Privacy and Truthful Equilibrium Selection for Aggregative Games

Rachel Cummings∗  Michael Kearns†  Aaron Roth†  Zhiwei Steven Wu†

Abstract

We study a very general class of games — multi-dimensional aggregative games — which in particular generalize both anonymous games and weighted congestion games. For any such game that is also large (meaning that the influence that any single player’s action has on the utility of others is diminishing with the number of players in the game), we solve the equilibrium selection problem in a strong sense. In particular, we give an efficient weak mediator: an algorithm or mechanism which has only the power to listen to reported types and provide non-binding suggested actions, such that (a) it is an asymptotic Nash equilibrium for every player to truthfully report their type to the mediator, and then follow its suggested action; and (b) that when players do so, they end up coordinating on a particular asymptotic pure strategy Nash equilibrium of the induced complete information game. In fact, truthful reporting is an ex-post Nash equilibrium of the mediated game, so our solution applies even in settings of incomplete information, and even when player types are arbitrary or worst-case (i.e. not drawn from a common prior). We achieve this by giving an efficient differentially private algorithm for computing a Nash equilibrium in such games. The rates of convergence to equilibrium in all of our results are inverse polynomial in the number of players \( n \). We also give similar results for a related class of one-dimensional games with weaker conditions on the aggregation function, and apply our main results to a multi-dimensional market game.

Our results can be viewed as giving, for a rich class of games, a more robust version of the Revelation Principle, in that we work with weaker informational assumptions (no common prior), yet provide a stronger solution concept (Nash versus Bayes Nash equilibrium). Previously, similar results were only known for the special case of unweighted congestion games.

In the process, we derive several algorithmic results that are of independent interest, and that further the connections between tools in differential privacy and truthfulness in game-theoretic settings. We give the first algorithm for efficiently computing Nash equilibria in aggregative games of constant dimension \( d > 1 \). We also give the first method for solving a particular class of linear programs under the constraint of joint differential privacy.

∗Computing and Mathematical Sciences, California Institute of Technology; rachelc@caltech.edu. Research performed while the author was visiting the University of Pennsylvania.
†Computer and Information Science, University of Pennsylvania; {mkearns,aaroth,wuzhiwei}@cis.upenn.edu
1 Introduction

Games with a large number of players are almost always played, but only sometimes modeled, in a setting of incomplete information. Consider, for example, the problem of selecting stocks for a 401k portfolio among the companies listed in the S&P500. Because stock prices are the result of the aggregate decisions of millions of investors, this is a large multi-player strategic interaction, but it is so decentralized that it is implausible to analyze it in a complete information setting (in which every player knows the types or utilities of all of his opponents), or even in a Bayesian setting (in which every agent shares common knowledge of a prior distribution from which player types are drawn). How players will behave in such interactions is unclear; even under settings of complete information, there remains the potential problem of coordinating or selecting a particular equilibrium among many.

One solution to this problem, recently proposed by [18], is to modify the game by introducing a weak mediator, which essentially only has the power to listen and to give advice. Players can ignore the mediator, and play in the original game as they otherwise would have. Alternately, they can use the mediator, in which case they can report their type to it (although they have the freedom to lie). The mediator provides them with a suggested action that they can play in the original game, but they have the freedom to disregard the suggestion, or to use it in some strategic way (not necessarily following it). The goal is to design a mediator such that good behavior – that is, deciding to use the mediator, truthfully reporting one's type, and then faithfully following the suggested action – forms an ex-post Nash equilibrium in the mediated game, and that the resulting play forms a Nash equilibrium of the original complete information game, induced by the actual (but unknown) player types. A way to approximately achieve this goal – which was shown in [18, 27] – is to design a mediator which computes a Nash equilibrium of the game defined by the reported player types under a stability constraint known as differential privacy [10]. Prior to our work, this was only known to be possible in the special case of large, unweighted congestion games.

In this paper, we extend this approach to a much more general class of games known as multi-dimensional aggregative games (which among other things, generalize anonymous and weighted congestion games). In such a game, there is a vector of linear functions of players’ joint actions called an aggregator. Each player’s utility is then a possibly non-linear function of the aggregator vector and their own action. For example, in an investing game, the imbalance between buyers and sellers of a stock, which is a linear function of actions, may be used in the utility functions to compute prices, which are a non-linear function of the imbalances (see Section 4 for details). In an anonymous game, the aggregator function represents the number of players playing each action. In a weighted congestion game, the aggregator function represents the total weight of players on each of the facilities. Our results apply to any large aggregative game, meaning that any player’s unilateral change in action can have at most a bounded influence on the utility of any other player, and the bound on this influence should be a diminishing function in the number of players in the game.

This line of work can be viewed as giving robust versions of the Revelation Principle, which can implement Nash equilibria of the complete information game using a “direct revelation mediator,” but without needing the existence of a prior type distribution. Compared to the Revelation Principle, which generically requires such a distribution and implements a Bayes Nash equilibrium, truth-telling forms an ex-post Nash equilibrium in our setting. A comparison to previous work can be found in Table 1.
| Mechanism               | Class of Games          | Common Prior? | Mediator Strength | Equilibrium Implemented |
|------------------------|-------------------------|---------------|-------------------|-------------------------|
| Revelation Principle   | Any Finite Game         | Yes           | Weak              | Bayes Nash              |
| Myerson [23]           |                         |               |                   |                         |
| Kearns et al. [18]     | Any Large Game          | No            | Strong            | Correlated              |
| Rogers and Roth [27]   | Large Congestion Games  | No            | Weak              | Nash                    |
| This Work              | Aggregative Games       | No            | Weak              | Nash                    |

Table 1: Summary of truthful mechanisms for various classes of games and solution concepts. Note that a “weak” mediator does not require the ability to verify player types. A “strong” mediator does. Weak mediators are preferred.

1.1 Our Results and Techniques

Our main result is the existence of a mediator which makes truthful reporting of one’s type and faithful following of the suggested action an ex-post Nash equilibrium in the mediated game, thus implementing a Nash equilibrium of the underlying game of complete information. Unlike the previous work in this line [18, 27], we do not have to implement an arbitrary (possibly worst-case) Nash equilibrium, but can implement a Nash equilibrium which optimizes any linear objective (in the player’s actions) of our choosing. We here state our results under the assumption that any player’s action has influence bounded by \((1/n)\) on other’s utility, but our results hold more generally, parametrized by the “largeness” of the game.

**Theorem 1** (Informal). In a \(d\)-dimensional aggregative game of \(n\) players and \(m\) actions, there exists a mediator that makes good behavior an \(\eta\)-approximate ex-post Nash equilibrium, and implements a Nash equilibrium of the underlying complete information game that optimizes any linear objective function to within \(\eta\), where

\[
\eta = O \left( \frac{\sqrt{d}}{n^{1/3}} \cdot \text{polylog}(n, m, d) \right).
\]

The underlying tool that we use is differential privacy. Our main technical contribution is designing a (jointly) differentially private algorithm for computing approximate Nash equilibria in aggregative games. The algorithm that we design runs in time polynomial in the number of players, but exponential in the dimension of the aggregator function. We note that since aggregative games generalize anonymous games, where the dimension of the aggregator function is the number of actions in the anonymous game, this essentially matches the best known running time for computing Nash equilibria in anonymous games even non-privately [9].

In the process of proving this result, we develop several techniques which may be of independent interest. First, we give the first algorithm for computing equilibria of multi-dimensional aggregative games (efficient for constant dimensional games) even in the absence of privacy constraints — past work in this area has focused on the single dimensional case [19, 4]. Second, in order to implement this algorithm privately, we develop the first technique for solving a certain class of linear programs under the constraint of joint differential privacy.

We also give similar results for a class of one-dimensional aggregative games that permit a more general aggregation function and rely on different techniques (Section 5), and we show how our main result can be applied to equilibrium selection in a multi-commodity market (Section 4).
1.2 Related Work

Conceptually, our work is related to the classic Revelation Principle of Myerson [23], in that we seek to implement equilibrium behavior in a game via a “mediated” direct revelation mechanism. Our work is part of a line, starting with [18] and continuing with [27], that attempts to give a more robust reduction, without the need to assume a prior on types. Kearns et al. [18] showed how to privately compute correlated equilibria (and hence implement this agenda) in arbitrary large games. The private computation of correlated equilibria turns out to give the desired reduction to a direct revelation mechanism only when the mediator has the power to verify types. Rogers and Roth [27] rectified this deficiency by privately computing Nash equilibria, but their result is limited to large unweighted congestion games. In this paper, we substantially generalize the class of games in which we can privately compute Nash equilibria (and hence solve the equilibrium selection problem with a direct-revelation mediator).

This line of work is also related to “strategyproofness in the large”, introduced by Azevedo and Budish [3], which has similar goals. In comparison to [3], we do not require that player types be drawn from a distribution over the type-space, do not require any smoothness condition on the set of equilibria of the game, are algorithmically constructive, and do not require our game to be nearly as large. Generally, the results of [3] require the number of agents $n$ to be larger than the size of the action set and the size of the type set. In contrast, we only require $n$ to be as large as the logarithm of the number of actions, and require no bound at all on the size of the type space (which can even be infinite).

Our work is also related to the literature on mediators in games [21, 22]. In contrast to our main goal (which is to implement solution concepts of the complete information game in settings of incomplete information), this line of work aims to modify the equilibrium structure of the complete information game. It does so by introducing a mediator, which can coordinate agent actions if they choose to opt in using the mediator. Mediators can be used to convert Nash equilibria into dominant strategy equilibria [21], or implement equilibrium that are robust to collusion [22]. Ashlagi et al. [2] considers mediators in games of incomplete information, in which agents can misrepresent their type to the mediators. Our notion of a mediator is related, but our mediators require substantially less power than the ones from this literature. For example, our mechanisms do not need the power to make payments [21], or the power to enforce suggested actions [22]. Like the mediators of [2], ours are designed to work in settings of incomplete information and so do not need the power to verify agent types — but our mediators are weaker, in that they can only make suggestions (i.e. players do not need to cede control to our weak mediators).

The computation of equilibria in aggregative games (also known as summarization games) was studied in [19], which gave efficient algorithms and learning dynamics converging to equilibria in the 1-dimensional case. Babichenko [4] also studies learning dynamics in this class of games and shows that in the 1-dimensional setting, sequential best response dynamics converge quickly to equilibrium. Our paper is the first to give algorithms for equilibrium computation in the multi-dimensional setting, which generalizes many well studied classes of games, including anonymous games. The running time of our algorithm is polynomial in the number of players $n$ and exponential in the dimension of the aggregation function $d$, which essentially matches the best known running time for equilibrium computation in anonymous games [9].

We use a number of tools from differential privacy [10], as well as develop some new ones. In particular, we use the advanced composition theorem of [13], the exponential mechanism from [20], and the sparse vector technique introduced by [11] (refined in [15] and abstracted into its current form in [12]). We introduce a new technique for solving linear programs under joint differential
privacy, which extends a line of work (solving linear programs under differential privacy) initiated by [17].

Finally, our work relates to a long line of work initiated by McSherry and Talwar [20] using differential privacy as a tool and desideratum in mechanism design. In addition to works already cited, this includes [25, 24, 28, 14, 7, 5] among others. For a survey of this area see [26].

2 Model and Preliminaries

2.1 Aggregative Games

Consider an $n$-player game with action set $\mathcal{A}$ consisting of $m$ actions and a (possibly infinite) type space $\mathcal{T}$ indexing utility functions. Let $\vec{x} = (x_i, \vec{x}_{-i})$ denote a strategy profile in which player $i$ plays action $x_i$ and the remaining players play strategy profile $\vec{x}_{-i}$. Each player $i$ has a utility function, $u: \mathcal{T} \times \mathcal{A}^n \rightarrow [-1, 1]$, where a player with type $t_i$ experiences utility $u(t_i, \vec{x})$ when players play according to $\vec{x}$. When it is clear from context, we will use shorthand and write $u_i(\vec{x})$ to denote $u(t_i, \vec{x})$, the utility of player $i$.

The utility functions in aggregative games, can be defined in terms of a multi-dimensional aggregator function $S: \mathcal{A}^n \rightarrow [-W, W]^d$, which represents a compact “sufficient statistic” to compute player utilities. In particular, each player’s utility function can be represented as a function only of her own action $x_i$ and the aggregator of the strategy profile $\vec{x}$: $u_i(\vec{x}) = u_i(x_i, S(\vec{x}))$. We also assume $W$ to be polynomially bounded by $n$ and $m$. In aggregative games, the function $S_k$ for each coordinate $k \in [d]$, is an additively separable function: $S_k(\vec{x}) = \sum_{i=1}^n f_i(x_i)$.

Similar to the setting of [19] and [4], we focus on $\gamma$-aggregative games, in which each player has a bounded influence on the aggregator:

$$\max_{x_i, x'_i \in \mathcal{A}} \|S(x_i, \vec{x}_{-i}) - S(x'_i, \vec{x}_{-i})\|_\infty \leq \gamma, \text{ for all } \vec{x}_{-i} \in \mathcal{A}^{n-1}.$$ 

That is, the greatest change a player can unilaterally cause in the aggregator is bounded by $\gamma$. With our motivation to study large games, we assume $\gamma$ diminishes with the population size $n$. We also assume that all utility functions are 1-Lipschitz with respect to the aggregator: for all $a \in \mathcal{A}$, $|u_i(a, s) - u_i(a, s')| \leq \|s - s'\|_\infty$. For $\gamma$-aggregative games, we can express the aggregator more explicitly as

$$S_k(\vec{x}) = \gamma \sum_{i=1}^n f_i^k(x_i),$$

where $f_i^k(a)$ is the influence of player $i$’s action $a$ on the $k$-th aggregator function, and also $|f_i^k(a)| \leq 1$ for all actions $a \in \mathcal{A}$. Let $f_{ij}^k = f_i^k(a_j)$, where $a_j$ denotes the $j$-th action in $\mathcal{A}$.

We say that player $i$ is playing an $\eta$-best response to $\vec{x}$ if

$$u_i(\vec{x}) \geq u_i(a, \vec{x}_{-i}) - \eta, \text{ for all } a \in \mathcal{A}.$$ 

A strategy profile $\vec{x}$ is an $\eta$-pure strategy Nash equilibrium if all players are playing an $\eta$-best response in $\vec{x}$. We also consider mixed strategies, which are defined by probability distributions over the action set. For any profile of mixed strategies, given by a product distribution $\vec{p}$, we can define expected utility $u_i(\vec{p}) = \mathbb{E}_{\vec{x} \sim \vec{p}} u_i(\vec{x})$ and the expected aggregator

$$S_k(\vec{p}) = \mathbb{E}_{\vec{x} \sim \vec{p}} S_k(\vec{x}) = \gamma \sum_{i=1}^n \sum_{j=1}^m f_{ij}^k p_{ij} = \gamma \langle f^k, \vec{p} \rangle.$$ 

\footnote{In some economics literature, aggregative games have more restricted aggregator function: $S_k(\vec{x}) = \sum_{i=1}^n x_i$. The games we study are more general, and sometimes referred to as generalized aggregative games.}
Similarly, a product distribution $\vec{p} = (p_1, \ldots, p_n)$ forms an $\eta$-mixed strategy Nash equilibrium if for every player $i$ and every action $a$, $u_i(\vec{p}) \geq u_i(a, \vec{p} - i) - \eta$.

For each aggregator $s$, we define the aggregative best response\footnote{Sometimes called best react [4], and apparent best response [19].} for player $i$ to $s$ as $\vec{B}\AA_i(s) = \arg \max_{a \in A} \{u_i(a, s)\}$, breaking ties arbitrarily. We define the $\eta$-aggregative best response set for player $i$ to $s$ as 

$$\eta-\vec{B}\AA_i(s) = \{a \in A | u_i(a, s) \geq \max_{a'} u_i(a', s) - \eta\}$$

to be the set of all actions that are at most $\eta$ worse than player $i$’s exact aggregative best response.

**Remark:** Note that best response is played against the other players’ actions $x_{-i}$, but aggregative best response is played against the aggregator value $s$. Aggregative best response ignores the effect of the player’s action on the aggregator, which is bounded by $\gamma$; the player reasons about the utility of playing different actions as if the aggregator value were promised to be $s$. Nevertheless, aggregative best response and best response can translate to each other with only an additive loss of $\gamma$ in the approximation factor. Furthermore, aggregative best responses to different aggregators can translate to each other as long as the corresponding aggregators are close. If $\|s - s'\|_\infty \leq \alpha$, then the actions in $\eta-\vec{B}\AA(s)$ are also in $(\eta + 2\alpha)-\vec{B}\AA(s')$. We state these results more formally in the following lemmas.

**Lemma 1.** Let $\vec{x}$ be a strategy profile such that player $i$’s action $x_i$ is an $\eta$-best response. Then $x_i$ is also an $($$\eta + \gamma$$)$-aggregative best response to $S(\vec{x})$.

**Proof.** Let $s = S(\vec{x})$ and $s' = S(a, x_{-i})$ for some deviation $a \neq x_i$. Since $x_i$ is an $\eta$-best response, we know that $u_i(x_i, s) \geq u_i(a, s) - \eta$. By the bounded influence of player $i$, we know that $\|s - s'\|_\infty \leq \gamma$.

Also, by the Lipschitz property of $u_i$, we have that $|u_i(a, s') - u_i(a, s)| \leq \gamma$. It follows that $u_i(a, s') \geq u_i(a, s) - \gamma$, and therefore $u(x_i, s) \geq u_i(a, s) - \gamma - \eta$. \hfill $\Box$

**Lemma 2.** Let $\vec{x}$ be a strategy profile such that every player is playing $\eta$-aggregative best response to $S(\vec{x})$. Then we know that each player is playing $(\eta + \gamma)$-best response, and hence $\vec{x}$ forms a $(\eta + \gamma)$-Nash equilibrium.

**Proof.** Let $s = S(\vec{x})$ and $s' = S(a, x_{-i})$ for some deviation $a \neq x_i$. Since $x_i$ is an $\eta$-aggregative best response, we know $u(x_i, s) \geq u_i(a, s) - \eta$. We know that $\|s - s'\|_\infty \leq \gamma$ by bounded influence of $i$. Then by the Lipschitz property of $u_i$, $u_i(a, s) \geq u_i(a, s') - \gamma$. It follows that $u_i(x_i, s) \geq u_i(a, s') - \gamma - \eta$. \hfill $\Box$

**Lemma 3.** Suppose action $x_i$ is an $\eta$-aggregative best response to $s$ for player $i$. Let $s'$ be aggregator such that $\|s - s'\|_\infty \leq \alpha$. Then $x_i$ is an $(\eta + 2\alpha)$-aggregative best response to $s'$.

**Proof.** Let $a \neq x_i$ be some deviation for player $i$. Since $x_i$ is an $\eta$-aggregative best response to $s$, we have $u_i(x_i, s) \geq u_i(a, s) - \eta$. By the Lipschitz property of $u_i$, $u_i(x_i, s') \geq u_i(x_i, s) - \alpha$ and also $u_i(a, s) \geq u_i(a, s') - \alpha$. Combining these inequalities, we have $u_i(x_i, s') \geq u_i(a, s') - \eta - 2\alpha$. \hfill $\Box$

### 2.2 Mediated Games

We now define games modified by the introduction of a mediator, following [18, 27]. A mediator is an algorithm $M : (T \cup \{\perp\})^n \to A^n$ which takes as input reported types (or $\perp$ for any player who declines to use the mediator), and outputs a suggested action to each player. Given an aggregative game $G$, we construct a new game $G_M$ induced by the mediator $M$. Informally, in $G_M$, players have
several options: they can opt-out of the mediator (i.e. report ⊥) and select an action independently of it. Alternately they can opt-in and report to it some type (not necessarily their true type), and receive a suggested action \( r_i \). They are free to follow this suggestion or use it in some other way: they play an action \( f_i(r_i) \) for some arbitrary function \( f_i : A → A \). Formally, the game \( G_M \) has an action set \( A_i \) for each player \( i \) defined as \( A_i = A_i' ∪ A_i'' \), where

\[
A_i' = \{(t_i, f_i) : t_i ∈ T, f_i : A → A\} \quad A_i'' = \{ (∐, f_i) : f_i \text{ is constant} \}
\]

Players’ utilities in the mediated game are simply their expected utilities induced by the actions they play in the original game. Formally, they have utility functions \( u'_i: u'_i(t, f) = \mathbb{E}_{x ∼ M(t)}[u_i(f(x))] \).

We are interested in finding mediators such that good behavior is an ex-post Nash equilibrium in the mediated game. We first define an ex-post Nash equilibrium.

**Definition 1.** A collection of strategies \( \{σ_i : T → A\}_{i=1}^n \) forms an \( η \)-approximate ex-post Nash equilibrium if for every type vector \( t ∈ T^n \), and for every player \( i \) and action \( a_i ∈ A \):

\[
u_i(σ_i(t_i), σ_{-i}(t_{-i})) ≥ u_i(a_i, σ_{-i}(t_{-i})) − η
\]

That is, it forms an \( η \)-approximate Nash equilibrium for every possible vector of types.

Note that ex-post Nash equilibrium is a very strong solution concept for incomplete information games because it does not require players to know a prior distribution over types.

In a mediated game, we would like players to truthfully report their type, and then faithfully follow the suggested action of the mediator. We call this good behavior. Formally, the good behavior strategy is defined as \( g_i(t_i) = (t_i, \text{id}) \) where \( \text{id} : A → A \) is the identity function – i.e. it truthfully reports a player’s type to the mediator, and applies the identity function to its suggested action.

In order to achieve this, we use the notion of joint differential privacy defined in [18] (adapted from differential privacy, defined in [10] and presented here in Appendix A), as a privacy measure for mechanisms on agents’ private data (types). Intuitively, it guarantees that the output to all other agents excluding player \( i \) is insensitive to \( i \)'s private type, so the mechanism protects \( i \)'s private information from arbitrary coalitions of adversaries.

**Definition 2** (Joint Differential Privacy [18]). Two type profiles \( t \) and \( t' \) are \( i \)-neighbors if they differ only in the \( i \)-th component. An algorithm \( M : T^n → A^n \) is \( (ε, δ) \)-joint differentially private if for every \( i \), for every pair of \( i \)-neighbors \( t, t' ∈ C^n \), and for every subset of outputs \( S ⊆ A^{n-1} \),

\[
\Pr[M(t)_{-i} ∈ S] ≤ \exp(ε) \Pr[M(t')_{-i} ∈ S] + δ.
\]

If \( δ = 0 \), we say that \( M \) is \( ε \)-jointly differentially private.

We here quote a theorem of [27], inspired by [18] which motivates our study of private equilibrium computation.

**Theorem 2** ([27, 18]). Let \( M \) be a mechanism satisfying \( (ε, δ) \)-joint differential privacy, that on any input type profile \( t \) with probability \( 1 − β \) computes an \( α \)-approximate pure strategy Nash Equilibrium of the complete information game \( G(t) \) defined by \( t \). Then the “good behavior” strategy \( g = (g_1, \ldots, g_n) \) forms an \( η \)-approximate ex-post Nash equilibrium of the mediated game \( G_M \) for

\[
η = α + 2(2ε + β + δ).
\]

Our private equilibrium computation relies on two algorithmic tools, sparse vector mechanism (called Sparse) and exponential mechanism (called EXP), which allows us to access agents’ private data in privacy-preserving manner. (Full details in Appendix A.)
3 Private Equilibrium Computation

In this section, we present the following main result.

**Theorem 3.** Let $G$ be a $d$-dimensional $\gamma$-aggregative game, and $L$ be a $\gamma$-Lipschitz linear objective (cost) function of the players’ actions. There exists a mediator $M$ that makes good behavior an $\eta$-approximate ex-post Nash equilibrium of the mediated game $G_M$, and implements a Nash equilibrium $\vec{x}$ of the underlying complete information game with $L(\vec{x}) \leq \text{OPT} + \eta$, where

$$\eta = O\left(\left(\frac{\eta^2}{d^{1/4}} + \frac{d^{1/2} \gamma^{1/3}}{n^{1/3}} \left(1 + n^{2/3} \gamma^{1/3}\right)\right) \text{polylog}(n,m,d)\right),$$

where $\text{OPT} = \min\{L(\vec{p}) \mid \vec{p} \text{ is an exact mixed strategy Nash equilibrium}\}$.

Recall that the quantity $\gamma$ is diminishing in $n$; whenever $\gamma = O(1/n^{1/2+\varepsilon})$ for $\varepsilon > 0$, the approximation factor $\eta$ tends towards zero as $n$ grows large. Plugging in $\gamma = 1/n$ recovers the bound in Theorem 1.

This result follows from instantiating Theorem 2 with an algorithm that computes an approximate equilibrium under joint differential privacy, presented in Algorithm 1 as **PRESL** (Private Equilibrium Selection).\(^3\)

We here give an informal description of our algorithm absent privacy concerns and then describe how we implement it privately, deferring the formal treatment to Appendix C. The main object of interest in our algorithm is the set-valued function $V(\hat{s}) = S(\xi - \text{BA}(\hat{s})) = \{s(x) : x \in \xi - \text{BA}(\hat{s})\}$ which maps aggregator values $\hat{s}$ to the set of aggregator values that result when players play approximate aggregative best responses to $\hat{s}$. This is a set-valued function, because players may have multiple approximate best responses to $\hat{s}$, and every such combination of player best responses may lead to a different aggregator. An approximate equilibrium will yield an aggregator $\hat{s}$ such that $\hat{s} \in V(\hat{s})$, and so we wish to find such a fixed point for $V$.

For every fixed value $\hat{s}$, the problem of determining whether $\hat{s} \in V(\hat{s})$ is a linear program (because the aggregator is linear), and although $p \in \xi - \text{BA}(\hat{s})$ is not a convex constraint in $\hat{s}$, the aggregative best responses are fixed for each fixed value of $\hat{s}$. The first step of our algorithm simply searches through a discretized grid of all possible aggregators $X = \{-W, -W + \alpha, \ldots, W - \alpha\}^d$, and solves this linear program to check if some point $\hat{s} \in V(\hat{s})$. This results in a set of aggregators $S$ that are induced by the approximate equilibria of the game. The linear program we need to solve is as follows:

$$\forall k \in [d], \quad \hat{s}_k - \alpha \leq \gamma \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^k p_{ij} \leq \hat{s}_k + \alpha$$

$$\forall i \in [n], \quad \forall j \in \xi - \text{BA}_i(\hat{s}), \quad 0 \leq p_{ij} \leq 1$$

$$\forall i \in [n], \quad \forall j \notin \xi - \text{BA}_i(\hat{s}), \quad p_{ij} = 0$$

(2)

in which the vector $p$ represents the mixed strategies of each of the players, and $\xi = \gamma + 2\alpha$ as suggested by Lemmas 1 and 3.

Next, we need to find a particular equilibrium (an assignment of actions to players) that optimizes our cost-objective function $L$. This is again a linear program (since the objective function is linear) for each $\hat{s}$: hence, for each $\hat{s} \in X$ we simply solve this linear program, and out of all of the candidate equilibria, output the one with the lowest cost. Finally, this results in mixed strategies for each of the players – we round this to a pure strategy Nash equilibrium by sampling from each

\(^3\)We also present the full details of the non-private algorithm in Appendix D.
player’s mixed strategy. This does not substantially harm the quality of the equilibrium; because of the low sensitivity of the aggregator, it is well concentrated around its expectation under this rounding. The running time of this algorithm is dominated by the grid search for the aggregator fixed point \( \hat{s} \), which takes time exponential in \( d \). Solving each linear program can be done in time polynomial in all of the game parameters.

Making this algorithm satisfy joint differential privacy is more difficult. There are two main steps. The first is to identify the fixed point \( \hat{s} \in \mathcal{V}(\hat{s}) \) that corresponds the lowest cost equilibrium. There are exponentially in \( d \) many candidate aggregators to check, and with naive noise addition we would have to pay for this exponential factor in our accuracy bound. However, we take advantage of the fact that we only need to output a single aggregator – the one corresponding to the lowest objective value equilibrium – and so the sparse vector mechanism \textbf{Sparse} (described in Appendix A.1) can be brought to bear, allowing us to pay only linearly in \( d \) in the accuracy bound.

The second step is more challenging, and requires a new technique: we must actually solve the linear program corresponding to \( \hat{s} \), and output to each player the strategy they should play in equilibrium. The output strategy profile must satisfy joint differential privacy. To do this, we give a general method for solving a class of linear programs (containing in particular, LPs of the form (2)) under joint differential privacy, which may be of independent interest. This algorithm, which we call \textbf{DistributedMW} (described in Appendix B), is a distributed version of the classic multiplicative weights (MW) technique for solving LPs [1]. The algorithm can be analyzed by viewing each agent as controlling the variables corresponding to their own mixed strategies, and performing their multiplicative weights updates in isolation (and ensuring that their mixed strategies always fall within their best response set \( \xi^{-}\mathbf{BA}(\hat{s}) \)). At every round, the algorithm aggregates the current solution maintained by each player, and then identifies a coordinate in which the constraints are far from being satisfied. The algorithm uses the \textbf{exponential mechanism EXP} (described in Appendix A.2) to pick such a coordinate while maintaining the privacy of the players’ actions. The identification of such a coordinate is sufficient for each player to update their own variables. Privacy then follows by combining the privacy guarantee of the exponential mechanism with a bound on the convergence time of the multiplicative weights update rule. The fact that we can solve this LP in a distributed manner to get joint differential privacy (rather than standard differential privacy) crucially depends on the fact that the sensitivity \( \gamma \) of the aggregator is small.

The linear program ends up finding a set of strategies that form an approximate fixed point of the aggregator best response function, up to error \( \alpha + E \) in each coordinate for

\[
E = O \left( \frac{n\gamma^2}{\epsilon} \text{polylog} \left( n, m, d, \frac{1}{\beta}, \frac{1}{\delta} \right) \right)^{1/2}.
\]

Consider a \( \gamma \)-Lipschitz linear loss function \( L: \mathcal{A}^n \rightarrow [0,1] \):

\[
L(\vec{x}) = \gamma \sum_t \ell_i(x_i) \quad \text{and} \quad L(\vec{p}) = \gamma \mathbb{E}_{\vec{x} \sim \vec{p}} L(\vec{x}) = \gamma \sum_i \langle p_{ij}, \ell_{ij} \rangle.
\]

where \( |\ell_i(a_j)| \leq 1 \) for all actions \( a_j \in \mathcal{A} \), and \( \ell_{ij} = \ell_i(a_j) \). Let our benchmark be

\[
\text{OPT} = \min \{ L(\vec{p}) \mid \vec{p} \text{ is an exact mixed strategy equilibrium} \}
\]

The algorithm \textbf{PRESL} has the following guarantee:

**Theorem 4.** \textbf{PRESL}(\( t, \epsilon, \delta, \beta \)) satisfies \( (2\epsilon, \delta) \)-joint differential privacy, and, with probability at least \( 1 - \beta \), computes a \( 12\alpha \)-approximate pure strategy equilibrium \( \vec{x} \) such that \( L(\vec{x}) < \text{OPT} + 5\alpha \), where

\[
\alpha = O \left( \frac{(\sqrt{n} + d)\gamma}{\epsilon} \text{polylog} (n, m, d, 1/\beta, 1/\delta) \right).
\]
Algorithm 1 Private Equilibrium Selection via LP

\textbf{PRESL}((t, \varepsilon, \delta, \beta))

\textbf{Input}: A type vector \( t \), privacy parameters \((\varepsilon, \delta)\), confidence parameter \( \beta \)

\textbf{Output}: An \( \tilde{O}\left(\frac{(\sqrt{n}+d)^{\frac{2\gamma}{3}}}{\varepsilon}\right)\)-approximate pure strategy Nash equilibrium

\textbf{Initialize}: discretization resolution \( \alpha = E_1 + E_2 \), where

\[
E_1 = \frac{100\gamma \left(d + 1\right) \log(2W) \log(n) + \log\left(\frac{6}{\beta}\right)}{\varepsilon}, \quad E_2 = 100 \left(\frac{n\gamma^2}{\varepsilon} \log\left(\frac{3d}{\beta}\right) \log(n) \sqrt{\log(m) \ln(1/\delta)}\right)^{1/2}
\]

\[
\text{let } \{a(\hat{s}, \hat{y})\} = \text{Sparse}(t, Q, \alpha + E_1, 1, \varepsilon)
\]

if all \( a(\hat{s}, \hat{y}) = \bot \) then Abort

else we have \( (\hat{s}, \hat{y}) \) such that \( a(\hat{s}) \neq \bot \)

let \( \tilde{p} = \text{DistributedMW}(LP(\hat{s}, \hat{y}), \varepsilon, \delta, \alpha, \beta/3) \)

let \( \vec{x} \) be an action profile sampled from the product distribution \( \tilde{p} \)

\textbf{Output}: \( \vec{x} \)

We defer the full proof and technical details to Appendix C.

Remark: The running time of this algorithm is exponential in \( d \), the dimension of the aggregative game. For games of fixed dimension (where \( d \) is constant), this yields a polynomial time algorithm. This exponential dependence on the dimension matches the best known running time for (non-privately) computing equilibrium in anonymous games by [9], which is a sub-class of aggregative games.

Theorem 3 then follows by instantiating Theorem 2 with \( \text{PRESL}((t, \frac{n^{1/3} \gamma^{2/3} d^{1/2}}{\varepsilon}; 1/n, 1/n)) \) – i.e. by setting \( \varepsilon = n^{1/3} \gamma^{2/3} d^{1/2} \) and \( \delta = \beta = \frac{1}{n} \).

4 An Application to Multi-Commodity Markets

Here we give an application of our main result to a natural market-based game, in which aggregator functions are used to compute non-linear prices.

Consider a market with \( d \) types of goods or contracts, which agents can either buy or sell short (i.e. on each contract, an agent can be either long, short, or neutral, and so we can think of actions as being vectors \( a \in A = \{-1, 0, 1\}^d \)). In aggregate, the actions of all \( n \) players will lead to a price for each contract, represented by a vector \( q \in [0, 1]^d \). Agents have (potentially complicated) valuation functions of their positions in the market, modeled as arbitrary functions \( v_i : A \to [-d, d] \), and utilities which are quasilinear in money. Note that such valuation functions can model arbitrary complementarity and substitute relationships between contracts. As a result, the equilibria in this market can be complex and diverse, and equilibrium selection becomes a problem.

Central to the game is a market maker who sets prices for each contract as a function of the demand. The precise pricing rule that the market maker uses determines the structure of the equilibria of the market. In both real markets and our idealized game, one of the market maker’s key objectives is to choose a pricing rule that minimizes his worst-case loss — i.e. the loss he might suffer over the buy and sell decisions of the market participants (defined precisely below). A natural and realistic goal is for this loss to be sublinear in the number \( n \) of participants or trades. For example, in random-walk models of price movements it is typical for market maker loss to be on the
order of $\sqrt{n}$ after $n$ steps or trades. Our model is agnostic to the nature of the commodities being bought and sold — these could, for example, be contracts paying off as a function of the realization of future events, making this a combinatorial prediction market. See [6] and [8] for analyses of market maker loss in both traditional finance and prediction market models, respectively.

In the following, we show how to phrase the market described here as an aggregative game. We implement a market maker that makes truthful reporting an approximate ex-post Nash equilibrium, and computes an asymptotic Nash equilibrium of the underlying market, all while guaranteeing that the market maker has loss bounded by $O(n^{1/2+\varepsilon})$ per commodity, for any constant $\varepsilon > 0$ (i.e. almost achieving an overall loss $O(d\sqrt{n})$).

4.1 Instantiation of the Market as an Aggregative Game

Consider a market with $d$ securities and $n$ players. We want to formalize this setting as a $\gamma$-aggregative game. The action set of each player is $A = \{-1, 0, 1\}^d$, where an action is a $d$-dimensional vector of long, short, or neutral decisions (a portfolio), where 1 and −1 in the $k$-th coordinate respectively indicate buying and selling a unit of the $k$-th security. Player $i$’s private type is described by her private valuation function $v_i : A \to [-d, d]$ that determines her value for any portfolio of $d$ securities, held in positive or negative unit quantities.

Given any strategy profile $\vec{x}$, the imbalance in each security is the number agents buying minus the number of agents selling: $I_k(\vec{x}) = \sum_i (x_i)_k$. If the price of security $k$ is $q_k$ and $a_j = 1$, then the player pays $q_k$; if $a_j = -1$, the player is paid $q_k$; otherwise, the player receives no payment. The price $q_k$ for each security is a (nonlinear) function of the imbalance vector $I$ parameterized by $\lambda$:

$$q_k(I) = \begin{cases} 
0 & \text{if } I_k < -\lambda/2 \\
I_k/\lambda + 1/2 & \text{if } -\lambda/2 \leq I_k \leq \lambda/2 \\
1 & \text{if } I_k > \lambda/2
\end{cases}$$

(3)

This simple “hinge” pricing rule is linear with slope $\lambda$ in a symmetric range of imbalances around 0, and saturates at a price of 1 the case of overdemand (too many buyers) or 0 in the case of underdemand (too many sellers). We note that all the results discussed here also hold for the standard exponential pricing rule often used in prediction markets [8].

![Figure 1: Hinge Pricing Rule](image)

To apply our main result, we define our aggregator to be $S(\vec{x}) = I(\vec{x})/\lambda$. Given an aggregator $s$, the imbalance vector is $\lambda s$ and each player $i$’s payoff function (after rescaling) for playing action $x_i$ is

$$u_i(x_i, s) = \frac{1}{2d} (v_i(x_i) - \langle x_i, q(\lambda s) \rangle).$$
Note that for a fixed action $x_i$, the payoff $u_i$ is a $1/2$-Lipschitz function of price vector $q$, and the price vector $q$ is an $(1/\lambda)$-Lipschitz function of the imbalance $I$, which in turn is a $\lambda$-Lipschitz function of the aggregator. Therefore, the payoff is a $1$-Lipschitz function of the aggregator. Here the range of the aggregator is $[-n/\lambda, n/\lambda]$ in each coordinate, and each player has bounded influence $\gamma = 1/\lambda$.

4.2 Equilibrium Selection and Market Maker’s Loss

For each security $k$, if $I_k$ is positive, then there are $I_k$ more buyers than sellers, and the market maker must sell to these players at price $q_k(I)$. The market maker will have to pay the maximum price of 1 to procure an extra copy of the item for each player, for a loss of $I_k(1 - q_k(I))$. Conversely, if $I_k$ is negative, then there are $I_k$ more sellers than buyers, so the market maker must buy from these players at price $q_k(I)$, for a potential loss of $-I_kq_k(I)$.

Now consider a mediated game in this market in which the market maker wishes to collect private valuation functions from all players and make buy/sell recommendations to each player. We have the freedom to set the pricing rule via choice of the parameter $\lambda$, but are also faced with a bicriteria problem; we need to set prices to minimize the potential loss of the market maker, while still incentivizing truthful reporting from the players. Here, we demonstrate a trade-off between incentives and market maker’s potential loss. First, we have the following lemma which bounds the market maker’s loss as a function of $\lambda$; it is proven in Appendix E.

**Lemma 4.** The loss for the market maker in each security under the pricing rule defined in Equation (3) is bounded by $\lambda/16$.

Furthermore, since we have a $(1/\lambda)$-aggregative game with each player’s action set consisting of $3^d$ actions, the market maker can use PRESL as a mediator. By Theorem 3, each player truthfully reporting their valuation function and following the market maker’s recommendation forms an $\eta$-approximate ex-post Nash equilibrium, where

$$\eta = O\left(\sqrt{d} \left[ \left( \frac{n}{\lambda^2} \right)^{1/3} + \left( \frac{1}{n\lambda} \right)^{1/3} \right] \text{polylog}(n, m, d) \right).$$

For a fixed number of commodities $d$, we get asymptotic truthfulness as long as the market maker sets $\lambda = O(n^{1/2+\varepsilon})$ for any $\varepsilon > 0$. With this setting of $\lambda$, we also guarantee that the market maker experiences worst-case loss at most $O(n^{1/2+\varepsilon})$ per good.

5 Single Dimensional (Quasi)-Aggregative Games

In this section, we consider a more general class of games – quasi-aggregative games, in which the aggregator $S$ is not required to have a linear structure as in aggregative games. We focus on $\gamma$-quasi-aggregative games with a one-dimensional aggregator $S: \mathcal{A}^n \to [-W, W]$, and assume the same properties of bounded influence and Lipschitz utilities.\(^5\) We have the following result:

**Theorem 5.** Let $G$ be a single dimensional $\gamma$-quasi-aggregative game for some $\gamma < 1$. There exists a mediator $M$ that makes good behavior an $\eta$-approximate ex-post Nash equilibrium of the mediated game $G_M$, and implements a Nash equilibrium $\vec{x}$ of the underlying complete information game, where

$$\eta = O((\sqrt{\gamma}) \cdot \text{polylog}(n, m)).$$

\(^4\)In fact, the payoff is a 1/2-Lipschitz function of the aggregator, which is a strictly stronger condition.

\(^5\)This is identical to the setting in [19] and [4].
Similar to Section 3, our mediator is a jointly differentially private algorithm that computes an approximate Nash equilibrium. The algorithm is a private implementation of existing algorithms [19, 4], so we use different techniques for these single-dimensional games. Under certain assumptions, we can also select equilibrium with respect to any Lipschitz objective function of the aggregator (Appendix F.1).

5.1 Private Equilibrium Computation

Our algorithm \texttt{PSummNash}, presented in Algorithm 2, is a privatized version of the \texttt{SummNash} algorithm proposed in [19], that computes an approximate Nash equilibrium under joint differential privacy. The primary change we make to \texttt{SummNash} is that our algorithm only accesses the private data through \texttt{Sparse}. In particular, the algorithm iterates through at most \((2W/\alpha + n)\) strategy profiles and halts as soon as it finds an approximate Nash equilibrium, where \(\alpha\) is the discretization resolution.

Similar to \texttt{PRESL}, \texttt{PSummNash} also finds an equilibrium by computing an approximate fixed point in the aggregator space. At each discretized value \(s\), we let each player \(i\) report her aggregative best response \(\overline{BA}_i(s)\), and we define \(V(s) = S(\overline{BA}(s))\). If \(|s - V(s)|\) is small, then in \(\overline{BA}(s)\), all players are playing an approximate aggregative best response to an aggregator \(V(s)\) that is close to \(s\), so \(\overline{BA}(s)\) forms an approximate Nash equilibrium (by Lemmas 1 to 3). \texttt{PSummNash} iterates through a discretized set of values \(Z = \{-W, -W + \alpha, \ldots, W - \alpha\}\) to find such an aggregator. The algorithm only accesses the players’ private information through calls to \texttt{Sparse}.

\texttt{PSummNash} has two stages. In the first stage, the algorithm checks if there exist any \(s \in Z\) that satisfy \(|V(s) - s| \leq \alpha\). If such an \(s\) is found, \(\overline{BA}(s)\) is a \(O(\gamma + \alpha)\)-Nash equilibrium. The algorithm would then output \(s\) and suggest that each player \(i\) play \(\overline{BA}_i(s)\).

Otherwise, the algorithm moves to the second stage, where it again iterates over \(Z\), this time to find two adjacent aggregators \(s\) and \(s + \alpha\) such that \(s > V(s) + \alpha\) and \(s < V(s + \alpha) - \alpha\). Such pair of \((s, s + \alpha)\) is guaranteed to exist because a failure in the first stage implies that the two endpoints satisfy \(V(-W) > -W + \alpha\) and \(V(W - \alpha) < W - 2\alpha\). Intuitively, the value of \(V\) is “too high” at the lower endpoint, and “too low” at the upper endpoint, so there must some “crossing point” \(s\) in the middle where \(V(s)\) is close to \(s\).

We can define a sequence of strategy profiles \(\mathcal{X} = \{x^0, \ldots, x^n\}\), where

\[
x^j_i = \begin{cases} 
\overline{BA}_i(s) & \text{if } i \leq j \\
\overline{BA}_i(s + \alpha) & \text{otherwise}
\end{cases}
\]

Each profile in \(\mathcal{X}\) is a combination of some prefix in \(\overline{BA}(s)\) and \(\overline{BA}(s + \alpha)\). The sequence of aggregators given by the profiles in \(\mathcal{X}\) is essentially a walk between \(V(s)\) and \(V(s + \alpha)\). By the assumption of bounded influence, changing the action of one player can change the aggregator value by at most \(\gamma\). Thus there must be an action profile \(x \in \mathcal{X}\) such that \(|V(x) - s| \leq \gamma\). Note that players’ actions in \(x\) come from both \(\overline{BA}(s)\) and \(\overline{BA}(s + \alpha)\), so all players are playing an \(O(\alpha)\)-aggregative best response to \(s\) in \(x\), and by Lemma 2, \(x\) forms an \(O(\alpha + \gamma)\)-approximate equilibrium. We here state the formal guarantee of the algorithm, and defer the full proof and technical details to Appendix F.

**Theorem 6.** \(\texttt{PSummNash}(t, \varepsilon, \alpha, \beta)\) satisfies \(\varepsilon\)-joint differential privacy, and with probability at least \(1 - \beta\), computes a \((10\alpha + 2\gamma)\)-approximate pure strategy Nash equilibrium as long as

\[
\alpha \geq O\left(\frac{\gamma}{\varepsilon} \text{polylog}(n, m, 1/\beta)\right).
\]
Algorithm 2 PSummNash$(t, \varepsilon, \alpha, \beta)$

**Input:** An $n$-player type vector $t$, privacy parameter $\varepsilon$, accuracy parameter $\alpha$, and confidence parameter $\beta$

**Output:** A $(10\alpha + 2\gamma)$-approximate Nash equilibrium

**Stage 1**

for each $-W/\alpha \leq k \leq W/\alpha - 1$, define a query $Q_k$ on the players’ private payoff functions:

$$Q_k = |V(k\alpha) - k\alpha|$$

let \{a_k\} = Sparse$(t, \{Q_k\}, 4\alpha, 1, \varepsilon/3)$

if we have some $a_k \neq \bot$ then Output $\vec{BA}(t\alpha)$

**Stage 2**

for each $-W/\alpha + 1 \leq k \leq W/\alpha - 1$, define query $Q'_k$ on the players’ private payoff functions:

$$Q'_k = \max(\min(0, k\alpha - V((k - 1)\alpha)), -2\alpha) + \max(\min(0, V(k\alpha) - k\alpha), -3\alpha)$$

let \{a'_k\} = Sparse$(t, \{Q'_k\}, -4\alpha, 1, \varepsilon/3)$

if all $a'_k = \bot$ then Abort.
else let $l$ be the index such that $a'_l \neq \bot$

for each $0 \leq j \leq n$, let strategy profile $x^j$ be defined as

$$x^j_i = \begin{cases} \vec{BA}_i(l\alpha) & \text{if } i \leq j \\ \vec{BA}_i((l - 1)\alpha) & \text{otherwise} \end{cases}$$

let query $Q''_j = S(x^j)$

let \{a''_j\} = Sparse$(t, \{Q''_j\}, \alpha + \gamma/2, 1, \varepsilon/3)$

if some $a''_j \neq \bot$ then Output $x^j$ else Abort.

Theorem 5 then follows by instantiating Theorem 2 with PSummNash by setting $\varepsilon = \sqrt{\gamma}$, $\beta = 1/n$ and $\alpha = 100\sqrt{\gamma}(12Wn^2)$. (Recall that $\gamma < 1$, so $\sqrt{\gamma}$ dominates $\gamma$.)

**Future Work**

The most interesting open question in this line of work is whether there exists a weak mediator that implements good behavior in every large game. Recall that in [18], it was shown that there exists a strong mediator that implements good behavior in any large game, by giving an algorithm that privately computes a correlated equilibrium in any large game. An equivalent result could be shown for weak mediators by giving an algorithm that is able to compute (under the constraint of joint differential privacy) a Nash equilibrium, subject only to a largeness condition on the game. Note that such an algorithm would not be expected to be computationally efficient in general. However, at the moment it remains open whether such an algorithm exists at all, independent of efficiency concerns.
References

[1] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[2] Itai Ashlagi, Dov Monderer, and Moshe Tennenholtz. Mediators in position auctions. *Games and Economic Behavior*, 67(1):2–21, 2009.

[3] Eduardo M Azevedo and Eric Budish. Strategyproofness in the large as a desideratum for market design. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 55–55. ACM, 2012.

[4] Yakov Babichenko. Best-reply dynamic in large aggregative games. *CoRR*, 2013.

[5] Avrim Blum, Jamie Morgenstern, Ankit Sharma, and Adam Smith. Privacy-preserving public information for sequential games. *arXiv preprint arXiv:1402.4488*, 2014.

[6] T. Chakraborty and M. Kearns. Market making and mean reversion. In *Proceedings of the Twelfth ACM Conference on Electronic Commerce*, 2011.

[7] Yiling Chen, Stephen Chong, Ian A Kash, Tal Moran, and Salil Vadhan. Truthful mechanisms for agents that value privacy. In *Proceedings of the fourteenth ACM conference on Electronic commerce*, pages 215–232. ACM, 2013.

[8] Yiling Chen and Jennifer Wortman Vaughan. A new understanding of prediction markets via no-regret learning. In *Proceedings of the Eleventh ACM Conference on Electronic Commerce*, 2010.

[9] Constantinos Daskalakis and Christos H. Papadimitriou. Discretized multinomial distributions and nash equilibria in anonymous games. In *FOCS*, pages 25–34, 2008.

[10] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. 2006.

[11] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N Rothblum, and Salil Vadhan. On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 381–390. ACM, 2009.

[12] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. 2013.

[13] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010.

[14] Arpita Ghosh and Katrina Ligett. Privacy and coordination: Computing on databases with endogenous participation. In *Proceedings of the Fourteenth ACM Conference on Electronic Commerce*, EC ’13, 2013.

[15] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS*, pages 61–70, Oct 2010.

[16] Justin Hsu, Zhiyi Huang, Aaron Roth, Tim Roughgarden, and Zhiwei Steven Wu. Private matchings and allocations. In *STOC*, pages 21–30, 2014.
We first state the formal definition of differential privacy [10], which is a measure of the privacy of computations on databases. In our setting, a database $D \in \mathcal{T}^n$ contains $n$ players’ private types, which determine their utility functions. Two databases are neighboring if they differ only in a single entry.

**Definition 3 (Differential Privacy [10]).** An algorithm $\mathcal{M}: \mathcal{T}^n \rightarrow \mathcal{R}$ is $(\varepsilon, \delta)$-differentially private if for every pair of neighboring databases $D, D' \in \mathcal{T}^n$ and for every subset of possible outputs $\mathcal{S} \subseteq \mathcal{R}$,

$$\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta.$$  

If $\delta = 0$, we say that $\mathcal{M}$ is $\varepsilon$-differentially private.

We will make use of the following composition theorem, which shows how the privacy parameters $\varepsilon$ and $\delta$ “compose” nicely.

We will make use of the following composition theorem, which shows how the privacy parameters $\varepsilon$ and $\delta$ “compose” nicely.
Theorem 7 (Adaptive Composition [13]). Let $\mathcal{M}: \mathcal{T}^n \to \mathcal{R}^T$ be a $T$-fold adaptive composition\footnote{For a more detailed discussion of $T$-fold adaptive composition, see [13].} of $(\varepsilon, \delta)$-differentially private mechanisms. Then $\mathcal{M}$ satisfies $(\varepsilon', T\delta + \delta')$-differential privacy for

$$
\varepsilon' = \varepsilon \sqrt{2T \ln(1/\delta')} + T\varepsilon(e^\varepsilon - 1).
$$

In particular, for any $\varepsilon \leq 1$, if $\mathcal{M}$ is a $T$-fold adaptive composition of $(\varepsilon/\sqrt{8T \ln(1/\delta)}, 0)$-differentially private mechanisms, then $\mathcal{M}$ satisfies $(\varepsilon, \delta)$-differential privacy.

In the remainder of this section, we review two tools from the differential privacy literature, namely the Sparse Vector Mechanism and the Exponential Mechanism. Both will be used in our algorithms.

A.1 Sparse Vector Mechanism

Our main tool from differential privacy is a slight modification of the sparse vector mechanism from [11] (we follow the presentation of [12]). The sparse vector mechanism \textbf{Sparse} takes in a sequence of low-sensitivity queries $\{Q_t\}$ on database $D$, and a threshold $T$. The mechanism only outputs answers to those queries with (noisy) answers below the (noisy) threshold,\footnote{The Sparse Vector Mechanism as presented in [12] only answers queries with answers above a certain threshold. For the purposes of this paper, we use it instead to answer queries with answers below threshold. This modification does not change the analysis.} and reports that all other queries were above threshold. There is also an upper bound $c$ on the number of queries that can be answered. If more than $c$ queries have answers below the threshold, the mechanism will abort and not produce an output.

This mechanism is especially useful if an analyst is facing a stream of queries and believes that only a small number of the queries will have small answers. The Sparse Vector Mechanism allows the analyst to identify and answer only the “important” queries, without having to incur privacy cost proportional to all queries in the stream.

The sensitivity of a query $Q$, denoted $\Delta(Q)$, is an upper bound over all pairs of neighboring databases on the amount that one entry can affect the answer to the query:

$$
\Delta(Q) = \max_{D, D'} \text{ s.t. } |D \Delta D'| \leq 1 |Q(D) - Q(D')|
$$

Note that because of our restriction to $(\gamma, \rho)$-aggregative games in section 2.1, each coordinate of the aggregative function is $\gamma$-sensitive.

Theorem 8 ([11]). For any sequence of $N$ queries $Q_1, \ldots, Q_N$ such that $|\{k : Q_k(D) \leq T + \alpha\} | \leq c$, \textbf{Sparse} satisfies $\varepsilon$-differential privacy and, with probability at least $1 - \beta$, releases answers such that for all $a_k \in \mathbb{R}$,

$$
|a_k - Q_k(D)| \leq \alpha,
$$

and for all $a_k = \perp$,

$$
Q_k(D) \geq T - \alpha,
$$

where

$$
\alpha = \frac{4c\gamma (\log N + \log(2c/\beta))}{\varepsilon}.
$$
Algorithm 3 Sparse Vector Mechanism

\textbf{Sparse}(D, \{Q_t\}, T, c, \varepsilon)

\textbf{Input:} A private database \(D\), an adaptively chosen stream of queries \(\{Q_t\}\) of sensitivity \(\gamma\), threshold \(T\), total number of numeric answers \(c\), and privacy parameter \(\varepsilon\)

\textbf{Output:} A stream of answers \(\{a_t\}\)

Let \(\hat{T} = T + \text{Lap}\left(\frac{2\gamma}{\varepsilon}\right)\)

let \(\sigma = \frac{2\gamma}{\varepsilon}\)

let count = 0

For each query \(Q_t\) on database \(D\) do

Let \(\nu_t = \text{Lap}(\sigma)\) and \(\hat{Q}_t = Q_t(D) + \nu_t\)

if \(\hat{Q}_t \leq \hat{T}\) and then

Output \(a_t = \hat{Q}_t\),

Update count = count + 1

if count \(\geq c\) then Abort

else Output \(\bot\)

\mA.2 Exponential Mechanism

The exponential mechanism \([20]\) is a powerful private mechanism for selecting approximately the best outcome from a set of alternatives, where the quality of an outcome is measured by a score function relating each alternative to the underlying secret data. Let \(\mathcal{T}^n\) be the domain of input databases, and \(\mathcal{R}\) be the set of possible outcomes, then a score function \(q: \mathcal{T}^n \times \mathcal{R} \rightarrow \mathbb{R}\) maps each database and outcome pair to a real-valued score. The exponential mechanism \(\text{EXP}\) instantiated with database \(D\), a score function \(q\), and a privacy parameter \(\varepsilon\) is defined as

\[
\text{EXP}(D, q, \varepsilon) = \text{output } r \text{ with probability proportional to } \exp\left(\frac{\varepsilon q(D, r)}{2\Delta(q)}\right),
\]

where \(\Delta(q)\) is the global sensitivity of score function \(q\) defined as

\[
\Delta(q) = \max_{r, D, D' \text{ s.t. } |D - D'| \leq 1} |q(D, r) - q(D', r)|.
\]

Then exponential mechanism has the following property:

\textbf{Theorem 9} ([20]). \(\text{EXP}(D, q, \varepsilon)\) satisfies \(\varepsilon\)-differential privacy and, with probability at least \(1 - \beta\), outputs an outcome \(r\) such that

\[
q(D, r) \geq \max_{r'} q(D, r') - \frac{2\Delta(q) \log(|\mathcal{R}|/\beta)}{\varepsilon}.
\]

\mB Distributed Multiplicative Weights Algorithm

In order to compute an equilibrium privately, we need to solve the linear program in (2) under joint differential privacy. This LP has some nice structural properties which allow this to be possible. In particular, the variables are well partitioned among the \(n\) players, such that each player independently controls a set of variables that must form a probability distribution. Each player also has a private restricted feasible set defined by her type (she needs to play approximate
aggregative best response to \( \widehat{s} \) according to her private utility function). This motivates us to solve the following more general linear program:

\[
\forall k \in [d] \quad \gamma \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^k p_{ij} = \gamma \langle f^k, p \rangle \leq b_k \tag{4}
\]

\[
\forall i \in [n] \quad p_i = (p_{i1}, \ldots, p_{im}) \in R_i \subseteq \left\{ x \in \mathbb{R}_{\geq 0}^m \mid \sum_{j} x_j = 1 \right\} \tag{5}
\]

where each \( |f_{ij}^k| \leq 1 \). In this LP, there are two types of constraints. Each agent has a private constraint (5) for her own variables, defined by the restricted feasible set \( R_i \). We also have \( d \) cross-agent constraints (4) that require coordination among the agents. Solving this LP under joint differential privacy guarantees that the output variables to the other agents, \( p_{-i} \), is insensitive in agent \( i \)'s restricted set \( R_i \). Our goal is to find a solution \( p \) that approximately satisfies all \( d \) cross-agent constraints (4) and exactly satisfies all private constraints (5).

Our algorithm DistributedMW is essentially a distributed version of the multiplicative weights (MW) update algorithm [1].

It proceeds in rounds and has each agent running an instantiation of MW over her own private variables. At each round \( t \), the algorithm collects the variables from all players \( p^t = (p^t_1, \ldots, p^t_n) \in \mathbb{R}^{mn} \), and then selects an approximately most violated cross-agent constraint under \( p^t \) using the Exponential Mechanism EXP (see Appendix A.2), where the score for a constraint \( \gamma \langle f, p^t \rangle \leq b \) is defined as

\[
q(p^t, (f, b)) = \gamma \langle f, p^t \rangle - b. \tag{6}
\]

Note that each cross-agent constraint takes the same form, so constraint \( \gamma \langle f, p^t \rangle \leq b \) can be fully described by the pair \( (f, b) \). The mechanism then “broadcasts” the selected constraint \( (f, b) \), and each agent \( i \) uses the \( i \)-th segment of \( f \) as the loss vector to update her instantiation of the MW distribution. After the re-weighting update at each round, each player projects her vector of variables into her private restricted set \( R_i \), so the solution always satisfies the private constraints. Finally, each agent takes the average of the distributions from all rounds to get her output distribution.

**Theorem 10.** DistributedMW\((\cdot, \varepsilon, \delta, \cdot, \cdot)\) satisfies \((\varepsilon, \delta)\)-joint differential privacy.

**Proof.** Our algorithm works in the billboard model introduced by [16]. In particular, the algorithm posts the violated constraint every round on a billboard as a differentially private signal to all agents, such that every agent can see the signal and perform the MW update.

**Lemma 5 (Billboard Lemma. [16]).** Suppose \( \mathcal{M} : \mathcal{T}^n \rightarrow \mathcal{R} \) is \((\varepsilon, \delta)\)-differentially private. Consider any set of functions \( F_i : \mathcal{T}_i \times \mathcal{R} \rightarrow \mathcal{R}' \), where \( \mathcal{T}_i \) is the \( i \)-th entry of the input data. The composition \( \{F_i(\prod_i D, \mathcal{M}(D))\} \) is \((\varepsilon, \delta)\)-jointly differentially private, where \( \prod_i \) is the projection to \( i \)'s data.

The only sub-routines of DistributedMW that access the private data (i.e. private constraints) are the constraint selection at every round using the Exponential Mechanism. Thus, our mechanism has \( T \) instantiations of an \((\varepsilon_0, 0)\)-differentially private mechanism, where \( \varepsilon_0 = \varepsilon/(2\sqrt{2T \ln(1/\delta)}) \). By the Adaptive Composition Theorem (Theorem 7 in Appendix A), we know that the selected constraints satisfy \((\varepsilon, \delta)\)-differential privacy. Note that the \( i \)-th component of the output is a function only of the selected constraints and the MW update rule. By Lemma 5, the algorithm satisfies \((\varepsilon, \delta)\)-joint differential privacy. \(\square\)
Algorithm 4 Distributed Multiplicative Weights for Solving Linear Program

\textbf{DistributedMW}(FeasLP, \varepsilon, \delta, \alpha, \beta)

\textbf{Input:} A feasibility LP FeasLP of with cross-agent constraints of the form (4), private constraints of the form (5), and quality score \( q \) of the form (6), privacy parameters (\( \varepsilon, \delta \)), accuracy parameter \( \alpha \), and confidence parameter \( \beta \)

\textbf{Output:} A solution \( p \) that satisfies all private constraints and only violates any public constraint by at most \( \alpha \)

\textbf{Initialize} \( p^1 \) : \( p^1_{ij} = 1/m \) for all \( i \in [n] \) and \( j \in [m] \)

Let \( T = \frac{16n^2\gamma^2\log m}{\alpha^2} \) \( \varepsilon_0 = \frac{\varepsilon}{2\sqrt{2T \ln(1/\delta)}} \) \( \eta = \alpha/4n\gamma \)

For each round \( t \in \{1, \ldots, T\} \)

Let \((f^t, b^t) = \text{EXP}(p^t, q, \varepsilon_0)\)

Each agent \( i \) performs MW update: for each \( j \)

\[ \hat{p}^{t+1}_{ij} = \exp(-\eta \cdot f^t_{ij}) \cdot p^t_{ij} \]

Projection with relative entropy:

\[ p^{t+1}_i = \text{arg min}_{x \in R_i} \text{RE}(x || \hat{p}^{t+1}_i) \]

\textbf{Output} the average vector \( \overline{p} = 1/T \sum_{t=1}^T p^t \)

\[ \alpha = O \left( \frac{n\gamma^2}{\varepsilon} \text{polylog} \left( n, m, d, \frac{1}{\beta}, \frac{1}{\delta} \right) \right)^{1/2} \]

\textbf{Proof.} Since each \( |f^t_{ij}| \leq 1 \), the MW algorithm gives a no-regret guarantee for each agent \( i \):

\[ \frac{1}{T} \sum_t \langle f^t_i, p^t_i \rangle \leq \min_{p_i \in R_i} \frac{1}{T} \sum_t \langle f^t_i, p_i \rangle + \frac{1}{T} \sum_t \langle f^t_i, p_i \rangle + \frac{\alpha}{2n\gamma} \]

for every agent \( i \). Let \( R = R_1 \times \ldots \times R_n \), then the joint play of all \( n \) agents satisfy

\[ \frac{1}{T} \sum_t \left( \gamma \langle f^t_i, p^t_i \rangle - b^t \right) \leq \min_{p \in R} \frac{1}{T} \sum_t \left( \gamma \langle f^t_i, p \rangle - b^t \right) + \alpha/2 \] (7)

Since there is feasible solution to the LP, we know that

\[ \min_{p \in R} \frac{1}{T} \sum_t \left( \gamma \langle f^t_i, p \rangle - b^t \right) \leq 0, \]

so from Equation (7),

\[ \frac{1}{T} \sum_t \left( \gamma \langle f^t_i, p^t_i \rangle - b^t \right) \leq \alpha/2. \] (8)
By Theorem 9, with probability at least $1 - \beta$, exponential mechanism gives

$$\sum_t (\gamma \langle f^t, p^t \rangle - b^t) \geq \max_{(f,b)} \sum_t \left( (\gamma \langle f, p^t \rangle - b) - \frac{2\gamma \log \left( \frac{dT^t}{\beta} \right)}{\varepsilon_0} \right)$$

and so,

$$\frac{1}{T} \sum_t (\gamma \langle f^t, p^t \rangle - b^t) \geq \max_{(f,b)} \frac{1}{T} \left[ \sum_t (\gamma \langle f, p^t \rangle - b) \right] - \frac{2\gamma \log \left( \frac{dT^t}{\beta} \right)}{\varepsilon_0} \quad (9)$$

Combining Equations (8) and (9) with the definition of $\bar{p}$, we get

$$\max_{(f,b)} (\gamma \langle f, \bar{p} \rangle - b) = \max_{(f,b)} \frac{1}{T} \sum_t (\gamma \langle f, p^t \rangle - b) \leq \frac{2\gamma \log \left( \frac{dT^t}{\beta} \right)}{\varepsilon_0} + \alpha/2 \leq \alpha,$$

as long as $\frac{2\gamma \log \left( \frac{dT^t}{\beta} \right)}{\varepsilon_0} \leq \alpha/2$. Plugging in for parameter $\varepsilon_0$, this condition is equivalent to

$$\alpha^2 \geq \frac{32\sqrt{2n}\gamma^2 \log \left( \frac{dT^t}{\beta} \right) \sqrt{\log m \ln \left( \frac{1}{\delta} \right)}}{\varepsilon} \quad (10)$$

Plugging in $T$, we get our desired bound

$$\alpha = \tilde{O} \left( \frac{n\gamma^2}{\varepsilon} \log \left( \frac{d}{\beta} \right) \sqrt{\log(m) \ln(1/\delta)} \right)^{1/2}.$$

For simplicity, we set the target accuracy to be

$$\alpha = 100 \left( \frac{n\gamma^2}{\varepsilon} \log \left( \frac{d}{\beta} \right) \log(n) \sqrt{\log(m) \ln(1/\delta)} \right)^{1/2} \quad (11)$$

when calling DistributedMW $(\cdot, \varepsilon, \delta, \alpha, \beta)$.

C Details for PRESL

To recap, our goal is to select an approximate pure strategy equilibrium of a $\gamma$-aggregative game such that the objective (cost) function $L$ is approximately minimized, with the benchmark

$$\text{OPT} = \min \{ L(\bar{p}) \mid \bar{p} \text{ is an exact mixed strategy equilibrium} \}.$$  

PRESL has two stages. In the first stage, we try to identify an aggregator $\hat{s} \in X$ such that there exists a mixed strategy profile $\bar{p}$ that satisfies three requirements: (1) all players are playing an approximate aggregative best response to $\hat{s}$; (2) $S(\bar{p})$ is close to $\hat{s}$; and (3) $L(\bar{p})$ is close to OPT.

\footnote{This accuracy level is achievable under mild conditions: as long as $\varepsilon = O(1)$, $\gamma < 1$ and $1/\gamma$ is polynomially smaller than $n^{50}$, then the $\alpha$ in Equation (11) satisfies Inequality (10). Note that for any $W \geq 1$, $\gamma = o(1/n)$ is not interesting since the aggregator is almost the same for any strategy profile, thus any game we consider should easily satisfy this requirement on $\gamma$.}

\footnote{0-mixed strategy equilibrium in our notation}
We find such an aggregator through the objective values in Equation (12). Each LP is defined by a candidate aggregator \( \hat{s} \in X \) and a candidate objective value for \( L : \hat{y} \in \{0, \alpha, \ldots, 1\} \), and so the sequence of queries we want to answer is,

\[
Q = \{ Q(\hat{s}, \hat{y}, \cdot) \mid \hat{s} \in X, \hat{y} \in \{0, \alpha, \ldots, 1\} \};
\]

\[
Q(\hat{s}, \hat{y}, t) = \min a
\]

such that

\[
\forall k, \quad \gamma \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^k p_{ij} \leq \hat{s}_k + a
\]

\[
\forall k, \quad -\gamma \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^k p_{ij} \leq -\hat{s}_k + a
\]

\[
L(\tilde{p}) \leq \hat{y} + a
\]

\[
\forall i, \quad \forall j \in \xi-B\tilde{A}_i(\hat{s}), \quad 0 \leq p_{ij} \leq 1
\]

\[
\forall i, \quad \forall j \notin \xi-B\tilde{A}_i(\hat{s}), \quad p_{ij} = 0
\]

\[
\forall i, \quad \sum_{j=1}^{m} p_{ij} = 1,
\]

We use \text{Sparse} to output the first \((\hat{s}, \hat{y})\) such that the approximate answer of \(Q(\hat{s}, \hat{y}, t)\) is below the threshold \(\alpha + E_1\), where \(E_1\) is the additive error bound for our instantiation of \text{Sparse} (given by Theorem 8). This will guarantee the actual value of \(Q(\hat{s}, \hat{y}, t)\) \(\leq \alpha + 2E_1\).

During the second stage, the algorithm uses \text{DistributedMW} to compute a mixed strategy equilibrium \(\tilde{p}\) by solving a modified version of the above LP, denoted \(LP(\hat{s}, \hat{y})\), without the objective (12) and with \(a\) replaced by \(\alpha + 2E_1\) in the constraints. That is, constraints (13)-(15) are replaced by the following:

\[
\forall k, \quad \gamma \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^k p_{ij} \leq \hat{s}_k + \alpha + 2E_1
\]

\[
\forall k, \quad -\gamma \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij}^k p_{ij} \leq -\hat{s}_k + \alpha + 2E_1
\]

\[
L(\tilde{p}) \leq \hat{y} + \alpha + 2E_1.
\]

**Theorem 4.** \(\text{PRESL}(t, \varepsilon, \delta, \beta)\) satisfies \((2\varepsilon, \delta)\)-joint differential privacy, and, with probability at least \(1 - \beta\), computes a \(12\alpha\)-approximate pure strategy equilibrium \(\tilde{x}\) such that \(L(\tilde{x}) < \text{OPT} + 5\alpha\), where

\[
\alpha = O \left( \left( \frac{\sqrt{n}\varepsilon + d}{\varepsilon} \right) \gamma \log \log \left( \frac{\frac{4\gamma}{\beta} \log \log \left( \frac{6W}{\alpha} \right)}{\frac{4\gamma}{\beta}} \right) \right).
\]

**Proof.** Similar to the privacy proof in Theorem 10, \(\text{PRESL}\) also works in the Billboard model. Each player’s action in the output is a function of only the “broadcast” information and her private type. Note that “broadcast” information comes from both \text{Sparse} and \text{DistributedMW}, which together satisfy \((2\varepsilon, \delta)\)-differential privacy. By the Billboard Lemma 5, our algorithm satisfies \((2\varepsilon, \delta)\)-joint differential privacy.

From Theorem 8, we know that the instantiation \text{Sparse} is \(\varepsilon\)-differentially private, and with probability at least \(1 - \beta/3\), has additive error bounded by

\[
err \leq \frac{4\gamma \left( (d + 1) \log \left( \frac{2W}{\alpha} \right) + \log \left( \frac{6}{\beta} \right) \right)}{\varepsilon} < \frac{100\gamma \left( (d + 1) \log(2W) \log(n) + \log \left( \frac{6}{\beta} \right) \right)}{\varepsilon} = E_1 < \alpha,
\]
as long as $n^{25} > 1/\alpha$. For the rest of the proof, we condition on this level of accuracy, which is the case except with probability at most $\beta/3$. We know there exists an optimal mixed strategy equilibrium\(^{10}\) $p'$ such that $L(p') = \text{OPT}$, and every player places positive probability only on actions that are best responses, which are also $\xi$-aggregative best responses (recall $\xi = \gamma + 2\alpha$) to some discretized point $\hat{s} \in X$, by Lemmas 1 and 3.

Since we set the threshold of Sparse to be $\alpha + E_1$, we are guaranteed that Sparse will output a pair $(\hat{s}, \hat{y})$ such that there exists some mixed strategy $\vec{p}$ with

$$\|S(\vec{p}) - \hat{s}\|_\infty \leq \alpha + 2E_1 \quad \text{and} \quad L(\vec{p}) \leq \hat{y} + \alpha + 2E_1 \leq \text{OPT} + 2\alpha + 2E_1,$$

and where every player only places weight on actions that are a $\xi$-aggregative best response to $\hat{s}$.

Then $\vec{p}$ is a feasible solution to the second stage linear program, $LP(\hat{s}, \hat{y})$. By Theorem 11, DistributedMW will output such a solution $\vec{p}$ where

$$\|S(\vec{p}) - \hat{s}\|_\infty \leq \alpha + 2E_1 + E_2 \quad \text{and} \quad L(\vec{p}) \leq \text{OPT} + 2\alpha + 2E_1 + E_2,$$

except with probability $\beta/3$, where

$$E_2 = 100 \left( \frac{n\gamma^2}{\epsilon} \log \left( \frac{3d}{\beta} \right) \log(n) \sqrt{\log(m) \ln(1/\delta)} \right)^{1/2} < \alpha.$$

Let $\vec{x}$ be the action profile sampled from the mixed strategy $\vec{p}$. By McDiarmid’s inequality, for each coordinate $k$:

$$\Pr[|S_k(\vec{x}) - S_k(\vec{p})| \geq t] \leq 2 \exp \left( \frac{-2t^2}{n\gamma^2} \right), \text{ and } \Pr[|L(\vec{x}) - L(\vec{p})| \geq t] \leq 2 \exp \left( \frac{-2t^2}{n\gamma^2} \right).$$

The union bound gives

$$\Pr[\|S(\vec{x}) - S(\vec{p})\|_\infty \geq t \text{ or } |L(\vec{x}) - L(\vec{p})| \geq t] \leq (d + 1)2 \exp \left( \frac{-2t^2}{n\gamma^2} \right).$$

Then with probability at least $(1 - \beta/3)$, we can guarantee

$$|L(\vec{x}) - L(\vec{p})|, \|S(\vec{x}) - S(\vec{p})\|_\infty \leq \left( \frac{n\gamma^2}{2} \ln \left( \frac{6d + 6}{\beta} \right) \right)^{1/2} \Delta E_3 < \alpha.$$

Overall, we can guarantee the following with probability at least $1 - \beta$

$$\|S(\vec{x}) - \hat{s}\|_\infty \leq \alpha + 2E_1 + E_2 + E_3 < 4\alpha$$

$$L(\vec{x}) \leq L(\vec{p}) + E_3 \leq \text{OPT} + 2\alpha + 2E_1 + E_2 + E_3 < \text{OPT} + 5\alpha.$$

In $\vec{x}$, all players are playing a $\xi$-aggregative best response to $\hat{s}$, so by Lemma 3 they are also playing a $(\xi + 8\alpha)$-aggregative best response to $S(\vec{x})$. By Lemma 2, all players in $\vec{x}$ are playing a $(\xi + \gamma + 8\alpha)$-best response. Since $\gamma < \alpha$ and $\xi + \gamma + 8\alpha < 12\alpha$, then $\vec{x}$ is an $O(\alpha)$-pure strategy Nash equilibrium. \(\square\)

\(^{10}\)An exact mixed strategy equilibrium is guaranteed to exist by Nash’s theorem.
D Non-Private Equilibrium Computation via LP

Here we show that similar techniques to those presented in Appendix C can be used to non-
privately compute and select approximate equilibrium in $\gamma$-aggregative games. In this setting, a
better approximation factor is possible because we no longer need to add noise to preserve privacy.

As before, we approximate the aggregator value domain with a discretized grid of all possible
aggregators $X = \{-W, -W + \alpha, \ldots, W - \alpha\}^d$, and consider a $\gamma$-Lipschitz linear loss function
$L: \mathcal{A}^n \to \mathbb{R}$:

\[
L(x) = \gamma \sum_i \ell_i(x_i) \quad \text{and} \quad L(p) = \gamma \mathbb{E}_{\tilde{x} \sim \tilde{p}} L(\tilde{x}) = \gamma \sum_i \langle p_{ij}, \ell_{ij} \rangle.
\]

where $|\ell_i(a_j)| \leq 1$ for all actions $a_j \in \mathcal{A}$, and $\ell_{ij} = \ell_i(a_j)$.

We first want to find the mixed strategy equilibrium that results in the optimal objective value.
This can be done by solving LP (22) for every $\hat{s} \in X$, denoted $LP(\hat{s})$, where $\xi = \gamma + 2\alpha$.

\[
\begin{align*}
\min_p L(p) \\
\forall k \in [d], \quad \hat{s}_k - \alpha &\leq \gamma \sum_{i=1}^n \sum_{j=1}^m f_{ij}^k p_{ij} \leq \hat{s}_k + \alpha \\
\forall i \in [n], \quad \forall j \in \xi - \tilde{\mathbb{B}}_1(\hat{s}), \quad 0 &\leq p_{ij} \leq 1 \\
\forall i \in [n], \quad \forall j \notin \xi - \tilde{\mathbb{B}}_1(\hat{s}), \quad p_{ij} &\equiv 0 \\
\forall i \in [n], \quad \sum_{j=1}^m p_{ij} &\equiv 1
\end{align*}
\]

(22)

After solving $LP(\hat{s})$ for all $\hat{s} \in X$, algorithm NPRESL selects the mixed strategy profile $\tilde{p}$ that
gives the smallest objective value among all the solutions to the LPs. The algorithm then rounds $\tilde{p}$ to get a pure strategy profile $\tilde{x}$.

Algorithm 5 Non-Private Equilibrium Selection via LP

\begin{algorithm}
NPRESL($t, \alpha, \beta$)
\begin{itemize}
\item \textbf{Input}: A type vector $t$, discretization parameter $\alpha$, and confidence parameter $\beta$
\item \textbf{Output}: An $\hat{O}(\alpha + \sqrt{n}\gamma)$-approximate pure strategy Nash equilibrium
\end{itemize}
\begin{itemize}
\item let $s$ be the aggregator in $X$ that achieves the smallest objective value in $LP(\hat{s})$
\item let $\tilde{p}$ be the solution to $LP(s)$
\item let $\tilde{x}$ be an action profile sampled from the product distribution $\tilde{p}$
\end{itemize}
\begin{itemize}
\item \textbf{Output}: $\tilde{x}$
\end{itemize}
\end{algorithm}

As before, we will define our benchmark OPT to be the optimal objective value that can be
achieved by any exact mixed strategy equilibrium.

Theorem 12. With probability at least $1 - \beta$, NPRESL($t, \alpha, \beta$) computes a $(4\alpha + 2\gamma + 2E)$-
approximate pure strategy Nash equilibrium $\tilde{x}$ such that $L(\tilde{x}) \leq OPT + E$, where

\[
E = \hat{O}\left(\sqrt{n}\gamma \text{polylog}(d, 1/\beta)\right).
\]

Proof. Since the (unrounded) mixed strategy profile $\tilde{p}$ is a feasible solution to $LP(s)$, we know

\[
\|S(\tilde{p}) - s\|_\infty \leq \alpha.
\]

23
We also know that in $\vec{p}$, every player is playing a $\xi$-aggregative best response to $s$, and by Lemma 3, it follows that she is playing a $(\xi + 2\alpha)$-aggregative best response to $S(\vec{p})$. By Lemma 1, every player is playing $(\xi + 2\alpha + \gamma)$-best response, and $\vec{p}$ forms a $(4\alpha + 2\gamma)$-mixed strategy equilibrium.

Note that the mixed strategy equilibrium $p'$ with objective value $\text{OPT}$ is also a feasible solution to $\text{LP}(\hat{s})$ for some $\hat{s} \in X$, because in profile $p'$ every player is playing an exact best response, and is therefore playing a $\xi$-aggregative best response for $\hat{s}$ by Lemmas 1 and 3. Thus the mixed strategy profile $p'$ must be the solution to some $\text{LP}(\hat{s})$ that we consider. Since $\text{NPRESL}$ selects the $\vec{p}$ with the best objective value, we must have $L(\vec{p}) \leq \text{OPT}$.

By McDiarmid’s inequality, for each coordinate $k$:

$$\Pr[|S_k(\vec{x}) - S_k(\vec{p})| \geq t] \leq 2 \exp \left(-\frac{2t^2}{n\gamma^2}\right), \text{ and } \Pr[|L(\vec{x}) - L(\vec{p})| \geq t] \leq 2 \exp \left(-\frac{2t^2}{n\gamma^2}\right).$$

The union bound gives

$$\Pr[\|S(\vec{x}) - S(\vec{p})\| \geq t \text{ or } |L(\vec{x}) - L(\vec{p})| \geq t] \leq (d+1)2 \exp \left(-\frac{2t^2}{n\gamma^2}\right).$$

Then with probability at least $1 - \beta$, we can guarantee

$$|L(\vec{x}) - L(\vec{p})|, \|S(\vec{x}) - S(\vec{p})\| \leq \left(\frac{n\gamma^2}{2} \ln \left(\frac{2d + 2}{\beta}\right)\right)^{1/2} \triangleq E.$$ 

Then with probability at least $1 - \beta$, we know that in $\vec{x}$, each player is playing a $(4\alpha + 2\gamma + 2E)$-best response, and that $L(\vec{x}) \leq \text{OPT} + E$.

\[\square\]

E Details for Multi-Commodity Market Application

**Lemma 4.** The loss for the market maker in each security under the pricing rule defined in Equation (3) is bounded by $\lambda/16$.

**Proof.** Suppose $I_k > 0$. The loss

$$I_k(1 - q_k(I)) \leq I_k(1 - I_k/\lambda - 1/2)$$

$$= I_k(1/2 - I_k/\lambda)$$

$$= -1/\lambda (I_k - \lambda/4)^2 + \lambda/16 \leq \lambda/16.$$ 

Suppose $I_k < 0$, we also have

$$I_k(-q_k(I)) \leq I_k(-I_k/\lambda - 1/2)$$

$$= -1/\lambda (I_k + \lambda/4)^2 + \lambda/16 \leq \lambda/16.$$ 

\[\square\]

F Details for Single Dimensional (Quasi)-Aggregative Games

**Theorem 6.** $PSummNash(t, \varepsilon, \alpha, \beta)$ satisfies $\varepsilon$-joint differential privacy, and with probability at least $1 - \beta$, computes a $(10\alpha + 2\gamma)$-approximate pure strategy Nash equilibrium as long as

$$\alpha \geq O\left(\frac{\gamma}{\varepsilon \text{polylog}(n,m,1/\beta)}\right).$$
Proof. Algorithm 2 only accesses the data through three instantiations of Sparse, each of which satisfy $\varepsilon/3$-differential privacy. By the Composition Theorem in [10], these three computations compose to satisfy $\varepsilon$-differential privacy. Each player’s action in the output strategy profile is a function only of the Sparse’s output and the player’s private data (type). Thus, the algorithm works in the Billboard model, and by Lemma 5, it satisfies $\varepsilon$-joint differential privacy.

We now prove that the algorithm computes an approximate Nash equilibrium. Let $err$ be the error in PSummNash due to its calls to Sparse. We know by Theorem 8 that with probability at least $1 - \beta$, all three instantiations of Sparse have error at most

$$err = \frac{100\gamma}{\varepsilon}(\log(2Wn) + \log(6/\beta)) \leq \alpha,$$

by our assumption on $\alpha$. For the rest of the proof we assume this level of accuracy, which is the case except with probability $\beta$.

First, consider the case that our algorithm outputs a strategy profile in stage 1. We claim that this gives an $(10\alpha + \gamma)$-approximate Nash equilibrium. Let $\vec{BA}(t\alpha)$ be the output. Then by the accuracy level of Sparse,

$$|V(t\alpha) - t\alpha| \leq 4\alpha + err \leq 5\alpha.$$

Since each player’s action is an aggregative best response to aggregator value $t\alpha$, it is also a $10\alpha$-aggregative best response to $V(t\alpha)$ by Lemma 3. Thus, each player is playing a $(10\alpha + \gamma)$-best response as desired by Lemma 2.

Now suppose that the algorithm does not output anything in stage 1. We argue that it will output a $(6\alpha + 2\gamma)$-approximate Nash equilibrium in stage 2.

We first show that the algorithm’s second Sparse outputs an index $l$ such that

$$V((l - 1)\alpha) - l\alpha < V(l\alpha).$$

Since Sparse failed to output a strategy profile in stage 1, we know that

$$|V(k\alpha) - k\alpha| \geq 4\alpha - err \geq 3\alpha$$

for all $-W/\alpha \leq k \leq W/\alpha - 1$.

Since $V(s)$ is in $[-W, W]$ for all $s$, we know that $V(-W) \geq 3\alpha - W$ and $V(W - \alpha) \leq W - 4\alpha$. Then there must exist an index $l$ such that

$$V((l - 1)\alpha) - l\alpha \geq 2\alpha,$n

which implies that $Q'_l = -5\alpha \leq -4\alpha - err$. Such an $l$ satisfies Equation (23) and will also be identified by Sparse as a below threshold query. Note that any index $i$ which does not satisfy Equation (23) must have $Q'_i \geq -3\alpha \geq -4\alpha + err$, so it will not be identified by the Sparse as a below threshold query (since we set the threshold to be $-4\alpha$).

Finally, we claim that we can find an approximate equilibrium between $\vec{BA}((l - 1)\alpha)$ and $\vec{BA}(l\alpha)$. Let $x^j$ be the profile defined in Algorithm 2. Then there exists $j'$ such that

$$|S(x^{j'}) - l\alpha| \leq \gamma/2.$$

Suppose not. Since $S(x^0) > l\alpha > S(x^n)$, there exists $r$ such that $S(x^r) > l\alpha > S(x^{r+1})$ such that

$$S(x^r) - l\alpha > \gamma/2, \text{ and } l\alpha - S(x^{r+1}) > \gamma/2.$$

However, this violates our bounded influence assumption: $|S(x^{r+1}) - S(x^r)| \leq \gamma$.
Since we set the threshold of the third Sparse to be \( \alpha + \gamma/2 \), we can find an index \( j' \) such that
\[
|S(x^{j'}) - l\alpha| \leq 2\alpha + \gamma/2.
\]
Note that players in \( x^{j'} \) are either playing an aggregative best response to aggregative value \( l\alpha \) or \((l - 1)\alpha \). It suffices to bound the payoff loss for the latter ones:
\[
|S(x^{j'}) - l\alpha| \leq |S(x^{j'}) - (l - 1)\alpha| \leq 3\alpha + \gamma/2.
\]
Thus an aggregative best response to \((l - 1)\alpha \) remains a \((6\alpha + \gamma)\)-aggregative best response to \(S(x^{j'})\) by Lemma 3. By Lemma 2, each player can only gain at most \( \gamma \) by deviating from an apparent best response, thus each player is playing \((6\alpha + 2\gamma)\)-best response, which gives at least a \((10\alpha + 2\gamma)\)-approximate equilibrium.

\[\square\]

### F.1 Private Equilibrium Computation with a Lipschitz Objective

The PSummNash algorithm presented in Section 5.1 allowed us to compute an approximate Nash equilibrium in any 1-dimensional \( \gamma \)-quasi-aggregative game. However, if the game has multiple approximate equilibria, it does not guarantee the quality of the equilibrium we obtain. In this section we propose an algorithm to select the approximate Nash equilibrium of the highest quality with respect to a given objective. Our algorithm requires the following assumptions on the quasi-aggregative game and objective:

**Assumption 1** Each player \( i \) has a complete ordering \( \succ_i \) over the action set \( \mathcal{A} \), where \( a \succ_i a' \) if and only if \( S(a, x_{-i}) \geq S(a', x_{-i}) \) for all \( x_{-i} \in \mathcal{A}^{n-1} \). We say player \( i \) is playing **optimistically** if she is maximizing the aggregator value with her action, and playing **pessimistically** if minimizing.

**Assumption 2** Let \( q : [-W, W] \to \mathbb{R} \) be a score function that measures the quality of a aggregator value, where \( q(s) \) is the quality of aggregator value \( s \). We assume that \( q \) is \( \lambda \)-Lipschitz in \( s \).

Kearns and Mansour [19] show that there exists an \( 4\gamma \)-approximate pure Nash equilibrium for any \( \gamma \)-quasi-aggregative game. Let \( \text{OPT}(4\gamma) \) be highest \( q(\bar{x}) \) for any \( 4\gamma \)-approximate Nash equilibrium \( \bar{x} \) of the game. We show that our algorithm can compute an \( O(\gamma) \)-approximate equilibrium that gets a \( O(\lambda\gamma) \) additive approximation to \( \text{OPT} \).

Similar to Algorithm 2, this algorithm also iterates through all \( s \in \mathbb{Z} \), with players submitting their approximate aggregative best response sets to \( s \), \( \zeta \)-BA\( (s)_i \). Let \( X(s) = \{ \bar{x} \mid \text{each player has } x_i \in \zeta \)-BA\( (s)_i \} \), where \( \zeta = 2\alpha + 5\gamma \). We are searching for an approximate pure strategy Nash equilibrium \( x' \in X(s) \) such that \( |S(x') - s| \leq \alpha \). Note that the cardinality of \( X(s) \) can potentially be \( \Omega(m^n) \), but by Assumption 1, we can simply compute the upper and lower bound
\[
S_{\text{max}}(s) = \max_{x \in X(s)} S(x), \text{ and } S_{\text{min}}(s) = \min_{x \in X(s)} S(x)
\]
by asking players to play optimistically and pessimistically, respectively.

If we find such an \( s \) that has an approximate Nash equilibrium strategy profile, there are three cases: either \( S_{\text{max}}(s) \) or \( S_{\text{min}}(s) \) is close to \( s \), or \( s \in [S_{\text{min}}(s), S_{\text{max}}(s)] \). If we are in the first two cases, the algorithm simply outputs the corresponding to the optimistic or pessimistic strategy profile, respectively. In the third case, the algorithm perform the same smooth walk as in Algorithm 2 from the optimistic profile to the pessimistic profile, and outputs an intermediate profile \( x' \) such that \( |S(x') - s| \) is small.
Since we are interested in computing the best equilibrium, we iterate through aggregators in order of their quality score. Let $s_1 > s_2 > \ldots > s_{2W/\alpha}$ be an ordering over the set of discretized aggregator values $Z = \{-W, -W + \alpha, \ldots, W - \alpha\}$, such that $q(s_i) \geq q(s_{i+1})$. The algorithm will sequentially consider $s_i$ according to this ordering $\succ$, to compute an approximate equilibrium with aggregator value that maximizes $q$.

**Algorithm 6** Private Equilibrium Selection with a Lipschitz Objective

**Input:** An $n$-player type vector $t$, privacy parameter $\varepsilon$, accuracy parameter $\alpha$, and confidence parameter $\beta$

**Output:** $(10\alpha + 7\gamma)$-approximate Nash equilibrium with quality score at least $\text{OPT}(4\gamma) - 5\alpha \lambda$

**Initialize:** $G$ such that $q(G) = -W$

For each aggregator value $s \in \{s_1, s_2, \ldots, s_{2W/\alpha}\}$

- let $X(s) = \{\vec{x} \mid \text{each player has } x_i \in (2\alpha + 5\gamma)\overline{BA}(s_i)\}$
- let $s_{\max}(s) = \max_{\vec{x} \in X(s)} S(\vec{x})$ and $s_{\min}(s) = \min_{\vec{x} \in X(s)} S(\vec{x})$

For $1 \leq k \leq 2W/\alpha$, let $\delta$ queries

$$Q_k = |s_{\max}(s_k) - s_k|, \quad Q'_k = |s_{\min}(s_k) - s_k|, \quad Q''_k = \max (\min (s_{\min}(s_k) - s_k, 0), -2\alpha) + \max (s_k - s_{\max}(s_k), 0), -2\alpha)$$

- let $\{a_k\} = \text{Sparse}(t, \{Q_k\}, 3\alpha, 1, \varepsilon/4)$
  - if some $a_i \not\perp$ then let $y = x_{\max}(s_i)$ and the associated aggregator $G = s_i$
- let $\{a'_k\} = \text{Sparse}(t, \{Q'_k\}, 3\alpha, 1, \varepsilon/4)$
  - if some $a'_i \not\perp$ then if $s_i \succ G$ let $y = x_{\min}(s_i)$ and $G = s_i$
- let $\{a''_k\} = \text{Sparse}(t, \{Q''_k\}, 3\alpha, 1, \varepsilon/4)$
  - if some $a''_i \not\perp$ then for each $0 \leq j \leq n$, let strategy profile $x^j$ be defined as
    $$x^j = \begin{cases} x_{\max}(s_i) & \text{if } i \leq j \\ x_{\min}(s_i) & \text{otherwise} \end{cases}$$
  - let query $Q''''_j = S(x^j)$
  - let $\{a'''_j\} = \text{Sparse}(t, \{Q''''_j\}, \alpha + \gamma/2, 1, \varepsilon/4)$
    - if some $a'''_j \not\perp$ then if $s_i \succ G$ let $y = x^{j'}$ if $y$ is defined output $y$ else Abort

**Theorem 13.** Algorithm 6 satisfies $\varepsilon$-joint differential privacy, and with probability at least $1 - \beta$, outputs a $(10\alpha + 7\gamma)$-approximate Nash equilibrium with quality score at least $\text{OPT}(4\gamma) - 5\alpha \lambda$, for any

$$\alpha \geq O\left(\frac{\gamma}{\varepsilon} \text{polylog}(n, m, 1/\beta)\right).$$

**Proof.** Algorithm 6 only accesses the data through four instantiations of Sparse, each of which answers at most one query with $\varepsilon/4$-differential privacy. Again, by the Composition Theorem in [10], these privacy parameters compose so that the strategy profile selection as a public message satisfies $\varepsilon$-differential privacy. The action of each player is a function of only the public message and her own private payoff data (type), so by Lemma 5, Algorithm 6 satisfies $\varepsilon$-joint differential privacy.
We now prove that Algorithm 6 computes an approximate equilibrium with quality score close to \( \text{OPT} \). We know that with probability at least \( 1 - \beta \), all four instantiations of \textbf{Sparse} have error at most
\[
\text{err} = \frac{100\gamma(\log(2Wn) + \log(8/\beta))}{\varepsilon} \leq \alpha,
\]
by our assumption on \( \alpha \). For the rest of the argument, we assume this level of accuracy, which is the case except with probability \( \beta \).

Suppose that the algorithm outputs a strategy profile \( \vec{y} \) from the first two instantiations of \textbf{Sparse}. Let \( m \) be the index of the corresponding query, then
\[
|S(\vec{y}) - s_m| \leq 3\alpha + \text{err} \leq 4\alpha.
\]
Since each player is playing a \((2\alpha + 5\gamma)\)-aggregative best response to \( s_m \) in the profile \( \vec{y} \), we know by Lemma 3 that each player is at least playing a \((10\alpha + 5\gamma)\)-aggregative best response to \( S(\vec{y}) \), and so by Lemma 2, a \((10\alpha + 6\gamma)\)-best response.

Suppose that the algorithm does not output anything in the first two instantiations of \textbf{Sparse}. By the accuracy guarantee, we know that for each \( s_i \in \{s_1, \ldots, s_{2W/\alpha}\} \),
\[
|S_{\text{max}}(s_i) - s_i| \geq 2\alpha \quad \text{and} \quad |S_{\text{min}}(s_i) - s_i| \geq 2\alpha.
\]
Furthermore, we know from [19] that there exists a \( 4\gamma \)-pure strategy Nash equilibrium \( \vec{x}^* \), so each player in \( \vec{x}^* \) must be playing a \( 5\gamma \)-aggregative best response to \( S(\vec{x}^*) \) by Lemma 1. Then by Lemma 3, the players are playing a \((2\alpha + 5\gamma)\)-aggregative best response for some \( s_l \in \mathbb{Z} \). For such an \( s_l \), it must be the case that \( |S(\vec{x}^*) - s_l| \leq \alpha \) and \( \vec{x}^* \in X(s_l) \), so,
\[
S_{\text{min}}(s_l) < s_l < S_{\text{max}}(s_l),
\]
otherwise the first two instantiations of \textbf{Sparse} would have output \( x_{\text{max}}(s_l) \) or \( x_{\text{min}}(s_l) \). The third instantiation of \textbf{Sparse} would find us such an \( s_l \). Now as with Theorem 6, we can find a strategy profile \( \vec{z} \) with the last instantiation of \textbf{Sparse} such that \( \vec{z} \) is between \( x_{\text{max}}(s_l) \) and \( x_{\text{min}}(s_l) \) and,
\[
|S(\vec{z}) - s_l| \leq \alpha + \gamma/2.
\]
By Lemma 2 and Lemma 3, each player in \( \vec{z} \) is playing a \((4\alpha + 6\gamma)\)-aggregative best response to \( s_l \), and so a \((4\alpha + 7\gamma)\)-best response.

Let \( \text{OPT} \) be the maximum \( q(S(\vec{y})) \) among all \( 4\gamma \)-approximate pure Nash equilibrium. Such an optimal strategy profile \( \vec{y}^* \) would be among the strategy profiles that our algorithm searches. Either we output the profile \( \vec{y}^* \), or we found a different profile \( \vec{x} \) corresponding to a discretized aggregator with higher quality score (because we enumerate aggregators with higher quality \( q \) first). Note that in our output strategy profile \( \vec{x} \), all players are playing a \((2\alpha + 5\gamma)\)-aggregative best response to some \( s \in \mathbb{Z} \). Thus \( |S(\vec{x}) - s| \leq 4\alpha \), and \( |S(\vec{y}) - s'| \leq \alpha \) for some discretized aggregator \( s' \in \mathbb{Z} \). Because of the order in which we consider aggregators, we know that \( q(s) \geq q(s') \), so \( q(S(\vec{x})) \geq \text{OPT} - 5\alpha\lambda \).

\( \square \)