Multiagent Evaluation Mechanisms

Tal Alon
Technion, Israel

Magdalen Dobson
Carnegie Mellon University, USA

Ariel D. Procaccia
Carnegie Mellon University, USA

Inbal Talgam-Cohen
Technion, Israel

Jamie Tucker-Foltz
University of Cambridge, UK

Abstract

We consider settings where agents are evaluated based on observed features, and assume they seek to achieve feature values that bring about good evaluations. Our goal is to craft evaluation mechanisms that incentivize the agents to invest effort in desirable actions; a notable application is the design of course grading schemes. Previous work has studied this problem in the case of a single agent. By contrast, we investigate the general, multi-agent model, and provide a complete characterization of its computational complexity.

1 Introduction

Any reader who has ever taught a course would have undoubtedly faced some variant of the grading-scheme dilemma: Should the final exam count for 30% of the grade, and the homework assignments for 40%? Or should these two components perhaps be weighted equally? Should the lowest homework grade be dropped? Admittedly, grades only serve as a proxy for students' (unobservable) learning outcomes. But once a grading scheme is in place, students will optimize their grades by investing effort accordingly. Therefore, the grading scheme must be designed to encourage desirable behaviors. For example, a participation grade may make some students come to class, but those same students — who are now short on time — may elect to cheat on their homework assignments.

The design of course grading schemes is an instance of a much broader challenge. Whenever an *evaluator* designs a scheme for evaluating an *agent* based on observed features, the agent is incentivized to achieve feature values that lead to a good evaluation. The hope is that the agent will do so through genuine self-improvement rather than blatant gaming. The evaluation of creditworthiness through credit history in the United States serves as an especially egregious example: instead of promoting true financial responsibility, it encourages idiosyncratic practices such as using a specific percentage of one's credit card limit.

In a very recent paper, Kleinberg and Raghavan (2019) model and analyze these scenarios. In their model, an agent has a given amount of *effort* that can be invested in different *actions* (e.g., attempt to solve a homework assignment or cheat). There are also *effort conversion* functions that map the levels of effort invested in each action to *features* (e.g., homework grade, exam grade, or participation grade). The evaluator's task is to design a *mechanism* that maps the feature values to a score, which coincides with the agent's *utility*. The agent seeks to distribute effort between actions to achieve maximum utility. The evaluator's goal, then, is to design the mechanism to elicit a desirable *effort profile*.

Importantly, an instance of the evaluation problem of Kleinberg and Raghavan (2019) includes only a single agent — or, equivalently, multiple agents that share the same model. However, in the domains of interest there are multiple *types* of agents. For example, one student might optimize exam grades by studying alone, whereas another would derive more benefit from studying with peers. With this in mind, we wish to extend the model and results of Kleinberg and Raghavan to the multi-agent case. In other words, our main research challenge is this:

> *Given a set of different agent models, design mechanisms that induce all agents, or as many agents as possible, to invest effort in desirable actions.*

1.1 Our Results

The foregoing challenge gives rise to multiple problems, each of which is determined by the answers to the following three questions.

First, what requirements must the mechanism satisfy? The minimal assumption made by Kleinberg and Raghavan is that the mechanism must be *monotone*, meaning that agents should receive higher payoffs for higher feature scores. A more restrictive requirement is that the mechanism be *linear*, meaning that the payoff is just a linear combination of all feature scores.

Second, what is the goal of the evaluator? Is there some specific *admissible profile* of effort investment the evaluator wishes to incentivize, or will they be content with any profile, as long as all of the actions which the agent invests a nonzero amount of effort into are *admissible actions*?

Third, are we interested in incentivizing *all agents* in a particular way, or just a *maximum number of agents*?

With only one agent, the main result of Kleinberg and Raghavan is that the answers to these questions do not matter: whenever a monotone mechanism exists a linear mechanism exists, and whether an effort profile can be incentivized

Problem #	Type of mechanism	Incentivize	Admissible set	Complexity
1	monotone	all agents	admissible profile	P
2	monotone	all agents	admissible actions	P (const. n) NP-c (gen.)
3	monotone	max # of agents	admissible profile	NP-c
4	monotone	max # of agents	admissible actions	NP-c
5	linear	all agents	admissible profile	P
6	linear	all agents	admissible actions	P (const. n) NP-c (gen.)
7	linear	max # of agents	admissible profile	P (const. n) NP-c (gen.)
8	linear	max # of agents	admissible actions	P (const. n) NP-c (gen.)

Table 1: Complexity of the 8 different variants of the evaluation problem. Note that n is the number of features, and "NP-c" stands for NP-complete.

depends only on the actions it is supported by. However, with multiple agents, we find striking differences between the problem variants, both qualitatively and in terms of computational complexity. We provide a complete classification of the complexity of each of these 8 problems, as shown in Table 1. For each problem, we also consider the complexity for the realistic restriction of a constant number of features n. (For example, even in MOOCs with massively-many students, the number of features factored into the final grade is likely to be held constant.) Problems 1-4 are analyzed in Section 3, and Problems 5-8 are analyzed in Section 4.

1.2 Related Work

There are two main lines of related work. First, evaluation can be viewed as *classifying* strategic agents (into classes such as "A students", "B students", etc.). Self-interested agents facing classification may invest in distorting their true attributes, in order to steer the classifier away from their "ground-truth" class. The goal in *strategic classification* is to build classifiers robust to such *gaming* (Hardt et al. 2016). Our goal is in some sense opposite to this line of work — we aim to *encourage* agents to invest in changing their features, but by choosing desirable actions like studying over undesirable ones like cheating. In other words, in our case the evaluation is not meant to expose some ground truth, but rather to incentivize worthwhile behavior. Strategic classification is part of a more general literature on learning in the presence of strategic behavior (Meir, Procaccia, and Rosenschein 2008; 2012; Dekel, Fischer, and Procaccia 2010).

A second line of research closely related to our work is *contract design*, a branch of microeconomics (Grossman and Hart 1983) that has recently gained interest in computer science (Babaioff, Feldman, and Nisan 2006; Dütting, Roughgarden, and Talgam-Cohen 2019). The precise relation between our model and the classic principal-agent, hidden-action[1] model from microeconomics is explained in Appendix A.[2] In a nutshell, the basic setting of our model can be reinterpreted as a simplified principal-agent one, in which the principal (the evaluator in our model) has no inherent interest in the agents' outputs except to incentivize the agents to choose permissible hidden actions. Given the connection between the models, to avoid confusion it is important to note here that we use the term *linear mechanism* for an entirely different object than the *linear contract* term that is standard in microeconomics — see the appendix for details.[3] We also diverge from previous work on contract design for multiple agents in our motivation for applying a *unified* approach to incentivizing the agents, instead of dealing with each of them separately: rather than aiming to encourage cooperation or optimize information as is common in the contract design literature, we are motivated by the fairness requirement that all agents face a single uniform evaluation mechanism (see the appendix for more details).

In parallel work, Xiao et al. (2020) also study the problem of incentivizing multiple agents under a single mechanism. In their model, actions are directly observable, and in designing the contract, the principal is motivated by profit and has to compensate the agents at personal expense. Hence, their model applies to an entirely different set of principal-agent problems than ours.

2 The Model

For consistency we adopt the notation of Kleinberg and Raghavan (2019) where possible. An instance of the *evaluation problem* consists of actions $1, 2, \ldots, m$ (indexed by j); features $F_1, F_2, \ldots, F_n$ (indexed by i); and agents (e.g., students) $S = \{s_1, s_2, \ldots, s_\ell\}$ (indexed by k). Each agent s_k has a matrix α^k in $\mathbb{R}_{\geq 0}^{m \times n}$ called their *effort conversion matrix*. Entry $\alpha_{j,i}^k \in \mathbb{R}_{\geq 0}$ (which we assume is described using a polynomial number of bits in m, n) specifies how effort put into action j translates into feature i (as specified in the next paragraph). We assume that every agent s_k has the ability to affect every feature, that is, no matrix α^k has an all-zero column.[4] We denote the j^{th} row and its entries by $\alpha_j^k = (\alpha_{j,1}^k, \ldots, \alpha_{j,n}^k)$. It is often convenient to describe an instance of the evaluation problem as an *effort graph* as depicted in Figure 1. The instance in Figure 1 has $m = 2$ actions, $n = 2$ features and $\ell = 2$ agents, and the conversion matrices are $\alpha_1^1 = (4, p), \alpha_2^1 = (0, 9)$ for the first agent and $\alpha_1^2 = (p, 4), \alpha_2^2 = (9, 0)$ for the second.

Each agent s_k has a budget of one unit of effort to divide

[1] As opposed to hidden-type models; note that while we deal with different types of agents, these are not hidden.

[2] The appendix is included in the full version of our paper, available at `http://procaccia.info`.

[3] We use the term linear mechanism to be consistent with (Kleinberg and Raghavan 2019).

[4] This is implicitly assumed in (Kleinberg and Raghavan 2019).

among different actions.[5] Their choice of how to divide their budget is specified by their *effort profile* x^k, where $x^k_j \geq 0$ (or x_j — we sometimes omit the k index where clear from context) is the effort they invest in action j, and $\sum_j x^k_j \leq 1$ (*feasibility*). We refer to the set of all feasible effort profiles as $\mathcal{X}$.

An effort profile is converted into the agent's n features as follows: $F^k(x^k)_i = \sum_j \alpha^k_{j,i} x^k_j$ for every $i \in [n]$. In words, for feature F_i, the effort s_k puts into action j is multiplied by $\alpha^k_{j,i}$, and this is summed over all actions. Note that Kleinberg and Raghavan (2019) introduce a generalization: they define $F^k(x^k)_i = f^k_i(\sum_j \alpha^k_{j,i} x^k_j)$, where f^k_i is a concave, strictly increasing function. We use the simpler form for ease of exposition, and indeed most of our results hold for the more general model — see Appendix I for details.

An *evaluation mechanism* is a function $M : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ that maps an agent's features to a score or *payoff*. The score coincides with the agent's utility. Given a mechanism, the agent chooses an effort profile x^k that maximizes their score; we then say x^k is *incentivized* by the mechanism. The design goal is to have the mechanism incentivize all agents, or as many agents as possible, to invest only in a prescribed set of *admissible* profiles (we assume that if several profiles are incentivized, the agent breaks ties in favor of admissible ones). We use $\mathcal{A} \subseteq \mathcal{X}$ to denote the set of admissible profiles, and consider two different problem variants depending on the form of $\mathcal{A}$: (1) In the *admissible profile* variant, $|\mathcal{A}| = 1$, meaning that the agents must be incentivized to choose a particular effort profile, which is given as part of the input. (2) In the *admissible actions* variant, $\mathcal{A}$ is implicitly specified by a subset of actions $A \subseteq [m]$, and an effort profile is admissible if and only if it is supported only over admissible actions ($x^k_j = 0$ for every $j \notin A$).

We consider two main classes of evaluation mechanisms: monotone mechanisms and their subclass of linear mechanisms. An evaluation mechanism is *monotone* if two conditions hold: (i) for every two feature vectors $F' \geq F$, it holds that $M(F') \geq M(F)$;[6] and (ii) for every feature vector F there exists a subset S of features such that increasing all features F_S in the subset strictly increases $M(F)$.[7] An evaluation mechanism is *linear* if $M(F)$ is a multilinear function in the features, namely, $M(F) = \sum_i \beta_i F_i$ where $\beta_i \geq 0$ for every i and $\beta_{i'} > 0$ for some i'.

[5] In Kleinberg and Raghavan (2019), the agent has an arbitrary effort budget B. Note it is without loss of generality to assume $B = 1$ for all agents as we do, since any discrepancies in effort budgets can instead be realized by scaling the effort conversion matrices.

[6] Throughout the paper, whenever we write an inequality between two vectors, it means that the inequality holds in each coordinate.

[7] Together with the assumption that no effort conversion matrix has a column of zeros, condition (ii) implies there is always *potential* to increase the score by investing more effort, so all agents are strictly incentivized to exhaust their effort budgets. Without condition (ii), the evaluation problem would be trivial, since we could always just use the mechanism that gives a payoff of zero no matter what the feature scores are.

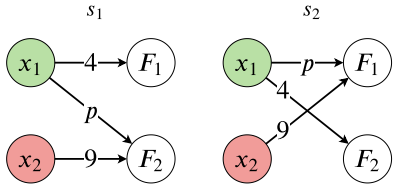

Figure 1: A two-agent instance of the evaluation problem, where $p \in [1, 8]$ is an arbitrary parameter.

2.1 Examples

The following examples illustrate the complexity that is added to the evaluation problem when there are multiple agents.

Example 2.1. Returning to the classroom setting, suppose that there are two types of students, s_1 and s_2, who can choose between studying (action 1) and cheating (action 2). Studying improves both test scores (feature F_1) and homework scores (feature F_2) for both types, while cheating improves just one of the scores by an even greater amount. The effort conversion rates in such a scenario might be as depicted in Figure 1, where $1 \leq p \leq 8$.

Since cheating only improves one kind of score, there are simple linear mechanisms that can incentivize studying for either student type in isolation, no matter what p is: for s_1, set $\beta := (1, 0)$ (final score depends only on the test), and for s_2, set $\beta := (0, 1)$ (final score depends only on the homework). But what if we wish to simultaneously incentivize both student types to study?

At $p = 6$, there is a still a linear mechanism that works. Taking $\beta := (1, 1)$, the marginal benefit toward studying is 10, while the marginal benefit toward cheating is 9, so both student types will invest all of their effort into studying.

At $p = 4$, no linear mechanism exists. For if some $\beta = (\beta_1, \beta_2)$ incentivizes s_1 to study, we must have $4\beta_1 + 4\beta_2 \geq 9\beta_2$, or in other words, $4\beta_1 \geq 5\beta_2$. Analogously, if that same β incentivizes s_2 to study, we must have $4\beta_2 \geq 5\beta_1$. This is only satisfied by $\beta = (0, 0)$, which violates the monotonicity requirement that at least one coordinate be strictly positive (and makes no sense as a classroom scoring method). Thus, no linear mechanism can simultaneously incentivize both student types to study. However, consider a *nonlinear*, monotone mechanism: $M(F_1, F_2) := \min(F_1, F_2)$. Neither type of student is incentivized to cheat under this mechanism, since that will not improve their minimum score.

At $p = 1$, there is no monotone mechanism at all, not even a nonlinear one. Supposing there was such an $M : \mathbb{R}^2_{\geq 0} \to \mathbb{R}_{\geq 0}$, consider the choice of an s_1 student between two different profiles: the admissible profile $(1, 0)$, and the inadmissible profile $(\frac{1}{2}, \frac{1}{2})$. If s_1 chooses the admissible profile, they obtain a feature vector $(4, 1)$, and if they choose the inadmissible profile, they obtain the feature vector $(2, 5)$. Since we are assuming M incentivizes only studying, we must therefore have $M(4, 1) \geq M(2, 5)$. Monotonicity implies $M(2, 5) > M(1, 4)$, so $M(4, 1) > M(1, 4)$. However, by a completely symmetric argument, for s_2 students to be incen-

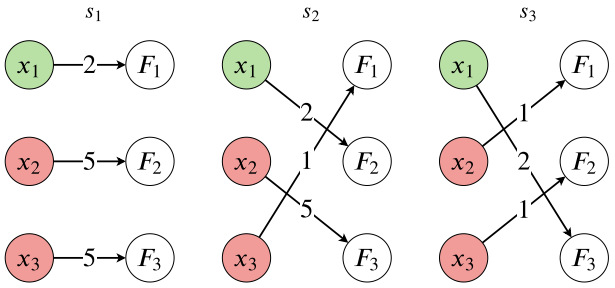

Figure 2: A three-agent instance of the evaluation problem exhibiting the power of nonlinear mechanisms. This construction can be generalized to any number of agents.

tivized to study we must have $M(1,4) > M(4,1)$, which is a contradiction. Thus, no monotone mechanism can exist.

One of the most remarkable conclusions from Example 2.1 is that, in stark contrast to the one-agent case, nonlinear mechanisms can succeed where linear mechanisms fail. One can interpret the scenarios where nonlinear mechanisms gain an advantage from a machine learning perspective. Nonlinear classifiers can see more complex relationships among data, such as the conjunction of two separate conditions like, "in order to be in the positive class, feature 1 must have value at least x *and* feature 2 must have value at least y." Analogously, a nonlinear evaluation mechanism can make more complicated distinctions between desirable and undesirable behavior, such as, "in order to get a high score, feature 1 must have value at least x *and* feature 2 must have value at least y." Nonlinear mechanisms are necessary when the effort conversion rates of several agents combine to form a complex boundary in the feature space between the results of desirable agent behavior and undesirable agent behavior, which cannot be linearly separated. This effect can be quite dramatic: as the following extreme example shows, there exist situations where the best linear mechanisms perform arbitrarily worse than the best nonlinear ones.

Example 2.2. For any positive integer n, define an instance of the evaluation problem with n actions, n features, and n agents, where only action 1 is admissible, and the rate of effort conversion for agent s_k from action j to feature F_i is

$$\alpha_{ji}^{k} := \begin{cases} 1 & \text{if } i < k \\ 2 & \text{if } i = k \\ 5 & \text{if } i > k \end{cases}$$

if $j+k-1 = i \mod n$, and zero otherwise. Figure 2 shows an example of this construction when $n = 3$.

Suppose some pair of agents $s_k, s_{k'}$ where $k' > k$ could be jointly incentivized to invest only in action 1 with some linear mechanism β. For s_k to be incentivized to invest only in action 1, we must have $2\beta_k \geq 5\beta_{k'}$, and for $s_{k'}$ we must have $2\beta_{k'} \geq \beta_k$. Therefore, $\beta_k \geq \frac{5}{4}\beta_k$, so $\beta_k = 0$, in which case monotonicity implies it is strictly preferable for s_k to invest in some other, inadmissible action yielding a nonzero payoff. Hence, no linear mechanism can incentivize more than one agent to invest only in action 1. However, we will see in Section 3.1 that *all* agents can be incentivized to invest only in action 1 with a monotone, nonlinear mechanism.

3 Monotone Mechanisms

In this section we first describe one of our main contributions — a useful characterization of when it is possible to jointly incentivize a given set of agents to choose admissible actions via a monotone mechanism. The proof is constructive, giving an efficiently computable $M : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ with the guarantee that if any monotone mechanism "works," so does M. Using this characterization, we present polynomial-time algorithms to solve Problem 1 for an unbounded number of features, and Problem 2 for a constant number of features. We then present hardness results for the remaining problems in the top half of Table 1.

3.1 Imitation Graphs

The central obstacle in the multi-agent evaluation problem is that one agent may be able to achieve good scores by choosing admissible actions, while another agent may be able to achieve even better scores by choosing inadmissible actions. In such scenarios, we say that the second agent is able to *imitate* the first one. The key observation is that, when this happens, the second agent must be given a greater payoff than the first agent, rewarding them for not acting in this undesirable way. This idea will allow us to characterize exactly when it is possible to jointly incentivize multiple agent types. Moreover it will imply that the incentivizing mechanism will have quite a natural form, ranking agents by their capability to emulate others' achievements, and assigning them payoffs according to this ranking.

To formally define imitation, we shall refer to two agents s_1 and s_2, where we somewhat abuse notation using s_1, s_2 for arbitrary agents as opposed to the agents with index $k = 1, 2$. We use this convention throughout Section 3 to avoid excessive subscripts.

Fix an admissible action profile $x^*(s)$ for each agent $s \in S$. We say that agent s_1 can *imitate* agent s_2 with respect to x^* if s_1 can play an inadmissible action profile x such that $F^{s_1}(x) \geq F^{s_2}(x^*(s_2))$. If x can be chosen so that $F^{s_1}(x) > F^{s_2}(x^*(s_2))$, we say that s_1 can *strictly* imitate s_2. The *imitation graph* with respect to x^* is the directed graph with vertex set S and an edge from s_1 to s_2 if and only if s_1 can imitate s_2. If s_1 can strictly imitate s_2, we say that (s_1, s_2) is a *strict edge*.

Theorem 3.1. *It is possible to incentivize all agents in S to choose effort profiles in $\mathcal{A}$ using a monotone mechanism if and only if there exists some $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges.*

Proof. For the forward direction, suppose $M : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a monotone mechanism that incentivizes all agents to invest only in admissible actions. Then for each $s \in S$, choose $x^*(s)$ to be any admissible best response of s under M. Suppose toward a contradiction that the imitation graph with respect to x^* contains a directed cycle $s_1, s_2, \ldots, s_q, s_1$, where

(s_q, s_1) is a strict edge. Then for each k from 1 to $q-1$, we must have

$$M(F^{s_k}(x^*(s_k))) \geq M(F^{s_{k+1}}(x^*(s_{k+1}))),$$

for otherwise, if $M(F^{s_{k+1}}(x^*(s_{k+1}))) > M(F^{s_k}(x^*(s_k)))$ then agent s_k could deviate from $x^*(s_k)$ and choose some inadmissible action x such that

$$F^{s_k}(x) \geq F^{s_{k+1}}(x^*(s_{k+1})),$$

receiving a strictly greater payoff:

$$M(F^{s_k}(x)) \geq M(F^{s_{k+1}}(x^*(s_{k+1}))) > M(F^{s_k}(x^*(s_k)))$$

(here the first inequality follows from monotonicity condition (i)). Additionally, we must have

$$M(F^{s_q}(x^*(s_q))) > M(F^{s_1}(x^*(s_1))),$$

for otherwise, if $M(F^{s_1}(x^*(s_1))) \geq M(F^{s_q}(x^*(s_q)))$, agent s_q could deviate from $x^*(s_q)$ and choose some inadmissible action x such that $F^{s_q}(x) > F^{s_1}(x^*(s_1))$, receiving a strictly greater payoff:

$$M(F^{s_q}(x)) > M(F^{s_1}(x^*(s_1))) \geq M(F^{s_q}(x^*(s_q)))$$

(here the first inequality follows from monotonicity condition (ii)). Thus, we have an inconsistent cycle of inequalities

$$M(F^{s_1}(x^*(s_1))) \geq M(F^{s_2}(x^*(s_2))) \geq \dots$$
$$\geq M(F^{s_q}(x^*(s_q))) > M(F^{s_1}(x^*(s_1))).$$

We have reached a contradiction, so it must be that there are no cycles containing any strict edges.

For the backward direction, let G be the imitation graph with respect to some x^*, and assume that G has no directed cycles containing any strict edges. We topologically sort the strongly connected components of G in decreasing order, and let $v : S \to \{1, 2, \dots, |S|\}$ give the index of each vertex's component in the topological sort (v already provides a rough ranking of the agents). Let $m : \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the function that takes the minimum value of all coordinates in a vector, and let $B \in \mathbb{R}_{>0}$ be a strict upper bound on $m(F)$ for any feature vector F that is attainable by any agent in S. Consider the mechanism

$$M(F) := \max\Big\{v(s) + \frac{m(F - F^s(x^*(s)))}{B} \;\Big|\; s \in S,\ F \geq F^s(x^*(s))\Big\} \tag{1}$$

It is not hard to verify that M satisfies both conditions for monotonicity.[8] We claim that M incentivizes every $s \in S$ to play the admissible action $x^*(s)$.

Suppose, toward a contradiction, that for some agent s_1, there existed some alternative, inadmissible effort profile x yielding a strictly higher payoff, i.e.,

$$M(F^{s_1}(x)) > M(F^{s_1}(x^*(s_1))).$$

By the definition of M, this means that

$$\max\Big\{v(s_2) + \frac{m(F^{s_1}(x) - F^{s_2}(x^*(s_2)))}{B} \;\Big|\; s_2 \in S,\ F^{s_1}(x) \geq F^{s_2}(x^*(s_2))\Big\}$$
$$> \max\Big\{v(s_3) + \frac{m(F^{s_1}(x^*(s_1)) - F^{s_3}(x^*(s_3)))}{B} \;\Big|\; s_3 \in S,\ F^{s_1}(x^*(s_1)) \geq F^{s_3}(x^*(s_3))\Big\}.$$

Taking any s_2 on the LHS that realizes the maximum, and plugging in s_1 for s_3 on the RHS, this becomes

$$v(s_2) + \frac{m(F^{s_1}(x) - F^{s_2}(x^*(s_2)))}{B} > v(s_1).$$

Since $m(F) < B$ for any attainable feature vector F, it follows that $v(s_2) + 1 > v(s_1)$. Since v is integer-valued, this means $v(s_2) \geq v(s_1)$.

On the other hand, $F^{s_1}(x) \geq F^{s_2}(x^*(s_2))$ implies $(s_1, s_2) \in E(G)$, so $v(s_1) \geq v(s_2)$. Thus, we have $v(s_1) = v(s_2)$, meaning that s_1 and s_2 are in the same strongly connected component of G. Also,

$$\frac{m(F^{s_1}(x) - F^{s_2}(x^*(s_2)))}{B} > v(s_1) - v(s_2) = 0,$$

which implies $F^{s_1}(x) > F^{s_2}(x^*(s_2))$, so (s_1, s_2) is a strict edge. But it is impossible to have a strict edge between two vertices in the same strongly connected component, as this would imply that G has a cycle containing that strict edge, contradicting our hypothesis. We have a contradiction, so M incentivizes all agents to play according to x^*. □

Notice that the imitation graph in Example 2.2 with respect to all agents investing all effort in action 1 consists of a strict edge $(s_{k'}, s_k)$ whenever $k' > k$. Since this graph has no cycles, Theorem 3.1 implies there is a monotone mechanism that incentivizes all agents to invest only in action 1, in particular the mechanism specified in (1). Ignoring the small payoff summand that is a fraction over B (which is only necessary for satisfying condition (ii) of monotonicity), the payoff of this mechanism is

$$M(F) \approx \max_{i \in [n]}\{n - i + 1 \mid F_i \geq 2\}.$$

In words, all agents are incentivized to focus all of their effort on raising the feature of smallest index in which they can score at least 2. For each agent, this feature is always the one with an edge from x_1 in the effort graph (see Figure 2), so all agents will invest only in action 1.

3.2 Incentivizing All Agents

Theorem 3.1 directly leads to a simple algorithm to solve Problem 1.

Corollary 3.2. *There is a polynomial-time algorithm to find a monotone mechanism that incentivizes all agents to choose a specific effort profile, or determine that no such mechanism exists.*

[8] Technically speaking, the mechanism is undefined for off-equilibrium-path strategies of scoring lower than any agent should ever score. This can be fixed by adding a dummy agent with an effort conversion rate of zero from every action to every feature.

Proof. Since $|\mathcal{A}| = 1$, there is only one possible $x^* : S \to \mathcal{A}$ to choose from, and thus only one possible imitation graph G. For each strict edge $(s_1, s_2) \in E(G)$, we just check to see if there is a path in G from s_2 to s_1. By Theorem 3.1, it is possible to jointly incentivize all agents if and only if none of these paths exist.

The only potential difficulty lies in constructing this imitation graph in the first place. We show in Lemma B.1 of Appendix B that this can be accomplished in polynomial time using linear programming. □

Solving Problem 2 is trickier, since to use our characterization we must search for an assignment $x^* : S \to \mathcal{A}$ prescribing which admissible action profile we would like each agent to choose. While at first glance this might appear hopelessly intractable, we make an observation that turns out to help when the number of features is constant: If there exists any $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges, then there exist profiles for some nonempty subset of agents $T \subseteq S$ such that

1. no agents in $S \setminus T$ can imitate any agents in T, and
2. no agents in T can strictly imitate any agents in T.

Informally, the observation follows from topologically sorting the strongly connected components of the imitation graph, and noticing that the first component must have these two properties. Given the observation, if such a subset of agents T and their effort profiles can be found in polynomial time, those agents can be removed, since they can no longer create a cycle with any of the remaining agents. If it is then possible to keep removing sets of agents in this manner, then the final imitation graph will have no cycles containing strict edges; otherwise, we can conclude impossibility for the given problem instance.

Finding a subset T and corresponding profiles can be achieved using an iterative marking algorithm, formally presented in Appendix C.1. However, it relies on the ability to efficiently answer the simple question, "Is there some admissible profile that s_1 can play that no agent in some given set R can (strictly) imitate?" Formally, this predicate is,

$$\exists x^1 \in \mathcal{A},\ \forall s_2 \in R,$$

$$\neg \left(\exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F^{s_2}(x^2)_i \geq F^{s_1}(x^1)_i\right)$$

(for the strict version, we have a strict inequality). It turns out that this is computable in polynomial time for a constant number of features n, but is NP-hard in general, and so is Problem 2 (see Appendix C for the details).

Theorem 3.3. *The problem of finding a monotone mechanism that incentivizes all agents to choose admissible actions, or determining that no such mechanism exists, is*

1. *solvable in polynomial time for a constant number of features n, and*
2. *NP-complete for unbounded n.*

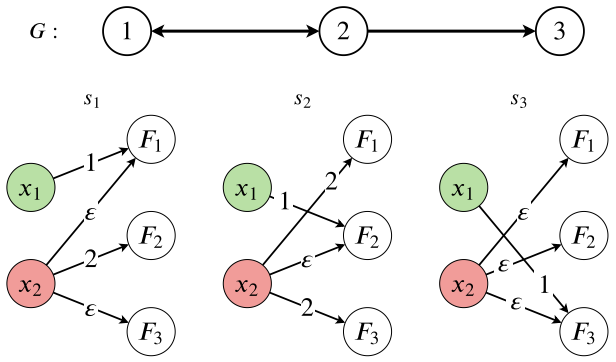

Figure 3: An input graph G and the corresponding evaluation problem produced by the reduction.

3.3 Incentivizing a Maximum Number of Agents

Once the imitation graph for all agents has been constructed, Theorem 3.1 implies that we can incentivize any subset of agents whose induced subgraph has no cycles with strict edges. When all edges are strict, this is an instance of the NP-complete Feedback Vertex Set problem: given a graph, it asks for a minimum-size subset of vertices whose deletion would eliminate all cycles. It turns out that there is a reduction in the other direction too, since all directed graphs can be constructed as imitation graphs — even with just two features! This proves that Problems 3 and 4 are NP-complete.

Theorem 3.4. *The problems of finding a monotone mechanism that incentivizes a maximum number of agents to choose admissible actions / a specific admissible profile are NP-complete even for a constant number of features.*

We will sketch the proof of NP-hardness for an *unbounded* number of features, leaving the more complicated reduction with only two features for Appendix E. Suppose we are given an instance of Feedback Vertex Set, that is, a graph G with n vertices. For convenience, assume $V(G) = [n]$. We construct an instance of the evaluation problem with agents $s_1, s_2, \ldots, s_n$, features $F_1, F_2, \ldots, F_n$, and 2 actions, where action 1 is admissible and action 2 is inadmissible. For each $i, j \in [n]$, define

$$\alpha^k_{1,i} := \begin{cases} 1 & i = k \\ 0 & \text{otherwise} \end{cases}, \quad \alpha^k_{2,i} := \begin{cases} 2 & (k,i) \in E(G) \\ \varepsilon & \text{otherwise} \end{cases}$$

(see Figure 3 for an example where $n = 3$).

It is proved in Appendix D that, for $0 < \varepsilon < 1$, G is the imitation graph with respect to the profile assignment of $(1, 0)$ for all agents, and all edges are strict edges. By Theorem 3.1, G has a feedback vertex set of size at most q if and only if at least $n - q$ agents (namely, those not in the feedback vertex set) can be jointly incentivized to invest only in action 1.

4 Linear Mechanisms

While nonlinear monotone mechanisms can incentivize arbitrarily more agents than linear ones (see Example 2.2),

there are still many reasons to consider the problem of finding a linear mechanism to incentivize multiple agents. For one, what we call a linear mechanism coincides with traditional contracts investigated in contract theory (see Appendix A). Linear mechanisms also bear more similarity to the kinds of grading schemes commonly used in practice. Furthermore, one might hope that, since linear mechanisms are simpler than monotone mechanisms, finding linear mechanisms might be an easier problem. This intuition turns out to be partially correct: under the very reasonable assumption that the number of features is held constant, each problem we consider in this section has a polynomial time algorithm, although some are hard in general.

4.1 Algorithmic Results

Observe first that, when responding to a linear mechanism, agents can simply compute the marginal payoff toward each of their actions, and invest effort only in the most profitable ones. Therefore, an agent may only split their effort among multiple actions if they are all tied for the highest marginal payoff. If we only care about an agent investing in any one of a set of multiple admissible actions, we need only ensure that one of those admissible actions gives the highest marginal payoff.

Motivated by these observations, we introduce the following notation: when, for a given agent s_k, some action j_1 yields a weakly greater marginal payoff under a linear mechanism $\beta \in \mathbb{R}^n$ than some other action j_2, we say that β satisfies the constraint $h(k, j_1, j_2)$. Since each $h(k, j_1, j_2)$ can be written as a linear constraint over the space of linear mechanisms $\mathbb{R}^n$, we immediately have an algorithm for Problem 5.

Theorem 4.1. *There is a polynomial-time algorithm to find a linear mechanism that incentivizes all agents to choose a specific effort profile, or determine that no such mechanism exists.*

Proof. To solve this problem, we must determine if there exists $\beta \in \mathbb{R}^n$ such that, for every agent $s_k \in S$ and every action j_1 in the support of the admissible profile, for every alternative action $j_2 \in [m]$ the constraint $h(k, j_1, j_2)$ is satisfied (i.e., $\alpha^k_{j_1} \cdot \beta \geq \alpha^k_{j_2} \cdot \beta$). This reduces to testing the feasibility of a linear program with n variables and at most ℓm^2 constraints (were ℓ is the number of agents), which is solvable in polynomial time. □

Jumping to Problem 7, we do not require that β satisfy these constraints for all $k \in [\ell]$, just for as many k as possible. Let $\mathcal{L}_k$ be the polytope in $\mathbb{R}^n$ consisting of all points β that satisfy $h(k, j_1, j_2)$ for all actions j_1 in the support of the admissible profile, and for all actions $j_2 \in [m]$. Our objective is then to find a point in the intersection of a maximum number of the $\mathcal{L}_k$ polytopes. This is no longer a convex optimization problem like Problem 5. Yet we can solve it efficiently when n is a constant, using a geometric data structure known as a *hyperplane arrangement* (Goodman and O'Rourke 1997, Chapter 28).

An arrangement decomposes $\mathbb{R}^n$ into connected open *cells*, where each cell is a maximal connected region in the intersection of a subset of the hyperplanes that is not intersected by any other hyperplane. The key property that we will use is that all points within a given cell are equivalent in terms of which of the linear constraints they satisfy. This implies that, to test whether a given predicate on the constraints holds for any point in $\mathbb{R}^n$, it suffices to check only one point from each cell. This is tractable when n is constant, since it is known that the arrangement of p hyperplanes decomposes $\mathbb{R}^n$ into $O(p^n)$ cells, and that the arrangement can be computed in $O(p^n)$ time.

Algorithm 1: An algorithm for Problem 7.

Input: An instance of the evaluation problem with a single admissible profile x^*
Output: A linear mechanism β that incentivizes a maximum number of agents to invest effort according to x^*

1 $\mathcal{R} \leftarrow$ arrangement of all hyperplanes for constraints $h(k, j_1, j_2)$ for all $k \in [\ell]$, $j_1 \in \mathcal{S}(x^*)$, and $j_2 \in [m]$;
2 `max` $\leftarrow -1$;
3 **for** *each cell* $C \in \mathcal{R}$ **do**
4 $\beta' \leftarrow$ any point in C;
5 `numIncentivized` $\leftarrow |\{s_k \in S \mid$ all actions in $\mathcal{S}(x^*)$ yield the (weakly) greatest marginal payoff for s_k under $\beta'\}|$;
6 **if** *`numIncentivized` $>$ `max`* **then**
7 `max` $\leftarrow$ `numIncentivized`;
8 $\beta \leftarrow \beta'$;
9 **end**
10 **end**
11 **return** β;

Theorem 4.2. *Assuming a constant number of features, there is a polynomial-time algorithm to find a linear mechanism that incentivizes a maximum number of agents to invest in a specific admissible profile.*

Proof. Using the notation of Kleinberg and Raghavan, for an effort profile x, let $\mathcal{S}(x)$ denote the support of x. Recall that, to incentivize a given profile x^* for all agents, we must ensure all actions in $\mathcal{S}(x^*)$ are weak best responses. Based on our discussion of arrangements above, Algorithm 1 solves this problem in polynomial time when n is constant. Note that the predicate on line 5 is easy to compute for a fixed β', and does not depend on which $\beta' \in C$ is chosen, since whether a given $s_k \in S$ satisfies the predicate is completely determined by the constraints from line 1. □

With very minor adjustments to Algorithm 1, this same technique can be used to solve Problems 6 and 8 as well (see Appendix G).

Theorem 4.3. *Assuming a constant number of features, there is a polynomial-time algorithm to find a linear mechanism that incentivizes a maximum number of agents to choose admissible actions (and consequently, to determine if all agents can be incentivized to choose admissible actions).*

4.2 Hardness Results

Since these algorithms for Problems 6, 7, and 8 all rely on the ability to efficiently enumerate all cells in a low-dimensional hyperplane arrangement, it is natural to ask what happens when the number of features is part of the input, making this technique no longer viable. As it turns out, all three problems are NP-complete in general.

Theorem 4.4. *The following problems are NP-complete:*

1. *Finding a linear mechanism that incentivizes a maximum number of agents to invest only in admissible actions / a specific admissible profile.*
2. *Finding a linear mechanism that incentivizes all agents to invest only in admissible actions.*[9]

Part (1) follows from the same reduction outlined in Section 3.3 since the instances produced by that reduction have a special property: whenever a particular subset of agents can be incentivized to choose admissible actions using a monotone mechanism, they can, in fact, be so incentivized using a linear mechanism. The hardness in part (2) is of a completely different nature, and is proved via a separate reduction from 3SAT. See Appendices D and F for the proofs.

5 Discussion

Designing an evaluation scheme for a group of agents is broad practical dilemma. It comes up in credit scoring, principal-agent relationships without money (commanders and soldiers, teachers and students), employment under collective agreements, etc. In these cases, designing a *single* evaluation rule for all agents is the only realistic approach. This paper addresses the challenge of multi-agent evaluation from a computational perspective, answering an open question of Kleinberg and Raghavan (2019). Our main contribution is in showing that the evaluation problem with more than one agent is "a whole new ball game": for example, monotone mechanisms now have more power than linear ones, and the goal of incentivizing admissible actions is now separate (and often harder) than incentivizing a particular effort profile.

There are many directions for future research. A natural one is *approximating* the optimal number of incentivized agents when maximizing is NP-hard. Our techniques are able to provide insights in this direction, since we show a close connection to the Feedback Vertex Set problem, for which both approximations and lower bounds are known (Bar-Yehuda et al. 1998). Other future directions include settings with hidden types as well as hidden actions, incentivizing agent cooperation (by allowing features like scores on a group project), or accommodating complex effects of combinations of agent actions.

Acknowledgments

This work was partially supported by the National Science Foundation under grants IIS-1350598, IIS-1714140, CCF-1525932, and CCF-1733556; by the Office of Naval Research under grants N00014-16-1-3075 and N00014-17-1-2428; by a J.P. Morgan AI Research Award; by a Guggenheim Fellowship; by the Israel Science Foundation (grant No. 336/18); and by a Taub Family Foundation Fellowship. Part of this work was done while Dobson was visiting the Technion via the MISTI-MIT Israel Program.

[9]The hardness of the problem in part (2) of the theorem implies the hardness of the first problem in part (1); we list these separately for consistency with Table 1.

References

Babaioff, M.; Feldman, M.; and Nisan, N. 2006. Combinatorial agency. In *Proceedings of the 7th ACM Conference on Electronic Commerce (EC)*, 18–28.

Bar-Yehuda, R.; Geiger, D.; Naor, J.; and Roth, R. M. 1998. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference. *SIAM Journal on Computing* 27(4):942–959.

Dekel, O.; Fischer, F.; and Procaccia, A. D. 2010. Incentive compatible regression learning. *Journal of Computer and System Sciences* 76(8):759–777.

Dütting, P.; Roughgarden, T.; and Talgam-Cohen, I. 2019. Simple versus optimal contracts. In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, 369–387.

Goodman, J. E., and O'Rourke, J., eds. 1997. *Handbook of Discrete and Computational Geometry*. CRC Press.

Grossman, S. J., and Hart, O. D. 1983. An analysis of the principal-agent problem. *Econometrica* 51(1):7–45.

Hardt, M.; Megiddo, N.; Papadimitriou, C. H.; and Wootters, M. 2016. Strategic classification. In *Proceedings of the 7th ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, 111–122.

Kleinberg, J. M., and Raghavan, M. 2019. How do classifiers induce agents to invest effort strategically? In *Proceedings of the 20th ACM Conference on Economics and Computation (EC)*, 825–844.

Meir, R.; Procaccia, A. D.; and Rosenschein, J. S. 2008. Strategyproof classification under constant hypotheses: A tale of two functions. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, 126–131.

Meir, R.; Procaccia, A. D.; and Rosenschein, J. S. 2012. Algorithms for strategyproof classification. *Artificial Intelligence* 186:123–156.

Xiao, S.; Wang, Z.; Chen, M.; Tang, P.; and Yang, X. 2020. Optimal common contract with heterogeneous agents. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. To appear.

Multiagent Evaluation Mechanisms: Supplemental Material

The purpose of this online appendix is to provide additional observations and present complete proofs of results stated in the manuscript.

A Relation to Contract Design

In this appendix we analyze the connection between our evaluation model and the standard principal-agent model in contract design. The connection described in Appendix A.1 does not appear to have been observed before.

A.1 Casting our Basic Model as a Principal-Agent Setting

We focus here on the basic setting with a single agent s_k (we omit the index k for simplicity) and linear mechanisms.

Evaluation model recap. Recall that in our *evaluation* problem, there are m actions $1, 2, \ldots, m$, and n features F_1,F_2,$\ldots$,F_n. There is a mapping α_j from action j to the features; we assume here without loss of generality that $\sum_i \alpha_{j,i} \leq 1$ (this is by normalization of the matrix α), so that α_j can be treated as a vector of probabilities.[9] In general, the agent plays a mixed strategy x over the actions, where $\sum_j x_j^k = 1$ (since the agent is strictly incentivized to exhaust their effort budget of size 1). But given a *linear* mechanism $\beta = (\beta_1, \ldots, \beta_n)$, recall from Section 4 that it is without loss to assume the agent plays a pure strategy and picks a single action j that maximizes their utility

$$\sum_i \alpha_{j,i}\beta_i.$$

The goal of the principal is to design the mechanism such that j is an admissible action.

Principal-agent model. Keeping the evaluation problem in mind, let us turn to the classic *principal-agent* problem: There are m action a_1,a_2,$\ldots$,a_m and n rewards r_1,r_2,$\ldots$,r_n to the principal. There is a distribution α_j mapping action a_j to the rewards. Given a contract $\beta = (\beta_1, \ldots, \beta_n)$ specifying the agent's payment for every reward, the agent chooses action a_j that maximizes their expected payment

$$\sum_i \alpha_{j,i}\beta_i.$$

The goal of the principal is to maximize the expected reward minus payment, i.e., $\sum_i \alpha_{j,i}(r_i - \beta_i)$.

Comparison. As can be seen from the descriptions above, the basic evaluation problem and the principal-agent problem are almost identical, up to the principal's goal. From the principal's perspective, evaluation is in some sense simpler: In the principal-agent model, the principal often wishes to incentivize a certain action a_j — like in the evaluation model — but needs to do so with minimum expected payment $\sum_i \alpha_{j,i}\beta_i$. In contrast, "paying" the agent in the evaluation model by awarding them a high score does not affect the utility of the principal. From the agent's perspective, the two models are equivalent! This means that some classic results from contract theory like the *characterization of incentivizable actions* apply to evaluation. The last observation helps explain the similarity between the classic characterization (see, for example, Proposition 1 in (Dütting, Roughgarden, and Talgam-Cohen 2019b)), and the characterization of Kleinberg and Raghavan (which holds more generally — see below).

We end this discussion of the basic evaluation setting by emphasizing that despite the connection to contract design theory, the term *linear mechanism* used in this paper is not the same as *linear contract*. In linear mechanisms, the score is a *multilinear* function of the n features with coefficients $\beta_1, \ldots, \beta_n$ — and this is referred to simply as a *contract* in microeconomics. In linear contracts, there is a single coefficient β that is multiplied by the reward r_j to specify the payment of the agent when their action leads to the realized reward r_j.

A.2 Generalizations Beyond Standard Principal-Agent Settings

Interestingly, two aspects of evaluation introduced by Kleinberg and Raghavan (2019) strictly generalize the classic principal-agent model: concave conversion functions, and non-linear mechanisms.

[9] If the entries sum up to 1 then we have a distribution, which can be interpreted as follows: the effort which the agent puts into action j adds to feature F_i with probability $\alpha_{j,i}$. If the entries sum up to less than 1, we interpret this as if with probability $1 - \sum_i \alpha_{j,i}$ the effort leads to nothing. This view of the model is nice as it captures the probabilistic aspects of one's actions. For example, a student may study for the exam and still fail with some small probability.

Concave conversion functions. Kleinberg and Raghavan (2019) introduce a more general way to translate the agent's effort profile x into features: instead of $F_i = \sum_j x_j \alpha_{j,i}$, they define $F_i = f_i(\sum_j x_j \alpha_{j,i})$, where f_i is a concave, strictly increasing function. This has an interesting effect — since now putting in only (say) 80% of the effort into a certain action can result in almost 100% of the feature value, the agent is strictly incentivized to split their effort among different actions. One of the main contributions of Kleinberg and Raghavan (2019) is in showing that a generalization of the classic incentivizability characterization holds in this generalized setting, and moreover that for a single agent there is no need to consider mechanisms beyond linear.

In Appendix I we discuss extensions of our results to concave conversion functions.

Non-linear mechanisms. In the evaluation model, the principal designing the contract is assumed to have access to more information about the agent's actions than in the classic model. In particular, they view the full vector of features rather than a single realization drawn from α_j. Thus, they are no longer confined to paying the agent β_i for the ith realization, and can choose any monotone function of the features. This extra freedom becomes significant when dealing with multiple agents, as demonstrated in the body of this paper.

A.3 Our Treatment of Multiple Agents versus Classic Literature

Our work's focus is on mechanisms/contracts for multiple agents. In the classic literature, multiple agents have been investigated under two main models (Salanie 2005, pp. 140-141). In one, a group of agents work together as a team and are evaluated and rewarded depending on the global outcome. The main design challenge is to discourage agents from free-riding on others' contribution. A variation is when agents have separate tasks but are expected to help one another. In the other model, the conversion of different agents' efforts to outcomes is interconnected, so the principal can learn about one agent's actions from the outcomes of another, and the design challenge lies in best utilizing the available information. In both models, each agent has a personal contract with the principal. We initiate a third model, in which what unifies the ℓ different principal-agent problems is not the need to incentivize teamwork or optimize information, but rather the fairness requirement that all agents face the *same* evaluation mechanism. In other words, a single contract should apply to the whole group, and still manage to incentivize desirable actions.

B Constructing Imitation Graphs

The efficiency of the algorithm presented in Corollary 3.2 relies on the following lemma, which we will now prove.

Lemma B.1. *Given an assignment of profiles $x^* : S \to \mathcal{A}$ and any pair of agents $(s_1, s_2) \in S$, it is possible to determine, in polynomial time, whether (s_1, s_2) is an edge of the imitation graph with respect to x^*, and if so, whether or not that edge is strict.*

Proof. Suppose $x^*(s_2) = x^2$. To determine if s_1 can imitate s_2, we must determine if the following set is nonempty:

$$\mathcal{I} := \{x^1 \in \mathcal{X} \mid x^1 \notin \mathcal{A} \text{ and } F^{s_1}(x^1) \geq F^{s_2}(x^2)\}.$$

In the admissible profile variant of the problem, $\mathcal{A}$ is a single point in $\mathbb{R}^m$, while in the admissible actions variant, $\mathcal{A}$ is a standard simplex supported over the admissible actions. In either of these two cases, we can express $\mathcal{A}$ as some polytope bounded by at most $2m$ hyperplanes

$$\mathcal{A} = \{x \in \mathbb{R}^m \mid \text{for all } j \in [2n],\, a_j \cdot x \leq b_j\},$$

where each a_j is a vector in $\mathbb{R}^m$ and each b_j is a scalar. For x^1 not to be contained in $\mathcal{A}$, it must violate one of these constraints. Therefore,

$$\begin{aligned}\mathcal{I} &= \{x^1 \in \mathcal{X} \mid \text{for some } j \in [2n],\, a_j \cdot x > b_j; \text{ and } F^{s_1}(x^1) \geq F^{s_2}(x^2)\} \\ &= \bigcup_{j \in [2n]} \{x^1 \in \mathcal{X} \mid a_j \cdot x > b_j \text{ and } F^{s_1}(x^1) \geq F^{s_2}(x^2)\}\end{aligned}$$

Determining whether each of these $2n$ sets is nonempty is an LP feasibility problem with a mixture of strong and weak inequalities, which can be solved in polynomial time using linear programming by Lemma B.2. Computing whether a given edge is a strict edge is the same, except that we have more strict inequalities. □

Lemma B.2. *Given matrices A and C, and vectors b and d, it is possible to determine, in polynomial time, whether there exists an $x \in \mathbb{R}^m$ satisfying the constraints*

$$\begin{aligned}Ax &\leq b \\ Cx &< d,\end{aligned}$$

The authors could not find a formal reference for this fact. This proof is inspired by (Lorenz 2012).

Proof. Suppose b and d are column vectors of length k and ℓ, respectively. If A_i refers to the i^{th} row of A and C_j refers to the j^{th} row of C, we can rewrite these constraints as

$$\begin{aligned} A_i x &\leq b_i && \forall i \in [k] \\ C_j x &< d_j && \forall j \in [\ell]. \end{aligned}$$

For a given x, these constraints are satisfied if and only if there is some $\varepsilon > 0$ satisfying

$$\begin{aligned} A_i x &\leq b_i && \forall i \in [k] \\ C_j x + \varepsilon &\leq d_j && \forall j \in [\ell]. \end{aligned}$$

This is possible if and only if the following LP, with variables $x \in \mathbb{R}^m$, $\varepsilon \in \mathbb{R}$ has an optimal value strictly greater than zero:

$$\begin{aligned} &\textbf{maximize} && \varepsilon \\ &\textbf{subject to} && \begin{array}{ll} A_i x \leq b_i & \forall i \in [k] \\ C_j x + \varepsilon \leq d_j & \forall j \in [\ell]. \end{array} \end{aligned}$$

□

C Problem 2: Algorithm and Hardness

In this appendix we prove both parts of Theorem 3.3.

C.1 Algorithm for Problem 2 with a Constant Number of Features

Proof of Theorem 3.3 Part 1. We claim that Algorithm 2 (together with the mechanism from Theorem 3.1) solves Problem 2 when there are a constant number of features.

Algorithm 2: Finds an assignment of profiles to solve Problem 2.

Input: $\mathcal{X}$, $\mathcal{A}$, $\{F^s \mid s \in S\}$
Output: An assignment of profiles $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges, or `IMPOSSIBLE` if no such assignment exists

1 $T_1 \leftarrow S$;
2 $x^* \leftarrow$ an empty map;
3 $j \leftarrow 1$;
4 **while** T_j *is nonempty* **do**
5 $T_{j+1} \leftarrow \emptyset$;
6 **while** $|T_j|$ *is decreasing* **do**
7 `move` $\leftarrow \emptyset$;
8 **for** $s_1 \in T_j$ **do**
9 **if** $\exists x^1 \in \mathcal{A},\ \left[\forall s_2 \in T_{j+1},\ \neg\left(\exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F^{s_2}(x^2)_i \geq F^{s_1}(x^1)_i\right)\right]$ $\wedge\ \left[\forall s_2 \in T_j,\ \neg\left(\exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F^{s_2}(x^2)_i > F^{s_1}(x^1)_i\right)\right]$ **then**
10 $x^*(s_1) \leftarrow$ some valid choice of x^1;
11 **else**
12 `move` $\leftarrow$ `move` $\cup \{s_1\}$;
13 **end**
14 **end**
15 $T_j \leftarrow T_j \setminus$ `move`;
16 $T_{j+1} \leftarrow T_{j+1} \cup$ `move`;
17 **end**
18 **if** $T_j = \emptyset$ **then**
19 **return** *IMPOSSIBLE*;
20 **else**
21 $j \leftarrow j + 1$;
22 **end**
23 **end**
24 **return** x^*;

In words, the conditional statement on line 9 says that there must exist some admissible profile x^1 such that, if s_1 plays x^1, no agent in T_{j+1} can imitate s_1, and no agent in T_j can strictly imitate s_1. We will return to the complexity of this predicate shortly. Assuming for now that it is possible to compute in polynomial time, it is not hard to see that the algorithm as a whole runs in polynomial time as well, since all three loops run for at most $|S|$ iterations, and consist only of elementary operations. We now argue that the algorithm is correct.

Suppose first that the algorithm returns on line 19 at some specific iteration j of the outer loop. We claim that there is no $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges. Suppose toward a contradiction that there was such an x^*, and let G be the induced subgraph of the imitation graph with respect to x^* generated by the vertices in T_j as it was at the beginning of the first iteration of the loop starting on line 6 (or equivalently, the vertices in T_{j+1} at the end of the last iteration of the loop). Let $H \subseteq V(G)$ be the first strongly connected component in some topologically-sorted order.

We claim that the following invariant holds throughout the loop starting on line 6: $H \subseteq T_j$. Clearly, it holds before the first iteration of the loop, since $V(G) = T_j$ by the definition of G. To prove that each iteration preserves this invariant, it suffices show that the only vertices that are removed from T_j on line 15 are ones which cannot possibly be in H. Suppose toward a contradiction that some s_1 falsifying the predicate on line 9 was contained in H. Since s_1 falsifies the predicate, we must have that either

1. some $s_2 \in T_{j+1}$ can imitate s_1 playing $x^*(s_1)$, or
2. some $s_2 \in T_j$ can strictly imitate s_1 playing $x^*(s_1)$.

In either case, $(s_2, s_1) \in E(G)$, implying that $s_2 \in H$ as well. In case (1) this contradicts the assumption that the invariant holds at the beginning of the iteration, since $s_2 \in T_{j+1} \implies s_2 \notin T_j \implies s_2 \notin H$. In case (2) this contradicts the assumption that G has no cycles containing any strict edges, since (s_2, s_1) is a strict edge between two vertices in the same strongly connected component H. Thus, in either case, we have a contradiction, so no $s_1 \in H$ is removed from T_j on line 15. It follows that $H \subseteq T_j$ is an invariant of the loop starting on line 6.

However, since we are assuming that the algorithm returns on line 19, at the end of this loop we must have $T_j = \emptyset$. Thus, we have $H \subseteq T_j = \emptyset$, implying H is empty. This is a contradiction, since the first strongly connected component of G cannot possibly be empty. Hence, there can be no $x^* : S \to \mathcal{A}$ such that the imitation graph with respect to x^* has no cycles containing any strict edges.

Now suppose that the algorithm returns on line 24. Let G be the imitation graph with respect to the assignment x^* returned by the algorithm. We must show that G has no cycles containing any strict edges. Suppose toward a contradiction that G contains a directed cycle s_1,s_2,$\ldots$,s_q,s_1, where (s_q, s_1) is a strict edge. For each $k \in [q]$, let j_k be the unique index such that $s_k \in T_{j_k}$.

Observe that for each $k \in [q-1]$, if $j_{k+1} < j_k$, then on iteration j_{k+1} of the outer loop, s_k must have been moved out of $T_{j_{k+1}}$ and into $T_{j_{k+1}+1}$, so the condition on line 9 implies that s_k should not be able to imitate s_{k+1} playing $x^1 = x^*(s_{k+1})$. Since $(s_k, s_{k+1}) \in E(G)$, we have by *modus tollens* that $j_k \leq j_{k+1}$.

Similarly, if $j_1 \leq j_q$ then on iteration j_1 of the outer loop, on the final iteration of the loop on line 6, $s_q \in T_{j_1} \cup T_{j_1+1}$, so the condition on line 9 implies that s_q should not be able to strictly imitate s_1 playing $x^1 = x^*(s_1)$. So again, by *modus tollens*, since $(s_q, s_1) \in E(G)$ is a strict edge, we have that $j_q < j_1$.

Thus, we have a contradiction in the form of an inconsistent cycle of inequalities

$$j_1 \leq j_2 \leq \cdots \leq j_q < j_1,$$

so it must be that G has no cycles containing any strict edges.

We have now proved that the algorithm is correct in both cases. All that is left to show is that we can compute the predicate on line 9 in polynomial time when n is constant.

It will be more convenient to consider the negation:

$$\begin{aligned} \forall x^1 \in \mathcal{A},\ \big[\exists s_2 \in T_{j+1},\ \exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F^{s_2}(x^2)_i \geq F^{s_1}(x^1)_i\big] \\ \vee \big[\exists s_2 \in T_j,\ \exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F^{s_2}(x^2)_i > F^{s_1}(x^1)_i\big] \end{aligned} \tag{2}$$

For each $s_2 \in T_{j+1}$, let

$$C(s_2) := \{F \in \mathbb{R}^n \mid \exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F_i \leq F^{s_2}(x^2)_i\}.$$

and for each $s_2 \in T_j$, let

$$C(s_2) := \{F \in \mathbb{R}^n \mid \exists x^2 \in \mathcal{X} \setminus \mathcal{A},\ \forall i \in [n],\ F_i < F^{s_2}(x^2)_i\}.$$

Then (2) becomes

$$\forall x^1 \in \mathcal{A},\ F^{s_1}(x^1) \in \bigcup_{s_2 \in T_j \cup T_{j+1}} C(s_2), \tag{3}$$

In other words, we need to determine if

$$F^{s_1}(\mathcal{A}) \subseteq \bigcup_{s_2 \in T_j \cup T_{j+1}} C(s_2). \tag{4}$$

If this is not the case, then we would additionally like to find specific evidence in the form of a point

$$F \in F^{s_1}(\mathcal{A}) \setminus \bigcup_{s_2 \in T_j \cup T_{j+1}} C(s_2), \tag{5}$$

as well as an $x^1 \in \mathcal{A}$ such that $F = F^{s_1}(x^1)$.

Since we are assuming F^{s_1} is linear, the set on the left-hand side of (4) is a polytope. As it turns out, we can write each $C(s_2)$ as a polytope as well (but with some faces open). Geometrically, $C(s_2)$ is the set you get by taking $F^{s_2}(\mathcal{X} \setminus \mathcal{A})$ and projecting downward in each coordinate. More formally, one can see that $F \in C(s_2)$ if and only if F is a convex combination of some point in $x^2 \in \mathcal{X} \setminus \mathcal{A}$ and each of its 2^n projections onto subspaces spanned by sets of coordinate axes. So, to compute the (closure of the) polytope $C(s_2)$, simply take the convex hull of all vertices of the polytope $F^{s_2}(\mathcal{X} \setminus \mathcal{A})$, and all of their 2^n projections. Since n is constant, it follows from (Kaibel and Pfetsch 2003, Problem 1) that there are a polynomial number of such vertices, and that they can be enumerated in polynomial time. By (Kaibel and Pfetsch 2003, Problem 2), we can then obtain a description of the polytope as the intersection of polynomially-many halfspaces. Note that, since some faces of $\mathcal{X} \setminus \mathcal{A}$ are open (and there are additional strict inequalities in the description of $C(s_2)$ for any $s_2 \in T_j$), some faces of $C(s_2)$, might be open as well.

So determining whether (4) holds is really the question of whether one polytope is contained in a union of other polytopes. To solve this, we construct an arrangement of all hyperplanes defining $F^{s_1}(\mathcal{A})$ and each $C(s_2)$. (See Section 4 for a more detailed discussion of arrangements). Since n is constant and we have polynomially-many hyperplanes, it follows from Goodman and O'Rourke (1997) that the arrangement has polynomially-many cells. To check if (4) holds, simply enumerate every cell contained in $F^{s_1}(A)$ and check if it is contained in some $C(s_2)$. These containments are easy to determine, since either all points of the first cell are contained in the second one, or none are, so we can just test one point per cell, as done in Algorithm 1. Note that having a mixture of closed and open faces does not matter here, since boundaries of cells in an arrangement are considered to be distinct cells. If we find some point in some cell that is contained in $F^{s_1}(\mathcal{A})$ but is not contained in the union of the $C(s_2)$ sets (as in (5)), then we can easily invert F^{s_1} to obtain an x^1 to set $x^*(s_1)$ to on line 10.

Hence, for constant n, the computation involved in lines 9 and 10 can be done in polynomial time, and thus the entire algorithm runs in polynomial time. □

C.2 Hardness of Problem 2

Proof of Theorem 3.3 Part 2. Problem 2 is in NP since we can easily verify that a given choice of preference profile for each agent gives an imitation graph with no cycles containing any strict edges by Lemma B.1.

To prove NP-hardness, we give a reduction from 3SAT, involving 3 different kinds of agents, 5 different kinds of actions, and 2 different kinds of features. An example is given in Figure 4. For a 3SAT formula

$$\varphi = c_1 \wedge c_2 \wedge \cdots \wedge c_m$$

where each clause is a disjunction of 3 literals in the set $\{v_1, \overline{v_1}, v_2, \overline{v_2}, \ldots, v_n, \overline{v_n}\}$, we construct an instance of Problem 2 with the following agents, actions, and features. Note that in this section we break a few of the notational conventions laid out in Section 2. First, we use lowercase f's for the feature values so as to avoid confusion with the symbol "F," which is used to denote the Boolean value "false." These features are indexed by 2 variables instead of 1. Second, we refer to some of the actions not as numbers between 1 and m, but instead by symbols $a_{i,b}$, indexed by two variables i and b. We refer to the level of effort invested by an agent in such an action as $x_{i,b}$.

Agents	Purpose/Meaning
s_0	Can always strictly imitate all other agents, so the goal is to make no other agents able to imitate s_0.
$s_{1,i}$ for $i \in [n]$	Ensure s_0 invests in either action $a_{i,T}$ or $a_{i,F}$ (corresponding to setting v_i to be true or false).
$s_{2,j}$ for $j \in [m]$	Ensure clause c_j is satisfied by the assignment of variables corresponding to the actions that s_0 invests in.

Actions	In $\mathcal{A}$?	Purpose/Meaning
$a_{i,b}$ for $i \in [n]$, $b \in \{T, F\}$	✓	If s_0 invests in action $a_{i,b}$, it corresponds to setting $v_i = b$ in φ.
1	×	The action that any $s_{1,i}$ could use to imitate s_0 if s_0 does not invest in either $a_{i,T}$ or $a_{i,F}$.
2	×	The action that any $s_{2,j}$ could use to imitate s_0 if s_0's choice of actions does not satisfy clause c_j.
3	✓	The action that each $s \neq s_0$ should take in equilibrium.
4	×	The action that s_0 can use to imitate any $s \neq s_0$.

Features	Purpose/Meaning
$f_{i,b}$ for $i \in [n]$, $b \in \{T, F\}$	The feature values by which some $s \neq s_0$ might be able to imitate s_0.
f_0	The feature value by which s_0 can imitate every $s \neq s_0$.

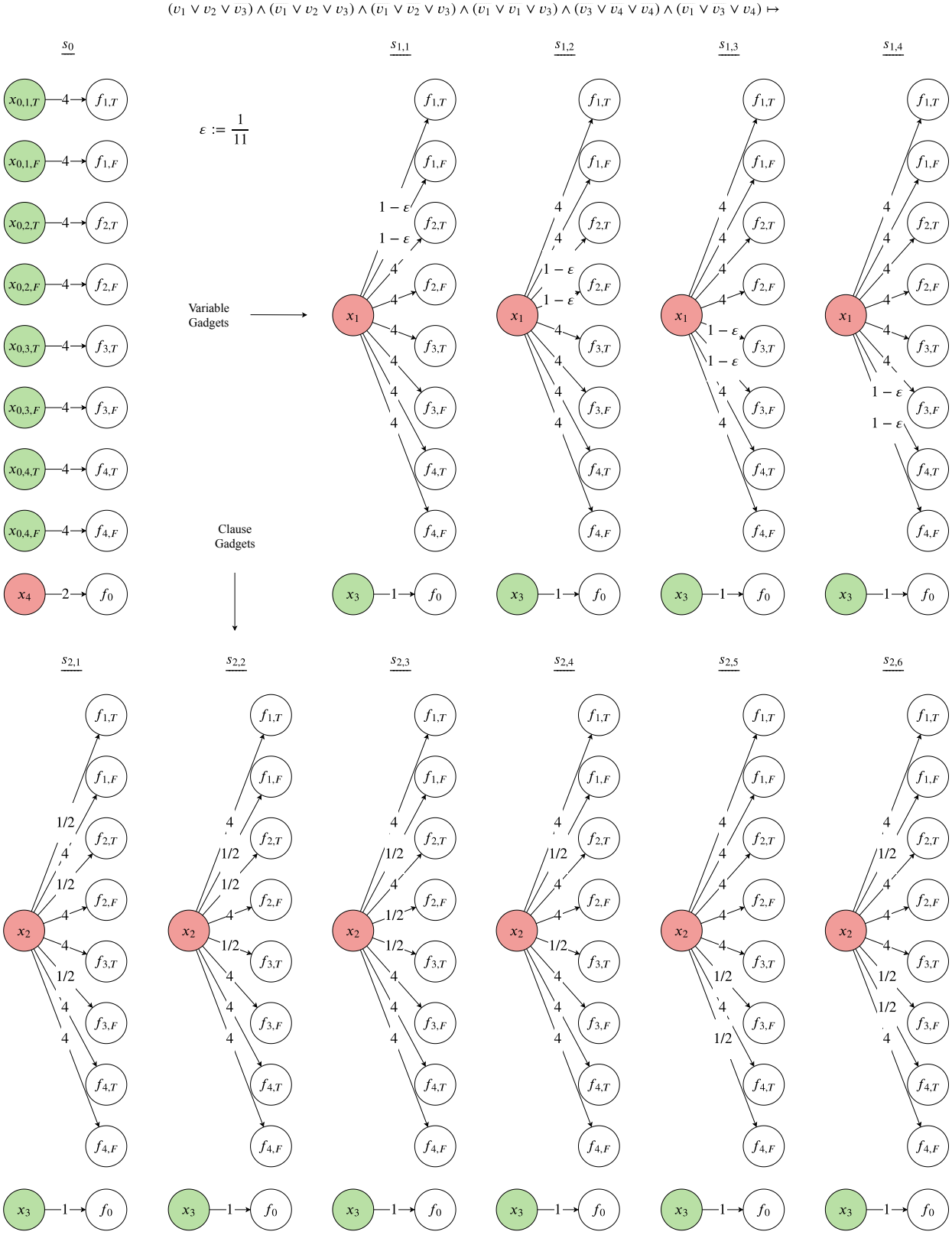

Figure 4: An example of the reduction for Theorem 3.3 part 2 where $n = 4$ and $m = 6$. The formula is uniquely satisfied by setting v_1 and v_4 to be false, and v_2 and v_3 to be true. Note that ε does not have to be $\frac{1}{11}$ exactly; it just has to be strictly smaller than $\frac{1}{10}$.

Formally, the coefficients for converting actions into features are given below, where, to avoid excessive subscripts, we denote by $\alpha(s, a, f)$ the effort conversion rate of action a into feature f, for agent s. Any α values that cannot be matched to an entry on the list are defined to be zero. Here ε can be any real number such that $0 < \varepsilon < \frac{1}{2(n+1)}$.

$$\alpha(s_0, a_{i,b}, f_{i,b}) := n \qquad \alpha(s_{1,i}, 3, f_0) := 1$$

$$\alpha(s_{1,i}, 1, f_{i',b}) := \begin{cases} 1-\varepsilon & \text{if } i = i' \\ n & \text{otherwise} \end{cases} \qquad \alpha(s_{2,j}, 3, f_0) := 1$$

$$\alpha(s_{2,j}, 2, f_{i,T}) := \begin{cases} \frac{1}{2} & \text{if literal } v_i \text{ appears in clause } c_j \\ n & \text{otherwise} \end{cases} \qquad \alpha(s_0, 4, f_0) := 2$$

$$\alpha(s_{2,j}, 2, f_{i,F}) := \begin{cases} \frac{1}{2} & \text{if literal } \overline{v_i} \text{ appears in clause } c_j \\ n & \text{otherwise} \end{cases}$$

To prove the theorem, it suffices to show that φ has a satisfying assignment if and only if there exists an effort profile $x^*(s)$ for each agent s such that the imitation graph with respect to x^* has no cycles containing any strict edges. It will be helpful for the reader to refer back to the "Purpose/Meaning" columns in the tables above as needed. Notice that, in this construction, all imitations can only occur between s_0 and other agents. This is because all features that are affected by s_0 choosing an admissible action can only be affected by inadmissible actions from other agents; and similarly, all features that are affected by agents other than s_0 choosing admissible actions can only be affected by inadmissible actions from s_0. Note also that, since agents other that s_0 have only one admissible action to choose from (action 3), and s_0 can strictly imitate this action by investing in the inadmissible action 4, no matter what, s_0 will always be able to strictly imitate all other agents. Therefore, to determine whether we can find an imitation graph with no cycles containing strict edges, it is equivalent to simply ensure that no agents can imitate s_0. So we must prove that φ has a satisfying assignment if and only if there exists a feasible $x^* \in \mathcal{A}$ such that no agents can imitate s_0 playing x^*.

For the forward direction, let v_i be a satisfying assignment of φ. Let x^* be the action profile defined by

$$x_{i,b} := \begin{cases} \frac{1}{n} & v_i = b \\ 0 & \text{otherwise} \end{cases},$$

$$x_1 := x_2 := x_3 := x_4 := 0.$$

Note that x^* is a feasible profile for s_0, since there are n nonzero values, each with investment $\frac{1}{n}$. Clearly, $x^* \in \mathcal{A}$, since all $a_{i,b}$ are admissible actions. No $s_{1,i}$ can imitate s_0, since even if they dedicated their entire budget to the one inadmissible action that affects feature f_{i,v_i} (namely, action 1), they will at most be able to achieve $f_{i,v_i} = 1 - \varepsilon$, whereas s_0 will achieve

$$f_{i,v_i} = \alpha(s_0, a_{i,v_i}, f_{i,v_i}) \cdot x_{i,v_i} = n \cdot \frac{1}{n} = 1.$$

No $s_{2,j}$ can imitate s_0 either, for if i is the index of the variable whose literal satisfies clause c_j, then the greatest value for f_{i,v_i} that $s_{2,j}$ can achieve is $\frac{1}{2}$ (by setting $x_2 = 1$), whereas s_0 will again achieve 1. Therefore, no agents can imitate s_0 playing x^*, so the forward direction is proved.

For the backward direction, suppose we have some admissible and feasible profile for s_0

$$x^* = (x_{1,T}, x_{1,F}, x_{2,T}, x_{2,F}, \ldots, x_{n,T}, x_{n,F}, x_1, x_2, x_3, x_4)$$

such that no $s \neq s_0$ can imitate s_0. We break this proof up into 3 claims.

Claim 1: For all $i \in [n]$, at least one element of the set $\{x_{i,T}, x_{i,F}\}$ is at least $\frac{1-\varepsilon}{n}$. Suppose, toward a contradiction, that there was some $i \in [n]$ such that $x_{i,T} < \frac{1-\varepsilon}{n}$ and $x_{i,F} < \frac{1-\varepsilon}{n}$. Then the feature values that s_0 achieves for $f_{i,T}$ and $f_{i,F}$ are both at most $1 - \varepsilon$. However, by setting $x_1 = 1$ (which is inadmissible), $s_{1,i}$ can achieve the feature vector where $f_{i,T} = f_{i,F} = 1 - \varepsilon$, and for all $i' \neq i$, $f_{i,T} = f_{i,F} = n$, which is weakly greater in all coordinates than the feature vector s_0 achieves by choosing x^* (recall that x_4 must be zero for x^* to be admissible). Thus, $s_{1,i}$ can imitate s_0, contradicting our hypothesis, so Claim 1 holds.

Claim 2: For each $i \in [n]$, one element of the set $\{x_{i,T}, x_{i,F}\}$ is at most $\frac{\frac{1}{2}-\varepsilon}{n}$, and the other is strictly greater than $\frac{\frac{1}{2}-\varepsilon}{n}$. It clearly follows from Claim 1 that at least one element is strictly greater than $\frac{\frac{1}{2}-\varepsilon}{n}$. Thus, we need only prove that they are not *both* strictly greater than $\frac{\frac{1}{2}-\varepsilon}{n}$, so suppose toward a contradiction that they are. Since at least one element of $\{x_{i,T}, x_{i,F}\}$ must be at least $\frac{1-\varepsilon}{n}$ by Claim 1, we have

$$x_{i,T} + x_{i,F} \geq \frac{1-\varepsilon}{n} + \frac{\frac{1}{2}-\varepsilon}{n} = \frac{\frac{3}{2}-2\varepsilon}{n}.$$

Therefore,

$$\begin{aligned}
\sum_{i' \in [n]} \sum_{b \in \{T,F\}} x_{i',b} &\geq \frac{\frac{3}{2} - 2\varepsilon}{n} + \sum_{i' \neq i} \sum_{b \in \{T,F\}} x_{i',b} \\
&\geq \frac{\frac{3}{2} - 2\varepsilon}{n} + (n-1)\frac{1-\varepsilon}{n} \quad \text{(by the previous claim)} \\
&= \frac{n + \left(\frac{1}{2} - (n+1)\varepsilon\right)}{n} \\
&> \frac{n+0}{n} \quad \text{(because } \varepsilon < \frac{1}{2(n+1)}\text{)} \\
&= 1
\end{aligned}$$

Since the sum of all effort is greater than 1, x^* is infeasible, contradicting our hypothesis. Thus, Claim 2 holds.

Claim 3: The following assignment of variables satisfies φ:

$$v_i := \begin{cases} T & \textbf{if } x_{i,T} > \frac{\frac{1}{2}-\varepsilon}{n} \textbf{ and } x_{i,F} \leq \frac{\frac{1}{2}-\varepsilon}{n} \\ F & \textbf{if } x_{i,F} > \frac{\frac{1}{2}-\varepsilon}{n} \textbf{ and } x_{i,T} \leq \frac{\frac{1}{2}-\varepsilon}{n} \end{cases}$$

Note that this is well-defined by Claim 2, as these are the only two possible cases. Suppose, toward a contradiction, that there was some clause c_j which is not satisfied. Let $i_1, i_2, i_3 \in [n]$ be the indices of the three variables appearing in c_j, and let $b_1, b_2, b_3 \in \{T, F\}$ be such that $b_k = T$ if the variable v_{i_k} appears positively, and $b_k = F$ if it appears as the negated literal $\overline{v_{i_k}}$. Then for each $k \in \{1, 2, 3\}$, $s_{2,j}$ achieves $f_{i_k,b_k} = \frac{1}{2}$ from investing all effort in the inadmissible action 2, whereas s_0 achieves at most $\frac{\frac{1}{2}-\varepsilon}{n}$ from choosing x^*, since if some $x_{i_k,b_k} > \frac{\frac{1}{2}-\varepsilon}{n}$, by the definition of the assignment v_{i_k} we would have that the literal in which v_{i_k} appears would satisfy the clause (either $v_{i_k} = b_k = T$ or $v_{i_k} = b_k = F$). By investing all effort in action 2, $s_{2,j}$ clearly dominates s_0 in all other features as well, achieving n in each coordinate. Thus, $s_{2,j}$ can imitate s_0, contradicting our hypothesis. It follows that the assignment v_i satisfies φ, so the proof of the backward direction is complete. □

D Hardness of Problems 3, 4, 7 and 8

In this section we prove that maximizing the number of agents that choose admissible action profiles is NP-complete in general. We give one reduction that works for all four variants of this optimization problem (Problems 3, 4, 7 and 8). However, the instances of the evaluation problem produced by this reduction have an unbounded number of features, so we can conclude nothing about the complexity of any of the four problems when there are a constant number of features. As it turns out, Problems 3 and 4 are still hard under this restriction; we present a separate reduction in Appendix E to prove this fact. On the other hand, we showed in Section 4 that Problems 7 and 8 are solvable in polynomial time when the number of features is constant.

Theorem D.1. *The problems of finding a monotone mechanism that incentivizes a maximum number of agents to choose admissible actions / a specific admissible profile, or determining that no such mechanism exists, are NP-complete.*

Proof. Problems 3 and 4 are in NP since, given a subset of agents and admissible profiles for each of them to play, we can easily verify that the imitation graph has with no cycles containing any strict edges by Lemma B.1. To prove that these problems are NP-hard, we give a reduction from Feedback Vertex Set.

This proof was outlined in Section 3.3, but we repeat the entire argument here for completeness. Suppose we are given an instance of Feedback Vertex Set, that is, a directed graph G with n vertices. For convenience, assume $V(G) = [n]$. We construct an instance of the evaluation problem with agents s_1,s_2,…,s_n, features F_1,F_2,…,F_n, and 2 actions, where action 1 is admissible and action 2 is inadmissible (for the admissible profile variant, just take the admissible profile to be $(1, 0)$, meaning all effort should be invested into action 1). For each $i, k \in [n]$, define

$$\alpha^k_{1,i} := \begin{cases} 1 & i = k \\ 0 & \text{otherwise} \end{cases}, \qquad \alpha^k_{2,i} := \begin{cases} 2 & (k,i) \in E(G) \\ \varepsilon & \text{otherwise} \end{cases}$$

(see Figure 3 in Section 3.3 for an example).

We claim that that, for $0 < \varepsilon < 1$, G is the imitation graph with respect to the profile assignment of $(1, 0)$ for all agents (which is the only potential admissible profile that can be weakly optimal), and all edges are strict edges. Since both graphs have the same vertex set, we need only show that they have the same edge sets. First suppose $(k, i) \in E(G)$. We show that $F^k((0, 1)) > F^i((1, 0))$, so that (k, i) is a strict edge of the imitation graph. Note that, at index i,

$$F^k((0,1))_i = 2 > 1 = F^i((1,0))_i,$$

and for all indices $j \neq i$,

$$F^k((0,1))_j \geq \varepsilon > 0 = F^i((1,0))_j$$

Thus, $F^k((0,1)) > F^i((1,0))$ as desired.

Conversely, suppose there is an edge from k to i in the imitation graph. That is, there is some inadmissible $x = (x_1, x_2) \in \mathcal{X}$ such that $F^k(x) \geq F^i((1,0))$. In particular, considering index i of both vectors, we know that

$$F^k(x)_i \geq F^i(1,0)_i = 1.$$

Suppose, toward a contradiction, that $(k,i) \notin E(G)$. Then $F^k((0,1))_i = \varepsilon$, so, starting from the equation above,

$$1 \leq F^k(x)_i = x_1 F^k((1,0))_i + x_2 F^k((0,1))_i = x_1 F^k((1,0))_i + x_2\varepsilon \leq x_1 + x_2\varepsilon,$$

where the final inequality follows because all entries of $F^k((1,0))$ are at most 1 by definition. The budget constraint implies $x_1 \leq 1 - x_2$, so it follows that $1 \leq (1 - x_2) + x_2\varepsilon$, so $(1-\varepsilon)x_2 \leq 0$. Since $\varepsilon < 1$, this means $x_2 \leq 0$, which can only happen if $x_2 = 0$. This contradicts our assumption that x is inadmissible, so we conclude that $(k,i) \in E(G)$.

Thus, by Theorem 3.1, G has a feedback vertex set of size at most q if and only if at least $n - q$ agents (namely, those not in the feedback vertex set) can be jointly incentivized to invest only in action 1. □

Proof of Theorem 4.4 Part 2. Problems 7 and 8 are in NP since it is easy to compute best responses to a given linear mechanism, and hence determine which agents will invest in admissible actions / an admissible profile. For hardness, we claim that, by specializing $\varepsilon < \frac{1}{n \cdot 3^n}$, the same reduction given in the proof of Theorem D.1 works for Problems 7 and 8 as well. To prove this, it suffices to show that, in this construction, there exists a monotone mechanism to incentivize a given subset of agents to invest only in action 1 if and only if there exists a linear mechanism. The backward direction is trivial. To prove the forward direction, we assume that the subgraph of the input graph G induced by some arbitrary subset $T \subseteq V(G)$ has no cycles, and let $v : T \to [n]$ be a reverse topological ordering of these vertices, as in the proof of Theorem 3.1. We claim that the following linear mechanism incentivizes all agents to invest only in action 1, because it yields a higher marginal payoff:

$$\beta_i := \begin{cases} 3^{v(i)} & \text{if } i \in T \\ 0 & \text{if } i \in S \setminus T \end{cases}$$

An arbitrary agent $i \in T$ gets a marginal payoff of $3^{v(i)}$ from investing in action 1, and

$$\sum_{j \in T \text{ s.t. } (i,j) \in E(G[T])} 2 \cdot 3^{v(j)} + \sum_{j \in T \text{ s.t. } (i,j) \notin E(G[T])} \varepsilon \cdot 3^{v(j)}$$

from investing in action 2. Since v is a reverse topological ordering, for any j that i has an edge to, $v(j) \leq v(i) - 1$. Also, note that each such j has to have a distinct index under v. Therefore, we can rewrite the first term above as summing over the positive v values that are realized in the exponent,

$$\sum_{k=1}^{v(i)-1} \begin{cases} 2 \cdot 3^k & \text{if there exists } j \text{ such that } (i,j) \in E(G) \text{ and } v(j) = k \\ 0 & \text{otherwise} \end{cases},$$

which is bounded above by

$$2 \cdot \sum_{k=1}^{v(i)-1} 3^k = 3^{v(i)} - 1.$$

As for the second term, since there are at most n terms in the sum, $v(j) \leq n$ for all n, and $\varepsilon < \frac{1}{n \cdot 3^n}$, we have that

$$\sum_{j \in T \text{ s.t. } (i,j) \notin E(G[T])} \varepsilon \cdot 3^{v(j)} \leq n\varepsilon \cdot 3^n < \frac{n \cdot 3^n}{n \cdot 3^n} = 1.$$

Adding together these two bounds, we have that the total marginal payoff of agent i from investing in action 2 is strictly less than

$$3^{v(i)} - 1 + 1 = 3^{v(i)},$$

which is the marginal payoff to investing in action 1. Therefore, all agents in T are strictly incentivized to invest only in action 1. □

Note the similarities between this mechanism, and the one in the proof of Theorem 3.1. Both mechanisms involve ranking the agents so that those with the power to imitate other agents do not want to exercise this power, since the other agents are lower-ranked, and thus receive lower payoffs. As long as there are no inconsistencies among the constraints that certain agents must be ranked higher than other ones, it is always possible to implement such a ranking system with a monotone mechanism. The main takeaway from this proof is that, with enough features to distinguish the agents, it is *sometimes* possible to achieve the same effect with a linear mechanism. However, as Example 2.2 showed, this is not always the case.

E Hardness of Problems 3 and 4 with a Constant Number of Features

Proof of Theorem 3.4. We follow the same proof strategy as in Theorem D.1. As noted in that proof, to reduce from Feedback Vertex Set it suffices to show that any directed graph can be constructed as an imitation graph (with only strict edges) in polynomial time. The only new constraint is that there must be a constant number of features in the instance we create; this construction uses only 2 features.

Let G be an arbitrary digraph with n vertices. We construct an instance of the evaluation problem where the set of agents is the same as the set of vertices, which we will label as $S = V(G) = [n]$. There will be $n+1$ actions, where only the last action, $n+1$, is admissible. (As in the proof of Theorem D.1, since there is only one admissible action, the admissible profile variant of the evaluation problem is equivalent to the admissible actions variant.) The effort invested in these actions converts into the 2 features as follows:

$$\alpha^k_{n+1,1} := \cos\frac{\pi k}{2(n+1)}$$

$$\alpha^k_{n+1,2} := \sin\frac{\pi k}{2(n+1)}$$

$$\text{for } j \in [n]\text{: } \alpha^k_{j,1} := \begin{cases} (1+\varepsilon)\cos\frac{\pi j}{2(n+1)} & \text{if } (k,j) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

$$\text{for } j \in [n]\text{: } \alpha^k_{j,2} := \begin{cases} (1+\varepsilon)\sin\frac{\pi j}{2(n+1)} & \text{if } (k,j) \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

It is clear that agent k can strictly imitate agent j if $(k,j) \in E(G)$. All that remains is to check that k cannot imitate any other agents. This is best seen geometrically, plotting F_1 on the x-axis and F_2 on the y-axis. In this construction, we have arranged the feature profiles that result from each agent playing the admissible action profile $(0,0,\ldots,0,1)$ on a unit circle, equally spaced throughout the first quadrant.

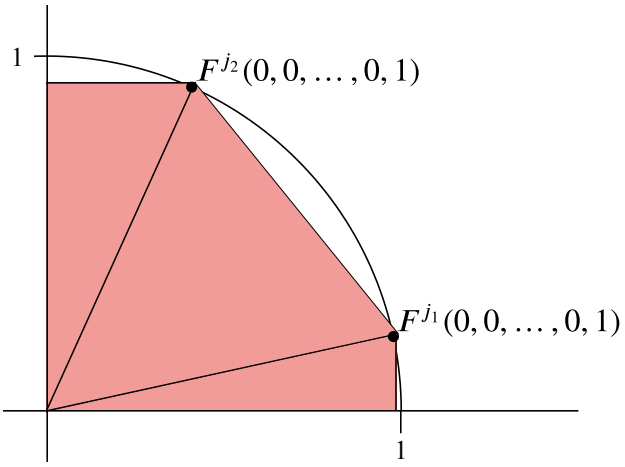

Figure 5: The region that an arbitrary agent corresponding to a vertex of degree 2 can imitate.

The shaded region in Figure 5 denotes the set of feature vectors that an arbitrary agent, with edges to j_1 and j_2 in G, can imitate. Note that it extends outside of the unit circle by ε at two points (dominating the feature vectors that agents j_1 and j_2 achieve by investing all effort in action $n+1$). However, due to the strict convexity of the circle, by making ε small enough, this region will not include any other feature vectors achieved by agents other than j_1 and j_2. This easily generalizes to vertices of arbitrary degree. Thus, the imitation graph of this instance is precisely G. □

F Hardness of Problem 6

Proof of Theorem 4.4 Part 2. Problem 6 is in NP since, given a linear mechanism, it is easy to compute the best responses of each agent by comparing the marginal payoffs of each action. One simply checks that some admissible action realizes the maximum marginal payoff over all actions.

To prove NP-hardness, we give a reduction from 3SAT. For a 3SAT formula

$$\varphi = c_1 \wedge c_2 \wedge \cdots \wedge c_m$$

where each clause is a disjunction of 3 literals in the set $\{v_1, \overline{v_1}, v_2, \overline{v_2}, \ldots, v_n, \overline{v_n}\}$, we construct an instance of the evaluation problem with 7 actions, where only actions 1 and 2 are admissible, and features $f_{i,j,b}$ for every $i \in [n]$, $j \in [m]$, and $b \in \{T, F\}$. Recall that, since we are looking for a linear mechanism, the objective is to choose coefficients $\beta_{i,j,b}$ for each $f_{i,j,b}$ such that all agents invest only in admissible actions. The main idea behind this reduction is that choosing a linear mechanism $\overrightarrow{\beta} \in \mathbb{R}^{2mn}$ in which $\beta_{i,j,T} > \beta_{i,j,F}$ will correspond to setting v_i to be true in φ, and choosing a $\overrightarrow{\beta}$ in which $\beta_{i,j,T} < \beta_{i,j,F}$ will correspond to setting v_i to be false in φ. There are three different kinds of agents which will be used to ensure that all agents are incentivized to choose admissible actions if and only if the assignment of variables obtained through this correspondence satisfies φ:

Agents	Purpose/Meaning
$s_{1,i,j}$ for $i \in [n]$, $j \in [m]$	Ensure that the coefficients of $f_{i,j,T}$ and $f_{i,j,F}$ differ by a substantial factor.
$s_{2,i,j}$ for $i \in [n]$, $j \in [m-1]$	Ensure that the choice of which feature is given greater weight among the features $\{f_{i,j,T}, f_{i,j,F}\}$ is consistent between j and $j+1$.
$s_{3,j,k}$ for $j \in [m]$, $k \in [3]$	These 3 agents create an inconsistent cycle of inequalities among the β values whenever clause j is not satisfied.

The effort conversion rates for each type of agent are as depicted in Figure 6. Note that, for the last three agents, i_1, i_2, i_3, b_1, b_2, and b_3 are such that clause c_j contains variables v_{i_1}, v_{i_2}, and v_{i_3}, in that order, negated according to b_1, b_2, and b_3, respectively. Also, all agents have one more inadmissible action 7 to choose from, which is not depicted in the figure, with a conversion rate of $\varepsilon := \frac{1}{120mn} \cdot \left(\frac{6}{11}\right)^n$ to every feature.

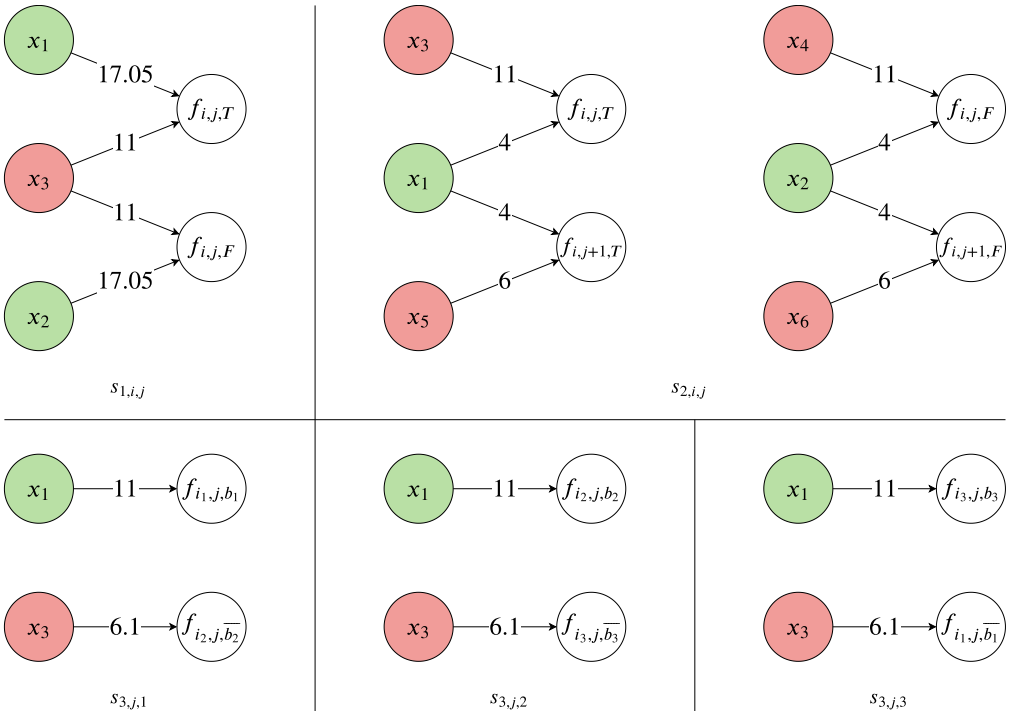

Figure 6: The effort conversion rates for each type of agent (excluding action 7).

We now prove that the reduction is correct. For the forward direction, suppose that v_i is a satisfying assignment of φ. We will show that the following linear mechanism $\overrightarrow{\beta} \in \mathbb{R}^{2mn}$ incentivizes all agents to invest only in the admissible actions 1 and 2:

$$\beta_{i,j,b} := \begin{cases} \gamma_{i,j}\left(\frac{11}{6}\right)^j & \text{if } v_i = b \\ \gamma_{i,j}\left(\frac{11}{6}\right)^{j-1} & \text{if } v_i \neq b \end{cases},$$

where

$$\gamma_{i,j} := \begin{cases} 60 & \text{if variable } v_i \text{ appears in } c_j \text{ as an unsatisfied literal, and the next} \\ & \qquad \text{literal to the right (cyclicly, within } c_j) \text{ is also unsatisfied} \\ 59 & \text{otherwise} \end{cases}.$$

First observe that, for all agents, it is never optimal to invest any effort into the inadmissible action 7. This is because there are $2mn$ features, each with a β value of at most $60\left(\frac{11}{6}\right)^n$, so the marginal payoff toward investing in action 7 is at most

$$\varepsilon \cdot 2mn \cdot 60\left(\frac{11}{6}\right)^n = 1.$$

Since all other actions have marginal benefits strictly greater than 1, no agent will be incentivized to invest any effort into action 7. So for each agent type, we need only consider how the marginal payoffs toward actions 1 and 2 compare to the marginal payoffs toward actions 3, 4, 5, and 6.

For an arbitrary agent $s_{1,i,j}$, the marginal payoff toward the inadmissible action 3 is

$$11\left(\gamma_{i,j}\left(\frac{11}{6}\right)^j + \gamma_{i,j}\left(\frac{11}{6}\right)^{j-1}\right) = 11\gamma_{i,j}\left(\frac{11}{6}\right)^j\left(1+\left(\frac{6}{11}\right)\right) = 17\gamma_{i,j}\left(\frac{11}{6}\right)^j,$$

while the payoffs toward the admissible actions 1 and 2 are

$$17.05\gamma_{i,j}\left(\frac{11}{6}\right)^j \qquad \text{and} \qquad 17.05\gamma_{i,j}\left(\frac{11}{6}\right)^{j-1}$$

(which action yields which marginal payoff depends on whether v_i is set to true or false). The first of these admissible actions will be the most profitable of the three choices, so agent $s_{1,i,j}$ will invest their entire budget into that admissible action.

For an arbitrary agent $s_{2,i,j}$, there are two completely analogous cases, depending on whether v_i is true or false. By symmetry, it is without loss of generality to only consider the case where $v_i = T$. In this case, the marginal payoff toward action 1 is

$$4\left(\gamma_{i,j}\left(\frac{11}{6}\right)^j + \gamma_{i,j+1}\left(\frac{11}{6}\right)^{j+1}\right) = 4\left(\frac{11}{6}\right)^j\left(\gamma_{i,j} + \frac{11}{6}\gamma_{i,j+1}\right)$$

Since $\gamma_{i,j}$ and $\gamma_{i,j+1}$ are each at least 58.5, this marginal payoff is at least

$$4 \cdot 58.5 \cdot \left(\frac{11}{6}\right)^j\left(1+\frac{11}{6}\right) = \frac{4 \cdot 58.5 \cdot 17}{6}\left(\frac{11}{6}\right)^j = 663\left(\frac{11}{6}\right)^j.$$

Since $\gamma_{i,j}$ and $\gamma_{i,j+1}$ are each at most 60, the marginal payoffs toward the four inadmissible actions are at most

$$11 \cdot 60\left(\frac{11}{6}\right)^j = 660\left(\frac{11}{6}\right)^j, \qquad 11 \cdot 60\left(\frac{11}{6}\right)^{j-1} = 660\left(\frac{11}{6}\right)^{j-1},$$

$$6 \cdot 60\left(\frac{11}{6}\right)^{j+1} = 660\left(\frac{11}{6}\right)^j, \quad \text{and} \quad 6 \cdot 60\left(\frac{11}{6}\right)^j = 660\left(\frac{11}{6}\right)^{j-1},$$

respectively. Since each of these are less than the marginal payoff toward action 1, we conclude that agent $s_{2,i,j}$ will invest their entire budget into that admissible action. (Analogously, if $v_i = F$, essentially the same proof shows that agent $s_{2,i,j}$ will invest their entire budget into the admissible action 2.)

Finally, we consider the $s_{3,j,k}$ agents. Again observe that, for each j, there is a symmetry among all three of these agents, so it suffices to consider only the first agent, $s_{3,j,1}$. We break this argument up into 3 cases, depending on which variable satisfies the clause c_j.

If v_{i_1} satisfies c_j, then, by definition, we have $v_{i_1} = b_1$, so the payoff toward action 1 is

$$11\gamma_{i_1,j}\left(\frac{11}{6}\right)^j \geq 649\left(\frac{11}{6}\right)^j$$

(because $\gamma_{i_1,j} \geq 59$). On the other hand, the payoff toward action 3 is either

$$6.1\gamma_{i_2,j}\left(\frac{11}{6}\right)^{j-1} \leq 366\left(\frac{11}{6}\right)^{j-1} \qquad \text{or} \qquad 6.1\gamma_{i_2,j}\left(\frac{11}{6}\right)^j \leq 366\left(\frac{11}{6}\right)^j,$$

(because $\gamma_{i_2,j} \leq 60$) depending on whether or not $v_{i_2} = b_2$. In either case, it is more profitable to invest in action 1.

If v_{i_2} satisfies c_j, then, by definition, we have $v_{i_2} = b_2 \neq \overline{b_2}$, so the payoff toward action 3 is

$$6.1\gamma_{i_2,j}\left(\frac{11}{6}\right)^{j-1} \leq 336\left(\frac{11}{6}\right)^{j-1}.$$

On the other hand, the payoff toward action 1 is either

$$11\gamma_{i_1,j}\left(\frac{11}{6}\right)^{j} \geq 649\left(\frac{11}{6}\right)^{j} \qquad \text{or} \qquad 11\gamma_{i_1,j}\left(\frac{11}{6}\right)^{j-1} \geq 649\left(\frac{11}{6}\right)^{j-1},$$

depending on whether or not $v_{i_1} = b_1$. So again, in either case, it is more profitable to invest in action 1.

Finally, consider the case where neither variables v_{i_1} nor v_{i_2} satisfy c_j, and v_{i_3} does satisfy c_j. This is the one place in the proof where we leverage the difference between the two possible values for $\gamma_{i,j}$. Since $v_{i_1} \neq b_1$ and neither v_{i_1} nor the next variable v_{i_2} satisfy c_j, the payoff toward action 1 is

$$11\gamma_{i_1,j}\left(\frac{11}{6}\right)^{j-1} = 11 \cdot 60\left(\frac{11}{6}\right)^{j-1} = 660\left(\frac{11}{6}\right)^{j-1}.$$

On the other hand, since $v_{i_2} = \overline{b_2}$, and the next variable v_{i_3} does satisfy c_j, the payoff toward action 3 is

$$6.1\gamma_{i_2,j}\left(\frac{11}{6}\right)^{j} = 6.1 \cdot 59 \cdot \left(\frac{11}{6}\right)^{j} = 359.9 \cdot \left(\frac{11}{6}\right)^{j} = 659.81\overline{6}\left(\frac{11}{6}\right)^{j-1} < 660\left(\frac{11}{6}\right)^{j-1}.$$

So, again, action 1 is the most profitable.

Thus, in all three cases, agent $s_{3,j,1}$ is incentivized to invest all of its effort into the admissible action 1, concluding the proof of the forward direction.

For the backward direction, suppose that we have some monotone mechanism $\overrightarrow{\beta} \in \mathbb{R}^{2mn}$ that incentivizes all agents to invest effort only in desirable actions. Note that monotonicity requires that every β value be nonnegative. We break this proof up into 3 claims, which show that each of the 3 types of agents function as stated in the "Purpose/Meaning" column of the table.

Claim 1: For all $i \in [n]$ and $j \in [m]$, the feature coefficients $\beta_{i,j,T}$ and $\beta_{i,j,F}$ are not both zero, and differ by a factor of at least $\frac{11}{6.05}$. It is clear that these two feature values are not both zero, for otherwise agent $s_{1,i,j}$ would be strictly incentivized to invest all effort into the inadmissible action 7. To prove the second part of the claim, observe that, for agent $s_{1,i,j}$, the marginal payoff to investing in the admissible action 1 is $17.05\beta_{i,j,T}$, the marginal payoff to investing in the admissible action 2 is $17.05\beta_{i,j,F}$, and the payoff to investing in the inadmissible action 3 is $11(\beta_{i,j,T} + \beta_{i,j,F})$. Since one of the admissible actions must be a best response for $s_{1,i,j}$, we know that

$$17.05\beta_{i,j,T} \geq 11(\beta_{i,j,T} + \beta_{i,j,F}) \qquad \text{or} \qquad 17.05\beta_{i,j,F} \geq 11(\beta_{i,j,T} + \beta_{i,j,F}).$$

In other words, we must have

$$\beta_{i,j,T} \geq \frac{11}{6.05}\beta_{i,j,F} \qquad \text{or} \qquad \beta_{i,j,F} \geq \frac{11}{6.05}\beta_{i,j,T}.$$

This proves Claim 1.

Claim 2: For all $i \in [n]$ and $j_1, j_2 \in [m]$, $\beta_{i,j_1,T} > \beta_{i,j_1,F}$ if and only if $\beta_{i,j_2,T} > \beta_{i,j_2,F}$. By induction, it suffices to prove that, for each $j \in [m-1]$, $\beta_{i,j,T} > \beta_{i,j,F}$ if and only if $\beta_{i,j+1,T} > \beta_{i,j+1,F}$. There are two completely analogous cases, depending on which admissible action $s_{2,i,j}$ is investing effort into as a best response. We will only discuss the case where they are investing effort into action 1. Since action 1 must yield a weakly higher marginal payoff than action 7, it must be that one of $\beta_{i,j,T}$ or $\beta_{i,j+1,T}$ is nonzero. But if one of them is nonzero, the other one must clearly be nonzero as well, for otherwise one of the two inadmissible actions, 3 or 5, is a strictly better response. Thus, by re-scaling $\overrightarrow{\beta}$, it is without loss of generality to assume that $\beta_{i,j,T} = 6$. (We could have chosen any positive constant, but 6 is convenient for avoiding non-integral values in the argument that follows.) One can check that we must then have $\beta_{i,j+1,T} \in [10.5, 12]$ for otherwise investing in action 3 or action 5 is a strictly better response. So the marginal payoff toward investing in action 1 is at most $4(6+12) = 72$. On the other hand, the payoff to investing in the inadmissible action x_4 is $11\beta_{i,j,F}$, and the payoff to investing in the inadmissible action 6 is $6\beta_{i,j+1,F}$. As neither of these can be better responses than investing in the admissible action 1, we derive that $\beta_{i,j,F} \leq \frac{72}{11} = 6.\overline{54}$ and $\beta_{i,j+1,F} \leq \frac{72}{6} = 12$. Since we have already established that $\beta_{i,j,T} = 6$ and $\beta_{i,j+1,T} \in [10.5, 12]$, it follows from Claim 1 that

$$\beta_{i,j,F} \leq \frac{6.05}{11}\beta_{i,j,T} < \beta_{i,j,T}$$

and

$$\beta_{i,j+1,F} \leq \frac{6.05}{11}\beta_{i,j+1,T} < \beta_{i,j+1,T}.$$

Thus, we have proved that $\beta_{i,j,T} > \beta_{i,j,F}$ if and only if $\beta_{i,1,T} > \beta_{i,1,F}$, as both statements are true; in the analogous case where $s_{2,i,j}$ invests in the other admissible action 2, we would derive that both statements are false. In both cases, Claim 2 holds.

Claim 3: The following assignment of variables satisfies φ:

$$v_i := \begin{cases} T & \textbf{if, for all } j \in [m], \beta_{i,j,T} \geq \frac{11}{6.05}\beta_{i,j,F} \\ F & \textbf{if, for all } j \in [m], \beta_{i,j,F} \geq \frac{11}{6.05}\beta_{i,j,T} \end{cases}.$$

Note that this is well-defined by Claims 1 and 2, as these are the only two possible cases. To prove the claim, suppose, toward a contradiction, that some clause c_j is not satisfied by this assignment. Suppose that c_j prescribes that v_{i_k} should have value $b_k \in \{T, F\}$, for at least one $k \in [3]$, and that the variables are listed in increasing order of k. Since c_j is not satisfied, it means that $v_{i_k} = \overline{b_k}$ for all $k \in [3]$. For each v_{i_k} appearing in c_j, if $v_{i_k} = T$ we have $\beta_{i_k,j,T} \geq \frac{11}{6.05}\beta_{i_k,j,F}$, and if $v_{i_k} = F$ we have $\beta_{i_k,j,F} \geq \frac{11}{6.05}\beta_{i_k,j,T}$. Thus, in either case, since $v_{i_k} = \overline{b_k}$, we know that $\beta_{i_k,j,\overline{b_k}} \geq \frac{11}{6.05}\beta_{i_k,j,b_k}$. It then follows that

$$\begin{aligned}
\beta_{i_1,j,b_1} &\geq \frac{6.1}{11} \cdot \beta_{i_2,j,\overline{b_2}} \\
&\geq \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \beta_{i_2,j,b_2} \\
&\geq \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \frac{6.1}{11} \cdot \beta_{i_3,j,\overline{b_3}} \\
&\geq \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \beta_{i_3,j,b_3} \\
&\geq \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \frac{6.1}{11} \cdot \beta_{i_1,j,\overline{b_1}} \\
&\geq \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \frac{6.1}{11} \cdot \frac{11}{6.05} \cdot \beta_{i_1,j,b_1},
\end{aligned}$$

where the first, third, and fifth inequalities follow from the conditions that $s_{3,j,1}$, $s_{3,j,2}$, and $s_{3,j,3}$ must be incentivized to invest only in action 1. Therefore,

$$\beta_{i_1,j,b_1} \geq \left(\frac{6.1}{6.05}\right)^3 \beta_{i_1,j,b_1},$$

implying that $\beta_{i_1,j,b_1} = 0$, and hence $\beta_{i_2,j,\overline{b_2}} = 0$ as well. But this means that agent $s_{3,j,1}$ receives a marginal payoff of zero from investing in action 1, so it would have been better for them to instead invest in the inadmissible action 7, which yields a strictly positive marginal payoff. We have a contradiction, so it follows that the assignment v_i defined above satisfies φ, concluding the proof of the backward direction. □

G Algorithms for Problems 6 and 8 with a Constant Number of Features

Proof of Theorem 4.3. To solve Problem 8, recall that, since agents are incentivized to choose the action(s) with the greatest marginal payoff, we need only ensure that some admissible action is the most profitable. Algorithm 3, which differs from Algorithm 1 only on lines 1 and 5, solves this problem. Since Problem 6 is a special case of Problem 8, we thus have an algorithm for Problem 6 as well.

Algorithm 3: An algorithm for Problem 8.

Input: An instance of the evaluation problem with admissible actions $A \subseteq [m]$
Output: A linear mechanism β that incentivizes a maximum number of agents to invest effort only in admissible actions

1 $\mathcal{R} \leftarrow$ arrangement of all hyperplanes for constraints $h(k, j_1, j_2)$ for all $k \in [\ell]$, $j_1 \in A$, and $j_2 \in [m]$;
2 `max` $\leftarrow -1$;
3 **for** *each cell* $C \in \mathcal{R}$ **do**
4 $\beta' \leftarrow$ any point in C;
5 `numIncentivized` $\leftarrow |\{s_k \in S \mid$ some action in A yields the (weakly) greatest marginal payoff for s_k under $\beta'\}|$;
6 **if** *numIncentivized* $>$ *max* **then**
7 `max` $\leftarrow$ `numIncentivized`;
8 $\beta \leftarrow \beta'$;
9 **end**
10 **end**
11 **return** β;

□

H Superiority of Nonlinear Mechanisms with a Constant Number of Features

Example 2.2 showed that the best nonlinear mechanisms can perform arbitrarily better than the best linear mechanisms. Here we give another example of this phenomenon, only this time, our construction uses only 2 features.

Example H.1. For any positive integer n, let G_n be the graph with vertex set $[2n+1]$, and an edge from i to j if and only if i is even and $|j-i|=1$. Then apply the reduction in the proof of Theorem 3.4 (see Appendix E), and consider the evaluation problem restricted to only the set of n even-numbered agents. Since G_n has no cycles, the induced subgraph of G_n generated by the even-numbered vertices has no cycles as well, so it follows from Theorem 3.1 that it is possible to incentivize all n agents to invest only in the admissible action using a monotone mechanism. However, we claim that at most 1 agent can be incentivized to invest only in the admissible action using a linear mechanism.

To see this, take any nontrivial linear mechanism $\beta \in \mathbb{R}^2_{\geq 0}$, and let $\theta \in [0, \frac{\pi}{2}]$ be the angle that β makes with the x-axis (which is identified with feature F_1). Without loss of generality we can assume that β is a unit vector. Suppose some arbitrary agent $k \in [2n+1]$ (where k is even) is incentivized to invest only in the admissible action. Then the marginal payoff toward the admissible action, which is

$$\left(\cos\frac{\pi k}{4(n+1)}, \sin\frac{\pi k}{4(n+1)}\right)\cdot\beta,$$

must be greater than the marginal payoffs toward the two inadmissible actions, which are

$$(1+\varepsilon)\left(\cos\frac{\pi(k-1)}{4(n+1)}, \sin\frac{\pi(k-1)}{4(n+1)}\right)\cdot\beta \qquad \text{and} \qquad (1+\varepsilon)\left(\cos\frac{\pi(k+1)}{4(n+1)}, \sin\frac{\pi(k+1)}{4(n+1)}\right)\cdot\beta.$$

We can drop the $(1+\varepsilon)$ terms, obtaining

$$\left(\cos\frac{\pi k}{4(n+1)}, \sin\frac{\pi k}{4(n+1)}\right)\cdot\beta \geq \left(\cos\frac{\pi(k-1)}{4(n+1)}, \sin\frac{\pi(k-1)}{4(n+1)}\right)\cdot\beta \implies \cos\left|\theta - \frac{\pi k}{4(n+1)}\right| \geq \cos\left|\theta - \frac{\pi(k-1)}{4(n+1)}\right|,$$
$$\left(\cos\frac{\pi k}{4(n+1)}, \sin\frac{\pi k}{4(n+1)}\right)\cdot\beta \geq \left(\cos\frac{\pi(k+1)}{4(n+1)}, \sin\frac{\pi(k+1)}{4(n+1)}\right)\cdot\beta \implies \cos\left|\theta - \frac{\pi k}{4(n+1)}\right| \geq \cos\left|\theta - \frac{\pi(k+1)}{4(n+1)}\right|.$$

Here we are using the fact that all vectors are unit vectors, so the dot products are equal to the cosines of the angles between them.

The first condition implies that θ is closer to the angle $\frac{\pi k}{4(n+1)}$ than the angle $\frac{\pi(k-1)}{4(n+1)}$, so it follows that

$$\theta \geq \frac{\pi(k-\frac{1}{2})}{4(n+1)}.$$

The second condition implies that θ is also closer to the angle $\frac{\pi k}{4(n+1)}$ than the angle $\frac{\pi(k+1)}{4(n+1)}$, so it follows that

$$\theta \leq \frac{\pi(k+\frac{1}{2})}{4(n+1)}.$$

Thus, for agent k to be incentivized to invest only in the admissible action, we must have

$$\theta \in \left[\frac{\pi(k-\frac{1}{2})}{4(n+1)}, \frac{\pi(k+\frac{1}{2})}{4(n+1)}\right].$$

These intervals do not overlap for values of k that differ by at least 2 (recall we are only considering the even-numbered agents), so we conclude that it is not possible to incentivize more than one agent with a linear mechanism.

I Extension of Algorithms: Concave Effort Conversion Functions

Kleinberg and Raghavan (2019) consider a slightly more general form of the evaluation problem than we do. They define $F^k(x^k)_i = f_i^k(\sum_j \alpha_{j,i}^k x_j^k)$, where f_i^k is a concave, strictly increasing function. In this setting, the model we have considered is the special case in which each f_i^k is the identity function. While all of our hardness results also hold for the more general model, our algorithms do not immediately generalize. Recall that we presented algorithms for 6 of the 8 problem variants; here we consider which of them generalize to the setting of concave effort conversion functions.

I.1 Problem 1

Recall that Problem 1 reduces to testing the feasibility of $2n$ separate linear programs with a mixture of strict and weak inequalities. (see Appendix B). Each feasible set has the form

$$\{x^1 \in \mathcal{X} \mid a_j \cdot x > b_j \text{ and } F^{s_1}(x^1) \geq F^{s_2}(x^2)\}$$

Algorithm 4: An algorithm for Problem 7 where the effort conversion functions are concave and strictly increasing.

Input: An instance of the evaluation problem with a single admissible profile x^*
Output: A linear mechanism β that incentivizes a maximum number of agents to invest effort according to x^*

1 $\mathcal{R} \leftarrow$ arrangement of all hyperplanes defining $\mathcal{L}^k(x^*)$ for all $k \in [\ell]$;
2 `max` $\leftarrow -1$;
3 **for** *each cell* $C \in \mathcal{R}$ **do**
4 $\quad \beta' \leftarrow$ any point in C;
5 $\quad$ `numIncentivized` $\leftarrow |\{k \in [\ell] \mid \beta' \in \mathcal{L}^k(x^*)\}|$;
6 $\quad$ **if** *numIncentivized* $>$ *max* **then**
7 $\quad\quad$ `max` $\leftarrow$ `numIncentivized`;
8 $\quad\quad \beta \leftarrow \beta'$;
9 $\quad$ **end**
10 **end**
11 **return** β;

(the weak inequality may also be a strict one; our argument works for that case as well). With concave effort conversion functions, the only potential worry is that the condition $F^{s_1}(x^1) \geq F^{s_2}(x^2)$ may no longer be expressible as a set of linear constraints. However, it turns out that it still is. Formally, these constraints are that, for all $i \in [n]$,

$$f_i^1\left(\sum_j \alpha_{j,i}^1 x_j^1\right) \geq f_i^2\left(\sum_j \alpha_{j,i}^2 x_j^2\right)$$

(we are using only agent indices 1 and 2 above to avoid excessive subscripts, and to keep consistent with Appendix B, but the equation has the same form different indices k, k', etc.). Since the fact that each f_i^1 function is increasing implies that it is invertible, we can rewrite this as

$$\sum_j \alpha_{j,i}^1 x_j^1 \geq (f_i^1)^{-1}\left(f_i^2\left(\sum_j \alpha_{j,i}^2 x_j^2\right)\right)$$

which is a linear constraint on the x_j^1 variables since the RHS is a constant. Thus, we can use the exact same technique to solve Problem 1.

I.2 Problem 2

The algorithm for Problem 2 does not immediately extend to the setting of concave effort conversion functions, since there is no clear way to compute the predicate on line 9 of Algorithm 2 in polynomial time, even when n is constant. The main obstacle is that the two sets in Equation 4 (see Appendix C.1) are no longer polytopes, so we cannot construct an arrangement of hyperplanes and enumerate all cells. For suitably nice effort conversion functions, however, it may still be possible, since arrangements can be computed for more general hypersurfaces (Goodman and O'Rourke 1997). This is really a computational geometry question, and the authors suspect that it should still be possible to compute this predicate in polynomial time for most effort conversion functions that one would encounter in practice.

I.3 Problem 5

For linear effort conversion functions, this problem reduced to testing the feasibility of a linear program, with explicit linear constraints $h(k, j_1, j_2)$ (see the proof of Theorem 4.1). While we can't use the same linear program with more general effort conversion functions, we can use a different linear program coming from the main result of Kleinberg and Raghavan. They prove that, even under their more general model of concave effort conversion functions, the set of linear mechanisms incentivizing profile x^* for a given agent k is still a polytope, $\mathcal{L}^k(x^*)$, bounded by polynomially-many linear constraints. Therefore, Problem 5 still reduces to testing the feasibility of a linear program (namely, the intersection of all $\mathcal{L}^k(x^*)$ for $k \in [\ell]$), so we can still solve it in polynomial time.

I.4 Problem 7

This algorithm extends in the same way that our algorithm for Problem 5 does. We just need to find a point of common intersection in a maximum number of $\mathcal{L}_k(x^*)$ polytopes. Algorithm 4, which differs from Algorithms 1 and 3 only on lines 1 and 5 (and is equivalent to Algorithm 1 when the effort conversion functions are linear), solves this problem in polynomial time for constant n.

I.5 Problems 6 and 8

Our algorithms for these two problems rely crucially on the fact that we can assume without loss of generality that the agents will invest their entire budgets into one action (namely, one of the actions with the greatest marginal payoff). Unfortunately, with more general concave effort conversion functions, we cannot analyze best responses purely in terms of marginal payoffs, so this fact no longer holds. Therefore, in their present forms, our algorithms for Problems 6 and 8 do not extend.

One possible way to get around this would be to sample polynomially-many profiles $x_1^*, x_2^*, \ldots, x_q^*$ supported by the admissible actions, and use an arrangement-based algorithm similar to Algorithms 1 and 3 to search for a β contained in $\bigcup_{i=1}^{q} \mathcal{L}^k(x_i^*)$ for a maximum number of indices k. It is uncertain what guarantees could be said for such an algorithm.