

# Computational principal agent problems

Pablo D. Azar\*  
MIT

Silvio Micali†  
MIT

July 8, 2017

## Abstract

Collecting and processing large amounts of data is becoming increasingly crucial in our society. We model this task as evaluating a function  $f$  over a large vector  $x = (x_1, \dots, x_n)$ , which is unknown, but drawn from a publicly known distribution  $X$ . In our model learning each component of the input  $x$  is costly, but computing the output  $f(x)$  has zero cost once  $x$  is known.

We consider the problem of a principal who wishes to delegate the evaluation of  $f$  to an agent, whose cost of learning any number of components of  $x$  is always lower than the corresponding cost of the principal.

We prove that, for every continuous function  $f$  and every  $\varepsilon > 0$ , the principal can—by learning a single component  $x_i$  of  $x$ —incentivize the agent to report the correct value  $f(x)$  with accuracy  $\varepsilon$ .

---

\*Email: pazar@mit.edu. Address: MIT Economics. 77 Massachusetts Ave. E52-300, Cambridge MA, 02139.

†Email: silvio@csail.mit.edu. Address: MIT Electrical Engineering and Computer Science. 32 Vassar Street, Cambridge, MA 02139.

# 1 Introduction

Our society is more and more data intensive. Every day, firms need to gather and process multiple pieces of data to make products and decisions. In this paper we investigate how to delegate to a rational agent the process of first obtaining multiple pieces of data and then aggregate them into a “compact” final result. In our model, obtaining the pieces of data is expensive, but the algorithmic operations to process them are free.

In our setting a risk-neutral principal wants to learn the output of a continuous function  $f : [0, 1]^n \rightarrow \mathbb{R}$  on an input  $x = (x_1, \dots, x_n)$ . The input  $x$  is unknown, but drawn from a known distribution  $X$ . Once the input vector  $(x_1, \dots, x_n)$  is known,  $f(x_1, \dots, x_n)$  can be computed at no cost. Learning the inputs, however, is costly. Specifically, the principal can learn any  $k$  coordinates of  $x$  at a cost of  $\gamma(k)$ . There also exists a risk-neutral agent, who can learn any  $k$  coordinates of  $x$  at a smaller cost,  $c(k) < \gamma(k)$ , where both  $c(k)$  and  $\gamma(k)$  are increasing with  $k$ . Since the agent has a lower cost, it is socially optimal for the agent to learn  $y = f(x)$ , and then report  $y$  to the principal. However, since the principal cannot monitor the agent’s actions, she cannot necessarily trust him to learn  $f(x)$  nor to truthfully report its value. Thus, we must find a way to incentivize the agent to act on the best interests of the principal in this new framework.

Let us illustrate the usefulness of our goals by means of two examples.

**EXAMPLE 1: SCIENTIFIC DATA.** Each input  $x_i$  is the result of an expensive but replicable physical experiment, which the agent can perform more cheaply than the principal.

**EXAMPLE 2: PROPRIETARY DATA.** An information company has a proprietary database about the behavior of  $n$  individuals, and the principal is an outsider wishing to obtain some aggregate function of these data in order—say—to price a new product. Here, the agent is the information company itself, and while the principal can recreate each individual’s data from scratch with non-trivial cost, the information company—after sinking a fixed cost into creating its database—can retrieve each record very cheaply.

Although our framework and results apply also when the number of inputs  $n > 1$  is small and each input component  $x_i$  consists of just a few digits, our results are most relevant when  $n$  is large and each  $x_i$  has a large number of significant digits, so that it would be impractical for an agent to report the entire input vector  $(x_1, \dots, x_n)$  to the principal. Typically, in fact, it is when  $(x_1, \dots, x_n)$  is huge that one wishes to deal instead with an aggregate value  $f(x_1, \dots, x_n)$ .<sup>1</sup> Indeed, when the input vector  $x$  is large,<sup>2</sup> the principal should not insist on the agent revealing all the data he has learned, but on his reporting the right answer  $f(x)$ !

---

<sup>1</sup>As an example, when  $f$  is Lipschitz continuous with Lipschitz constant 1, so that  $|f(x) - f(x')| < \delta$  when  $\|x - x'\|_\infty < \delta$ , the number of significant digits that we would need to send to communicate  $f(x)$  increases with  $1/\delta$ , while the number of significant digits needed to communicate the entire vector  $(x_1, \dots, x_n)$  increases at a rate that is  $n$  times larger.

<sup>2</sup>E.g., the Large Hadron Collider at CERN produces 30 petabytes of data every year [15].

We investigate delegation of computation over costly inputs in two settings, described below.

## 1.1 Exact Computation

In our first setting, the *exact computation case*, the principal wants to learn  $y = f(x_1, \dots, x_n)$  *exactly*. We show that, for an important class of functions, the principal can incentivize the agent to reveal  $f(x)$  using a *direct mechanism*. Informally, in a direct mechanism

1. The agent
  - a. chooses a subset  $S_A$  of the input coordinates and learns the corresponding subvector  $x_{S_A} = (x_i)_{i \in S_A}$  at a cost of  $c(|S_A|)$ , and
  - b. reports a value  $z(x_{S_A})$ , allegedly equal to  $y = f(x_1, \dots, x_n)$ .
2. The principal
  - a. chooses a (random) subset  $S$  of the input coordinates and learns  $x_S = (x_i)_{i \in S}$ , and
  - b. pays the agent a reward  $R(z, x_S)$  which is a continuous function of  $z$  and  $x_S$ .

A direct mechanism is

- (a) *incentive compatible* if the agent maximizes his expected payoff  $\mathbb{E}_S[R(z, x_S) - c(|S_A|)]$  only by learning  $f(x_1, \dots, x_n)$  exactly, and reporting  $z = f(x_1, \dots, x_n)$ ,
- (b) *individually rational* if the agent's expected reward minus his costs are greater than or equal to zero.

**Our First Result: The Mechanism  $\mathcal{M}_1$ .** A function  $f : [0, 1]^n \rightarrow \mathbb{R}$  is separable if  $f(x_1, \dots, x_n) = \sum_{i=1}^n f_i(x_i)$ . We show that if  $f$  is separable and each  $f_i$  is bounded, then the principal can correctly learn  $f(x_1, \dots, x_n)$  by means of a direct, incentive compatible, and individually rational mechanism  $\mathcal{M}_1$  in which the principal queries *just one input* herself.

As we shall see, mechanism  $\mathcal{M}_1$  is a crucial subroutine for our more general mechanisms.

**Our Second Result: The Limits of Direct Contracts.** Our first result raises the question of whether direct contracts can be used to delegate the exact computation of functions that are not separable. In our second theorem, we show that this is not the case even for a very simple non-separable function. Indeed we show that for

$$f(x) = \prod_{i=1}^n x_i$$

no direct contract can incentivize the agent to reveal  $f(x)$  unless the principal queries the entire input vector  $(x_1, \dots, x_n)$  herself.<sup>3</sup>

---

<sup>3</sup>It is important to remark that our positive results (Theorems 1 and 3) in this paper are proved for functions defined over  $[0, 1]^n$ , while the counterexample function  $f(x) = \prod_{i=1}^n x_i$  we use to prove our negative results (Theorems 2 and 4) will be defined over  $[-1, 1]^n$ . We note that we can always scale and shift this domain and use the function  $\tilde{f}(x) = \prod_{i=1}^n (2x_i - 1)$  defined over  $[0, 1]^n$ , but we prefer not to do so to simplify the notation in the proofs of Theorems 2 and 4.

## 1.2 Approximate Computation

In our second setting, *the  $\epsilon$ -approximate computation case*, the principal wants to learn  $z$  such that  $|z - f(x)| \leq \epsilon$ , for some arbitrarily small  $\epsilon > 0$ .

### **Our Third Result: Approximately Delegating Arbitrary Continuous Functions.**

We show that for any continuous function  $f : [0, 1]^n \rightarrow \mathbb{R}$ , any input  $x \in [0, 1]^n$  and any  $\epsilon > 0$ , there exists a (non direct) mechanism  $\mathcal{M}_2$  that incentivizes the agent to reveal a  $z$  which is within distance  $\epsilon$  of  $f(x)$ .

In mechanism  $\mathcal{M}_2$ :

- The principal queries only one coordinate of the input vector  $x$
- The principal and the agent interact in 2 rounds, in each of which one sends a message to the other.

**Our Fourth Result: Round Optimality of  $\mathcal{M}_2$**  In our fourth result, we show that one-round mechanisms cannot be used for approximate delegation of arbitrary continuous functions unless the principal queries the whole input vector  $(x_1, \dots, x_n)$ , which would defeat the purpose of delegating the computation in the first place. Accordingly, our mechanism  $\mathcal{M}_2$  simultaneously minimizes the number of queries that the principal makes to the input vector, and the number of rounds of interaction between the principal and the agent.

## 1.3 Optimality and Unlimited liability

Our mechanisms  $\mathcal{M}_1$  minimizes the number of queries and interaction rounds, among all incentive-compatible mechanisms that delegate separable functions, and  $\mathcal{M}_2$  minimizes the number of queries and interaction rounds, among all incentive-compatible mechanisms that approximately delegate continuous functions. These properties hold for any environment where the agent and principal are risk neutral. In particular, they hold when the agent has limited liability and can only be paid a positive amount ex-post.

Of course, queries and rounds of interaction are not the only costs that the principal faces. She must also bear the monetary cost of paying the agent's reward. We highlight that there is one setting where we can simultaneously minimize all three of these costs: namely, when the agent also has unlimited liability.

With unlimited liability, the technique for minimizing the expected payment to the agent is well known [11]: the principal charges the agent a fixed participation fee, equal to the expected utility that the agent gets from the mechanism.<sup>4</sup> In this way, the principal can always find a mechanism that makes the agent's IR constraint bind, without affecting the number of queries or the number of communication rounds.

---

<sup>4</sup>The principal can compute this expected reward at zero cost because she knows the random variable  $X$  from which the true input  $x$  is drawn and can take expectations with respect to this distribution without gathering any costly information about the true input  $x$ . Of course, this would not be true in a model where the principal has to pay a cost for every algorithmic operation that she makes.

**Auditing and Optimality** When the agent has unlimited liability, there is a trivial incentive compatible, individually rational, and one-round mechanism in which the principal learns  $f(x_1, \dots, x_n)$  exactly, minimizes the expected reward, and makes (in expectation) arbitrarily few queries. Namely,

THE  $\varepsilon$ -AUDITING MECHANISM

1. The agent reports a value  $z$ , allegedly equal to  $y = f(x_1, \dots, x_n)$
2. With probability  $(1 - \varepsilon)$ , the principal pays the agent 0.
3. With probability  $\varepsilon$ , the principal queries the entire input vector  $(x_1, \dots, x_n)$  and pays the agent  $c^{(n)}/\varepsilon$ , if  $f(x_1, \dots, x_n) = z$ , and 0 otherwise.

The  $\varepsilon$ -Auditing mechanism, however, is not meaningful in many cases: for instance, when  $\gamma(k) = +\infty$  for some  $k > 1$ . Consider first our example 1, where each  $x_i$  is the result of a replicable scientific experiment. Assume  $n = 100$ , and each experiment takes one day for the agent to perform, but one year for the principal. Then, for every possible choice of  $\varepsilon$ , the agent should never agree to participate in the  $\varepsilon$ -Auditing Mechanism, as the principal will die before he could pay the agent a positive amount.

Consider now our example 2, where  $(x_1, \dots, x_n)$  is a proprietary dataset owned by an information company. If the principal does not have enough money (or time) to learn the entire vector  $(x_1, \dots, x_n)$ , then the agent has no hope to be paid. (Problems also arise even when the agent is able to reveal and prove to the principal the value of each  $x_i$  in a very cheap manner. Indeed, the information company must reveal all of its proprietary data in order to get paid, and thus destroys its own business by enabling a competitor in the process.)

In any case, even when the  $\varepsilon$ -Auditing Mechanism can be meaningfully used, we must ask what is the “optimal mechanism to use once the auditing state is reached.” This is the mechanism that would be observable (when there is no auditing, we only observe that there is no auditing). Thus, we will focus on mechanisms which—under any possible realization of the random choices they make—query at least one input. As we shall prove in sections 3 and 4,  $\mathcal{M}_1$  and  $\mathcal{M}_2$  both make only one query to the input distribution and, under unlimited liability, are therefore “optimal once the auditing state is reached” for their respective settings.

## 1.4 Additional Related Work

Delegating computation to an expert can be interpreted as a moral hazard problem. The agent’s effort corresponds to the number of components of  $x$  that he queries. The project is successful only if the principal learns  $f(x)$  correctly, which requires maximum effort from the agent. Furthermore, the agent has to decide *ex-ante* how much effort to exert, before observing any components of  $x$ .

There are many results in the proper scoring rule literature that consider costly information acquisition. Osband [19], Clemen [6] and Lambert [14] consider modifications of strictly proper scoring rules where the agent can increase the precision of his signal by paying a cost. These models are like ours because the agent must exert some costly effort to acquire more

information, and the decision to expend this effort is made ex-ante. In fact, our mechanisms use strictly proper scoring rules as a crucial component to incentivize the agent to acquire information about  $x$ , and then reveal  $f(x)$ .

Demski and Sappington [7] consider a moral hazard model where a (weakly) risk-averse agent must be incentivized to (a) acquire information about a random state of the world  $s$  (b) take an action  $a$  that produce some outcome  $y = p(s, a)$ . The principal is risk-neutral and can only observe the outcome  $y$ . Her payoff is given by  $y - t(y)$ , where  $t$  is some transfer to the agent that only depends on the observed outcome  $y$ . This model—like ours—captures scenarios where expertise is both costly to acquire and costly to communicate. In fact, in Demski and Sappington’s model, communication is *infinitely expensive*: the principal and agent cannot exchange messages, and the transfer can only depend on the observed outcome. Our paper is similar to Demski and Sappington’s model in the sense that the agent learns a function of the state of the world, but due to communication costs cannot share his entire expertise with the principal. Instead, he must take an action depending on his acquired expertise which produces a low-dimensional outcome which is the only information observable by the principal.

There are many other papers that consider moral hazard with costly information acquisition (see Zermeno[20], Carroll [5], Malcolmson[16] and the references therein). The main conceptual difference between our results and the moral hazard with costly information acquisition literature is that our model allows *both* the principal and the agent to acquire information at a cost, with the information asymmetry arising from the fact that the principal’s cost for acquiring information is higher than the agent’s respective cost. Furthermore, we focus on a problem that has been overlooked so far, that is, how to delegate the evaluation of any continuous function while observing as few of the inputs as possible.

Our model can also be viewed as a way of capturing the complexity of contracts. One other way to account for computation in the design of contracts has been studied by Anderlini and Felli [2], who show that when contracts are generated by a Turing Machine, then in some situations computable contracts will be sub-optimal. In subsequent work, Al-Najjar, Anderlini and Felli [1] show that when events cannot be finitely described, sometimes the best contract is no contract at all. Since our model focuses on the computation of continuous functions, where the complexity of a contract depends only on the number of inputs queried, we bypass these impossibility results. Another way in which complexity in contracts can be modeled is via how much time it takes to resolve uncertainty in the state of the world. Using this, MacLeod studies what type of contracts are more efficient in low and high complexity environments [17].

In a previous paper of ours [3], we study the problem of delegating the computation of a function, where both principal and agent have *zero* cost for computing, but the principal cannot perform more than a given amount of computation. In contrast, the model we present in this paper captures the fact that computation is costly for both the principal and the agent, and that differences in cost should capture realistic differences in computational power.

Furthermore, our earlier results [3], and follow-up work by Guo, Hubacek, Rosen and Wald [10] focus on giving alternative characterizations of computational complexity classes

as sets of problems that can be delegated to a rational agent. In particular, when the agent is computationally unlimited, the set of problems that can be delegated to a computationally unbounded and rational agent in one-round of communication includes the complexity class  $\#P$  [3]. This is a set of canonical counting problems which includes counting the number of satisfying assignments to a boolean formula, and counting the number of perfect matchings in a bipartite graph. When the principal is restricted to act in *sublinear time* and the agent is restricted to act in polynomial time (but both still have zero cost of computation), the set of problems that the principal can delegate to the agent is the complexity class  $P$  [10]. This is the set of problems which can be solved in polynomial time.

We remark that—like our results in this paper—the results in [3] and [10] use proper scoring rules to incentivize the agent to compute a large sum in a truthful manner.

## 2 The Model

**Notation** We denote the set  $\{1, \dots, n\}$  by  $[n]$ . For any  $S \subset [n]$ , we denote by  $x_S$  the vector  $(x_i)_{i \in S}$  and by  $x_{-S}$  the vector  $(x_i)_{i \notin S}$ . We emphasize that knowledge of  $x_S$  implies knowledge of the underlying coordinate set  $S$ . We refer to each subvector  $x_S$  as a partial input.

If  $X = (X_1, \dots, X_n)$  is a random variable over  $\mathbb{R}^n$ , we define  $X_S \stackrel{\text{def}}{=} (X_i)_{i \in S}$  and denote the conditional random variable “ $X$  given that  $X_S = x_S$ ” by  $X|x_S$ . We refer to the process of learning  $x_S$  as *querying* the subset  $S$ .

If  $g$  is a function of a random variable  $X$ , we will write the expectation of  $g$  as  $\mathbb{E}_{x \leftarrow X}[g(x)]$ . We write the expectation of a function  $g(\cdot, \cdot)$  with respect to two independent random variables  $X$  and  $Y$  as  $\mathbb{E}_{x \leftarrow X, y \leftarrow Y}[g(x, y)]$ .

**Computational Environments** A *computational environment* is a tuple,  $\mathcal{E} = (f, X, x, c, \gamma)$ , where

- $f : [0, 1]^n \rightarrow \mathbb{R}$  is the *target function*.
- $X \in \Delta([0, 1]^n)$  is a continuous random variable, with a distribution which is common knowledge to the principal and the agent.
- $x = (x_1, \dots, x_n)$  is a realization of the random variable  $X$ , a priori unknown to the principal or agent.
- $c : \{0, \dots, n\} \rightarrow \mathbb{R}$  is the *agent’s cost function*. For every set  $S \subset [n]$ ,  $c(|S|)$  is the cost to the agent of learning  $x_S$ .
- $\gamma : \{0, \dots, n\} \rightarrow \mathbb{R}$ , the *principal’s cost function*. For every set  $S \subset [n]$ ,  $\gamma(|S|)$  is the cost to the principal of learning  $x_S$ .

**Assumptions.** Throughout the paper, we will have the following assumptions

0 Any sequence of purely algorithmic operations has zero cost for both the principal and agent. (Thus, evaluating any function on known inputs, or computing an expectation over a known distribution, can be performed at zero cost.)

1. The cost functions are monotonic. For all  $\ell < k$  we have

$$\gamma(\ell) \leq \gamma(k), c(\ell) \leq c(k).$$

2. The cost of querying zero inputs is zero

$$\gamma(0) = c(0) = 0.$$

3. The cost functions satisfy the increasing differences condition. For all  $\ell, k$  we have

$$c(k + \ell) - c(k) \leq \gamma(k + \ell) - \gamma(k).$$

(I.e., it is always cheaper for the agent than for the principal to query  $\ell$  extra inputs.)

### 3 Exact Computation

In this section, we prove our results for the exact computation setting. Before stating our results, we define direct computational contracts, define boundedly separable functions, and recall some facts about proper scoring rules.

#### 3.1 Preliminaries

**Definition 1.** A set  $S \subset [n]$  determines  $f$  exactly if the support of the random variable  $f(X)|_{x_S}$  is a singleton for every possible value of  $x_S$ .

**Definition 2.**

For all  $k \leq n$ , a **k-query direct computational contract** is a mechanism  $\mathcal{M}$  specified by

- a function  $\mathcal{D}$  mapping a real number  $z$  to a distribution  $\mathcal{D}(z)$  over  $\{S \subset [n] : |S| = k\}$ , that is, over all input coordinate subsets of size  $k$
- a continuous reward function  $R$  mapping a real number  $z$ , and a partial input  $x_S$  to a real number  $R(z, x_S)$ .

Such a mechanism  $\mathcal{M} = (\mathcal{D}, R)$  has only one player, the agent, and is played as follows  
*Stage 0.* Nature draws  $x \leftarrow X$ . (The agent does not observe  $x$ .)

*Stage 1.* The agent queries a subset  $S_A$  of the inputs (updating his beliefs about  $x$  to  $X|x_{S_A}$ ).

*Stage 2.* The agent reports a value  $z = z(x_{S_A})$  to the mechanism.

*Stage 3.* The mechanism draws a subset  $S$  of inputs from the distribution  $\mathcal{D}(z)$ .



In such a play, the agent's reward is  $r = R(z, x_S)$  and his utility is  $r - c(|S_A|)$ .

$\mathcal{M}$  is **incentive compatible** for a function  $f : [0, 1]^n \rightarrow \mathbb{R}$  if the agent strictly maximizes his utility by choosing  $S_A$  such that  $S_A$  determines  $f$  exactly, and reporting  $z = f(x)$  in stage 2.

$\mathcal{M}$  is **individually rational** for  $f$  if  $\mathbb{E}_{S \leftarrow \mathcal{D}(f(x)), x \leftarrow X} [R(f(x), x_S) - c(n)] \geq 0$

**Definition 3.** A function  $f : [0, 1]^n \rightarrow \mathbb{R}$  is **boundedly separable** if there exist bounded functions  $f_1, \dots, f_n : [0, 1] \rightarrow \mathbb{R}$  such that  $f(x_1, \dots, x_n) = \sum_{i=1}^n f_i(x_i)$ .

**Scoring Rules** A strictly proper scoring rule<sup>5</sup> is a function  $S : \Delta(K) \times K \rightarrow \mathbb{R}$  which takes as input a distribution  $\Omega$  over a finite set  $K$  and a sample  $\omega \in K$  and which satisfies the property

$$\mathbb{E}_{\omega \leftarrow \Omega} S(\Omega, \omega) > \mathbb{E}_{\omega \leftarrow \Omega} S(\Omega', \omega)$$

for any  $\Omega' \neq \Omega$ . A function which satisfies this property incentivizes a rational expert to state his true beliefs about the distribution of  $\omega$ . Many such rules are known.<sup>6</sup> We will use Brier's scoring rule, defined by

$$\text{BSR}(\Omega, \omega) = 2Pr(\Omega = \omega) - \sum_{\alpha} Pr(\Omega = \alpha)^2 + 1$$

where  $Pr(\Omega = \alpha)$  is the probability that distribution  $\Omega$  assigns to the element  $\alpha$ . The Brier Scoring Rule is always in the interval  $[0, 3]$ .<sup>7</sup>

### 3.2 Our First Theorem

**Theorem 1.** Let  $f : [0, 1]^n \rightarrow \mathbb{R}$  be a boundedly separable function. Then there exists a one-query, incentive compatible and individually rational direct computational contract  $\mathcal{M}_1 = (R, \mathcal{D})$  for  $f$ .

*Proof.* We first prove theorem 1 assuming that the agent has zero costs: that is,  $c(|S_A|) = 0$  for all  $S_A \subset [n]$ . Let  $f(x_1, \dots, x_n) = f_1(x_1) + \dots + f_n(x_n)$  and let  $B > 0$  be a bound such that  $|f_i(x)| < B$  for every  $x \in [0, 1]$  and every  $i \in [n]$ . Let  $g_i(x) = f_i(x_i) + B$  and note that  $0 \leq g_i(x) \leq 2B$ . Let  $g(x_1, \dots, x_n) = \sum_{i=1}^n g_i(x) = f(x_1, \dots, x_n) + nB$ . We now give an incentive compatible contract for  $g$ . Since, by construction, we have ensured that each term

<sup>5</sup>For brevity and when the context is clear, we will often refer to strictly proper scoring rules simply as *proper scoring rules* or *scoring rules*.

<sup>6</sup>The interested reader is referred to a paper by Gneiting and Raftery [8], which includes a comprehensive survey.

<sup>7</sup>Usually, the Brier Scoring Rule is defined as  $\text{BSR}(\Omega, \omega) = 2Pr(\Omega = \omega) - \sum_{\alpha} Pr(\Omega = \alpha)^2 - 1$ . Our formula is the usual definition, plus 2. The reason we add 2 to the usual formulation of the scoring rule is to ensure that the reward to the expert is non-negative. Note that adding a constant to this reward does not affect incentives.

$g_i(x)$  in the sum is non-negative, our mechanism can scale this term so as to interpret it as a probability, and use some techniques from scoring rules to incentivize the agent.

Our mechanism  $\mathcal{M}_1$  takes the agent's report  $z$  as an input and produces a distribution  $\mathcal{D}(z)$  and reward function  $R(z, \cdot)$  as follows

$$\text{MECHANISM } \mathcal{M}_1(z) = (\mathcal{D}(z), R(z, \cdot))$$

- $\mathcal{D}(z)$  is the uniform distribution over the singleton sets  $\{\{1\}, \dots, \{n\}\}$ .
- The reward function  $R(z, \cdot)$  is defined as follows
  - If  $z \notin [-Bn, Bn]$ , then  $R_1(z, x_S) = -\infty$  for all  $x_S$ .  
(I.e., since the range of  $f$  is  $[-Bn, Bn]$ , a  $z$  outside this range must be a lie.)
  - If  $z \in [-Bn, Bn]$ , then the mechanism
    - \* Draws  $S = \{i\}$  from  $\mathcal{D}$ . Since  $S$  is a singleton we denote it by  $S = i$ .
    - \* Queries  $x_i$  and computes  $g_i(x_i)$ .
    - \* Draws a realization from the random variable  $\omega$  which is equal to 1 with probability  $g_i(x_i)/2B$  and equal to 0 with probability  $1 - g_i(x_i)/2B$ .
    - \* Interprets  $z$  as a random variable  $\Omega_z$  over  $\{0, 1\}$  which is equal to 1 with probability  $(z+Bn)/2Bn$  and equal to 0 with probability  $1 - (z+Bn)/2Bn$ .
    - \* Returns  $R_1(z, x_i) = BSR(\Omega_z, \omega)$ .

Let us now show that mechanism  $\mathcal{M}_1$  is incentive compatible. Given a report  $z$ , the agent's expected reward is  $\mathbb{E}_i R(z, x_i) = \mathbb{E}_\omega BSR(\Omega_z, \omega)$ . Since Brier's scoring rule is strictly proper, this expected reward is maximized only when the agent reports  $z$  so that  $\Omega_z$  is equal to the distribution from which the principal is drawing  $\omega$ .

Let us now consider the probability that  $\omega$  is equal to 1. This probability is equal to  $\sum_{i=1}^n Pr(\omega = 1 | S = i) Pr(S = i)$ . Since  $Pr(\omega = 1 | S = i) = g_i(x_i)/2B$  and  $Pr(S = i) = 1/n$ , we have that

$$Pr(\omega = 1) = \sum_{i=1}^n Pr(\omega = 1 | S = i) Pr(S = i) = \frac{1}{2Bn} \sum_{i=1}^n g_i(x_i).$$

Since  $\Omega_z$  is a random variable with  $Pr(\Omega_z = 1) = (z+Bn)/2Bn$ , the agent maximizes his reward by announcing  $z$  such that  $z + Bn = \sum_{i=1}^n g_i(x_i)$ . Note that this is equivalent to announcing  $z = \sum_i f_i(x_i)$ , because each  $g_i(x_i) = f_i(x_i) + B$ . Thus, the agent maximizes his reward by announcing  $z = f(x_1, \dots, x_n)$ . Since we are currently assuming that the agent's cost function is identically 0, the agent can learn the value of  $f((x_1, \dots, x_n))$  at no cost by querying  $S = [n]$ .

The above argument only applies when the agent's cost function  $c$  is identically 0. When this is not the case, we can scale the reward  $R(z, x)$  by a large enough constant so that the

agent is incentivized to learn the value of  $f(x_1, \dots, x_n)$  exactly. This part of our proof is standard in the costly information acquisition literature [14, 19, 6] and proceeds as follows.

For every partial input  $x_{S_A}$ , let

$$z^*(x_{S_A}) \in \operatorname{argmax}_z \mathbb{E}_{i \leftarrow \mathcal{D}(z), x \leftarrow X | x_{S_A}} [R(z, x_i)]$$

$$\nu(S_A) = \mathbb{E}_{i \leftarrow \mathcal{D}(z), x \leftarrow X} [R(z^*(x_{S_A}), x_i)]$$

so that  $\nu(S_A)$  is the reward that the agent ultimately expects to receive when he chooses to query set  $S_A$ . Let  $S_A^* \in \operatorname{argmax}_S \nu(S)$ . Since the agent maximizes his reward only if he reports  $z = f(x)$  exactly, the random variable  $f(X)|x_{S_A}$  must satisfy  $Pr(f(X) = z|x_{S_A}) = 1$  for any possible realization of  $x_{S_A}$ . This implies that the agent learns  $f(x)$  exactly when he queries the set  $S_A^*$ .<sup>8</sup>

Depending on the distribution of the random variable  $X$  and the choice of  $f$ , there might be multiple sets in  $\operatorname{argmax}_S \nu(S)$ . Let  $S_A^*$  be a set in  $\operatorname{argmax}_S \nu(S)$  which has minimum cardinality.<sup>9</sup> Let  $\kappa > 0$  be such that, for any  $S$  such that  $S_A \notin \operatorname{argmax}_S \nu(S)$ , we have

$$\kappa \cdot \nu(S_A^*) - c(|S_A^*|) > \kappa \cdot \nu(S_A) - c(|S_A|). \quad (1)$$

Since  $\nu(S_A)$  can be computed by taking expectations over the commonly known distribution  $X$ , and without the need to learn the true input  $x$ , the principal can compute  $\kappa$  at zero cost. Let  $\tilde{R}(\cdot, \cdot) \stackrel{\text{def}}{=} \kappa \cdot R(\cdot, \cdot)$  be a scaled reward function, and let  $\tilde{\nu}(S_A)$  be the reward that the agent expects to receive when he chooses to query  $S_A$  and the reward function is  $\tilde{R}$ . Then, by construction, we have

$$\tilde{\nu}(S_A^*) - c(|S_A^*|) > \tilde{\nu}(S_A) - c(|S_A|)$$

for any  $S_A \notin \operatorname{argmax}_S \nu(S)$  and any  $S_A \in \operatorname{argmax}_S \nu(S)$  such that  $c(|S_A|) > c(|S_A^*|)$ . Thus, by changing the reward of function of  $\mathcal{M}_1$  to be  $\kappa \cdot R$ , we can incentivize the agent to learn  $f(x_1, \dots, x_n)$  exactly at the minimum cost, and to report  $z = f(x_1, \dots, x_n)$ .

Finally, since  $\nu(S_A^*)$  is positive, the principal can choose  $\kappa$  so that inequality (1) holds, and also  $\kappa \cdot \nu(S_A^*) - c(|S_A^*|) \geq 0$ . That is, so that  $\mathcal{M}_1$  using the reward function  $\kappa \cdot R$  is also individually rational.

*Q.E.D.*

### 3.3 Query Optimality and Our Second Theorem

Since the mechanism  $\mathcal{M}_1$  only makes one query we have the following corollary

**Corollary 1.** *No direct computational contract that is incentive-compatible for separable functions makes fewer queries than  $\mathcal{M}_1$ .*

<sup>8</sup>Equivalently, for any  $x$  in the support of  $X$ , the value of the function  $f(x)$  must not depend on  $x_{-S_A^*}$ .

<sup>9</sup>Note that the agent is incentivized to query a set with minimum cardinality. For any two sets  $S_A^*, S_A^{**} \in \operatorname{argmax}_S \nu(S)$  such that  $|S_A^*| < |S_A^{**}|$ , the agent will always prefer to query  $S_A^*$  than to query  $S_A^{**}$ .

We now argue that the query optimality of mechanism  $\mathcal{M}_1$  is intrinsically linked to the fact that  $f$  is separable. Indeed, we show the following theorem

**Theorem 2.** *There exists a continuous, bounded and non-separable function  $f : [-1, 1]^n \rightarrow \mathbb{R}$  such that any direct computational contract  $\mathcal{M} = (\mathcal{D}, R)$  that is incentive compatible for  $f$  must query the whole input vector  $(x_1, \dots, x_n)$ .*

Theorem 2, which is proved in appendix A, provides the motivation to broaden either our definition of computational contracts or the notion of delegation itself.

In the next section, we define *multi-round* contracts and show that they can be used to delegate any continuous function *approximately*. That is, the agent can be incentivized, for all continuous  $f$  and all  $x \in \mathbb{R}^n$ , to reveal some value  $z \in [f(x) - \epsilon, f(x) + \epsilon]$ , where  $\epsilon > 0$  is an arbitrary small number.

The fact that we broaden both the definition of a contract as well as the notion of exact delegation raises the question of whether multi-round mechanisms are really necessary for the approximate delegation of an arbitrary continuous function. In section 5, we prove a generalization of Theorem 2 showing that multi-round mechanisms are necessary. That is, even if we allow approximate answers, we cannot incentivize the agent to reveal an approximate value to  $f(x)$  with a one-round mechanism.

## 4 Approximate Computation

Direct computational contracts are sufficient for delegating boundedly separable functions. But to handle the delegation of arbitrary continuous functions in an approximate manner, we need to define computational contract more generally, so as to allow the agent to report not only (an approximation to) the true answer  $f(x)$ , but some additional evidence that helps the principal compute the reward. In addition, before proving our theorems, we recall a result of Kolmogorov about representing continuous functions, as well as a basic fact about Brier's Scoring Rule.

### 4.1 Preliminaries

**Definition 4.**

A **k-query, T-round computational contract** for a function  $f : [0, 1]^n \rightarrow \mathbb{R}$  is a mechanism  $\mathcal{M}$  specified by

- A collection of functions  $\{\mathcal{D}_t\}_{t=1}^T$  where  $\mathcal{D}_t : \mathbb{R}^{2(t-1)+1} \rightarrow \Delta(D)$  maps a vector of length  $2(t-1) + 1$  to a distribution  $\mathcal{D}_t(m)$  over a finite support  $D$ .
- A function  $\mathcal{S} : \mathbb{R}^{2T} \rightarrow [n]$  mapping a vector of size  $2T$  to a set of input coordinates  $\mathcal{S}(m)$  of size  $k$ ,
- a continuous reward function  $R : \mathbb{R}^{2T} \times \cup_{\ell=1}^n \mathbb{R}^\ell$  mapping, a vector of size  $2T$  and a partial input  $x_{\mathcal{S}}$  to a real number  $R(m, x_{\mathcal{S}})$ .

Such a mechanism  $\mathcal{M} = (\{\mathcal{D}_t\}_{t=1}^T, \mathcal{S}, R)$  has a single player, the agent, and it is played over  $T$  rounds. In each round, the agent sends a message to the mechanism and then receives a random message from the mechanism.

At any round  $t \in \{1, \dots, T\}$ , the information available to the agent consists of the set  $S_A^t$  of all inputs he has queried so far, and of the vector  $m^t$  of all messages exchanged with the mechanism so far. Initially,  $S_A^0 = \emptyset$  and  $m^0$  is the empty vector. A play of the mechanism proceeds as follows.

*Stage 0.* Nature draws  $x \leftarrow X$ . The agent does not observe  $x$ .

*Round  $t$ .* For each  $t \in \{1, \dots, T\}$ , round  $t$  consists of the following stages:

*Stage  $2(t-1) + 1$ .* The agent

- chooses a function  $a_t(\cdot)$  and a subset  $S_{A,t} \subset [n]$ ,
- queries the set  $S_{A,t}$  and updates  $S_A^t = S_A^{t-1} \cup S_{A,t}$ ,
- sends the mechanism the message  $m_t = a_t(x_{S_A^t})$ .

*Stage  $2(t-1) + 2$ .* The mechanism draws a random element  $r_t$  from the distribution  $\mathcal{D}_t(m_1, r_1, \dots, r_{t-1}, m_t)$ . The vector of messages at the end of this round is  $m^t = (m_1, r_1, \dots, m_t, r_t)$ .

At the end of this play, the mechanism queries the set  $S = \mathcal{S}(m^T)$  and pays the agent the reward  $r = R(m^T, x_S)$ . The agent's utility is  $r - c(|S_A^T|)$ .

$\mathcal{M}$  is  $\epsilon$ -incentive compatible for  $f$  if

$$|m_1 - f(x)| \leq \epsilon$$

where the sequence of messages  $m_1, \dots, m_T$  is generated by an agent that, at each round  $t \in \{1, \dots, T\}$  chooses  $a_t(\cdot)$  and  $S_{A,t}$  to maximize his expected utility given the information  $x_{S_A^{t-1}}, m^{t-1}$  available to him at the beginning of the round.

$\mathcal{M}$  is *individually rational* if

$$\max_{a_1(\cdot), S_{A,1}} \mathbb{E}_{x \leftarrow X, r_1 \leftarrow \mathcal{D}_1} \dots \max_{a_T(\cdot), S_A^T} \mathbb{E}_{x \leftarrow X | (x_{S_A^{T-1}}, m^{T-1}), r_T \leftarrow \mathcal{D}_T} [R((m_1, r_1, \dots, m_T, r_T), x_{S(m)})] - c(|S_A^T|) > 0$$

**Kolmogorov's Superposition Theorem [13]** In Theorem 3, we will make use of the following representation of continuous functions over compact sets.

**Theorem ([13]).** Let  $f : [0, 1]^n \rightarrow \mathbb{R}$  be an arbitrary continuous function. Then  $f$  has the representation

$$f(x) = \sum_{q=0}^{2n} \Phi_q \left( \sum_{p=1}^n \psi_{q,p}(x_p) \right) \quad (2)$$

where  $\Phi_q, \psi_{q,p}$  are continuous one-dimensional functions and the functions  $\psi_{q,p}$  are Lipschitz continuous and independent of the function  $f$ .

**A Basic Property of Brier’s Scoring Rule** In our proof of Theorem 3, we will use the following well known property of Brier’s scoring rule on binary distributions

LEMMA 1 *Let  $v, w$  be real numbers in  $[0, 1]$  and let  $V, W$  be random variables over  $\{0, 1\}$  such that  $Pr(V = 1) = v, Pr(W = 1) = w$ . Then*

$$\mathbb{E}_{\omega \leftarrow V} [BSR(V, \omega) - BSR(W, \omega)] = 2(v - w)^2$$

For completeness, the proof is given in Appendix B.

## 4.2 Our Third Theorem

**Theorem 3.** *For all continuous functions  $f : [0, 1]^n \rightarrow \mathbb{R}$  and all  $\epsilon > 0$ , there exists a 1-query, 2-round, computational contract  $\mathcal{M}_2$  that is  $\epsilon$ -incentive compatible and individually rational.*

*Proof.* As in the proof of Theorem 1, we first prove Theorem 3 assuming that the agent’s cost function is identically zero. We then use this result to prove the more general result when the agent’s cost function is arbitrary.

**Case 1:  $\mathbf{c}(\cdot) \equiv \mathbf{0}$ .** Let  $f(x_1, \dots, x_n) = \sum_{q=0}^{2^n} \Phi_q(\sum_{p=1}^n \psi_{q,p}(x_p))$ . Since the functions  $\psi_{q,p}$  are Lipschitz continuous, there exists an  $M$  such that  $|\psi_{q,p}(x) - \psi_{q,p}(x')| < M|x - x'|$  for any  $x, x' \in [0, 1]$ . Furthermore, because of this Lipschitz condition, the family of functions  $\{\psi_{q,p}\}_{q,p}$  has the following two properties

- Uniform boundedness: there exists a constant  $B > 0$  such that  $|\psi_{q,p}(x)| \leq B$  for every  $x \in [0, 1]$  and every  $q, p$ .
- Uniform equicontinuity: for every  $\epsilon > 0$  there exists a  $\delta > 0$  such that  $|x - x'| < \delta$  implies  $|\psi_{q,p}(x) - \psi_{q,p}(x')| < \epsilon$  for every  $q, p$  and every  $x, x' \in [0, 1]$ .

Because of uniform boundedness, we can interpret the domain of the “outer” function  $\Phi_q$  as the compact set  $[-nB, nB]$ . Since any continuous function with a compact domain is bounded and uniformly continuous, we have that each  $\Phi_q$  is bounded and uniformly continuous. Let  $C$  be a bound such that the image of each  $\Phi_q$  is contained in  $[-C, C]$ .

INTUITION. The intuition behind our contract is to interpret  $f(x) = \sum_{q=0}^{2^n} \Phi_q(\sum_{p=1}^n \psi_{q,p}(x_p))$  as a boundedly separable function  $\tilde{f}(w_0, \dots, w_{2^n}) = \sum_{q=0}^{2^n} \Phi_q(w_q)$  where each  $w_q(x) = \sum_{p=1}^n \psi_{q,p}(x_p)$  is itself a function of  $x$ . If the principal knew the value  $w_q$  for a random index  $q$ —then she could use the computational contract from Theorem 1 to incentivize the agent to reveal  $f(x) = \tilde{f}(w) = \sum_{q=0}^{2^n} \Phi_q(w_q)$  using the mechanism  $\mathcal{M}_1$ .

However, the principal does not know the value of  $w_q(x)$ . Since  $w_q(x)$  is itself a boundedly separable function of  $x$ , the principal might attempt the following mechanism

- Round 1. Use mechanism  $\mathcal{M}_1$  to incentivize the agent to reveal  $\tilde{f}(w(x)) = \sum_{q=0}^{2^n} \Phi_q(w_q(x))$ . This mechanism needs to query  $w_q(x)$  for a uniformly random  $q$ . To obtain  $w_q(x)$ , go to round 2.

Round 2. Use mechanism  $\mathcal{M}_1$  to incentivize the agent to reveal the boundedly separable function  $w_q(x) = \sum_{p=1}^n \psi_{q,p}(x_p)$  by querying a uniformly random input coordinate  $p$ .

The problem with this approach is that the agent gets two rewards: one for announcing  $\tilde{f}(w)$  and one for announcing  $w_q(x)$ . Accordingly, it is possible that the agent would lie about  $w_q$  (thus, getting a lower reward in round 2), to manipulate the mechanism in round 1 and receive a higher reward overall.

The way to avoid this problem is to make the reward from round 2 so high that the agent has no incentive in round 2 to reveal a value  $v_q$  whose distance from  $w_q(x)$  is greater than  $\delta$ , for some  $\delta$  that we will choose. We will argue that the agent will be incentivized in step 1 to announce  $\tilde{f}(v_1, \dots, v_n) = \sum_{q=0}^{2^n} \Phi_q(v_q)$ , instead of the true value  $\tilde{f}(w_1, \dots, w_n) = \sum_{q=0}^{2^n} \Phi_q(w_q)$ . Nevertheless, by using the uniform continuity of  $\tilde{f}$ , we will guarantee that the agent's announcement is  $\epsilon$ -close to the true value  $\tilde{f}(w) = f(x)$ .

THE CONTRACT  $\mathcal{M}_2$ . We formalize the above intuition via the following contract and analysis.

MECHANISM  $\mathcal{M}_2 = (\mathcal{D}, \mathcal{S}, R)$

A play of  $\mathcal{M}_2$  proceeds as follows

- In stage 1, the agent announces  $z$ , allegedly in the set  $[f(x) - \epsilon, f(x) + \epsilon]$
- In stage 2, the mechanism announces  $q$ , drawn from  $\mathcal{D}_1 = \text{Uniform}(\{0, \dots, 2n\})$
- In stage 3, the agent announces  $v_q$ , allegedly close to  $\sum_{p=1}^n \psi_{p,q}(x_p)$
- In stage 4, the mechanism announces  $p$ , drawn from  $\mathcal{D}_2 = \text{Uniform}(\{1, \dots, n\})$ .

Given the message vector  $m = (z, q, v_q, p)$ , define

- $\mathcal{S}(m) = \{p\} \subset [n]$ , and
- $R(m, x_{\mathcal{S}(m)})$  the value computed as follows
  - Let  $\omega_1(v_q)$  be a realization of a random variable which is equal to 1 with probability  $(\Phi_q(v_q) + C)/2C$  and equal to 0 with probability  $1 - (\Phi_q(v_q) + C)/2C$ .
  - Let  $\omega_2$  be a realization of a random variable which is equal to 1 with probability  $\psi_{p,q}(x_p)/2B$  and equal to 0 with probability  $1 - \psi_{p,q}(x_p)/2B$ .
  - Interpret  $z$  as a random variable  $\Omega_z$  which is equal to 1 with probability  $(z + (2n+1)C)/((2n+1)2C)$  and equal to 0 with probability  $1 - (z + (2n+1)C)/((2n+1)2C)$
  - Interpret  $v_q$  as a random variable  $\Omega_{v_q}$  which is equal to 1 with probability  $(v_q + nB)/2Bn$  and equal to 0 with probability  $1 - (v_q + nB)/2Bn$ .
  - Return the reward

$$R(m, x_{\mathcal{S}(m)}) = R((z, q, v_q, p), x_p) = BSR(\Omega_z, \omega_1) + \theta \cdot BSR(\Omega_{v_q}, \omega_2)$$

where  $\theta > 0$  is a constant, which we determine later.

$\epsilon$ -INCENTIVE COMPATIBILITY. The reward function depends on  $z$  only through  $BSR(\Omega_z, \omega_1(v_q))$ . Thus—taking his choice of  $v_q$  in stage 3 as given—the agent is incentivized to reveal  $z$  in stage 1 such that the distribution  $\Omega_z$  equals the distribution from which  $\omega_1$  is drawn. We have that

$$Pr(\omega_1 = 1) = \sum_{q=0}^{2n} Pr(q) Pr(\omega_1 = 1|q) = \frac{1}{2n+1} \sum_{q=0}^{2n} \frac{\Phi_q(v_q) + C}{2C}$$



$$= \frac{1}{(2n+1)2C} \left( \sum_{q=0}^{2n} \Phi_q(v_q) + (2n+1)C \right).$$

Since  $Pr(\Omega_z = 1) = (z + (2n+1)C) / ((2n+1)2C)$ , we conclude that  $Pr(\Omega_z = 1) = Pr(\omega_1 = 1)$  if and only if

$$z = \sum_{q=0}^{2n} \Phi_q(v_q).$$

Note that  $z$  is not necessarily the true value  $f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q(\sum_{p=1}^n \psi_{p,q}(x_p))$ , since  $v_q \neq \sum_{p=1}^n \psi_{p,q}(x_p)$ . Note furthermore that the agent *is not incentivized* to announce  $v_q = \sum_{p=1}^n \psi_{p,q}(x_p)$ , since  $v_q$  enters his reward both via  $\Omega_{v_q}$  and via  $\omega_1$ , which is defined in terms of  $v_q$ . While announcing  $v_q \neq \sum_{p=1}^n \psi_{p,q}(x_p)$  always decreases the term  $\theta BSR(\Omega_{v_q}, \omega_2)$  in the agent's reward, it may increase the term  $BSR(\Omega_z, \omega_1)$  enough to make such a deviation profitable.

We now give a bound on how much the agent can profit by announcing  $v_q$  instead of  $w_q = \sum_{p=1}^n \psi_{p,q}(x_p)$ . There are two effects that this deviation has on the expected reward:

- The first term of the expected reward changes by  $\mathbb{E}_{\omega_1}[BSR(\Omega_z, \omega_1(v_q)) - BSR(\Omega_z, \omega_1(w_q))] \leq 3$  since the Brier scoring rule is bounded above by 3 and below by 0.
- Given a fixed  $q$ , by lemma 1, the second term of the expected reward changes by

$$\theta \cdot \mathbb{E}_{\omega_2}[BSR(\Omega_{v_q}, \omega_2) - BSR(\Omega_{w_q}, \omega_2)] = -\frac{2\theta}{(2Bn)^2} (v_q - w_q)^2.$$

Taking expectations over  $q$ , which is drawn uniformly from  $\{0, 1, \dots, 2n\}$ , the total expected loss in reward is  $-2\theta / (2Bn)^2 \mathbb{E} \|v - w\|_2^2$ .

This analysis shows that when the agent reports  $v$  such that  $\|v - w\|_2^2 > \delta$ , the change in his reward is bounded by

$$3 - \frac{2\theta}{(2n+1)(2Bn)^2} \delta.$$

When  $\theta$  is high enough, this change in reward is always negative and the agent always prefers not to announce a value  $v$  far away from  $w$ . In particular, if we set

$$\theta > \frac{3(2n+1)(2Bn)^2}{2\delta},$$

the agent is incentivized to report  $v$  such that  $\|v - w\|_2^2 \leq \delta$ .

Finally, we use the above analysis to show that the principal can incentivize the agent to announce  $z$  such that  $|z - f(x)| < \epsilon$ . For the desired approximation factor  $\epsilon$ , let  $\delta(\epsilon)$  be such that when  $\|v - w\|_2^2 < \delta(\epsilon)$ , we have  $|\sum_{q=0}^{2n} \Phi_q(v_q) - \sum_{q=0}^{2n} \Phi_q(w_q)| < \epsilon$ .<sup>10</sup> Since the agent is incentivized to announce  $v$  such that  $\|v - w\|_2^2 < \delta(\epsilon)$  when  $\theta > (3(2Bn)^2 / 2\delta(\epsilon))$  and he

<sup>10</sup>Note that  $\delta(\epsilon)$  does not depend on  $x$  since  $\sum_{q=0}^{2n} \Phi_q(\cdot)$  is uniformly continuous.

is always incentivized to announce  $z = \sum_{q=0}^{2n} \Phi_q(v_q)$ , we conclude that the agent is always incentivized to announce  $z$  such that  $|z - f(x_1, \dots, x_n)| < \epsilon$ .

**Case 2:  $c(\cdot) \not\equiv 0$**  The above argument only applies when the agent's cost function  $c$  is identically 0. When this is not the case, we prove that we can scale the reward  $R((z, q, v_q, p), x_p)$  by a large enough constant  $\kappa$  so that the agent is incentivized to learn enough inputs from the vector  $(x_1, \dots, x_n)$  in order to provide an  $\epsilon$ -approximation to  $f$ . As mentioned above, once the agent has learned this  $\epsilon$ -approximation, the scoring rule guarantees that he will maximize his reward by reporting it to the mechanism. In contrast with the proof of Theorem 1, the agent

- does not need to learn the whole input  $(x_1, \dots, x_n)$ , since he only needs to give an approximation to  $f(x)$ , and
- learns his set of inputs  $S_A$  in two stages: by querying  $S_{A,1}$  in round 1 of the mechanism and querying  $S_{A,2}(z, q, x_{S_{A,1}})$  in round 2, after making his first-round announcement  $z$ , and learning the mechanism's random choice  $q$  and the partial input  $x_{S_{A,1}}$ . The final set of inputs  $S_A$  that the agent learns is the union  $S_{A,1} \cup S_{A,2}$ .

Our argument proceeds by analyzing the agent's optimization using backwards induction. For any partial inputs  $x_{S_{A,1}}, x_{S_{A,2}}$  that the agent could learn, any  $z$  that the agent announces in stage 1, and any random  $q$  that the mechanism could draw in stage 2, let

$$v_q^*(z, x_{S_{A,1}}, x_{S_{A,2}}) \in \operatorname{argmax}_v \mathbb{E}_{x \leftarrow X | (x_{S_{A,1}}, x_{S_{A,2}}), p \leftarrow \mathcal{D}_2} [R((z, q, v, p), x_p)]$$

$$\nu_2(z, q, x_{S_{A,1}}, S_{A,2}) = \mathbb{E}_{x \leftarrow X | x_{S_{A,1}}, p \leftarrow \mathcal{D}_2} [R((z, q, v_q^*(x_{S_{A,1}}, q, x_{S_{A,2}}), p), x_p)]$$

so that  $\nu_2(z, q, x_{S_{A,1}}, S_{A,2})$  is the reward that the agent ultimately expects to receive when he learns  $x_{S_{A,1}}$  and announces  $z$  in stage 1, receives  $q$  from the mechanism in stage 2, and chooses  $S_{A,2}$  in stage 3. For any  $z, x_{S_{A,1}}$  and any  $q$ , let

$$S_{A,2}^*(z, q, x_{S_{A,1}}) \in \operatorname{argmax}_{S_{A,2}} \nu_2(z, q, x_{S_{A,1}}, S_{A,2}).^{11}$$

Proceeding by backwards induction, for any set  $S_{A,1}$  we define

$$z^*(x_{S_{A,1}}) \in \operatorname{argmax}_z \mathbb{E}_{x \leftarrow X | x_{S_{A,1}}, q \leftarrow \mathcal{D}_1} [\nu_2(z, q, x_{S_{A,1}}, S_{A,2}^*(z, q, x_{S_{A,1}}))],$$

$$\nu_1(S_{A,1}) = \mathbb{E}_{x \leftarrow X, q \leftarrow \mathcal{D}_1} [\nu_2(z^*(x_{S_{A,1}}), q, x_{S_{A,1}}, S_{A,2}^*(z, q, x_{S_{A,1}}))]$$

so that  $\nu_1(S_{A,1})$  is the agent's expected reward when he chooses set  $S_{A,1}$ .

Let  $S_{A,1}^* \in \operatorname{argmax}_S \nu_1(S)$ . From case 1, we can infer that

---

<sup>11</sup>There may be multiple such sets, but we shall argue that this multiplicity does not matter for our argument.

- For any  $S_{A,1}$ , the agent's choice of set  $S_{A,2}^*(z^*(x_{S_{A,1}}), q, S_{A,1})$  in stage 3 will give him enough information to report  $v_q$  such that  $\|v - w\|_2^2 < \delta$ .
- Given that the agent is reporting such a  $v_q$  in stage 3, the agent chooses  $S_{A,1}^*$  in stage 1 to get enough information to learn  $z = z^*(x_{S_{A,1}})$  such that  $z = \sum_{q=0}^{2^n} \Phi_q(v_q)$ .

We now proceed to scale the reward so that, even when there is cost, the agent is always incentivized to choose  $S_{A,1}^*$  in stage 1 and is incentivized to choose  $S_{A,2}^*(\cdot, \cdot, \cdot)$  in stage 3. Choose  $\kappa > 0$  such that

- For any  $S_{A,1} \neq S_{A,1}^*$  and any  $S_{A,2}$ , we have

$$\kappa \cdot \nu_1(S_{A,1}^*) - c(|S_{A,1}^*| + |S_{A,2}|) > \kappa \cdot \nu(S_{A,1}) - c(|S_{A,1}| + |S_{A,2}|)$$

- For any  $z$ , any  $q$  and any  $x_{S_{A,1}}$ , and any  $S_{A,2} \neq S_{A,2}^*(z, q, S_{A,1})$

$$\begin{aligned} \kappa \nu_2(z, q, x_{S_{A,1}}, S_{A,2}^*(z^*(x_{S_{A,1}}), q, S_{A,1})) - c(|S_{A,1}| + |S_{A,2}^*(z, q, S_{A,1})|) > \\ \kappa \nu_2(z, q, x_{S_{A,1}}, S_{A,2}) - c(|S_{A,1}| + |S_{A,2}|) \end{aligned}$$

Such a  $\kappa$  exists because  $S_{A,2}^*(z, q, S_{A,1})$ ,  $S_{A,1}^*$  are chosen to maximize  $\nu_1$  and  $\nu_2$ , respectively. Let  $\tilde{R}(\cdot, \cdot) \stackrel{\text{def}}{=} \kappa \cdot R(\cdot, \cdot)$  be a scaled reward function. Let  $\tilde{\nu}_1 \stackrel{\text{def}}{=} \kappa \cdot \nu_1$ ,  $\tilde{\nu}_2 \stackrel{\text{def}}{=} \kappa \cdot \nu_2$ . Then, by our choice of  $\kappa$ , we have

- For any  $S_{A,1} \neq S_{A,1}^*$  and any  $S_{A,2}$ , we have

$$\tilde{\nu}_1(S_{A,1}^*) - c(|S_{A,1}^*| + |S_{A,2}|) > \tilde{\nu}(S_{A,1}) - c(|S_{A,1}| + |S_{A,2}|)$$

- For any  $z$ , any  $q$  and any  $x_{S_{A,1}}$ , and any  $S_{A,2} \neq S_{A,2}^*(z^*(x_{S_{A,1}}), q, S_{A,1})$

$$\begin{aligned} \tilde{\nu}_2(z, q, x_{S_{A,1}}, S_{A,2}^*(z^*(x_{S_{A,1}}), q, S_{A,1})) - c(|S_{A,1}| + |S_{A,2}^*(z^*(x_{S_{A,1}}), q, S_{A,1})|) > \\ \tilde{\nu}_2(z, q, x_{S_{A,1}}, S_{A,2}) - c(|S_{A,1}| + |S_{A,2}|). \end{aligned}$$

Thus, by changing the reward of function of  $\mathcal{M}_2$  to be  $\kappa \cdot R$ , we can incentivize the agent to gather enough information to be able to report  $v$  such that  $\|v - w\|_2^2 < \delta$  and  $z$  such that  $|z - f(x)| < \epsilon$ .

We conclude by noting that

- We can further increase  $\kappa$  to ensure individual rationality.
- Even if there are multiple optimal choices for  $S_{A,1}^*$  and  $S_{A,2}^*$ , all such choices guarantee that the agent gathers enough information to report  $z$  within  $\epsilon$  of  $f(x)$ .

*Q.E.D.*

### 4.3 Our Fourth Theorem

Our mechanism  $\mathcal{M}_2$  queries only one coordinate  $x_p$  from the input vector, and uses two rounds of interaction between the principal and the agent. This raises the question of whether there exist *one round* contracts which can be used to approximately delegate continuous functions and query few inputs. We show in the following theorem that this cannot be the case as long as the reward function is concave in the agent’s reported value  $z$ . We remark that the reward function  $R((z, q, v_q, p), x_p)$  used in Theorem 3 is concave in the agent’s reported value  $z$ , but it is not concave as a function of the agent’s additional message  $v_q$ .

**Theorem 4.** *There exists a continuous function  $f : [-1, 1]^n \rightarrow \mathbb{R}$  such that every one-round,  $\epsilon$ -incentive compatible contract  $\mathcal{M} = (\mathcal{D}, R)$  with a concave reward function must query the whole input  $(x_1, \dots, x_n)$ .*

The proof of Theorem 4 is given in appendix C.

## 5 Discussion

**Optimality and Unlimited Liability** In the above discussions we have been careful to minimize the number of queries that the mechanism makes to the input (as well as the number of interaction rounds between the principal and the agent). In addition to the query cost, the principal of course bears the monetary cost of paying the agent’s reward. As we mentioned in the introduction, this monetary cost can be minimized for any of our mechanisms when the agent has unlimited liability. The principal can simply charge the agent a fixed fee equal to the agent’s expected utility from participating in the mechanism. Since this fixed fee can be computed using expectations over  $X$  (rather than querying the actual input  $x$ ), the principal can compute this reward at no extra cost. Since the principal minimizes her expected payment to the agent and the number of queries she makes, our contracts are optimal in the unlimited liability scenario.

In the classical setting, this type of argument is known as “selling the firm” [11] and makes the problem trivial. In this setting, the problem is still non-trivial because the principal still needs to be convinced that the agent has computed a correct approximation to  $f(x)$  and therefore needs to query some set  $S$  of inputs in order to verify the agent’s answer. Thus, even when we sell the firm and minimize the principal’s monetary cost, the problem of minimizing the principal’s query cost still stands.

**Beyond  $\epsilon$ -incentive compatibility** In this paper, we focused on the case where the principal wants to obtain a value  $z$  arbitrarily close to the true value of  $f(x)$ . While we believe that this is an important model for delegation of computation, it is not the only one. For instance, one may also investigate various tradeoffs between the quality of the agent’s answer  $z$ , and the cost of obtaining this answer.

**Accounting for Algorithmic Cost** In our results we assumed that data collection is expensive but, once the data is available, running an algorithm on the collected data is free. An advantage of this model over other ways of measuring computational complexity is that we can precisely analyze how many pieces of data a principal has to observe in order to incentivize the agent to evaluate a continuous function of the entire dataset. Indeed, we showed that the principal only needs to observe *a single piece* of data.

An alternative approach would be to assume that the data is already available (and thus cost free), while running algorithms on the available data has a cost that increases with the number of operations performed. We are also interested in delegating computation to a rational agent in this model. We believe that doing so is likely to require techniques from computer science, such as computationally sound interactive proof systems [18] [12]—where the agent gives a verifiable evidence to the principal that his answer is correct. This verifiable evidence guarantees to the principal that no malicious agent (who may go out of his way to cheat her) will be able to deceive her. Such guarantees are very strong but require very complicated protocols. We believe that by properly modeling the rationality of the agent, as we have done in this paper, we can vastly simplify these protocols.

## Acknowledgements

We thank Gabriel Carroll, Tommaso Denti, Harry Di Pei, Juuso Toikka and Vira Semenova for productive discussions, as well as the referees and editors for their helpful suggestions. We give special thanks to Georgios Vlachos for his input on the proof of Theorem 2. Pablo D. Azar is grateful for financial support from the Robert Solow Fellowship and the Stanley and Rhoda Fischer Fellowship.

## References

- [1] Nabil I Al-Najjar, Luca Anderlini and Leonardo Felli Undescribable Events. *The Review of Economic Studies*, 73(4): 849-868, 2006.
- [2] Luca Anderlini and Leonardo Felli Incomplete written contracts: Undescribable states of nature. *The Quarterly Journal of Economics*, 109(4):1085-1124, 1994.
- [3] Pablo D. Azar and Silvio Micali Rational Proofs. *Symposium on the Theory of Computing*. 44:1017-1028, 2012.
- [4] Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 1950.
- [5] Gabriel D. Carroll Robust Incentives for Information Acquisition *Manuscript, 2016*. <http://web.stanford.edu/~gdc/infoscore.pdf>.
- [6] Robert T. Clemen Incentive contracts and strictly proper scoring rules. *Test* 11(1): 167-189, 2002.

- [7] Joel S. Demski and David E.M. Sappington Delegated Expertise *Journal of Accounting Research* 25(1):68-89 ,1987.
- [8] Tilmann Gneiting and Adrain E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- [9] Irving J. Good. Rational Decisions. *Journal of the Royal Statistical Society*, Ser. B 14:107-114.
- [10] Siyao Guo, Pavel Hubáček, Alon Rosen and Margarita Vald. Rational Sumchecks. *Theory of Cryptography* 13:319-351, 2016.
- [11] Bengt Holmstrom. Moral hazard and observability. *The Bell Journal of Economics*, 10(1):74-91,1979.
- [12] Joe Kilian. A note on efficient zero-knowledge proofs and arguments. *ACM symposium on Theory of computing* 24: 723–732. ACM, 1992.
- [13] Andrey N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Amer. Math. Soc. Transl* 28: 55-59,1963.
- [14] Nicolas S. Lambert. Elicitation and evaluation of statistical forecasts. *Manuscript*, 2011.
- [15] CERN Computing <http://home.web.cern.ch/about/computing>
- [16] James M. Malcolmson. Principal and Expert Agent *B.E. Journal of Theoretical Economics*, 9(1): Article 17,2009.
- [17] William B. MacLeod. Complexity and contract. *Revue d'économie Industrielle* 92(1): 149-178, 2000.
- [18] Silvio Micali. Computationally sound proofs. *SIAM Journal on Computing*, 30(4):1253–1298, 2001.
- [19] Kent Osband. Optimal forecasting incentives. *The Journal of Political Economy* 97(5):1091-1112, 1989.
- [20] Luis Zermeno. Essays on the Principal-Expert Problem. *Doctoral Dissertation*,2012. <http://hdl.handle.net/1721.1/72843>

## Appendix

## A Proof of Theorem 2

**Theorem 2.** *There exists a continuous, bounded and non-separable function  $f : [-1, 1]^n \rightarrow \mathbb{R}$  such that any direct computational contract  $\mathcal{M} = (\mathcal{D}, R)$  that is incentive compatible for  $f$  must query the whole input vector  $(x_1, \dots, x_n)$ .*

*Proof.* We consider the function  $f(x_1, \dots, x_n) = x_1 x_2 \dots x_n$ . Assume that there is a direct computational contract  $\mathcal{M} = (\mathcal{D}, R)$  that is incentive compatible for  $f$  and queries  $n - 1$  inputs. Let  $S_{-i} = \{1, \dots, n\} - \{i\}$  and let  $x_{-i} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$ .

We can write the function  $f(x_1, \dots, x_n) = x_1 x_2 \dots x_n$  as

$$x_1 x_2 \dots x_n = \operatorname{argmax}_z \sum_{i=1}^n \operatorname{Prob}(\mathcal{D} = S_{-i}) R(z, x_{-i}) \quad (3)$$

Let  $\phi_{-i}(x_{-i}, z) = \operatorname{Prob}(\mathcal{D} = S_{-i}) R(x_{-i}, z)$  and let  $\rho(x_1, \dots, x_n, z) = \sum_{i=1}^n \phi_{-i}(x_{-i}, z)$ . Then we can write  $f(x_1, \dots, x_n)$  as

$$x_1 \dots x_n = \operatorname{argmax}_z \rho(x_1, \dots, x_n, z). \quad (4)$$

Now consider the set  $A = \{-1, 1\}^n = \{(x_1, \dots, x_n) : x_i \in \{1, -1\}\}$  and notice that  $f(x_1, \dots, x_n) \in \{1, -1\}$  for any vector  $x \in A$ . Let  $A_1 = \{x \in A : f(x) = 1\}$ ,  $A_{-1} = \{x \in A : f(x) = -1\}$ . For any vector  $x \in A_1$ , we must have that  $1 = f(x) = \operatorname{argmax}_z \rho(x, z)$ , so that  $\rho(x, 1) > \rho(x, -1)$ . Similarly, for any  $x \in A_{-1}$ , we must have that  $\rho(x, -1) > \rho(x, 1)$ .

The above argument implies that

$$\sum_{x \in A_1} \rho(x, 1) > \sum_{x \in A_1} \rho(x, -1) \quad (5)$$

$$\sum_{x \in A_{-1}} \rho(x, -1) > \sum_{x \in A_{-1}} \rho(x, 1) \quad (6)$$

We proceed to obtain a contradiction by showing that

$$\sum_{x \in A_1} \rho(x, 1) = \sum_{x \in A_{-1}} \rho(x, 1) \quad (7)$$

$$\sum_{x \in A_1} \rho(x, -1) = \sum_{x \in A_{-1}} \rho(x, -1) \quad (8)$$

Note that for every  $x \in A_1$  and index  $i \in \{1, \dots, n\}$ , there exists a unique  $y(x, i) \in A_{-1}$  such that  $y_{-i}(x, i) = x_{-i}$ . Such a  $y(x, i)$  can be constructed by setting  $y_j(x, i) = x_j$  for all  $j \neq i$  and  $y_i(x, i) = -x_i$ . It is clear from this construction that  $x_1 \dots x_n = -y_1 \dots y_n$ , and since

$x_1 \dots x_n = 1$  we must have  $y_1 \dots y_n = -1$ . For a fixed  $i$ , it is clear that the function mapping  $x$  to  $y(x, i)$  is a bijection between  $A_1$  and  $A_{-1}$ .

To show that equation 7 holds, write

$$\begin{aligned} \sum_{x \in A_1} \rho(x, 1) &= \sum_{x \in A_1} \sum_{i=1}^n \phi_{-i}(x_{-i}, 1) = \sum_{i=1}^n \sum_{x \in A_1} \phi_{-i}(x_{-i}, 1) = \\ \sum_{i=1}^n \sum_{x \in A_1} \phi_{-i}(y_{-i}(x, i), 1) &= \sum_{i=1}^n \sum_{y \in A_{-1}} \phi_{-i}(y_{-i}, 1) = \sum_{y \in A_{-1}} \sum_{i=1}^n \phi_{-i}(y_{-i}, 1) = \sum_{y \in A_{-1}} \rho(y, 1). \end{aligned} \quad (9)$$

Analogously, we can show equation 8 holds. Now note that from equations 5,6,7 and 8 we can derive the contradiction

$$\sum_{x \in A_1} \rho(x, 1) > \sum_{x \in A_1} \rho(x, -1) = \sum_{x \in A_{-1}} \rho(x, -1) > \sum_{x \in A_{-1}} \rho(x, 1) = \sum_{x \in A_1} \rho(x, 1). \quad (10)$$

This contradiction arose from assuming that  $x_1 \dots x_n$  could be written in the form of equation 3. Thus, we conclude that  $x_1 \dots x_n$  cannot be delegated with an  $(n - 1)$ -query direct revelation contract.

*Q.E.D.*

To prove this theorem we have used the fact that the contract is *strictly* incentive-compatible. The proof would still work with weakly incentive-compatible contracts.<sup>12</sup>

## B Proof of Lemma 1

**Lemma 1.** *Let  $v, w$  be real numbers in  $[0, 1]$  and let  $V, W$  be random variables over  $\{0, 1\}$  such that  $Pr(V = 1) = v, Pr(W = 1) = w$ . Then*

$$\mathbb{E}_{\omega \leftarrow V} [BSR(V, \omega) - BSR(W, \omega)] = 2(v - w)^2$$

---

<sup>12</sup>We illustrate this informally. Suppose that the distribution of inputs  $X$  is such that  $x \in A_1$  probability  $1 - \epsilon$  and  $x \in A_{-1}$  with probability  $\epsilon$ . Suppose furthermore, that the contract is weakly incentive-compatible so that for any  $x \in A_1$  and any  $\tilde{x} \in A_{-1}$  we have

$$\begin{aligned} \mathbb{E}_{S \leftarrow \mathcal{D}} [R(f(x), x_S)] &\geq \mathbb{E}_{S \leftarrow \mathcal{D}} [R(f(\tilde{x}), x_S)] \\ \mathbb{E}_{S \leftarrow \mathcal{D}} [R(f(\tilde{x}), \tilde{x}_S)] &\geq \mathbb{E}_{S \leftarrow \mathcal{D}} [R(f(x), \tilde{x}_S)]. \end{aligned}$$

If any one of these equalities holds strictly, we derive a contradiction just as in our proof above. If both hold as equalities, then the contract cannot be weakly-incentive compatible because the agent knows that the value of  $f$  is equal to  $f(x) = 1$  with extremely high probability and therefore will maximize his expected payoff by always reporting the value 1 without querying any inputs, and thus obtaining the same reward with less cost.



*Proof.* Note that we have  $\|V\|_2^2 = v^2 + (1 - v)^2$  and  $\|W\|_2^2 = w^2 + (1 - w)^2$ . We have

$$\begin{aligned} \mathbb{E}_{\omega \leftarrow V} BSR(V, \omega) &= Pr(\omega = 1)(2Pr(V = 1) - \|V\|_2^2 + 1) + Pr(\omega = 0)(2Pr(V = 0) - \|V\|_2^2 + 1) \\ &= 2v^2 + 2(1 - v)^2 - v^2 - (1 - v)^2 + 1 = v^2 + (1 - v)^2 + 1 = 2v^2 - 2v + 2. \end{aligned}$$

We also have

$$\begin{aligned} \mathbb{E}_{\omega \leftarrow V} BSR(W, \omega) &= Pr(\omega = 1)(2Pr(W = 1) - \|W\|_2^2 + 1) + Pr(\omega = 0)(2Pr(W = 0) - \|W\|_2^2 + 1) \\ &= 2vw + 2(1 - v)(1 - w) - w^2 - (1 - w)^2 + 1 = 4vw + 2 - 2v - 2w^2. \end{aligned}$$

Taking differences, we have

$$\mathbb{E}_{\omega \leftarrow V} [BSR(V, \omega) - BSR(W, \omega)] = 2(v^2 + w^2) - 2v - 4vw + 2v = 2v^2 + 2w^2 - 4vw = 2(v - w)^2$$

*Q.E.D.*

## C Proof of Theorem 4

The following lemma will be useful for the proof of Theorem 4.

**Lemma 2.** *Let  $\phi_1, \dots, \phi_n : \mathbb{R} \rightarrow \mathbb{R}$  be continuous concave functions, and let  $\phi : \mathbb{R} \rightarrow \mathbb{R}$  be a function such that  $\phi(z) = \sum_{i=1}^n \phi_i(z)$ . Let  $z_i^* = \operatorname{argmax}_z \phi_i(z)$  and assume that there exists an interval  $[a, b]$  such that for all  $i$ ,  $z_i^* \in [a, b]$ . Then*

$$\operatorname{argmax}_z \phi(z) \in [a, b].$$

*Proof.* Since each  $\phi_i$  is continuous and concave, it is superdifferentiable everywhere.<sup>13</sup> Furthermore—since each  $\phi_i$  is maximized in the interval  $[a, b]$  for each  $i$  there exist

- a supergradient  $\alpha_i \geq 0$  such that  $\phi_i(z) - \phi_i(a) \leq \alpha_i(z - a)$  for every  $z \in \mathbb{R}$ ,
- a supergradient  $\beta_i \leq 0$  such that  $\phi_i(z) - \phi_i(b) \leq \beta_i(z - b)$  for every  $z \in \mathbb{R}$ .

Adding up over all  $i$ , we conclude that there exist supergradients  $\alpha = \sum_{i=1}^n \alpha_i \geq 0$ ,  $\beta = \sum_{i=1}^n \beta_i \leq 0$  for  $\phi$  such that  $\phi(z) - \phi(a) \leq \alpha(z - a)$  and  $\phi(z) - \phi(b) \leq \beta(z - b)$  for every  $z \in \mathbb{R}$ . Since  $\phi$  is concave, this implies that there exists a point  $z^* \in [a, b]$  such that 0 is a supergradient of  $\phi$  at  $z^*$ . That is  $\phi(z) - \phi(z^*) \leq 0(z - z^*)$  for every  $z \in \mathbb{R}$ . This implies that

$$\phi(z) \leq \phi(z^*)$$

for every  $z \in \mathbb{R}$ . Therefore, the maximum of  $\phi$  is in the interval  $[a, b]$ . *Q.E.D.*

<sup>13</sup>A function  $f : \mathbb{R} \rightarrow \mathbb{R}$  is superdifferentiable at  $x$  if there exists a real number  $c$  such that for every  $z \in \mathbb{R}$  we have  $f(z) - f(x) \leq c(z - x)$ .

**Theorem 4.** *There exists a continuous function  $f : [-1, 1]^n \rightarrow \mathbb{R}$  such that every one-round,  $\epsilon$ -incentive compatible contract  $\mathcal{M} = (\mathcal{D}, R)$  with a concave reward function must query the whole input  $(x_1, \dots, x_n)$ .*

*Proof.* As in the proof of Theorem 2, we consider the function  $f(x_1, \dots, x_n) = x_1 x_2 \dots x_n$ . Let  $\epsilon < 1$  and assume that an  $(n - 1)$ -query,  $\epsilon$ -incentive compatible contract  $\mathcal{M} = (\mathcal{D}, R)$  exists for delegating  $f$ . We can then write the agent's expected reward from sending a message  $m$  as

$$\sum_{i=1}^n \text{Prob}(\mathcal{D} = S_{-i}) R(m, x_{-i}) = \sum_{i=1}^n \phi_{-i}(m, x_{-i}). \quad (11)$$

where  $\phi_{-i}(m, x_{-i}) = \text{Prob}(\mathcal{D} = S_{-i}) R(m, x_{-i})$ . Note that each  $\phi_{-i}(m, x_{-i})$  is concave. Therefore

$$\rho(m, x) = \sum_{i=1}^n \phi_{-i}(m, x_{-i}) \quad (12)$$

will also be concave.

Define the set  $A = \{-1, 1\}^n = \{(x_1, \dots, x_n) : x_i \in \{1, -1\}\}$  and notice that  $f(x_1, \dots, x_n) \in \{1, -1\}$  for any vector  $x \in A$ . Let  $A_1 = \{x \in A : f(x) = 1\}$ ,  $A_{-1} = \{x \in A : f(x) = -1\}$ . Recall that for every  $x \in A_1$  and every index  $i \in \{1, \dots, n\}$ , there exists a unique  $y(x, i) \in A_{-1}$  such that  $x_{-i} = y_{-i}(x, i)$ . The key to proving Theorem 4 is using this bijection to derive the following equation for any possible message  $m$ .

$$\sum_{x \in A_1} \rho(m, x) = \sum_{y \in A_{-1}} \rho(m, y). \quad (13)$$

This equation holds because

$$\begin{aligned} \sum_{x \in A_1} \rho(m, x) &= \sum_{x \in A_1} \sum_{i=1}^n \phi_{-i}(m, x_{-i}) = \sum_{i=1}^n \sum_{x \in A_1} \phi_{-i}(m, x_{-i}) = \\ &= \sum_{i=1}^n \sum_{x \in A_1} \phi_{-i}(m, y_{-i}(x, i)) = \sum_{i=1}^n \sum_{y \in A_{-1}} \phi_{-i}(m, y_{-i}) = \sum_{y \in A_{-1}} \sum_{i=1}^n \phi_{-i}(m, y_{-i}) = \sum_{y \in A_{-1}} \rho(m, y). \end{aligned}$$

We now show that equation 13 leads to a contradiction. Since  $\rho(m, x)$  is a continuous and concave function, there exists a continuous function  $m(x)$  such that  $m(x) = \text{argmax}_m \rho(m, x)$ . Since the mechanism is  $\epsilon$ -incentive compatible,  $m(x) \in [f(x) - \epsilon, f(x) + \epsilon]$ . In particular, for any  $x \in A_1$  we must have  $m(x) \in [1 - \epsilon, 1 + \epsilon]$  and for any  $y \in A_{-1}$  we must have  $m(y) \in [-1 - \epsilon, -1 + \epsilon]$ .

The sum  $\sum_{x \in A_1} \rho(m, x)$  is itself a concave function of  $m$ . Let  $m_1^* = \text{argmax}_m \sum_{x \in A_1} \rho(m, x)$ . Since each  $m(x) = \text{argmax}_m \rho(m, x)$  belongs to the interval  $(1 - \epsilon, 1 + \epsilon)$ , lemma 2 tells us that

$m_1^* \in [1 - \epsilon, 1 + \epsilon]$ . Analogously, the same lemma tells us that  $m_{-1}^* = \operatorname{argmax}_m \sum_{y \in A_1} \rho(m, y)$  must belong in the interval  $[-1 - \epsilon, -1 + \epsilon]$ .

But equation 13 tells us that

$$\begin{aligned} \sum_{x \in A_1} \rho(m_1^*, x) &= \sum_{y \in A_{-1}} \rho(m_1^*, y) \\ \sum_{y \in A_{-1}} \rho(m_{-1}^*, y) &= \sum_{x \in A_1} \rho(m_{-1}^*, x) \end{aligned}$$

The strict concavity of  $\rho$  implies that  $\sum_{y \in A_{-1}} \rho(m_{-1}^*, y) > \sum_{y \in A_{-1}} \rho(m_1^*, y)$  and  $\sum_{x \in A_1} \rho(m_1^*, x) > \sum_{x \in A_1} \rho(m_{-1}^*, x)$ . Putting these equations together we derive the contradiction

$$\sum_{y \in A_{-1}} \rho(m_{-1}^*, y) = \sum_{x \in A_1} \rho(m_{-1}^*, x) < \sum_{x \in A_1} \rho(m_1^*, x) = \sum_{y \in A_{-1}} \rho(m_1^*, y) < \sum_{y \in A_{-1}} \rho(m_{-1}^*, y).$$

From this contradiction, we conclude that an  $(n - 1)$ -query,  $\epsilon$ -incentive compatible contract for  $f(x_1, \dots, x_n) = x_1 \dots x_n$  does not exist. *Q.E.D.*

As a final remark, note that we do not need  $f$  to be symmetric for Theorem 4 to hold. The proof tells us that there exists a *symmetric function*  $f$  such that any one-round  $\epsilon$ -incentive compatible contract for  $f$  requires the principal to query all  $n$  coordinates of  $x$ . However, we can construct a non-symmetric perturbation  $\tilde{f}$  of  $f$  for which Theorem 4 also holds. Let  $g(x)$  be a non-symmetric continuous function such that  $0 \leq g(x) \leq 1$ . Let  $\tilde{f}(x) = f(x) + \epsilon/2 \cdot g(x)$ . Then we have that  $\tilde{f}$  is not symmetric, and that  $|f(x) - \tilde{f}(x)| \leq \epsilon/2$ . Even though  $\tilde{f}$  is not symmetric, there does not exist a  $\epsilon/2$ -incentive compatible one-round computational contract  $(\tilde{\mathcal{D}}, \tilde{\mathcal{R}})$  for  $\tilde{f}$  which queries  $k < n$  coordinates of  $x$ . If such a contract existed, then it would also be an  $\epsilon$ -incentive compatible contract for  $f$ . This is because the contract would incentivize the agent to reveal some value  $\tilde{m}(x)$  such that  $|\tilde{m}(x) - \tilde{f}(x)| \leq \epsilon/2$ . The revealed value  $\tilde{m}(x)$  would also satisfy  $|\tilde{m}(x) - f(x)| \leq \epsilon$ .

We conclude that there is a very general class of functions for which no one-round mechanisms that make  $k < n$  queries can be  $\epsilon$ -incentive compatible.