if the worker chooses to retain her answer for the N^{th} question, while the right hand side is the expected payment if she chooses to copy the reference answer. Now, note that for any real valued variable q, and for any constants a, b and c,

$$ay\ \underset{q>c}{\overset{q<c}{\lessgtr}}\ b\quad\Rightarrow\quad a>0,\ c=\frac{b}{a},\ b>0\,.$$

[7] For ease of exposition, we consider $\{+\mathfrak{M},+\mathfrak{R},+\mathfrak{C}\}$ as "positive" values, and $\{-\mathfrak{M},-\mathfrak{R},-\mathfrak{C}\}$ as the corresponding "negative" values with inverted signs.

Applying this fact and making some simple algebraic manipulations gives

$$\begin{aligned}
&(1-T)\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1 y_1,\ldots,-\epsilon_{N-1}y_{N-1},+\mathfrak{R})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)\\
&+T\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1 y_1,\ldots,-\epsilon_{N-1}y_{N-1},-\mathfrak{R})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)\\
&-T\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1 y_1,\ldots,-\epsilon_{N-1}y_{N-1},+\mathfrak{C})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)\\
&-(1-T)\sum_{E\in\{-1,1\}^{N-1}}\left(g(-\epsilon_1 y_1,\ldots,-\epsilon_{N-1}y_{N-1},-\mathfrak{C})\prod_{j\in[N-1]}r_j^{\frac{1+\epsilon_j}{2}}(1-r_j)^{\frac{1-\epsilon_j}{2}}\right)=0. \qquad (28)
\end{aligned}$$

The left hand side of this equation is a polynomial in $\{r_1,\ldots,r_{N-1}\}$ which evaluates to zero for a solid $(N-1)$-dimensional box of values of $\{r_1,\ldots,r_{N-1}\}$. It follows that the coefficients of all monomials in this polynomial must be zero, and in particular, the constant term must be zero. The constant term appears when $\epsilon_j=-1\ \forall\ j$ in the summations. This argument thus yields the relation

$$\begin{aligned}
(1-T)g(y_1,\ldots,y_{N-1},+\mathfrak{R})+Tg(y_1,\ldots,y_{N-1},-\mathfrak{R})&\\
=Tg(y_1,\ldots,y_{N-1},+\mathfrak{C})+(1-T)g(y_1,\ldots,y_{N-1},-\mathfrak{C}),&
\end{aligned}$$

as claimed. Furthermore, since the arguments above are invariant to any permutation of the questions, we get that for any $i\in[N]$, any $(y_1,\ldots,y_{i-1},y_{i+1},\ldots,y_N)\in\{+\mathfrak{M},-\mathfrak{M},+\mathfrak{R},-\mathfrak{R},+\mathfrak{C},-\mathfrak{C}\}^{N-1}$, any incentive compatible mechanism must satisfy

$$\begin{aligned}
(1-T)g(y_1,\ldots,y_{i-1},+\mathfrak{R},y_{i+1},\ldots,y_N)+Tg(y_1,\ldots,y_{i-1},-\mathfrak{R},y_{i+1},\ldots,y_N)&\\
=Tg(y_1,\ldots,y_{i-1},+\mathfrak{C},y_{i+1},\ldots,y_N)+(1-T)g(y_1,\ldots,y_{i-1},-\mathfrak{C},y_{i+1},\ldots,y_N). \qquad (29)&
\end{aligned}$$

It remains to convert the result of Equation (29) to an equivalent condition on the function f as in the statement of the lemma. To this end, suppose that $y_{G+1}=\ldots=y_N=+\mathfrak{R}$. Also suppose without loss of generality that $i=1$. Then expanding the function g in (29) in terms of its constituent components f, we obtain the relation

$$\begin{aligned}
\sum_{\substack{j\in\{0,\ldots,G-1\}\\ i_1,\ldots,i_j\subseteq\{2,\ldots,G\}}}&\Big\{\alpha_{i_1,\ldots,i_j}\big((1-T)f(+\mathfrak{R},y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})+Tf(-\mathfrak{R},y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})\\
&\qquad -Tf(+\mathfrak{C},y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})-(1-T)f(-\mathfrak{C},y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})\big)\\
&+\alpha'_{i_1,\ldots,i_j}\big((1-T)f(y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})+Tf(y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})\\
&\qquad -Tf(y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})-(1-T)f(y_{i_1},\ldots,y_{i_j},+\mathfrak{R},\ldots,+\mathfrak{R})\big)\Big\}=0, \qquad (30)
\end{aligned}$$

where $\left\{\alpha_{i_1,\ldots,i_j},\alpha'_{i_1,\ldots,i_j}\right\}$ are all positive constants. We complete the proof with inductive argument on the values of $(y_2,\ldots,y_G)$. We begin by considering the base case $y_2=\ldots=y_G=+\mathfrak{R}$, for which we obtain the result

$$(1-T)f(+\mathfrak{R},+\mathfrak{R},\ldots,+\mathfrak{R})+Tf(-\mathfrak{R},+\mathfrak{R},\ldots,+\mathfrak{R})=Tf(+\mathfrak{C},+\mathfrak{R},\ldots,+\mathfrak{R})+(1-T)f(-\mathfrak{C},+\mathfrak{R},\ldots,+\mathfrak{R}).$$

from (30). We further induct on the number of values in $y_2,\ldots,y_G$ that equal $+\mathfrak{R}$ in (30), and this inductive argument thus shows that

$$(1-T)f(+\mathfrak{R},y_2,\ldots,y_G)+Tf(-\mathfrak{R},y_2,\ldots,y_G)=Tf(+\mathfrak{C},y_2,\ldots,y_G)+(1-T)f(-\mathfrak{C},y_2,\ldots,y_G),$$

for all possible values of $y_2,\ldots,y_G$. Finally, all arguments above are invariant to any permutation of the questions, and consequently we get the claimed result.

Proof of Lemma 3 We will induct on the number of entries in $\mathbf{y}$ whose values equal either $-\mathfrak{M}$ or $-\mathfrak{R}$ or $-\mathfrak{C}$ or $+\mathfrak{C}$; let us use the notation γ to denote the number of such entries. When $\gamma = G$, the no-free-lunch axiom implies $f(\mathbf{y}) = 0$, where we have used the assumption that $\mathbf{y} \notin \{+\mathfrak{C}, +\mathfrak{R}\}^G$ when applying the no-free-lunch axiom. The statement of the lemma is thus satisfied in this case.

Now suppose that $f(\mathbf{y}) = 0$ whenever $\gamma \geq \gamma_0 + 1$ for some integer $\gamma_0 > 0$. Consider any evaluation $\mathbf{y}$ such that $y_1, \ldots, y_{\gamma_0} \in \{-\mathfrak{M}, -\mathfrak{R}, -\mathfrak{C}, +\mathfrak{C}\}$. Then from the induction hypothesis stated above, we will have $f(\mathbf{y}) = 0$ if additionally we had $y_{\gamma_0+1} \in \{-\mathfrak{M}, -\mathfrak{R}, -\mathfrak{C}, +\mathfrak{C}\}$. Applying Lemma 2 with $i = \gamma_0 + 1$ gives $f(\mathbf{y}) = 0$ when $y_{\gamma_0+1} = +\mathfrak{R}$. This inductive argument completes the proof of the lemma.

C.8. Proof of Theorem 6: No-free-lunch cannot be stronger

Suppose that for every question, the worker has $p_A = \frac{3}{4} = 1 - p_B$, and further suppose that as desired, the worker selects option A for every question in the first stage. Suppose that there is a mismatch for every question, and hence all the questions go to the second stage. Now suppose that in the second stage, the worker has an updated belief $p'_{A|B} = \frac{1}{4}$ for every question. In this case, we wish to incentivize the worker to change her answer for every question. However, strong-no-free-lunch mandates that $f(\mathbf{x}) = 0$ for every $\mathbf{x} \in \{+\mathfrak{C}, -\mathfrak{C}\}^G$, and consequently, the worker will necessarily be paid a zero amount under such an action. Since any other action will also fetch an amount no less than zero, the worker is not incentivized to change her answers as required. Consequently, the strong-no-free lunch is too strong for the existence of any incentive-compatible mechanism.